whether the player should be prevented from purchasing or otherwise entering into any contract in the future. In another embodiment, determining whether a player should be prevented from purchasing or otherwise entering into a contract in the future may comprise determining whether the player should be prevented from purchasing or otherwise entering into a specific contract and / or a contract defining one or more specific terms or value thereof.

In one embodiment, determining whether a player should be prevented from purchasing or otherwise entering into a contract in the future may further comprise causing the player to be prevented from purchasing or otherwise entering into a contract in the future. For example, an indication of the player's ineligibility to establish a contract, a particular type of contract, and / or a contract defining certain terms may be stored in an appropriate record of a database (e.g., the record corresponding to the player in the player database). In one embodiment, if the player is to be prevented from purchasing or otherwise entering into a contract in the future for a predetermined period of time, an indication of the period of time may also be stored in association with the player's ineligibility.

In one embodiment, the determination of whether a player should be prevented from purchasing or otherwise entering into a contract in the future, based on play of a gaming device monitored previously, may be performed in response to a player's request to establish a contract. In such an embodiment, an analysis of the stored data may be performed and a message informing the player of his ineligibility may be output based on the analysis at the time of the request. This may be done in some embodiments in lieu of the determination being made based on retrieving a previously stored indication of the player's ineligibility from a record of a database).

In one embodiment, determining whether a player should be prevented from purchasing or otherwise entering into a contract in the future may comprise determining whether the player has complied with one or more terms of the contract the player has previously established. For example, a player whose rate of play during the contract period is below a minimum acceptable rate of play defined by a contract may be prevented from establishing future contracts and / or from establishing future contracts with the minimum acceptable rate of play (in one example, a lower minimum rate of play may be defined by a future contract provided to a player). To make a determination of whether the game play data indicates a compliance with one or more terms of a contract, a controller may access game play data and compare it with the terms defined by the subject contract. In the current example, the play data may indicate that the player made 120 handle pulls per hour during the period of time defined by the contract while the contract associated with the player may require a minimum acceptable rate of 200 handle pulls per hour. In one

embodiment, while the player may still receive a refund of losses despite not having complied with the minimum acceptable rate of play, the player may be prevented from establishing a contract in the future and / or from establishing a contract that has a minimum acceptable rate of play of 200 handle pulls per hour or any greater minimum acceptable rate of play.

It should be noted that determining whether a player should be prevented from purchasing or otherwise entering into a contract in the future may comprise determining that a player should be prevented from entering into a contract for a predetermined period of time and / or until a predetermined condition is satisfied. For example, the player may be prevented from entering into a contract for one month, one year, until the player's average minimum rate of play is at least a predetermined average minimum rate of play, until a status associated with the player changed, etc. In one embodiment, step 1610 may further comprise determining the period of time for which the player is to be prevented from purchasing or otherwise entering into a contract.

In one embodiment, monitored game data may indicate that the player has left his contract play card three times in machines that have not seen play for five minutes or more. While this may be an indication of a forgetful customer, it could also be a signal that the player may be intentionally leaving the contract play card in unattended machines in the hopes that another player will unknowingly sit down at that machine and begin a non-contract gaming session. In this way, the player who sat down might thus generate losses associated with that particular gaming contract without costing the first player any money (since the first player did not post the wagers for the game plays while the contract card was in the gaming device). These false losses might then result in substantial refund payments to the first player at the conclusion of the period of time defined by the contract. To address this issue, a controller might direct casino personnel to talk with the player when an orphaned card was detected more than three times. Alternatively or in addition, the controller may prevent such a player from establishing contracts in the future (e.g., by flagging the player, in the record of the player database that corresponds to the player, as a player who is not eligible to establish contracts). In another embodiment, a player might be required to visit a slot club booth to reactivate a contract card if it had been sitting idle in a machine for five minutes or more.

In one embodiment, a controller or other device may adjust game play data associated with the subject contract (e.g., to bring it in line with one or more terms of the contract and / or to compensate for one or more behaviors of a player that appear to be aimed at inappropriately maximizing the value of the benefit defined by the contract). For example, if it is determined that the player is removing his contract card after being dealt a strong initial hand in video poker (e.g. four queens) in an effort to generate large false losses (due to the fact that the controller would not track

the final outcome if the card had been removed), then the controller may be programmed to automatically add outcomes subsequently achieved by the player after the card was removed. The controller may also be operable to track the initial hand dealt to the player and keep tracking (for the purposes of determining win/loss) until the hand was completed. In one embodiment, a warning message may also be output to casino personnel whenever a player withdrew his contract card after being dealt an initially winning (or likely to result in a winning outcome) video poker hand. Such tactics may also be used in other multi-stage games, such as video blackjack or even reel-based games if the player has an ability to pull a contract card out of a gaming device after seeing winning results on the first three reels of a five reel game.

In one embodiment, determining whether a player should be prevented from purchasing or otherwise entering into a contract in the future may be based on a determination of whether a player's gambling behavior changed in a certain manner during the period of time defined by the contract and / or as compared to the player's gambling behavior while the player was not participating in play covered by a contract. For example, monitored game play data may indicate that a player began a period of time defined by a contract by posting relatively small wagers per game play but then increased the size of the wager per game play dramatically as the contract neared the end of the period of time. This behavior of the player might be consistent with a player attempting to artificially decrease the effective standard deviation of the period of time defined by the contract. While this may not violate the terms of the contract, the casino may choose not to enter into contracts with this player in the future.

In one embodiment, determining whether a player should be prevented from purchasing or otherwise entering into gaming contracts in the future may be based on a determination of whether monitored game play data indicates that the player identifier presented at the reconciliation of a contract matches that provided at the establishment of the contract. Non-matching identification may indicate that a player has sold his contract to anther player (perhaps exploiting the fact that contracts may be limited in number and thus may be highly desirable on some days or that only certain player may be eligible to enter into contract in general or into certain highly desirable contracts). The selling player may be prevented from entering into a contract in the future. Similarly, game play data may indicate that a first player has handed his contract card to another player (perhaps determined through the review of surveillance cameras). If such behavior is contrary to the terms of the contract, the player may have restrictions placed on his ability to enter into a contract in the future.

In accordance with some embodiments, a gaming device may have a feature which allows a player to build up equity, such as by collecting diamonds or other benefits or symbols upon an occurrence of a predetermined event during game play (e.g., whenever a special symbol appears on the payline and / or a predetermined outcome is achieved). At such machines, a player participating in contract play may advantageously play with his contract card in the machine until achieving a number of diamond symbols. At this point the may remove the contract card and quickly achieve a bonus payout for accumulating enough diamond symbols. This method of play may be determined by the controller based on the monitored game play data and the player may be prevented from using that particular type of equity machine in future contract play and / or may have other restrictions place on his ability to purchase or otherwise enter into contracts.

In one embodiment, a controller may be operable to terminate contract play under a contract based on the monitored play of the gaming device. The controller may be operable to do so addition to or in lieu of preventing a player from purchasing or otherwise entering into a contract in the future. Alternatively, the controller may reduce benefits to the player associated with the contract. For example, if the contract defines a refund rate of 100% of a player's losses, the controller may determine to pay back only 50% of the player's losses based on the monitored play of the gaming device.

As described, in one or more embodiments, rather than preventing the player from purchasing or otherwise entering into a contract in the future, the controller may be operable to add rules/restrictions regarding the player's eligibility to enter into contracts in the future. Such rules/restrictions might include, for example:

(i) that the player must pay 50% more for a contract than the player would otherwise be required to pay;

(ii) that the player is eligible for some types of contracts or contract terms, but ineligible for other types of contracts or contract terms;

(iii) that the player may continue to be eligible to enter into a contract, but not for a contract that is eligible for contract play on video poker devices;

(iv) that the player may only establish one contract per month or per other unit of time;

(v) that the player is eligible only for certain value or value ranges of certain contract terms (e.g., a refund rate of up to 50% of losses); and / or

(vi) that the player must maintain a minimum of 500 handle pulls per hour during all contract play.

The foregoing description discloses only exemplary embodiments of the invention; modifications of the above disclosed apparatus and methods which fall within the scope of the invention will be readily apparent to those of ordinary skill in the art.

For example, it should be understood that aspects of the invention may be utilized in connection with a device or devices located at a table game which facilitate placement of bets or other activities at a table game while reducing or eliminating actions required on a part of a player of the table game. For example, the MP21 table manufactured by Bally Table Management Systems (TMS) division of Bally Systems® is an advanced blackjack table that includes an array of state-of-the-art optical and electronic sensors. The MP21 constantly captures real-time data to instantly track and record every card dealt and every wager made to determine an accurate reporting of table game results. A device such as the MP21 may be used in embodiments of the present invention to monitor contract play at a table game. Other products manufactured by Bally TMS may also be used in embodiments of the present invention to monitor contract play at a table game. For example, Bally TMS produces the technology formerly known as MindPlay, which includes touch-screen data ratings products (formerly known as eTABLE), card security (Bally MPBaccarat(TM), UCS intelligent card shoe), and the MP21. Any and all of these products may be used in an embodiment of the present invention. Further, U.S. Patent No. 6,460,848 to Soltys et al., entitled "Method and Apparatus for Monitoring Casinos and Gaming" describes a "system [that] automatically monitors playing and wagering of a game, including the gaming habits of players and the performance of employees." (Abstract). The systems and methods of this invention may be used some embodiments described herein to monitor gaming activity of a player (e.g., at a table game) and is hereby incorporated by reference herein for all purposes.

It should also be understood that aspects of the present invention may be applicable to games in which the skill of the player and / or player input may partially or completely determine the outcomes. Such games may include video poker and video blackjack and may also include other games not usually present in casinos. For example, such games may include a simulation of a golf

putting game, in which player input causes a simulated golf ball to be propelled toward a simulated golf hole. If the simulated ball lands in the simulated hole, a prize may be awarded. A machine which allows playing of such a simulated golf game is to be included in the term "gaming device" as used herein.

Accordingly, while the present invention has been disclosed in connection with exemplary embodiments thereof, it should be understood that other embodiments may fall within the spirit and scope of the invention as defined by the following claims.

What is claimed:

1. A method comprising:

determining that play of a gaming device qualifies for coverage under a contract previously purchased;

monitoring the play; and

reconciling the contract based on the monitored play.

2. The method of claim 1, wherein determining is based on receiving, from a gaming device, an indication that an identifier that is associated with the contract has been inserted into the gaming device, and further comprising:

determining terms of the contract; and

determining that play of the gaming device satisfies the terms.

3. The method of claim 1, wherein reconciling comprises:

determining whether the contract has been complied with; and

providing a benefit to a player who purchased the contract only if the contract has been complied with.

4. The method of claim 3, wherein the benefit is a refund of at least a portion of losses incurred by the player during a period of time defined by the contract.

5. The method of claim 3, further comprising:

determining that the contract has been complied with.

6. The method of claim 5, wherein determining that the contract has been complied with comprises:

determining that an amount of play defined by the contract has been completed in a satisfactory manner.

7. The method of claim 6, wherein determining that the amount of play defined by the contract has been completed in a satisfactory manner comprises at least one of:

determining that the amount of play is not less than a minimum amount of play;

determining that the amount of play is not more than a maximum amount of play;

determining that the amount of play equals a specified amount of play;

determining that the play was conducted on a gaming device approved for play under the contract;

determining that the play was conducted within a period of time defined by the contract;

determining that the play required a minimum sum of wagers;

determining that the play was conducted at a minimum required rate; and

determining that a minimum wager amount was posed for at least one game play encompassed by the play.

8. A method, comprising:

receiving a contract initiation signal from a gaming device;

storing data associated with game play of the gaming device until one of a contract play termination signal is received and an end of a contract period is determined; and

providing a benefit defined by the contract based on whether the data indicates a compliance with terms of the contract.

9. The method of claim 8, wherein the benefit comprises a refund of at least a portion of losses incurred by a player associated with the contract during a period of time defined by the contract.

10. The method of claim 8, wherein the benefit comprises a refund of at least a portion of wagers posted by a player associated with the contract during a period of time defined by the contract.

11. The method of claim 8, wherein the benefit comprises authorizing the gaming device to allow play of the gaming device once a credit meter balance of the gaming device has been depleted below a predefined level.

12. The method of claim 11, wherein authorizing the gaming device to allow play of the gaming device comprises authorizing the gaming device to allow the credit meter balance to be a negative number.

13. The method of claim 11, wherein authorizing the gaming device to allow play of the gaming device comprises authorizing the gaming device to add credits to the credit meter balance without requiring payment therefore from a player playing the gaming device.

14. A system, comprising:

a computing device operable to communicate with a plurality of gaming devices, each of the gaming devices operable to facilitate a wagering game, the computing device being further operable to:

determine an initiation of a game play at one of the gaming devices;

determine an identifier of a contract associated with the game play,

wherein the contract has been entered into by a player prior to the initiation of the game play, and

wherein the contract defines a contract period and a benefit to which the player is entitled if the terms of the contract have been satisfied;

determine data associated with the game play; and

determine whether the player is entitled to the benefit based on the data and the terms.

15. The system of claim 14, the system further comprising:

the plurality of gaming devices.

16. The system of claim 14, wherein the computing device is further operable to:

store the data in association with data from at least one other game play, thereby storing data indicative of a plurality of game plays; and

determine whether the player is entitled to the benefit based on the data indicative of the plurality of game plays.

17. The system of claim 14, wherein the benefit is provided to a player at an end of a period defined by the contract, provided the player has complied with the terms of the contract.

18. The system of claim 14, wherein the benefit is provided to a player during a period of time defined by the contract, provided the player is currently complying with the terms of the contract.

19. The system of claim 14, wherein the contract comprises a contract in exchange for which the player provided payment.

20. The system of claim 14, wherein the computing device is further operable to:
determine whether the game play qualifies as a game play that is covered under the terms of a contract.

21. The system of claim 20, wherein the computing device is further operable to:
store the data associated with the game play in association with the identifier only if the game play qualifies as a game play that is covered under the terms of a contract.

22. The system of claim 14, wherein the computing device is further operable to:
identify the contract based on an identifier provided by the player to the gaming device.

23. The system of claim 22, wherein the identifier is an identifier uniquely identifying the contract.

24. The system of claim 22, wherein the identifier uniquely identifying the player.

25. The system of claim 14, wherein the computing device is further operable to:
provide the benefit to the player.

26. The system of claim 14, wherein the computing device being operable to provide the benefit to the player comprises the computing device being operable to cause a gaming device of the plurality of gaming devices to output the benefit to the player.

27. The system of claim 14, wherein the computing device being operable to provide the benefit to the player comprises the computing device being operable to authorize a casino employee to provide the benefit to the player.

28. The system of claim 14, wherein the benefit is a monetary payment.

29. The system of claim 28, wherein the monetary payment is an amount based on a refund of at least a portion of losses incurred by the player during the contract period.

30. The system of claim 14, wherein the computing device is further operable to:
receive a request from the player for the benefit; and
determine, in response to the request, whether the player is entitled to the benefit based on the data and the terms.

31. The system of claim 14, wherein the computing device is further operable to:
determine the benefit based on the data and the terms of the contract.

32. The system of claim 31, wherein the computing device being operable to determine the benefit based on the data and the terms of the contract comprises the computing device being operable to determine a value of the benefit based on the data and the terms of the contract.

33. The system of claim 14, wherein the computing device is further operable to:
authorize a sale of the contract to the player.

34. A method, comprising:
determining data associated with play of a gaming device by a player; and
determining, based on the data, whether to prevent the player from entering into a contract based on the data.

35. The method of claim 34, wherein the contract allows a player to earn a benefit defined by the contract in exchange for participating in play of a gaming device in accordance with terms of the contract and during a period of time defined by the contract.

36. The method of claim 34, wherein the data associated with the play of the gaming device is also associated with a contract previously entered into by the player.

37. The method of claim 34, wherein determining whether to prevent the player from entering into a contract comprises at least one of:

determining whether to prevent the player from entering into a particular contract;

determining whether to prevent the player from entering into a contract defining a particular term;

determining whether to prevent the player from entering a particular type of contract; and

determining whether to prevent, for a period of time, the player from entering a contract.

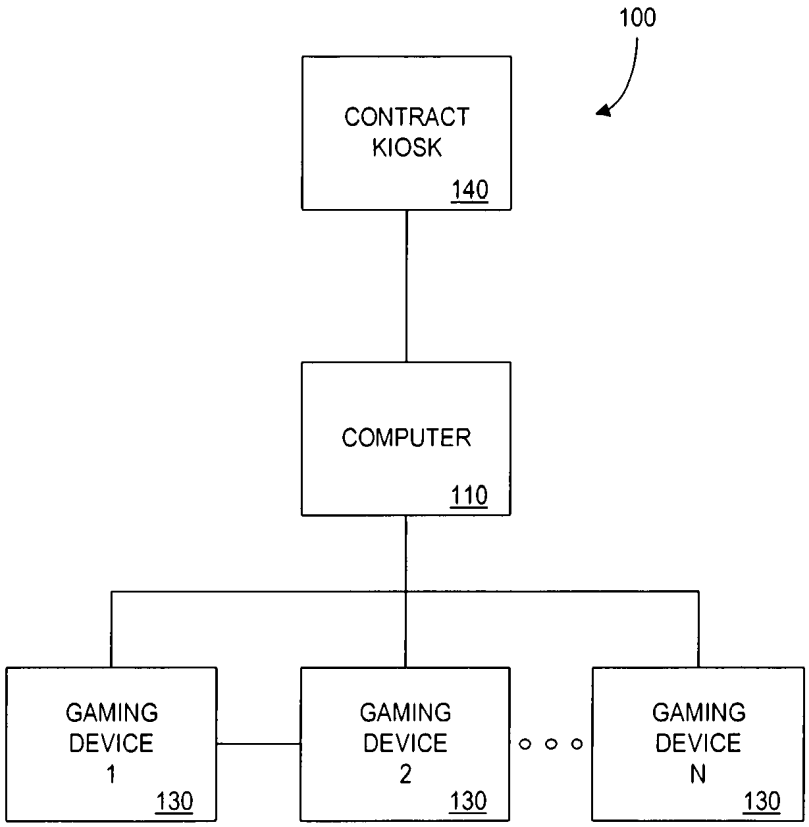

100
CONTRACT KIOSK
140
COMPUTER
110
GAMING DEVICE 1
130
GAMING DEVICE 2
130
GAMING DEVICE N
130

FIG. 1

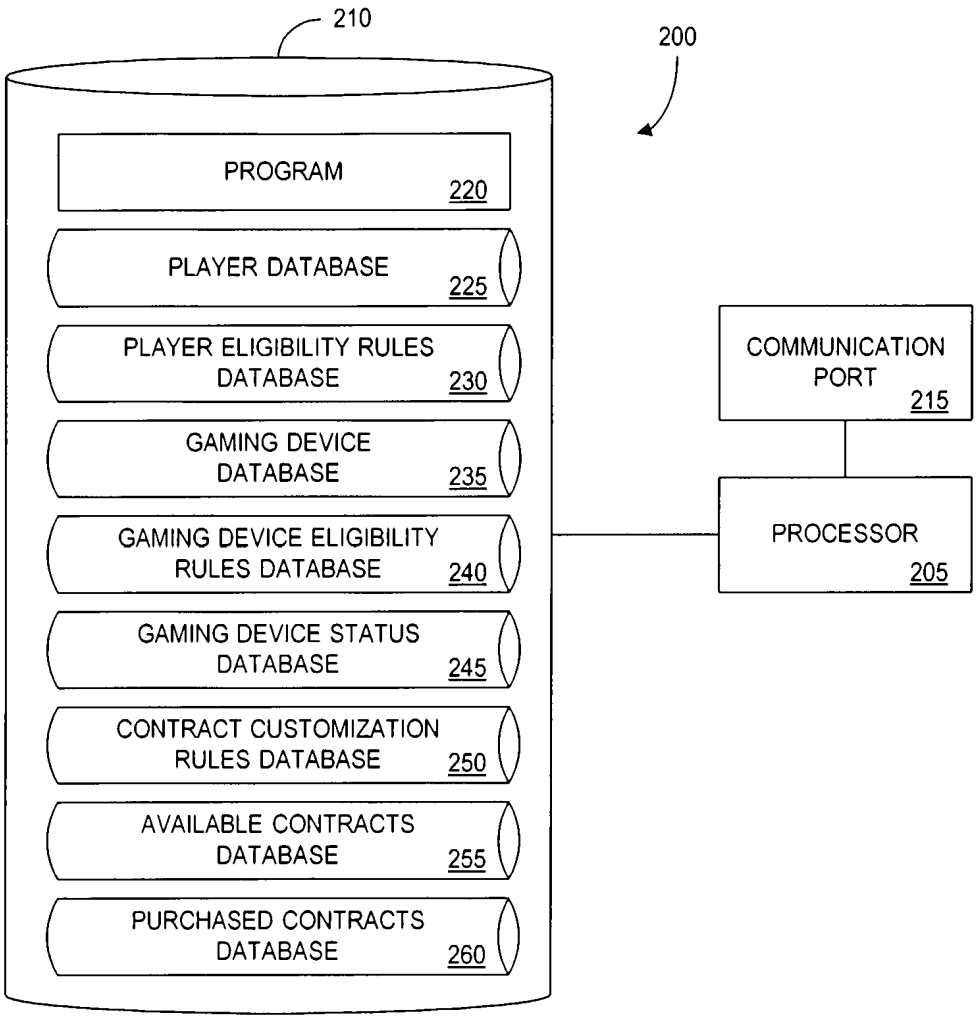

210
200
PROGRAM
220
PLAYER DATABASE
225
PLAYER ELIGIBILITY RULES DATABASE
230
GAMING DEVICE DATABASE
235
GAMING DEVICE ELIGIBILITY RULES DATABASE
240
GAMING DEVICE STATUS DATABASE
245
CONTRACT CUSTOMIZATION RULES DATABASE
250
AVAILABLE CONTRACTS DATABASE
255
PURCHASED CONTRACTS DATABASE
260
COMMUNICATION PORT
215
PROCESSOR
205

FIG. 2

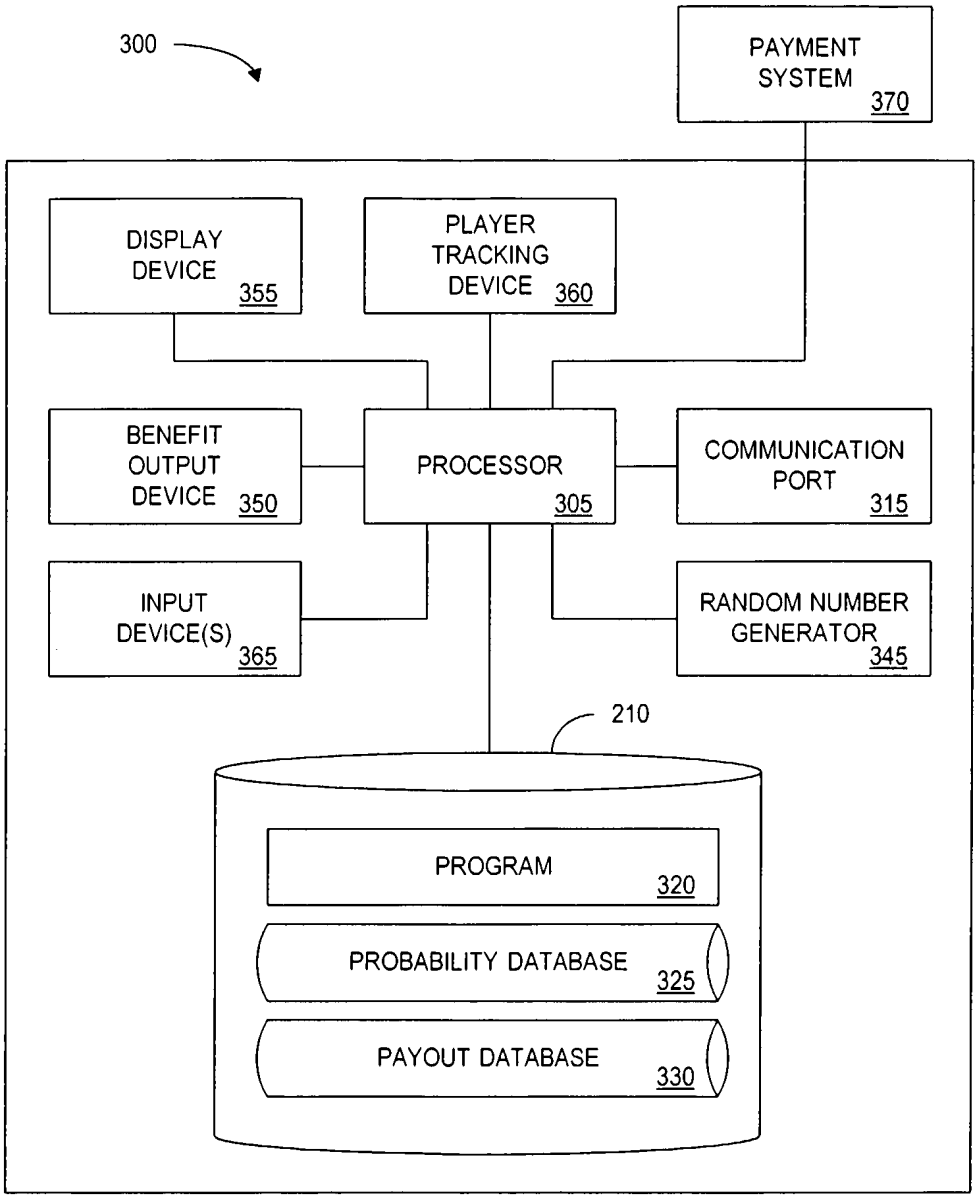

300
PAYMENT SYSTEM 370
DISPLAY DEVICE 355
PLAYER TRACKING DEVICE 360
BENEFIT OUTPUT DEVICE 350
PROCESSOR 305
COMMUNICATION PORT 315
INPUT DEVICE(S) 365
RANDOM NUMBER GENERATOR 345
210
PROGRAM 320
PROBABILITY DATABASE 325
PAYOUT DATABASE 330

FIG. 3

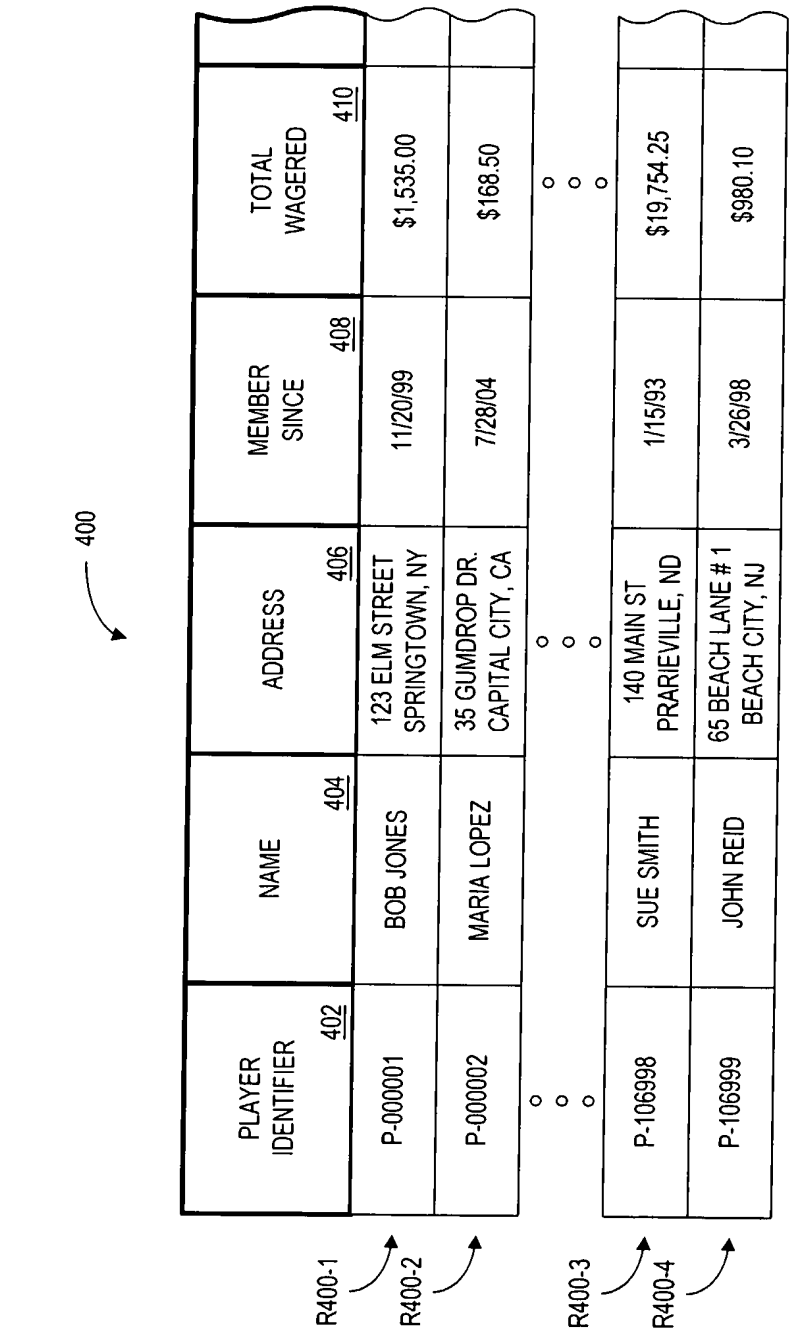

400
PLAYER IDENTIFIER 402
NAME 404
ADDRESS 406
MEMBER SINCE 408
TOTAL WAGERED 410
P-000001
BOB JONES
123 ELM STREET SPRINGTOWN, NY
11/20/99
$1,535.00
P-000002
MARIA LOPEZ
35 GUMDROP DR. CAPITAL CITY, CA
7/28/04
$168.50
P-106998
SUE SMITH
140 MAIN ST PRARIEVILLE, ND
1/15/93
$19,754.25
P-106999
JOHN REID
65 BEACH LANE # 1 BEACH CITY, NJ
3/26/98
$980.10
R400-1
R400-2
R400-3
R400-4

FIG. 4A

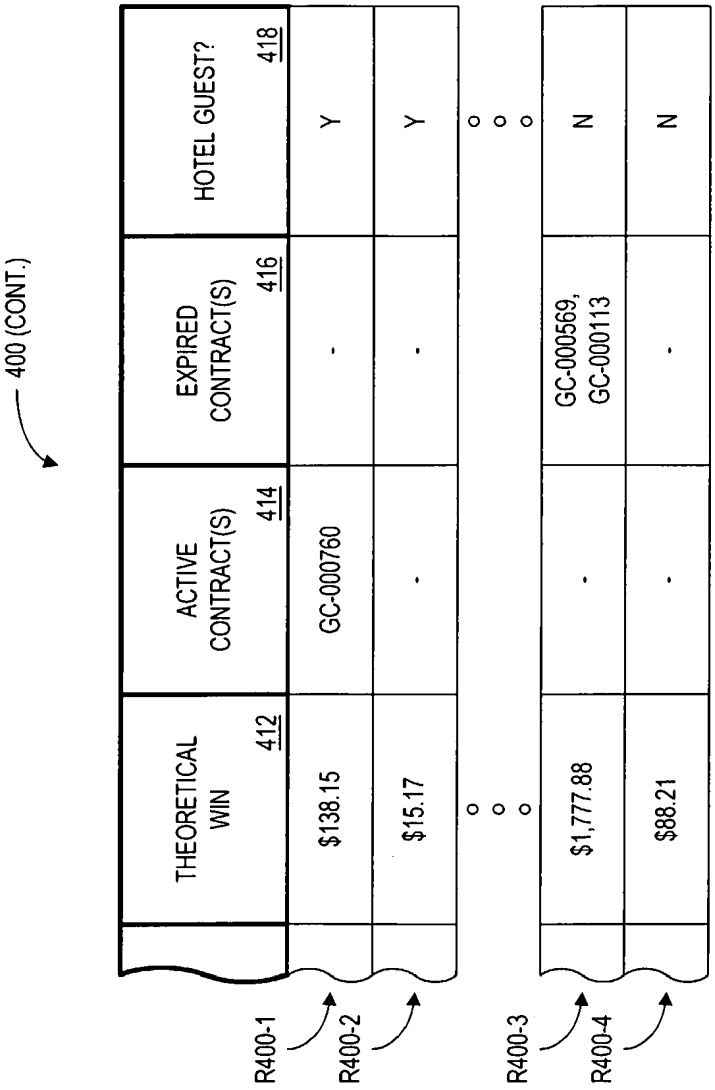

400 (CONT.)
THEORETICAL WIN 412
ACTIVE CONTRACT(S) 414
EXPIRED CONTRACT(S) 416
HOTEL GUEST? 418
R400-1
$138.15
GC-000760
-
Y
R400-2
$15.17
-
-
Y
R400-3
$1,777.88
-
GC-000569, GC-000113
N
R400-4
$88.21
-
-
N

FIG. 4B

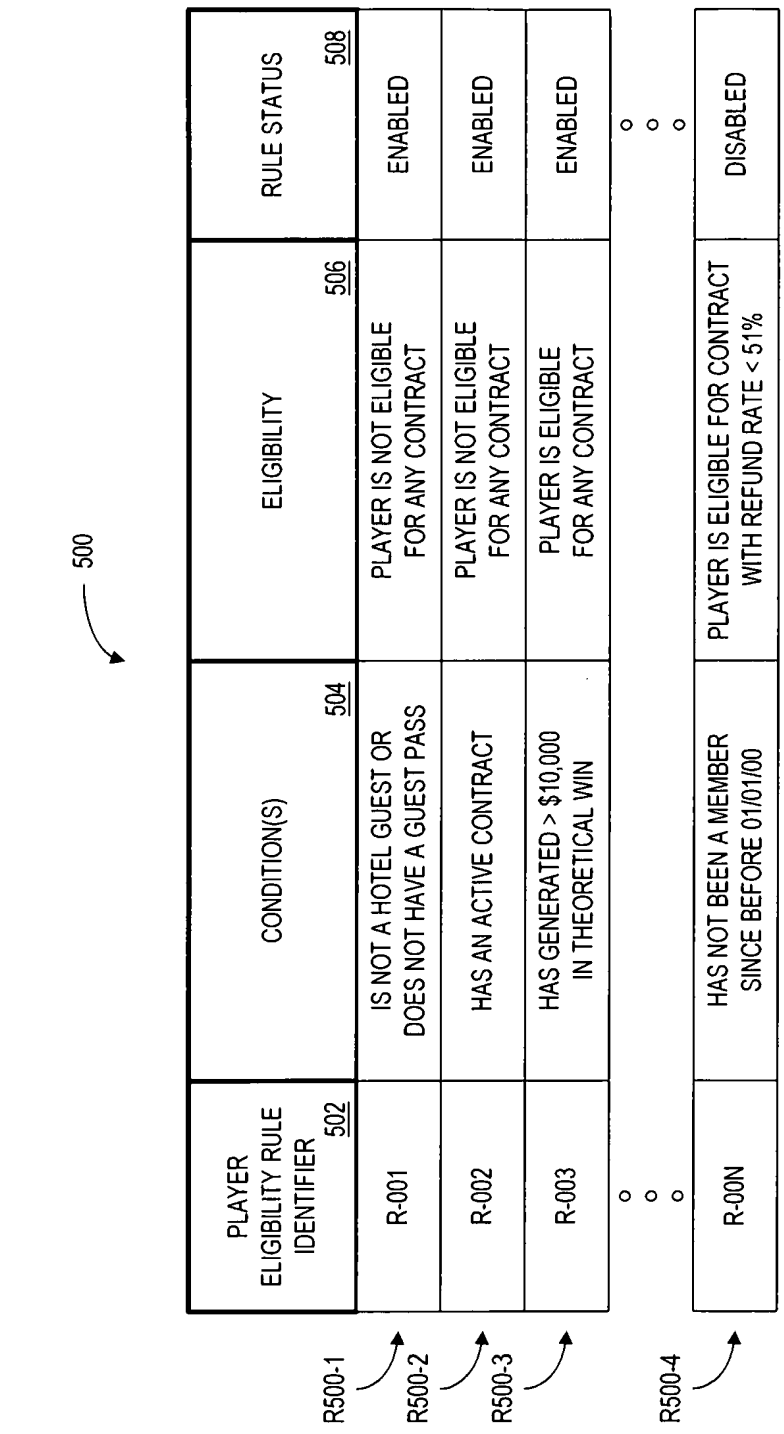

500
PLAYER ELIGIBILITY RULE IDENTIFIER 502
CONDITION(S) 504
ELIGIBILITY 506
RULE STATUS 508
R-001
IS NOT A HOTEL GUEST OR DOES NOT HAVE A GUEST PASS
PLAYER IS NOT ELIGIBLE FOR ANY CONTRACT
ENABLED
R-002
HAS AN ACTIVE CONTRACT
PLAYER IS NOT ELIGIBLE FOR ANY CONTRACT
ENABLED
R-003
HAS GENERATED > $10,000 IN THEORETICAL WIN
PLAYER IS ELIGIBLE FOR ANY CONTRACT
ENABLED
R-00N
HAS NOT BEEN A MEMBER SINCE BEFORE 01/01/00
PLAYER IS ELIGIBLE FOR CONTRACT WITH REFUND RATE < 51%
DISABLED
R500-1
R500-2
R500-3
R500-4

FIG. 5

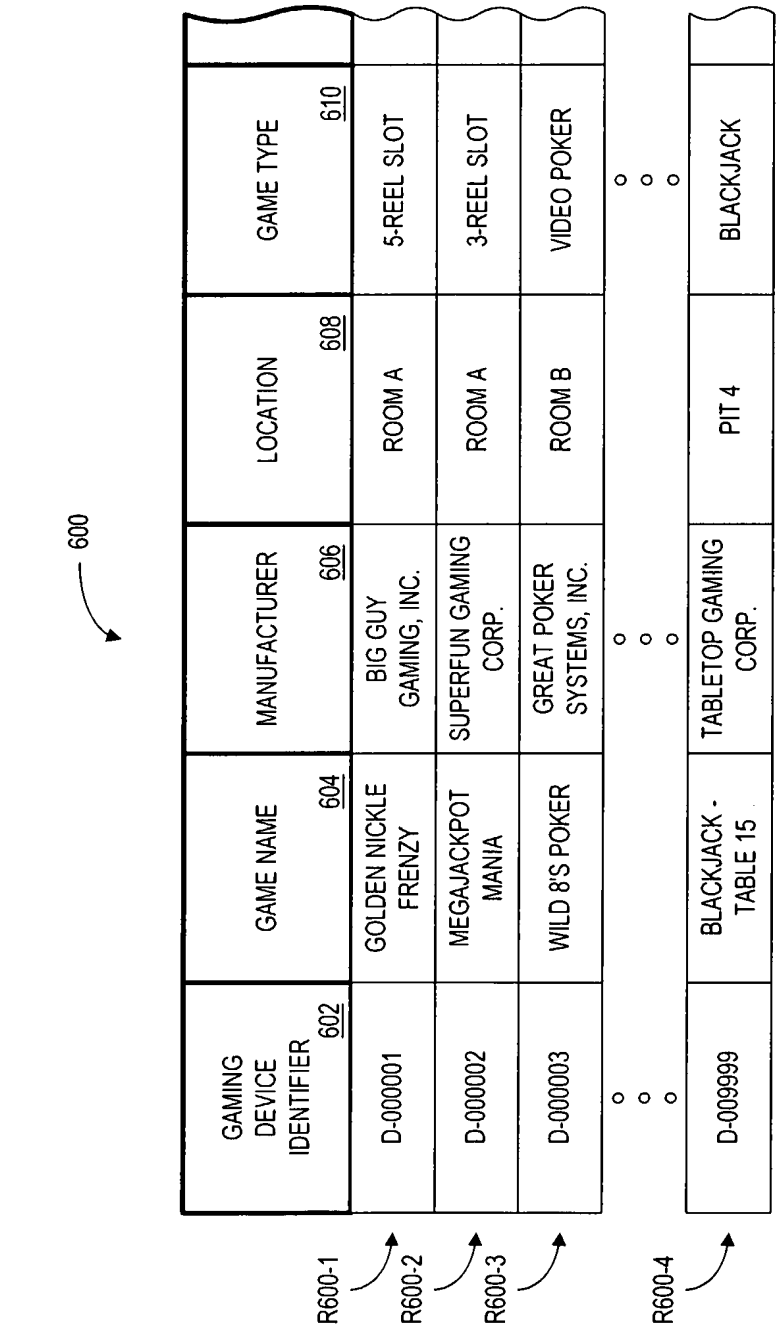

600
GAMING DEVICE IDENTIFIER 602
GAME NAME 604
MANUFACTURER 606
LOCATION 608
GAME TYPE 610
D-000001
GOLDEN NICKLE FRENZY
BIG GUY GAMING, INC.
ROOM A
5-REEL SLOT
D-000002
MEGAJACKPOT MANIA
SUPERFUN GAMING CORP.
ROOM A
3-REEL SLOT
D-000003
WILD 8'S POKER
GREAT POKER SYSTEMS, INC.
ROOM B
VIDEO POKER
D-009999
BLACKJACK - TABLE 15
TABLETOP GAMING CORP.
PIT 4
BLACKJACK
R600-1
R600-2
R600-3
R600-4

FIG. 6A

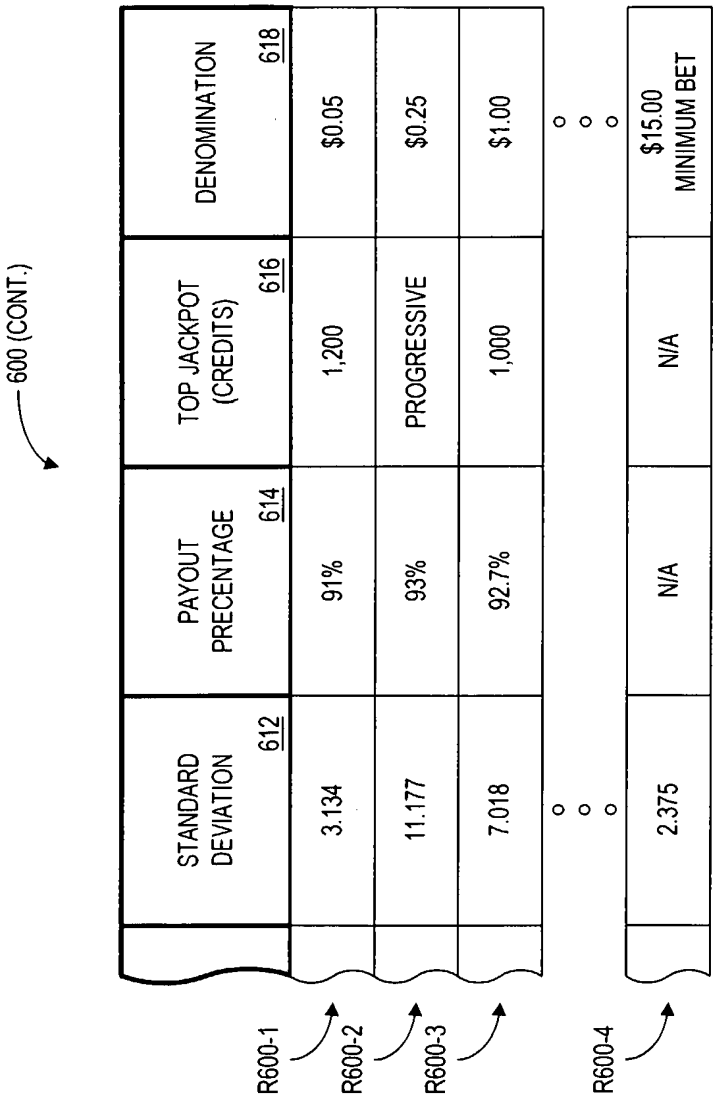

600 (CONT.)
STANDARD DEVIATION 612
PAYOUT PRECENTAGE 614
TOP JACKPOT (CREDITS) 616
DENOMINATION 618
3.134
91%
1,200
$0.05
11.177
93%
PROGRESSIVE
$0.25
7.018
92.7%
1,000
$1.00
2.375
N/A
N/A
$15.00 MINIMUM BET
R600-1
R600-2
R600-3
R600-4

FIG. 6B

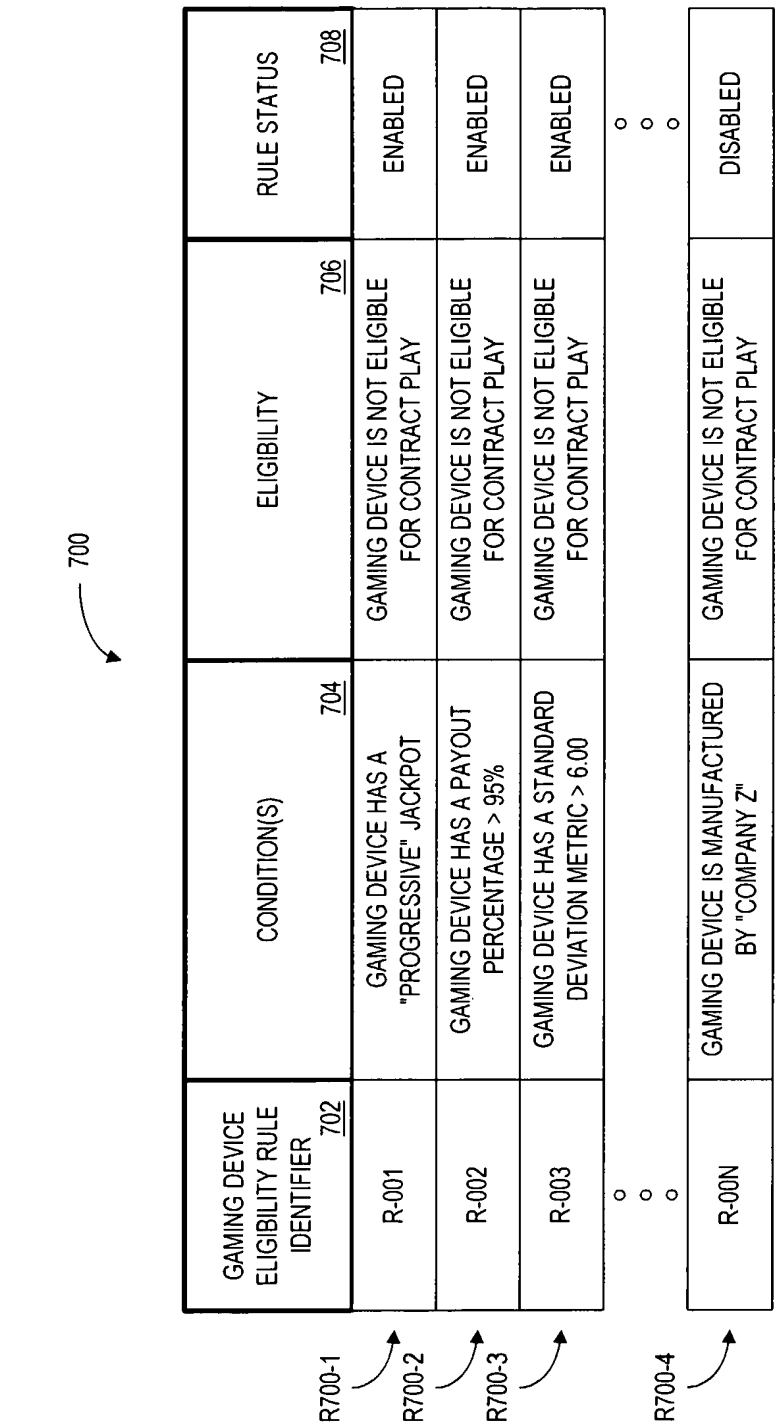

700
GAMING DEVICE ELIGIBILITY RULE IDENTIFIER 702
CONDITION(S) 704
ELIGIBILITY 706
RULE STATUS 708
R-001
GAMING DEVICE HAS A "PROGRESSIVE" JACKPOT
GAMING DEVICE IS NOT ELIGIBLE FOR CONTRACT PLAY
ENABLED
R-002
GAMING DEVICE HAS A PAYOUT PERCENTAGE > 95%
GAMING DEVICE IS NOT ELIGIBLE FOR CONTRACT PLAY
ENABLED
R-003
GAMING DEVICE HAS A STANDARD DEVIATION METRIC > 6.00
GAMING DEVICE IS NOT ELIGIBLE FOR CONTRACT PLAY
ENABLED
R-00N
GAMING DEVICE IS MANUFACTURED BY "COMPANY Z"
GAMING DEVICE IS NOT ELIGIBLE FOR CONTRACT PLAY
DISABLED
R700-1
R700-2
R700-3
R700-4

FIG. 7

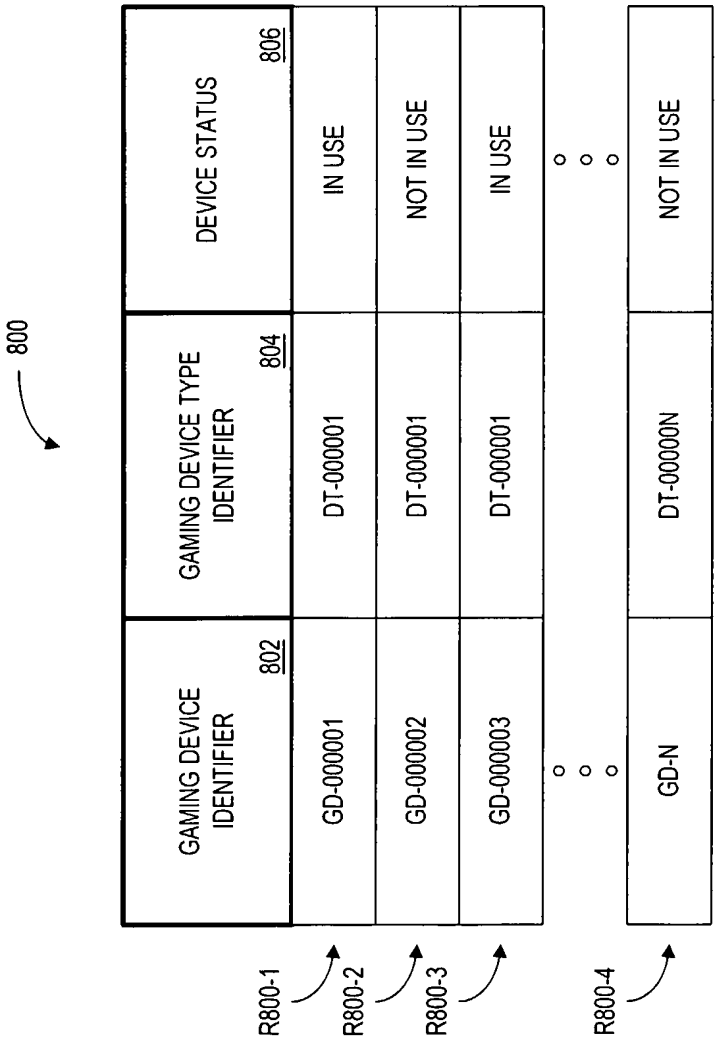

800
GAMING DEVICE IDENTIFIER 802
GAMING DEVICE TYPE IDENTIFIER 804
DEVICE STATUS 806
GD-000001
GD-000002
GD-000003
GD-N
DT-000001
DT-000001
DT-000001
DT-00000N
IN USE
NOT IN USE
IN USE
NOT IN USE
R800-1
R800-2
R800-3
R800-4

FIG. 8

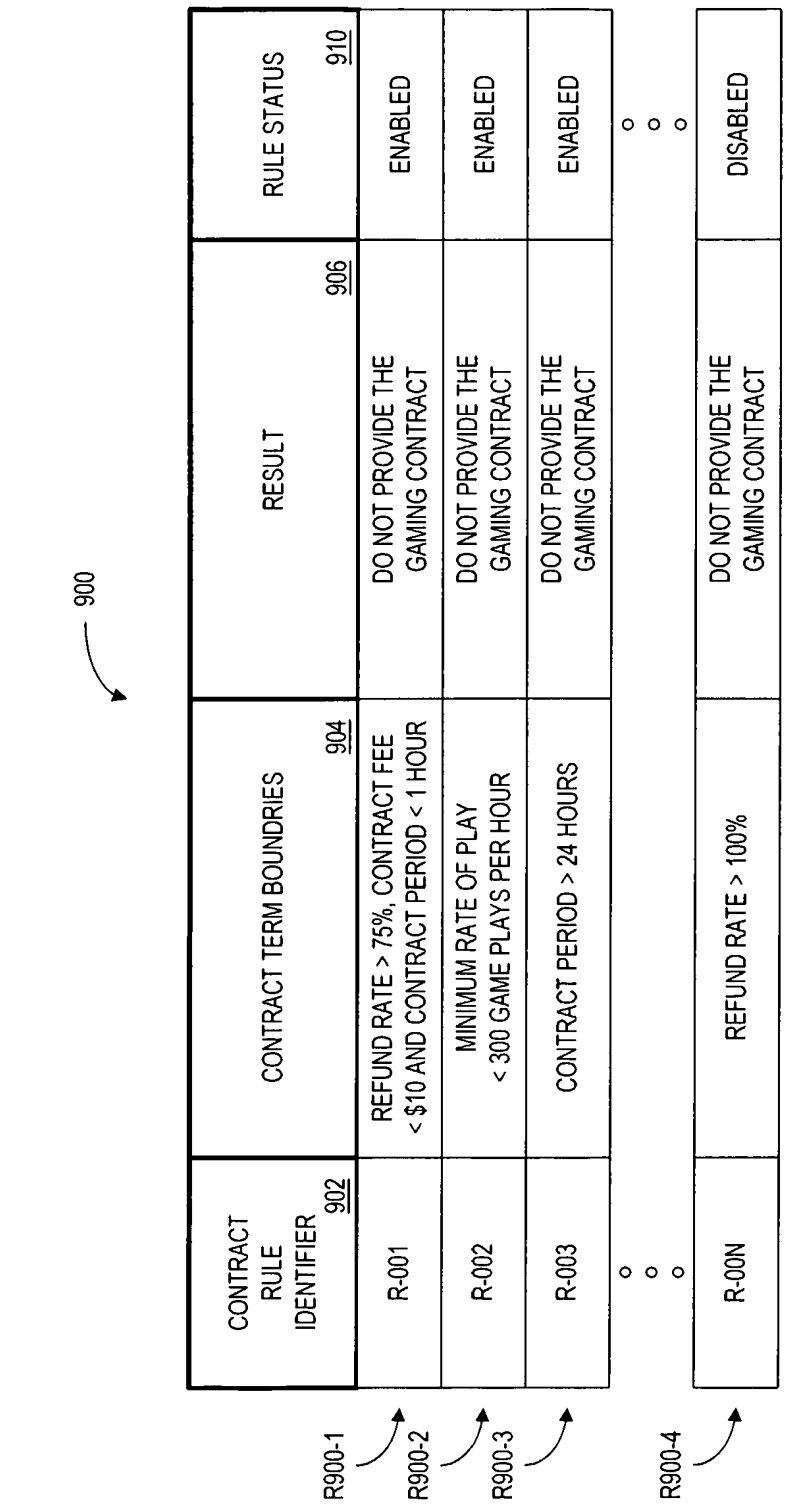

900
CONTRACT RULE IDENTIFIER 902
CONTRACT TERM BOUNDRIES 904
RESULT 906
RULE STATUS 910
R900-1
R-001
REFUND RATE > 75%, CONTRACT FEE < $10 AND CONTRACT PERIOD < 1 HOUR
DO NOT PROVIDE THE GAMING CONTRACT
ENABLED
R900-2
R-002
MINIMUM RATE OF PLAY < 300 GAME PLAYS PER HOUR
DO NOT PROVIDE THE GAMING CONTRACT
ENABLED
R900-3
R-003
CONTRACT PERIOD > 24 HOURS
DO NOT PROVIDE THE GAMING CONTRACT
ENABLED
R900-4
R-00N
REFUND RATE > 100%
DO NOT PROVIDE THE GAMING CONTRACT
DISABLED

FIG. 9

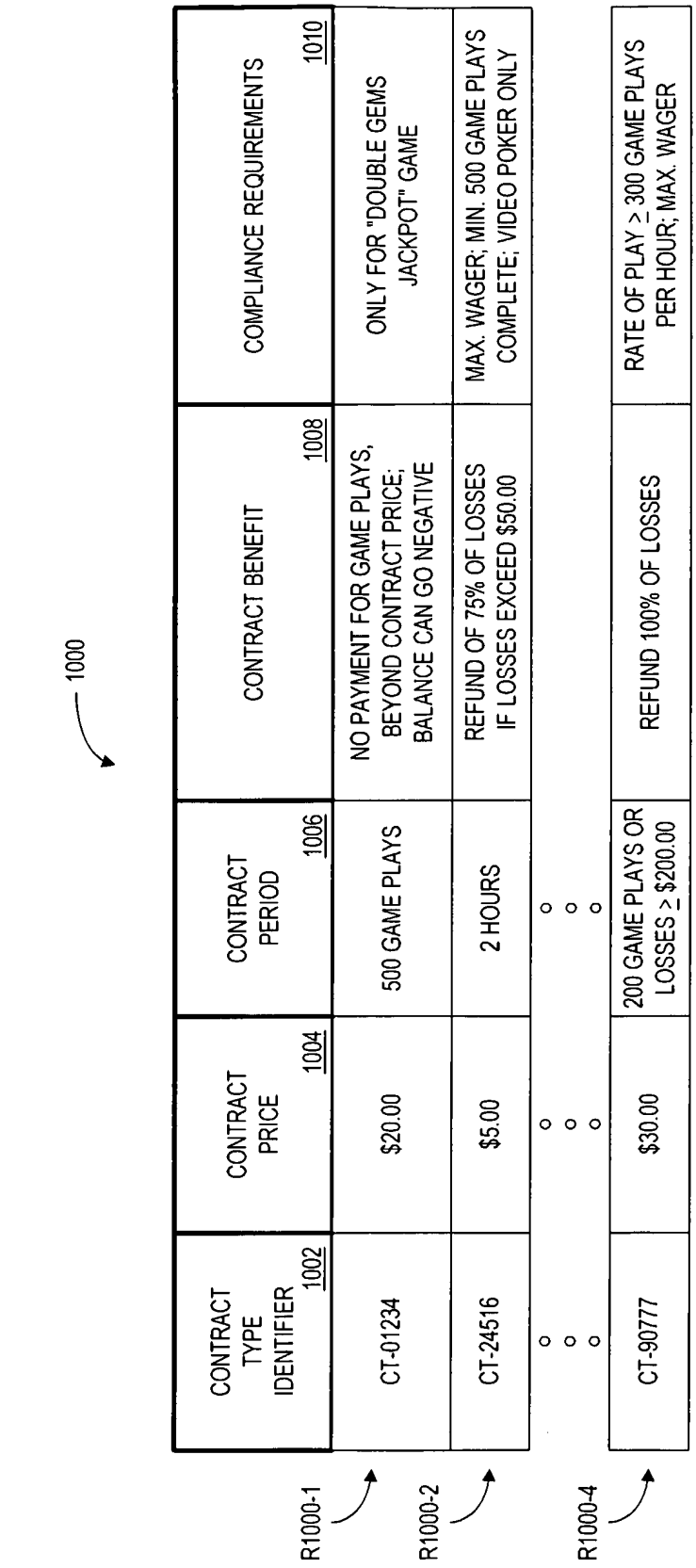

1000
CONTRACT TYPE IDENTIFIER 1002
CONTRACT PRICE 1004
CONTRACT PERIOD 1006
CONTRACT BENEFIT 1008
COMPLIANCE REQUIREMENTS 1010
CT-01234
$20.00
500 GAME PLAYS
NO PAYMENT FOR GAME PLAYS, BEYOND CONTRACT PRICE; BALANCE CAN GO NEGATIVE
ONLY FOR "DOUBLE GEMS JACKPOT" GAME
CT-24516
$5.00
2 HOURS
REFUND OF 75% OF LOSSES IF LOSSES EXCEED $50.00
MAX. WAGER; MIN. 500 GAME PLAYS COMPLETE; VIDEO POKER ONLY
CT-90777
$30.00
200 GAME PLAYS OR LOSSES ≥ $200.00
REFUND 100% OF LOSSES
RATE OF PLAY ≥ 300 GAME PLAYS PER HOUR; MAX. WAGER
R1000-1
R1000-2
R1000-4

FIG. 10

13 / 18

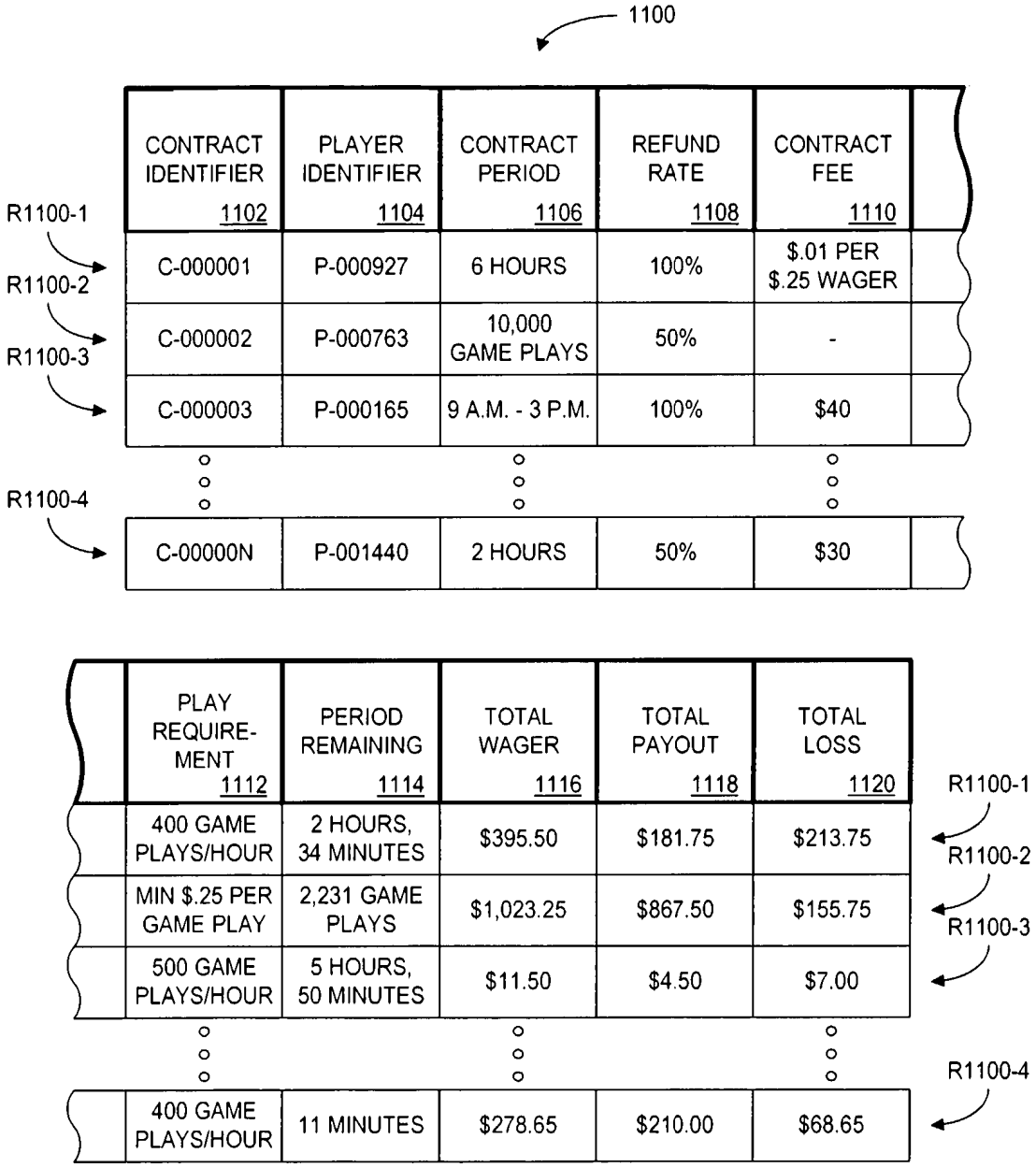

1100
CONTRACT IDENTIFIER 1102
PLAYER IDENTIFIER 1104
CONTRACT PERIOD 1106
REFUND RATE 1108
CONTRACT FEE 1110
R1100-1
R1100-2
R1100-3
R1100-4
C-000001
P-000927
6 HOURS
100%
$.01 PER $.25 WAGER
C-000002
P-000763
10,000 GAME PLAYS
50%
-
C-000003
P-000165
9 A.M. - 3 P.M.
100%
$40
C-00000N
P-001440
2 HOURS
50%
$30
PLAY REQUIRE-MENT 1112
PERIOD REMAINING 1114
TOTAL WAGER 1116
TOTAL PAYOUT 1118
TOTAL LOSS 1120
R1100-1
R1100-2
R1100-3
R1100-4
400 GAME PLAYS/HOUR
2 HOURS, 34 MINUTES
$395.50
$181.75
$213.75
MIN $.25 PER GAME PLAY
2,231 GAME PLAYS
$1,023.25
$867.50
$155.75
500 GAME PLAYS/HOUR
5 HOURS, 50 MINUTES
$11.50
$4.50
$7.00
400 GAME PLAYS/HOUR
11 MINUTES
$278.65
$210.00
$68.65

FIG. 11

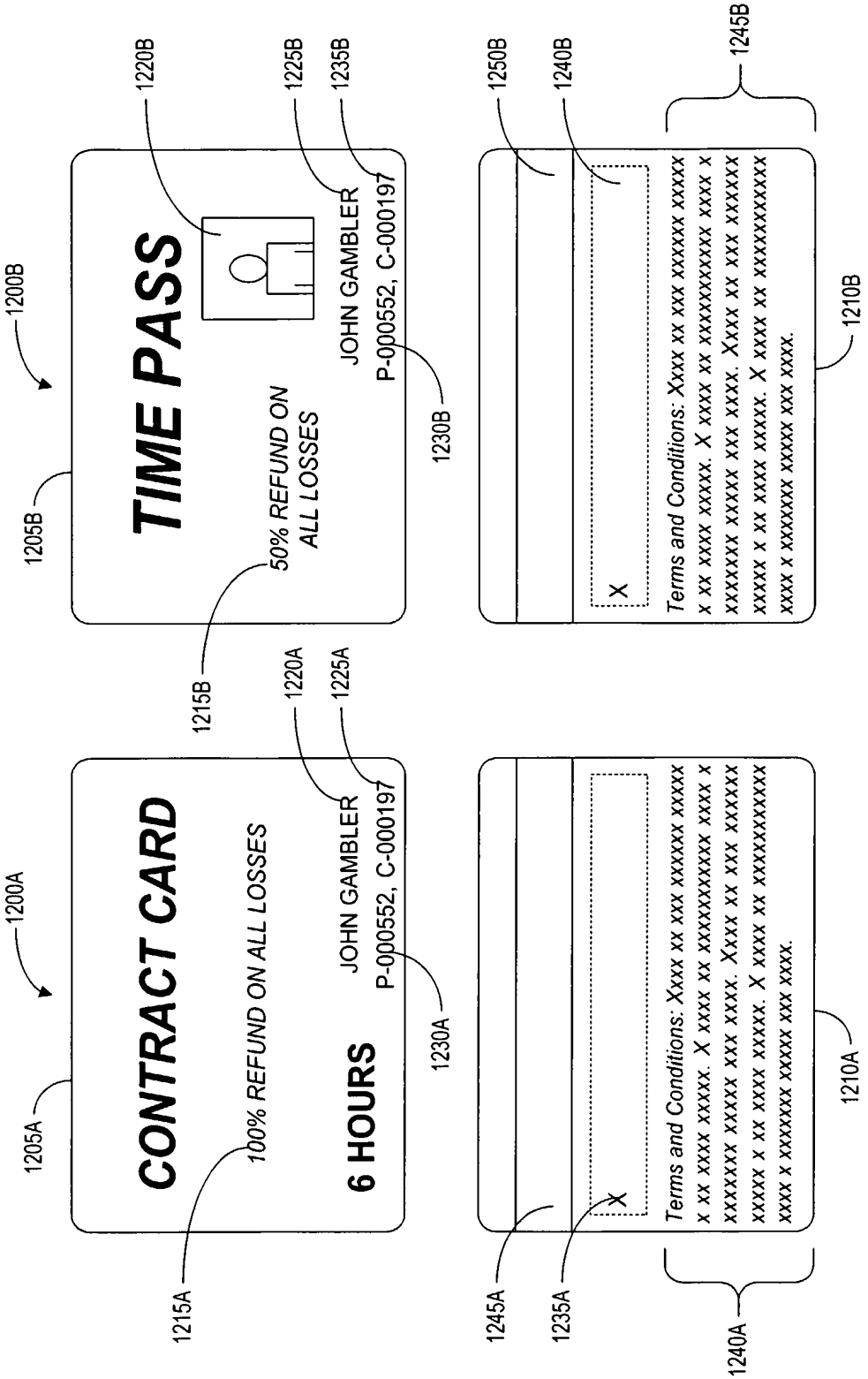

1200A
1205A
CONTRACT CARD
100% REFUND ON ALL LOSSES
6 HOURS
JOHN GAMBLER
P-000552, C-000197
1215A
1220A
1225A
1230A
1200B
1205B
TIME PASS
50% REFUND ON ALL LOSSES
JOHN GAMBLER
P-000552, C-000197
1215B
1220B
1225B
1230B
1235B
1240A
1245A
1235A
Terms and Conditions: Xxxx xx xxx xxxxx xxxxx x xx xxxx xxxxx. X xxxx xx xxxxxxxxxx xxxx x xxxxxxx xxxxx xxx xxxx. Xxxx xx xxx xxxxxx xxxxx x xx xxxx xxxxx. X xxxx xx xxxxxxxxxx xxxx x xxxxxxx xxxxx xxx xxxx.
1210A
1240B
1250B
1245B
Terms and Conditions: Xxxx xx xxx xxxxx xxxxx x xx xxxx xxxxx. X xxxx xx xxxxxxxxxx xxxx x xxxxxxx xxxxx xxx xxxx. Xxxx xx xxx xxxxxx xxxxx x xx xxxx xxxxx. X xxxx xx xxxxxxxxxx xxxx x xxxxxxx xxxxx xxx xxxx.
1210B

FIG. 12

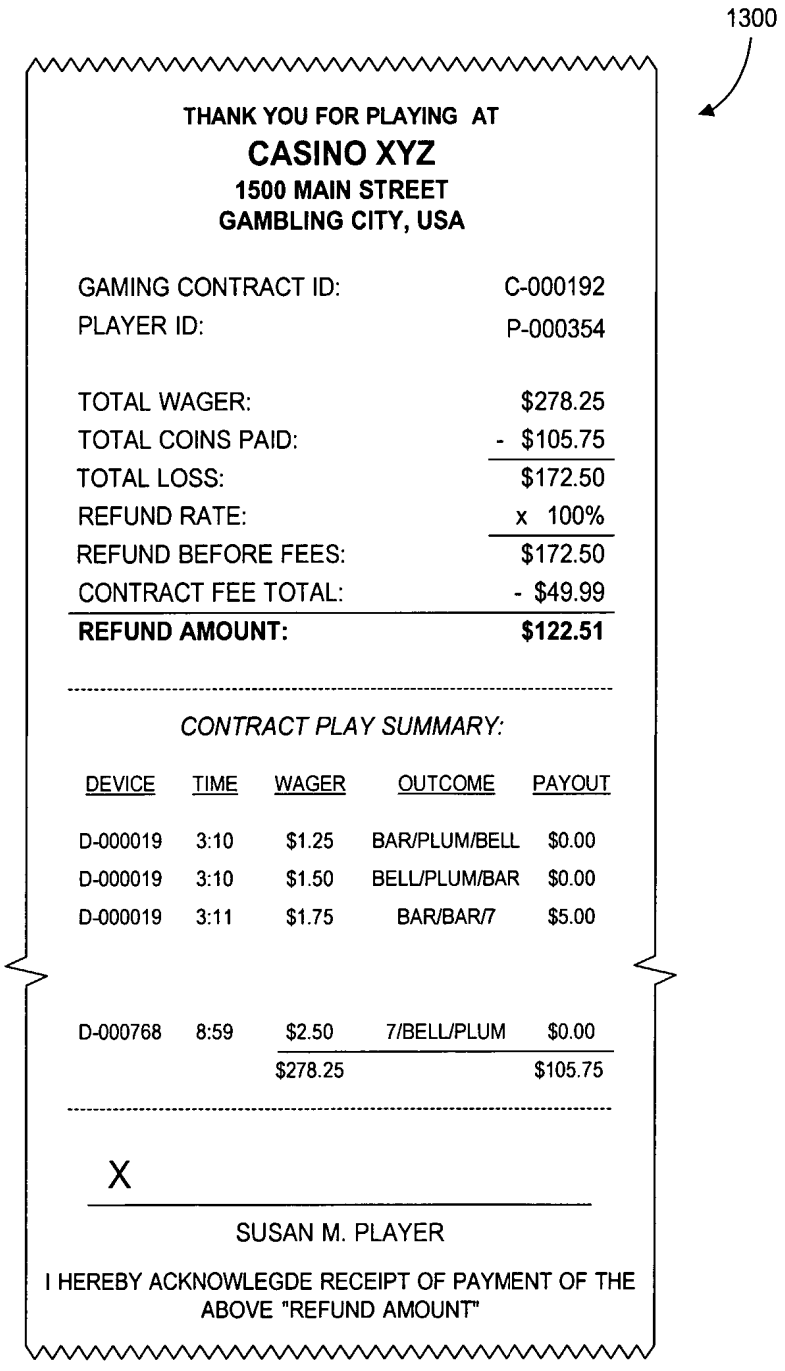

1300
THANK YOU FOR PLAYING AT
CASINO XYZ
1500 MAIN STREET
GAMBLING CITY, USA
GAMING CONTRACT ID: C-000192
PLAYER ID: P-000354
TOTAL WAGER: $278.25
TOTAL COINS PAID: - $105.75
TOTAL LOSS: $172.50
REFUND RATE: x 100%
REFUND BEFORE FEES: $172.50
CONTRACT FEE TOTAL: - $49.99
REFUND AMOUNT: $122.51
CONTRACT PLAY SUMMARY:
DEVICE TIME WAGER OUTCOME PAYOUT
D-000019 3:10 $1.25 BAR/PLUM/BELL $0.00
D-000019 3:10 $1.50 BELL/PLUM/BAR $0.00
D-000019 3:11 $1.75 BAR/BAR/7 $5.00
D-000768 8:59 $2.50 7/BELL/PLUM $0.00
$278.25 $105.75
X
SUSAN M. PLAYER
I HEREBY ACKNOWLEGDE RECEIPT OF PAYMENT OF THE ABOVE "REFUND AMOUNT"

FIG.13

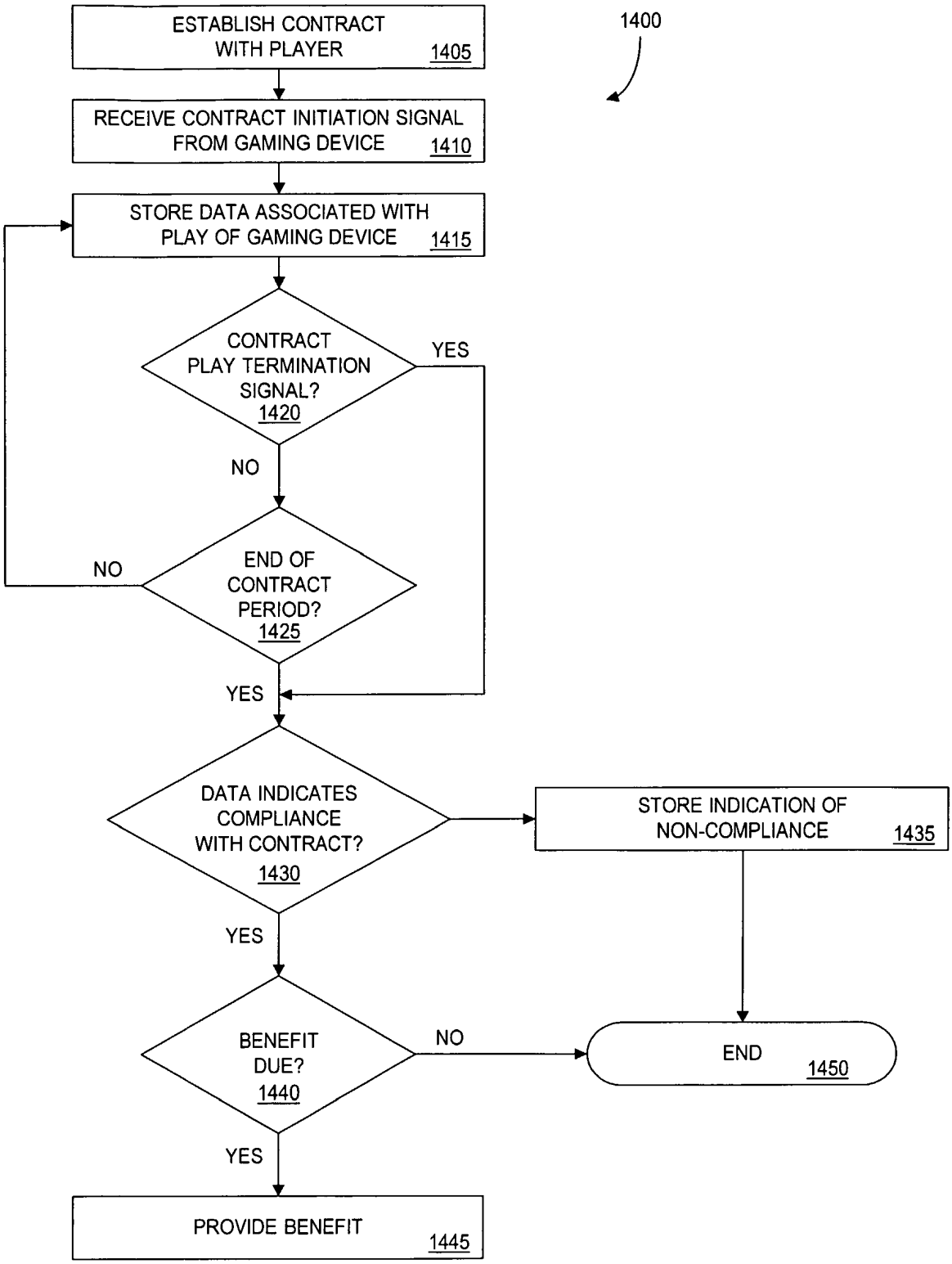

1400
ESTABLISH CONTRACT WITH PLAYER 1405
RECEIVE CONTRACT INITIATION SIGNAL FROM GAMING DEVICE 1410
STORE DATA ASSOCIATED WITH PLAY OF GAMING DEVICE 1415
CONTRACT PLAY TERMINATION SIGNAL? 1420
YES
NO
END OF CONTRACT PERIOD? 1425
NO
YES
DATA INDICATES COMPLIANCE WITH CONTRACT? 1430
STORE INDICATION OF NON-COMPLIANCE 1435
YES
BENEFIT DUE? 1440
NO
END 1450
YES
PROVIDE BENEFIT 1445

FIG. 14

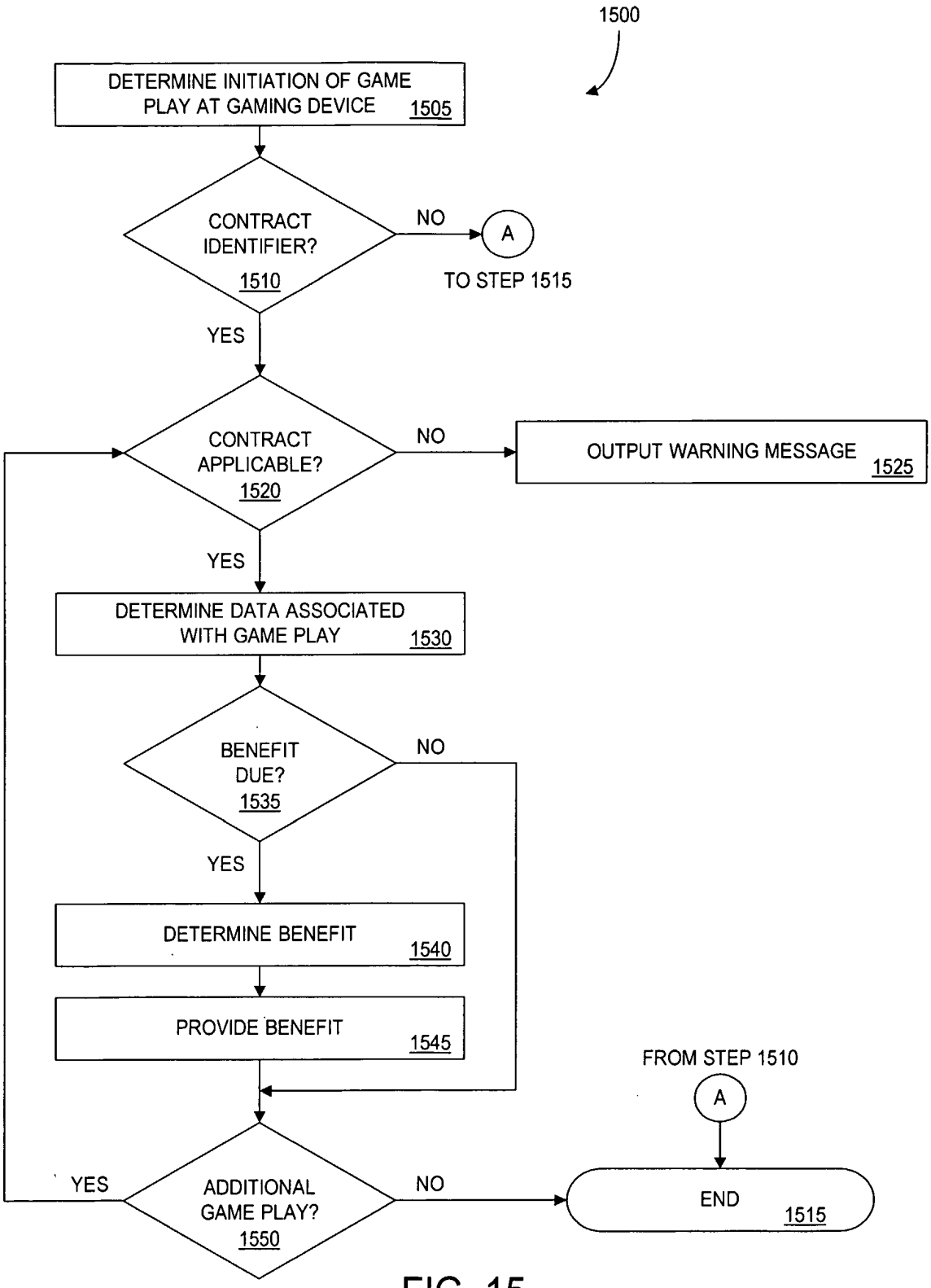

1500
DETERMINE INITIATION OF GAME PLAY AT GAMING DEVICE 1505
CONTRACT IDENTIFIER? 1510
NO
A
TO STEP 1515
YES
CONTRACT APPLICABLE? 1520
NO
OUTPUT WARNING MESSAGE 1525
YES
DETERMINE DATA ASSOCIATED WITH GAME PLAY 1530
BENEFIT DUE? 1535
NO
YES
DETERMINE BENEFIT 1540
PROVIDE BENEFIT 1545
FROM STEP 1510
A
YES
ADDITIONAL GAME PLAY? 1550
NO
END 1515

FIG. 15

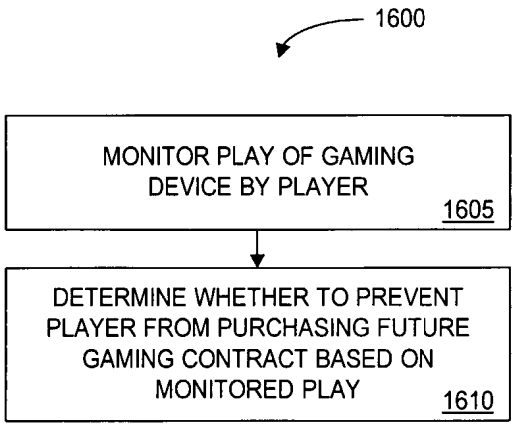

1600
MONITOR PLAY OF GAMING DEVICE BY PLAYER
1605
DETERMINE WHETHER TO PREVENT PLAYER FROM PURCHASING FUTURE GAMING CONTRACT BASED ON MONITORED PLAY
1610

FIG. 16

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 901 880 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
27.10.2021 Bulletin 2021/43

(21) Application number: **21165692.1**

(22) Date of filing: **18.12.2014**

(51) Int Cl.:
G06Q 20/32 *(2012.01)* ***G06Q 20/36*** *(2012.01)*
G06Q 20/40 *(2012.01)* ***G06Q 30/06*** *(2012.01)*
G06Q 20/18 *(2012.01)* ***G06Q 20/38*** *(2012.01)*
G07F 9/00 *(2006.01)* ***G07F 9/02*** *(2006.01)*

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(30) Priority: **18.12.2013 US 201361917936 P**
14.03.2014 US 201414214644

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
14828617.2 / 3 084 699

(71) Applicant: **PayRange Inc.**
Portland, OR 97220 (US)

(72) Inventors:
- **PATEL, Paresh K.**
Portland, Washington 97220 (US)
- **DOAN, Chau M.**
Beaverton, Oregon 97005 (US)
- **SOKOL, Christopher Michael**
Portland, Oregon 97212 (US)
- **FRAUENGLASS, Corin Premoe**
Seattle, Washington 98102 (US)

(74) Representative: **Mewburn Ellis LLP**
Aurora Building
Counterslip
Bristol BS1 6BX (GB)

Remarks:
•This application was filed on 29-03-2021 as a divisional application to the application mentioned under INID code 62.
•Claims filed after the date of filing of the application / after the date of receipt of the divisional application (Rule 68(4) EPC)

(54) MOBILE DEVICE-TO-MACHINE PAYMENT SYSTEMS

(57) Described herein is a mobile-device-to-machine payment system and method for facilitating a cashless transaction for purchase of at least one product or service by a user from a payment accepting unit.

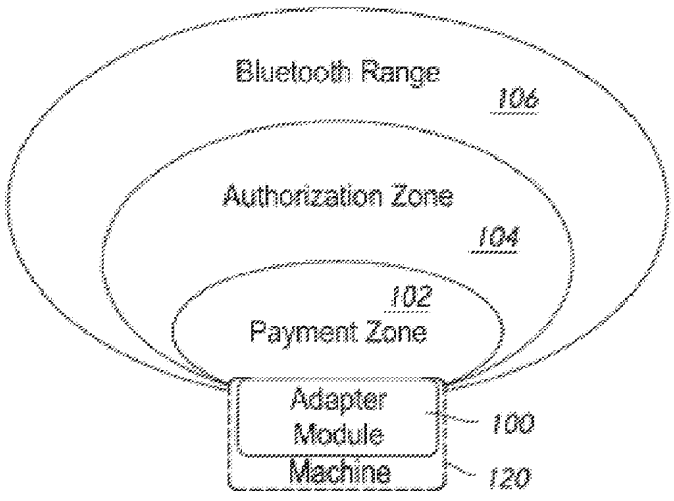


Figure 1

EP 3 901 880 A1

Petitioner Exhibit 1002-0476

Description

TECHNICAL FIELD

[0001] The present application relates to the field of payment processing systems, and in particular, to a mobile-device-to-machine payment processing system over a non-persistent network connection.

BACKGROUND

[0002] Vending machines (or "automatic retailing" machines), in the broadest sense, have been around for thousands of years. The first simple mechanical coin operated vending machines were introduced in the 1880s. Modern vending machines stock many different types of products including, but not limited to drinks (e.g., water, juice, coffee, and soda) and edible food products/items (e.g., snacks, candy, fruit, and frozen meals), as well as a wide variety of non-food items. In this fast paced world, vending machines are ubiquitous.

[0003] Vending machines are one type of "payment accepting unit" (payment accepting units are also referred to herein generically as "machines"). A payment accepting unit (or machine) is equipment that requires payment for the dispensing of products and/or services. In addition to vending machines, payment accepting units can also be other machines that require payment for the dispensing of a product and/or services including, but not limited to parking meters, toll booths, laundromat washers and dryers, arcade games, kiosks, photo booths, toll booths, transit ticket dispensing machines, and other known or yet to be discovered payment accepting units.

[0004] In using a payment accepting unit, a user will (1) approach the payment accepting unit, (2) determine from the face of the payment accepting unit the product (or service) he/she desires, (3) insert payment (e.g., coins, bills, or payment cards), and (4) input his/her selection into the payment accepting unit using a user interface (e.g., a series of buttons, a key pad, touch screen, or other input mechanism using, for example, the column and row at which a product is located). Based on the user's inputted selection, technology within the payment accepting unit provides the desired product (or service) to the user.

[0005] As the number of people with Internet-connected mobile devices proliferates, so does the variety of uses for such devices. Mobile payment is a logical extension. There is a large development effort around bringing mobile payment to the retail sector in an effort to not only provide options to the user, but also increased convenience.

SUMMARY

[0006] Disclosed herein is a payment processing system or, more specifically, a mobile-device-to-machine payment processing system over a non-persistent network connection with hands-free and manual (sometimes also herein called "swipe" or "swipe-to-pay" mode) modes.

[0007] In some implementations, a method of payment processing is performed at a device (e.g., the mobile device 150, Figures 5 and 21) with one or more processors, memory, and two or more communication capabilities. The method includes obtaining, from a payment module (e.g., the adapter module 100, Figures 5 and 20), advertised information via a first communication capability (e.g., a short-range communication technology/protocol such as BLE), where the advertised information at least includes an authorization code. The method includes sending, to a server (e.g., the server 130, Figures 5 and 22), at least the authorization code from the advertised information via a second communication capability distinct from the first communication capability (e.g., a long-range communication technology/protocol such as GSM, CDMA, or Wi-Fi). In response to sending at least the authorization code, the method includes obtaining, from the server, authorization information via the second communication capability, where the authorization information at least includes an authorization grant token. After obtaining the authorization information, the method includes detecting a trigger condition to perform a first transaction with a payment accepting unit (e.g., the payment accepting unit 120 (sometimes also herein called "machine 120") (Figures 5 and 19) such as a vending machine or kiosk for dispensing goods and/or services) associated with the payment module. In response to detecting the trigger condition, the method includes sending, to the payment module, the authorization grant token via the first communication capability.

[0008] In some implementations, a method of transmitting machine status information is performed at a device (e.g., the mobile device 150, Figures 5 and 21) with one or more processors, memory, and two or more communication capabilities. The method includes obtaining, from a payment module (e.g., the adapter module 100, Figures 5 and 20), advertised information via a first communication capability (e.g., the short-range communication technology/protocol such as BLE), where the advertised information at least includes status information indicating one or more states of at least one of a payment module and a payment accepting unit associated with the payment module. The method includes sending, to a server (e.g., the server 130, Figures 5 and 22), at least the status information from the advertised information via a second communication capability distinct from the first communication capability (e.g., the long-range communication technology/protocol such as GSM, CDMA, or Wi-Fi).

[0009] In some implementations, a method of payment processing acknowledgment information is performed at a payment module (e.g., the adapter module 100, Figures 5 and 20) coupled with a payment accepting unit (e.g., the payment accepting unit 120 (sometimes also herein called "machine 120") (Figures 5 and 19) such as a vend-

ing machine or kiosk for dispensing goods and/or services), the payment module including one or more processors, memory, and one or more first communication capabilities. The method includes obtaining, from the payment accepting unit, a first notification indicating completion of a first transaction performed by a first user of a first mobile device (e.g., the mobile device 150, Figures 5 and 21) at the payment accepting unit and an amount of the first transaction. In response to receiving the notification, the method includes: generating first transaction information based at least in part on the first notification; storing the generated first transaction information; and sending the generated first transaction information to the first mobile device via one of the one or more first communication capabilities (e.g., the short-range communication technology/protocol such as BLE). After sending the first transaction information to the first mobile device, the method includes: deleting the stored first transaction information generated for the first transaction performed by the first user of the first mobile device in accordance with a determination that first acknowledgement information has been received from the first mobile device within a predetermined time period; and maintaining the stored first transaction information generated for the first transaction performed by the first user of the first mobile device in accordance with a determination that the first acknowledgement information has not been received from the first mobile device within the predetermined time period.

[0010] In some implementations, a method of updating firmware is performed at a first device (e.g., the mobile device 150, Figures 5 and 21) with one or more processors, memory, and two or more communication capabilities. The method includes obtaining, from a payment module (e.g., the adapter module 100, Figures 5 and 20), advertised information via a first communication capability (e.g., the short-range communication technology/protocol such as BLE), where the advertised information at least includes a current firmware version of the payment module. In accordance with a determination that the current firmware version of the payment module satisfies one or more predefined firmware criteria (i.e., indicating that the payment module's firmware needs updating), the method includes sending, to the payment module, firmware update information via the first communication capability, where the firmware update information includes one or more data packets for updating the current firmware version of the payment module.

[0011] In some implementations, a device (e.g., the adapter module 100 (Figures 5 and 20), the mobile device 150 (Figures 5 and 21), the server 130 (Figures 5 and 22), or a combination thereof) includes one or more processors and memory storing one or more programs for execution by the one or more processors, the one or more programs include instructions for performing, or controlling performance of, the operations of any of the methods described herein. In some implementations, a non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which, when executed by a device (e.g., the adapter module 100 (Figure 20), the mobile device 150 (Figure 21), the server 130 (Figure 22), or a combination thereof) with one or more processors, cause the computer system to perform, or control performance of, the operations of any of the methods described herein. In some implementations, a device (e.g., the adapter module 100 (Figure 20), the mobile device 150 (Figure 21), the server 130 (Figure 22), or a combination thereof) includes means for performing, or controlling performance of, the operations of any of the methods described herein.

[0012] The subject matter described herein is particularly pointed out and distinctly claimed in the concluding portion of this specification. Objectives, features, combinations, and advantages described and implied herein will be more readily understood upon consideration of the following detailed description of the invention, taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013]

Figure 1 is a schematic diagram that shows three zones: a "communication zone" (e.g., Bluetooth range), an "authorization zone," and a "payment zone" in accordance with some implementations.

Figure 2 is a schematic diagram that shows the three zones of Figure 1 with multiple users therein in accordance with some implementations.

Figure 3 is a table that illustrates the hands-free credit or alert user principle in accordance with some implementations.

Figure 4 is a flow chart showing the logging received signal strength indicator (RSSI) information in accordance with some implementations.

Figure 5 is a block schematic that shows elements of the payment processing system including, but not limited to, the adapter module, the machine, the mobile device, and servers, as well as communications therebetween in accordance with some implementations.

Figure 6 is a block schematic that shows three areas of encryption used (each is bi-directional) between the adapter module, the machine, the mobile device, and/or servers in accordance with some implementations.

Figure 7 is a block diagram that shows communications, messaging, vending sequence, and purchase flow between the adapter module, the mobile device, and a system management server in accordance with some implementations.

Figure 8A is a schematic process flow diagram that shows additional elements and features of the payment processing system (e.g., communications, messaging, vending sequence, and purchase flow) when the user enters the "communication zone" (e.g., Bluetooth range) in accordance with some implementations.

Figure 8B is a schematic process flow diagram that shows additional elements and features of the payment processing system (e.g., communications, messaging, vending sequence, and purchase flow) when the user enters the "authorization zone" in accordance with some implementations.

Figure 8C is a schematic process flow diagram that shows additional elements and features of the payment processing system (e.g., communications, messaging, vending sequence, and purchase flow) when the user enters the "payment zone" and, in particular, detailing a hands-free mode embodiment and a swipe mode embodiment in accordance with some implementations.

Figure 8D is a schematic process flow diagram that shows additional elements and features of the payment processing system (e.g., communications, messaging, vending sequence, and purchase flow) in a vending transaction including a loop for multiple transactions in accordance with some implementations.

Figure 8E is a schematic process flow diagram that shows additional elements and features of the payment processing system (e.g., communications, messaging, vending sequence, and purchase flow) in the login mode in accordance with some implementations.

Figure 8F is a schematic process flow diagram that shows additional elements and features of the payment processing system (e.g., communications, messaging, vending sequence, and purchase flow) during boot-up of the adapter module in accordance with some implementations.

Figure 8G is a schematic process flow diagram that shows additional elements and features of the payment processing system (e.g., communications, messaging, vending sequence, and purchase flow) during an account check/update process in accordance with some implementations.

Figures 9A-9E are flow charts that show example steps and features of the payment processing system (e.g., communications, messaging, vending sequence, and purchase flow) in accordance with some implementations.

Figures 10A-10D show a mobile device with a graphical representation of a mobile application shown thereon, the mobile application being used as part of the mobile-device-to-machine payment processing system in accordance with some implementations.

Figure 11 is a perspective view of the in-line dongle adapter module in accordance with some implementations.

Figure 12 is a front plan view of the in-line dongle adapter module of Figure 11 in accordance with some implementations.

Figure 13 is a back plan view of the in-line dongle adapter module of Figure 11 in accordance with some implementations.

Figure 15 is a first end view of a connector receptacle of the in-line dongle adapter module of Figure 11 in accordance with some implementations.

Figure 16 is a second end view of a connector receptacle of the in-line dongle adapter module of Figure 11 in accordance with some implementations.

Figure 17 is a perspective view taken from the first end of the in-line dongle adapter module of Figure 11, the connectors and cables between which the in-line dongle adapter module is inserted being shown in broken lines for illustrative purposes in accordance with some implementations.

Figure 18 is a perspective view taken from the second end of the in-line dongle adapter module of Figure 11, the connectors and cables between which the in-line dongle adapter module is inserted being shown in broken lines for illustrative purposes in accordance with some implementations.

Figure 19 is a perspective view of the in-line dongle adapter module of Figure 11 within a vending machine in accordance with some implementations.

Figure 20 is a block diagram of an adapter module in accordance with some implementations.

Figure 21 is a block diagram of a mobile device in accordance with some implementations.

Figure 22 is a block diagram of a server in accordance with some implementations.

Figure 23 is a schematic flow diagram of a process for authenticating a user to perform a transaction in the payment processing system in accordance with some implementations.

Figure 24A is a block diagram of a packet of information broadcast by the payment module (sometimes also herein called the "adapter module") in accordance with some implementations.

Figure 24B is a block diagram of an authorization request in accordance with some implementations.

Figure 24C is a block diagram of an authorization grant token in accordance with some implementations.

Figure 24D is a block diagram of transaction information generated by the payment module in accordance with some implementations.

Figure 25A is a schematic flow diagram of a process for processing acknowledgment information in the payment processing system in accordance with some implementations.

Figure 25B is a schematic flow diagram of a process for processing interrupted transactions in the payment processing system in accordance with some implementations.

Figures 26A-26C show schematic flow diagrams of processes for updating firmware of the payment module in the payment processing system in accordance with some implementations.

Figures 27A-27C illustrate a flowchart diagram of a method of payment processing in accordance with some implementations.

Figures 28A-28B illustrate a flowchart diagram of a method of transmitting machine status information in accordance with some implementations.

Figures 29A-29C illustrate a flowchart diagram of a method of payment processing acknowledgment in accordance with some implementations.

Figures 30A-30D illustrate a flowchart diagram of a method of updating firmware in accordance with some implementations.

Figure 31A illustrates a schematic flow diagram of a process for providing a representation of a machine event at a mobile device in accordance with some implementations.

Figure 31B is a schematic flow diagram of a process for processing acknowledgment information in the payment processing system in accordance with some implementations.

Figures 32A-32D illustrate example user interfaces for providing a representation of a machine event at a mobile device in accordance with some implementations.

Figures 33A-33B illustrate a flowchart diagram of a method of presenting representations of payment accepting unit events in accordance with some implementations.

Figure 34A illustrates a block diagram of an offline-payment operated machine in accordance with some implementations.

Figure 34B illustrates signals sampled by the payment module in accordance with some implementations.

Figures 35A-35B illustrate a flowchart diagram of a method of retrofitting an offline-payment operated machine to accept electronic payments in accordance with some implementations.

Figure 36 illustrates a flowchart diagram of a method of enabling a payment

Figure 37 is a block diagram of a device for retrofitting a payment accepting unit to accommodate a plurality of payment peripherals in accordance with some implementations.

Figure 38 is a schematic flow diagram of a payment peripheral registration process in accordance with some implementations.

Figures 39A-39B illustrate a schematic flow diagram of a payment process in accordance with some implementations.

Figures 40A-40D illustrate a flowchart diagram of a method of retrofitting a payment accepting unit to accommodate a plurality of payment peripherals in accordance with some implementations.

[0014] Like reference numerals refer to corresponding parts throughout the several views of the drawings.

DESCRIPTION OF EMBODIMENTS

[0015] Disclosed herein is a payment processing system or, more specifically, a mobile-device-to-machine payment processing system for processing transactions over a non-persistent network connection. The mobile-device-to-machine payment processing system disclosed herein focuses on the unattended retail space (e.g., a payment accepting unit 120, sometimes also herein called a "machine 120"). More specifically, the mobile-device-to-machine payment processing system disclosed herein allows a user (having a mobile device 150

with a mobile application 140 thereon) to make a cashless purchase from a payment accepting unit 120 (having an adapter module 100 associated therewith).

[0016] The mobile-device-to-machine payment processing system described herein can be implemented with one or more of the following features: easy installation feature, a non-persistent network connection feature; a manual (swipe to pay) mode feature; a hands-free mode feature; and a multiple vending transactions (multi-vend) feature.

[0017] Easy Installation: Installation is very easy, requires no tools, requires no configuration, and takes as little as 30 seconds. This is accomplished by using an adapter module 100 (sometimes also herein called "payment module 100") such as an in-line dongle (a hardware device with software thereon) design for in-line insertion within a multi-drop bus (MDB) of a payment accepting unit 120 (e.g., a vending machine) (sometimes also herein called 'the machine 120"). Installation is as simple as "powering down" (turning off) the machine 120, identifying the "wire" that connects with a payment receiving mechanism (e.g., the coin mechanism), disconnecting the wire (so that there are two loose ends, such as a male connection end or adapter of an MDB and a female connection end or adapter of an MDB), plugging (inserting) the adapter module 100 in serial ("in-line") with the wire (e.g., connecting the MDB female adapter to a male adapter of the adapter module 100 and connecting the MDB male adapter to a female adapter of the adapter module 100), tucking the wire and the installed adapter module 100 back into position, and "powering up" (turning on) the machine 120. Most vending machines made since 1995 have this industry standard MDB technology that would allow this easy 30-second installation. On machines without MDB technology, the adapter module 100 can be configured or designed to work with other serial protocols or activate a switch. In essence the adapter module 100 simulates establishing payment on payment accepting unit 120 in much the same manner as other alternative forms of payment (e.g., cash).

[0018] Non-persistent Network Connection: Although payment accepting units (or "machines") that accept only cash (e.g., paper currency and coins) may not require a connection (persistent or non-persistent) to a network, traditional payment accepting units that accept cashless payments (e.g., credit cards, debit cards, and alternative mobile device payment methods using, for example, smart phones) require a persistent connection to a network (wired or wireless) to facilitate the cashless payments. In other words, without a persistent (ongoing or accessible on demand) network connection, traditional payment accepting units cannot accept cashless payments. Most traditional payment accepting units that accept cashless payments include the technology to accomplish this persistent network connection that allows them to connect to a remote server. If the network connection to a traditional machine is temporarily interrupted, cashless payments will be temporarily unavailable. If the machine is located in a location where no network connection is available, cashless payments is not possible. In addition to using a mobile device 150 as an intermediary between the payment accepting units 120 and the server 130, the mobile-device-to-machine payment processing system described herein minimizes (i.e., the manual mode) or eliminates (i.e., the hands-free mode) user interaction with the mobile device 150. Further, in some implementations, the mobile-device-to-machine payment processing system described herein facilitates the acceptance of cashless payments without requiring any network connection near the payment accepting unit 120. In some implementations, when the mobile-device-to-machine payment processing system described herein is located in a remote location where network connection is unavailable, the mobile-device-to-machine payment processing system, therefore, can still accept cashless payments.

[0019] Manual (Swipe-to-Pay) Mode: Using a "swipe-to-pay" feature (or just "swipe") refers to a user's action implemented on his/her mobile device 150 where he/she quickly brushes his/her finger (or other pre-determined interaction) on the mobile device's touch screen 152 (Figures 10A-10D) or other input devices associated with the mobile device 150. From the user's perspective, when the user is within range, a pre-installed mobile application 140 automatically connects to the payment accepting unit 120 (e.g., a vending machine). The mobile application 140 might display (on the touch screen 152) a prepaid balance that the user "swipes" to transfer payment to the payment accepting unit 120. The user could observe the transferred funds on the touch screen 152 of the mobile device 150 and/or on the display 122, 124 (Figure 19) of the payment accepting unit 120. The transaction is completed just as if cash was inserted in the machine 120 with the user inputting his selection on the payment accepting unit 120 and the payment accepting unit 120 dispensing the product or service. After the selection is made, the change is returned to the mobile device 150 and this may be shown on the touch screen 152 of the mobile device 150.

[0020] Hands-Free Mode: A "hands-free pay" feature (or just "hands-free") would most likely be used with "favorite" payment accepting units 120 (e.g., a frequently used vending machine at a user's work or school). From the user's perspective, he/she would approach the favorite payment accepting unit 120 and notice that the display 122, 124 (Figure 19) of the payment accepting unit 120 shows funds available, he/she would select the product or service using the payment accepting unit's input mechanisms (e.g., buttons 126 or a touch screen display 124 shown in Figure 19), and he/she would retrieve dispensed services or products. It would be that simple. More specifically, when the user is within range, a pre-installed mobile application 140 automatically connects to the payment accepting unit 120 (e.g., a vending machine). The user may leave the mobile device 150 in a pocket, purse, briefcase, backpack, or other carrier. As

the user approaches the payment accepting unit 120 and is in approximately "arm's-length" distance (e.g., 3 to 5 feet) of the payment accepting unit 120, the user could observe the transferred funds on the display 122, 124 (Figure 19) of the payment accepting unit 120. The transaction is completed just as if cash was inserted into the payment accepting unit 120 with the user inputting his/her selection on the payment accepting unit 120 and the payment accepting unit 120 dispensing the product or service. After the selection is made, the change is returned to the mobile device 150. Figure 3 details when the hands-free mode would be available.

[0021] Multiple Vending Transactions (Multi-Vend): Both the manual and hands-free modes could be used multiple times in sequence (implemented, for example, as a loop) so that a user may make multiple purchases. After making his/her first selection and receiving his product (or service), the user would observe that additional funds were available on the display 122, 124 (Figure 19) on the payment accepting unit 120. He/she could make another selection (or multiple selections) and receive additional product(s) (or service(s)). More specifically, the display 122, 124 (Figure 19) may reset as if the transaction is complete, but then, because the user is still standing in range, the mobile application 140 would send another credit to the payment accepting unit 120, allowing for a second purchase. When the user walks away, the system clears (e.g., returns unused funds to the application 140 on the mobile device 150).

[0022] The features described above, alone or in combination with other features described herein will revolutionize the hundred billion dollar automated retail industry. The hardware is very low cost and there are no reoccurring fees because no cellular connection is required on the machine 120. Using the mobile-device-to-machine payment processing system described herein, operators of machines 120 can increase frequency of visits by purchasers and items sold with each visit.

[0023] The mobile-device-to-machine payment processing system described herein may be implemented as an apparatus, system, and/or method for enabling payments to a machine 120 via a mobile device 150. The mobile-device-to-machine payment processing system may be better understood with reference to the drawings, but the shown mobile-device-to-machine payment processing system is not intended to be of a limiting nature.

DEFINITIONS

[0024] Before describing the mobile-device-to-machine payment processing system and the figures, some of the terminology should be clarified. Please note that the terms and phrases may have additional definitions and/or examples throughout the specification. Where otherwise not specifically defined, words, phrases, and acronyms are given their ordinary meaning in the art. The following paragraphs provide some of the definitions for terms and phrases used herein.

[0025] Adapter Module 100: As shown in Figures 1 and 2, the adapter module 100 (sometimes also herein called the "payment module 100") is a physical device that is installed in a machine 120 (a payment accepting unit 120). The shown adapter module 100 is an in-line dongle (a hardware device with software thereon) device that may be inserted in-line within a multi-drop bus (MDB) of a machine 120. The adapter module 100 bridges the communication between the machine 120 and a mobile device 150. Although described as a unique component, it should be noted that the adapter module 100 could be implemented as a plurality of devices or integrated into other devices (e.g., components of a machine 120). In its unique component form, the adapter module 100 can be easily inserted into a machine 120 so that the machine 120 is able to perform new features with the assistance of the adapter module 100. Figure 20 shows components associated with the adapter module 100. As shown in Figure 20, the communications unit 770 of the adapter module 100 includes short-range communication capability 776 (e.g., Bluetooth mechanisms).The shown example may be divided into multiple distinct components that are associated with each other or the example may be incorporated into or drawn from other technology (e.g., a computer or a payment accepting unit) as long as the components are associated with each other.

[0026] Mobile Device 150 and Application 140 (also referred to as a "mobile application," "mobile app," or "app"): In general, a mobile device 150 may be a user's personal mobile device 150. The mobile device 150 (with a mobile application 140 thereon) acts as a communication bridge between the adapter module 100 (associated with a payment accepting unit 120) and the server 130. The mobile device 150 and the application 140, however, are not "trusted" in that the communications (transmissions) it passes are encrypted. Encrypted (secured) communications are undecipherable (unencryptable, unreadable, and/or unuseable) by the mobile device 150. This keeps the communications passed between the adapter module 100 and the server 130 secured and safe from hacking. Mobile devices include, but are not limited to smart phones, tablet or laptop computers, or personal digital assistants (PDAs), smart cards, or other technology (e.g., a hardware-software combination) known or yet to be discovered that has structure and/or capabilities similar to the mobile devices described herein. The mobile device 150 preferably has an application (e.g., the application 140) running on it. The term "app" is used broadly to include any software program(s) capable of implementing the features described herein. Figures 10A-10D show user interfaces for the application 140 displayed by the mobile device 150. It should be noted that the phrase "mobile device" can be assumed to include the relevant app unless specifically stated otherwise. Similarly, it should be noted that an "app" can be assumed to be running on an associated mobile device unless specifically stated otherwise. Figure 21 shows

components associated with the mobile device 150. The shown example may be divided into multiple distinct components that are associated with each other or the example may be incorporated into or drawn from other technology (e.g., the cell phone itself) as long as the components are associated with each other.

[0027] Payment Accepting Unit 120 (or Machine 120): A payment accepting unit 120 (or the machine 120) is equipment that requires payment for the dispensing of an product and/or service. Payment accepting units 120 may be vending machines, parking meters, toll booths, laundromat washers and dryers, arcade games, kiosks, photo booths, toll booths, transit ticket dispensing machines, and other known or yet to be discovered payment accepting units 120. Some payment accepting units 120 can accept cashless payments (payments other than cash (paper currency and coins)) by accepting payment from, for example, credit cards, debit cards, and mobile devices.

[0028] Network Connections: For purposes of this discussion, a persistent network connection is a wired or wireless communications connection that is ongoing (e.g., a dedicated connection, a dedicated online connection, and/or a hardwired connection) or accessible on demand (e.g., the ability for the machine to make a temporary connection to a server or the ability for the user to contact a server from his mobile device). Typically the persistent network connection has been conducted over "long-range communication technology" or "long-range communication protocol" (e.g., hardwired, telephone network technology, cellular technology (e.g., GSM, CDMA, or the like), Wi-Fi technology, wide area network (WAN), local area network (LAN), or any wired or wireless communication technology over the Internet that is known or yet to be discovered). Traditionally, machines that accept payment other than cash require a persistent (ongoing or accessible on demand) connection to a network to facilitate payment. This is true for machines that accept, for example, credit cards and debit cards. The payment accepting units 120 described herein do not require a traditional persistent network connection. The user's mobile device 150 acts as a communication bridge between the adapter module 100 and the server 130. Communications between user mobile devices 150 and the servers (e.g., a system management server 130 and/or a funding source server 160) take place using long-range communication technology. Communications between user mobile devices 150 and the adapter module 100 of the payment accepting unit 120 take place using "short-range communication technology" or "short-range communication protocol" (e.g., Bluetooth (such as Bluetooth 4.0, Bluetooth Smart, Bluetooth Low Energy (BLE)), near-field communication (NFC), Ultra Wideband (UWB), radio frequency identification (RFID), infrared wireless, induction wireless, or any wired or wireless technology that could be used to communicate a small distance (approximately a hundred feet or closer) that is known or yet to be discovered). Therefore, neither the adapter module 100 nor the payment accepting unit 120 requires a traditional persistent long-range wireless network connection. The communications technology shown in the figures may be replaced with alternative like communications technology and, therefore, specific shown communications technologies are not meant to be limiting. For example, Wi-Fi technology could be replaced with another long-range communication technology.

[0029] Server: A server is the host processing server that may be operated by the company running the payment processing system. For each user, the server 130 preferably maintains at least one "virtual wallet" having at least one "balance" (which can be $0) of designated funds for which the server 130 keeps an accounting. The balance may represent, for example, "cash" or it may be a "promotional value" that represents funds that may be spent under certain circumstances. If these funds begin to be depleted, the user may be notified (e.g., via the application 140 on the mobile device 150) that additional funds need to be designated and/or transferred. Alternatively, funds from other sources (e.g., the funding source server 160) may be automatically transferred to restore a predetermined balance. The balance may also be increased based on a promotion (e.g., points earned or coupons). As shown in Figure 22, the server includes appropriate processors 950, memory 960 (which would keep an accounting of the user's balance in a manner similar to a gift card), and communication systems 970. As shown in Figure 22, the communications unit 970 of the server 130 includes long-range communication capability 972 (e.g., cellular technology and/or Wi-Fi mechanisms). The server 130 also includes a security unit 955 for encrypting and decrypting messages. The server 130 receives an authorization request (sometimes also herein called an "AuthRequest") from the adapter module 100 (via a mobile device 150) and, if funds are available, returns an authorization grant (sometimes also herein called an "AuthGrant" or an "authorization grant token") for funds. Figure 22 shows components associated with the server 130. The shown example may be divided into multiple distinct components that are associated with each other or the example may be incorporated into or drawn from other technology (e.g., a computer or a main frame) as long as the components are associated with each other.

[0030] Advertise Presence: Each adapter module 100 advertises its presence by broadcasting signals (advertising broadcast signals) to mobile devices in the zones 102, 104, 106. Each adapter module 100 can listen to other adapter modules' advertisements.

[0031] Received Signal Strength Indicator (RSSI): The adapter module 100 may have a self-calibrating signal strength to determine zone thresholds (e.g., a payment zone threshold and an authentication zone threshold). At the time the user selects an item (product or service) from the payment accepting unit 120, the Received Signal Strength Indicator (RSSI) is logged. At this moment, it is presumed the user is within "arm's-length" (which

may be a predetermined length approximating the distance of a user standing in front of a machine for the purpose of making a purchase) from the payment accepting unit 120. A mathematical computation (i.e., In-Range Heuristics) is conducted to derive the optimal RSSI threshold at which point payment should be triggered by an application 140 on a mobile device 150. The threshold may be payment accepting unit specific and can vary over a period of time. This optimal zone threshold is preferably reported to the mobile device 150 during an initial handshake.

[0032] In-Range Heuristics: A mathematical computation that determines the RSSI threshold to determine when a user is in the authorization zone 104 and/or the payment zone 102. This computation can take into consideration numerous historical data points as well as transaction specific information such as which the mobile device 150 is being used, payment accepting unit type, among other factors. Preferably the RSSI is logged while the user is making his selection (this is the one time in the entire process that the user definitely will be "in range" (e.g., they will be arm's length from the machine 120 because they are physically interacting with the machine 120). The type of user mobile device 150, accelerometer data (e.g., is the user moving or stationary), and/or other information may also be logged while the user is making his selection. The adapter module 100 can give a reference RSSI for the payment zone 102 for the machine 120, and the application 140 can make a +/- adjustment based on the specific mobile device 150 on which it is installed. Over a period of time, the payment processing system continues to improve itself based on additional data points.

[0033] Authorization Request ("AuthRequest:): When a user enters the authorization zone 104, the mobile device 150 notifies the adapter module 100 and the adapter module 100 sends a secured authorization request (e.g., the encrypted authorization request) as a "message" (also referred to as a communication or transmissions) to the server 130 via the mobile device 150. Encryption may be performed by a security unit 755 (Figure 20) with security technology (e.g., encryption and decryption means) that may be associated with the processing unit 750 and/or the memory 760. Significantly, the AuthRequest is a request for authorization of funds, not a request for authorization of a transaction. The purpose of the funds is irrelevant to the server 130.

[0034] Authorization Grant Token ("AuthGrant"): This is a "message" (also referred to as a communication or transmissions) encrypted by the security unit 955 (Figure 22) with security technology (e.g., encryption and decryption means) of the server 130 with the unique private key corresponding to the adapter module 100. The secured authorization grant (e.g., the encrypted authorization grant) is passed from the server 130 to the adapter module 100 via the mobile device 150 in the form of a message. The mobile device 150, however, is not able to decrypt and/or read the message. The authorization grant is in response to the authorization request. The amount of the funds granted by the AuthGrant may be determined by factors including, but not limited to, the amount of funds available (or, if funds are not available, a mini-loan could be granted), a pre-authorized amount (e.g., set by the server, set by the user during set-up, set by the funding source, or a standard amount), limited by time (e.g., only a certain amount per hour, or a predetermined amount at specific times of the day), limited to the maximum amount of an item on the machine (or enough for two or three items in the machine), or one or more of these and other factors. Significantly, the AuthGrant makes the funds available, but does not authorize a transaction. The AuthGrant may have an associated expiration period in that it may expire if it is not used in a predetermined time period. The length of time before the AuthGrant expires may be determined by factors including, but not limited to, the trustworthiness of the user (e.g., the user has a long history with the payment processing system or some known provider (e.g., credit card provider, bank, or credit union), the user has a good credit rating, or the user has a large wallet balance), a pre-authorized time period (e.g., set by the server, set by the user during set-up, set by the funding source, or a standard time period), limited by time (e.g., predetermined time periods at specific times of the day such as longer times during breakfast, lunch, and dinner), limited by the machine or the products or services sold in the machine, limited by the number of other users near the machine (e.g., if it is a crowded machine, the AuthGrant may expire faster), or one or more of these and other factors. The AuthGrant remains valid until it expires or some other event occurs to end its validity (e.g., the user cancels it). This means that under normal circumstances the mobile device 150 will hold the AuthGrant authorizing use of funds for a pre-determined time period that will allow the user sufficient time to make a purchase. The authorized amount may be considered to be the "wallet balance" that is held in a virtual "wallet."

[0035] Synchronization: Time may be synchronized to the adapter module 100 from the server 130. The server 130 sends time information with encrypted messages and the adapter module 100 uses the time encoded in the messages for synchronization.

[0036] The mobile-device-to-machine payment processing system and components thereof may have associated hardware, software, and/or firmware (a variation, subset, or hybrid of hardware and/or software). The term "hardware" includes at least one "processing unit," "processor," "computer," "programmable apparatus," and/or other known or yet to be discovered technology capable of executing instructions or steps (shown as the processing unit 750 in Figure 20, the processing unit 850 in Figure 21, and the processing unit 950 in Figure 22). The term "software" includes at least one "program," "subprogram," "series of instructions," or other known or yet to be discovered hardware instructions or hardware-readable program code. Software may be loaded onto

hardware (or firmware) to produce a "machine," such that the software executes on the hardware to create structures for implementing the functions described herein. Further, the software may be loaded onto the hardware (or firmware) so as to direct the mobile-device-to-machine payment processing system (and components thereof) to function in a particular manner described herein or to perform a series of operational steps as described herein. "Hardware" such as the adapter module 100, the mobile device 150, and the payment accepting unit 120 may have software (e.g., programs and apps) loaded thereon. The phrase "loaded onto the hardware" also includes being loaded into memory (shown as the memory 760 in Figure 20, the memory 860 in Figure 21, and the memory 960 in Figure 22) associated with or accessible by the hardware. The term "memory" is defined to include any type of hardware (or other technology) -readable media (also referred to as computer-readable storage medium) including, but not limited to, attached storage media (e.g., hard disk drives, network disk drives, servers), internal storage media (e.g., RAM, ROM, EPROM, FLASH-EPROM, or any other memory chip or cartridge), removable storage media (e.g., CDs, DVDs, flash drives, memory cards, floppy disks, flexible disks), firmware, and/or other known or yet to be discovered storage media. Depending on its purpose, the memory may be transitory and/or non-transitory. Appropriate "messages," "communications," "signals," and/or "transmissions" (that includes various types of information and/or instructions including, but not limited to, data, commands, bits, symbols, voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, and/or any combination thereof) over appropriate "communication paths," "transmission paths," and other means for signal transmission including any type of connection between two elements on the payment processing system (e.g., the adapter module 100, the mobile device 150, the payment accepting unit 120, hardware systems and subsystems, and memory) would be used as appropriate to facilitate controls and communications.

[0037] It should be noted that the terms "programs" and "subprograms" are defined as a series of instructions that may be implemented as software (i.e. computer program instructions or computer-readable program code) that may be loaded onto a computer to produce a "machine," such that the instructions that execute on the computer create structures for implementing the functions described herein or shown in the figures. Further, these programs and subprograms may be loaded onto a computer so that they can direct the computer to function in a particular manner, such that the instructions produce an article of manufacture including instruction structures that implement the function specified in the flow chart block or blocks. The programs and subprograms may also be loaded onto a computer to cause a series of operational steps to be performed on or by the computer to produce a computer implemented process such that the instructions that execute on the computer provide steps for implementing the functions specified in the flow chart block or blocks. The phrase "loaded onto a computer" also includes being loaded into the memory of the computer or a memory associated with or accessible by the computer. Separate, albeit interacting, programs and subprograms may be associated with the adapter modules 100, the server 130, and the mobile device 150 (including the mobile application 140) and these programs and subprograms may be divided into smaller subprograms to perform specific functions.

[0038] The terms "messages," "communications," "signals," and/or "transmissions" include various types of information and/or instructions including, but not limited to, data, commands, bits, symbols, voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, and/or any combination thereof. Appropriate technology may be used to implement the "communications," "signals," and/or "transmissions" including, for example, transmitters, receivers, and transceivers. "Communications," "signals," and/or "transmissions" described herein would use appropriate technology for their intended purpose. For example, hard-wired communications (e.g., wired serial communications) would use technology appropriate for hard-wired communications, short-range communications (e.g., Bluetooth) would use technology appropriate for close communications, and long-range communications (e.g., GSM, CDMA, Wi-Fi, or the like) would use technology appropriate for remote communications over a distance. Appropriate security (e.g., SSL or TLS) for each type of communication is included herein. The security units 755 and 955 include technology for securing messages. The security technology may be, for example, encryption/decryption technology (e.g., software or hardware). Although encryption/decryption is discussed primarily as being performed using a unique private key, alternative strategies include, but are not limited to encryption/decryption performed using public/private keys (i.e., asymmetric cryptography), or other encryption/decryption strategies known or yet to be discovered. Appropriate input mechanisms and/or output mechanisms, even if not specifically described, are considered to be part of the technology described herein. The communications unit 770 (shown in Figure 20) of the adapter module 100 is shown as including appropriate input and output mechanisms 772, 774 that may be implemented in association (e.g., directly or indirectly in functional communication) with male and female adapters 720, 730 of the adapter module 100. The communications unit 870 (shown in Figure 21) of the mobile device 150 includes mechanisms for both long-range communications (shown as the long-range communication capability 872 such as cellular and/or Wi-Fi mechanisms) for communicating with the server 130 and short-range communications (shown as the short-range communication capability 876 such as Bluetooth mechanisms) for communicating with the adapter module 100.

[0039] When used in relation to "communications,"

"signals," and/or "transmissions," the terms "provide" and "providing" (and variations thereof) are meant to include standard means of provision including "transmit" and "transmitting," but can also be used for non-traditional provisions as long as the "communications," "signals," and/or "transmissions" are "received" (that can also mean obtained). The terms "transmit" and "transmitting" (and variations thereof) are meant to include standard means of transmission, but can also be used for non-traditional transmissions as long as the "communications," "signals," and/or "transmissions" are "sent." The terms "receive" and "receiving" (and variations thereof) are meant to include standard means of reception, but can also be used for non-traditional methods of obtaining as long as the "communications," "signals," and/or "transmissions" are "obtained."

[0040] The term "associated" is defined to mean integral or original, retrofitted, attached, connected (including functionally connected), positioned near, and/or accessible by. For example, if the user interface (e.g., a traditional display 122 (Figure 19), a touch screen display 124 (Figure 19), a key pad 126 (Figure 19), buttons 126 (Figure 19, shown as part of the key pad 126), a keyboard (not shown), and/or other input or output mechanism) is associated with a payment accepting unit 120, the user interface may be original to the payment accepting unit 120, retrofitted into the payment accepting unit 120, attached to the payment accepting unit 120, and/or a nearby the payment accepting unit 120. Similarly, adapter modules 100 may be associated with payment accepting units 120 in that the adapter modules 100 may be original to the payment accepting unit 120, retrofitted into the payment accepting unit 120, attached to the payment accepting unit 120, and/or a nearby the payment accepting unit 120.

SYSTEM OVERVIEW

[0041] Figures 5, 6, and 7 together show major components of the mobile-device-to-machine payment system and the interactions there-between.

[0042] As shown, the adapter module 100 functionally connected bi-directionally to the payment accepting unit 120 via a wired serial connection such that no security is necessary. The adapter module 100 is also functionally connected bi-directionally to the mobile device 150 (and its installed mobile application 140) via short-range communication technology (e.g., a Bluetooth connection). Because the mobile device 150 is not a "trusted" link (e.g., it could be hacked by a user), only secured communications (transmissions) are passed between the adapter module 100 and the mobile device 150. This keeps communications secured and safe from hacking. The mobile device 150 (and its installed mobile application 140) is also functionally connected bi-directionally to a system management server 130 and/or a funding source server 160 via long-range communication technology (e.g., Wi-Fi or Cellular connection) that preferably has appropriate security (e.g., SSL security). Security between the mobile device 150 and the system management server 130 has the advantage of protecting communications from the mobile device 150 to the system management server 130 that may include sensitive data and may not be encrypted. The system management server 130 and the funding source server 160 may be connected via a wired Internet connection with SSL security. The system management server 130 may be connected via a wired Internet connection with SSL security to an operators' server 170. Although not necessary to implement a purchase transaction, for other purposes (e.g., inventory), the operators' server 170 may be connected to the payment accepting unit 120 using a handheld computer sync or a cellular connection.

[0043] Also, a unique private key may be used to securely transmit encrypted messages between the adapter module 100 and the system management server 130 (although the encrypted transmissions would most likely be routed through the mobile device 150). The server 130 stores a private key for each adapter module 100, and this key is only known to the adapter module 100 and the server 130. No intermediary is privy to this key (especially not the mobile device 150). When the adapter module 100 and the server 130 communicate messages (e.g., AuthRequest and AuthGrant), the security unit 755 of the adapter module 100 encrypts the message with its private key and passes the message to the mobile device 150. The mobile device 150 (which preferably cannot decrypt the message) passes the encrypted message to the server 130. The server 130 is able to decrypt the message using the security unit 955 of the adapter module 100 and the unique private key. The security unit 955 of the server 130 uses this same unique private key to encrypt messages to the adapter module 100 and sends the message to the mobile device 150 to relay to the adapter module 100 that is able to decrypt the message using the security unit 755 of the adapter module 100 and the unique private key.

[0044] Figure 7 shows specific communications and messaging with a vending sequence (the numbers to the left of the communications and messaging) between the adapter module 100, the mobile device 150, and the system management server 130. These communications are discussed in more detail in the discussion pertaining to the schematic flow diagrams (Figures 8A-8G) and the flow charts (Figures 9A-9E).

[0045] It should be noted that Figures 5, 6, and 7 are examples, and are meant to help in the understanding of the mobile-device-to-machine payment system. For example, the shown long-range communications technology may be replaced with alternative long-range communications technology known or yet to be discovered, the shown short-range communication technology may be replaced with alternative short-range communication technology known or yet to be discovered, and the shown security may be replaced with alternative security known or yet to be discovered. The shown connections are

meant to be examples, and there may be intermediaries that are not shown. The shown components have been simplified in that, for example, only one mobile device 150 (or machine 120, adapter module 100, or server 130) is shown where many may be included. Finally, the order of the steps may be changed and some steps may be eliminated.

ADAPTER MODULE

[0046] Figures 11-18 show views of adapter module 100a (referred to generally as adapter module 100). Adapter module 100 is a relatively low cost hardware component that is pre-configured to work with the industry standard multi-drop bus (MDB). On machines without MDB technology, the adapter module 100 can be configured or designed to work with other serial protocols or activate a switch. In essence the adapter module 100 simulates establishing payment on payment accepting unit 120 in much the same manner as other alternative forms of payment (e.g., cash).

[0047] The shown adapter modules 100 are preferably designed to be used as an in-line dongle for in-line insertion within, for example, a MDB of a machine 120. The wire used in MDB technology uses male and female connection ends or adapters to allow the attachment of peripherals. In the case of a vending machine, the wire with the connection ends or adapters would be present to allow the attachment of a payment receiving mechanism (e.g., a coin mechanism). The MDB male and female adapters 700, 710 may be separated (as shown in Figures 17-18). The adapter module 100a in Figures 11 and 17-18 has a male adapter 720 and a female adapter 730. The adapter module 100a may be plugged (inserted) in serial ("in-line") with the wire. For example, the MDB female adapter 710 may be connected to the male adapter 720 of the adapter module 100 and the MDB male adapter 700 may be connected to the female adapter 730 of the adapter module 100. The resulting in-line configuration is shown in Figure 19. It should be noted that the adapter modules 100 are designed to allow pass-through communications so that if the mobile-device-to-machine payment processing system is not enabled (e.g., for a particular purchase or simply turned off) the MDB functions as though the adapter module 100 is not there and the machine 120 can function normally.

HANDS-FREE MODE

[0048] Summarily, if it is available, a hands-free mode, from the user's perspective, would allow the user to approach a favorite payment accepting unit 120 and notice that the display (e.g., the displays 122 or 124 shown in Figure 19) associated with the payment accepting unit 120 shows funds available (e.g., the wallet balance), he would select the product or service using input mechanisms (e.g., buttons 126 or a touch screen display 124 shown in Figure 19) associated with the payment accepting unit 120, and he would retrieve his dispensed services or products.

[0049] During an initial handshake with the mobile device 150 (when the user is within range), the adapter module 100 reports to the mobile device 150 whether or not hands-free mode is available. If it is available, the installed mobile application 140 automatically connects to the payment accepting unit 120 without the user having to interact with the mobile device 150. The user observes that funds are available on the display 122, 124 of the payment accepting unit 120 and completes the purchase transaction as if cash was inserted in the machine 120 by inputting his selection on the payment accepting unit 120. The payment accepting unit 120 dispenses the product or service. After the selection is made, the change is returned to the mobile device 150.

[0050] Whether hands-free payment is available is determined by factors including, but not limited to whether if other mobile devices 150 are in range, if other adapter modules 100 are in range, if there are any alerts, if the payment trigger threshold is having wide variances and so deemed unstable, or if the payment accepting unit operator (e.g., a vending machine operator) has opted to disable hands-free mode for the payment accepting unit 120. In the latter instance, operators can disable via a maintenance mobile device 150, as well as through the operators' server 170 and/or the system management server 130.

[0051] Figure 3 is a table that shows considerations, conditions, or factors that may be used to determine whether the hands-free pay feature is available. Starting at the "Favorite?" column, this indicates whether the payment accepting unit 120 is a favorite machine. Preferably the hands-free pay feature is only available for use with "favorite" payment accepting units 120 (e.g., a vending machine at work or school). The "Alert" column has to do with whether there is some reason (e.g., there are too many users in range) that the hands-free pay feature should not work and, if there is such a reason, the user will be notified (alerted) and may be able to use the manual mode to resolve the alert and/or complete the transaction. Figure 3 shows situations in which a user is or is not able to make hands-free purchases from a machine 120 using a mobile application 140 on his mobile device 150. It should be noted that the shown interface is an example. For example, some of the features could be automated or pre-selected. (It should be noted that the left hand column, the "Tab" column, relates to whether the selected tab on the mobile application 140 is "all" or "favorite." Figures 10A-10D all show these tabs. Unlike the other columns in Figure 3, this column has more to do with the functionality and view of the application 140 than specifically with the hands-free feature. The tabs would allow a user to select whether he wanted to be alerted when he was in range of all payment accepting units 120 or just "favorite" payment accepting units 120 and the application 140 would show the appropriate view.)

[0052] Balance Display: An optional feature of the mobile-device-to-machine payment system that is particularly helpful in the hands-free mode (although it may be available in the manual mode and/or in a multiple-vend scenarios) is when the user's mobile device 150 sends "credit" to the payment accepting unit 120 (either via hands-free payment or through a manual swipe), the wallet balance is sent to the payment accepting unit 120 that is then displayed to the user on a display 122, 124 of the machine 120. This is particularly beneficial during hands-free mode when the user does not retrieve the mobile device 150 and, therefore, may not know the balance. Also, in a multiple-vend scenario the user would not have to calculate a remaining balance.

[0053] An example of a hands-free, multiple-vend scenario where a balance is displayed by the payment accepting unit 120, follows: The user has $5.00 in his/her virtual wallet as that is the amount that has been authorized (the AuthGrant being stored on the mobile device 150). The user walks up to the payment accepting unit 120 and $5.00 is displayed on the display 122, 124 of the payment accepting unit 120 since hands-free mode was enabled and credit was sent (e.g., via the short-range communication capability) to the payment accepting unit 120. The user makes a selection of $1.50 by interacting (e.g., pressing buttons) with the machine 120. The item (product or service) is dispensed and the "change" is "returned" (e.g., via the short-range communication capability) to the virtual wallet. But since the user is still standing in the payment zone 102, the remaining wallet balance of $3.50 is sent to the payment accepting unit 120 and displayed so that the user can now see that he/she has a $3.50 balance. (It should be noted that the authorized funds may remain on the machine 120 and not be transferred back to the mobile device 150 between transactions.) The user decides to purchase a $1.50 item, and the transaction is completed as usual (e.g., by interacting with the machine 120). Now the user is still standing in the payment zone 102 and he/she sees the wallet balance of $2.00 on the display 122, 124 of the payment accepting unit 120. The user decides that he/she does not wish to purchase anything else and simply walks away. As he/she walks out of the payment zone 102, the credit is cleared from the machine 120, but he/she is left with the knowledge that his wallet balance is $2.00 even though he/she never touched the mobile device 150. Communications between the payment accepting unit 120 and the adapter module 100 (via the mobile device 150) handle the accounting incidental to the transaction. The remaining balance ($2.00) is technically stored on the server 130, and may be reflected on the application 140 on the mobile device 150.

MULTIPLE DISTINCT ZONES

[0054] As shown in Figures 1-2, the functions performed by the adapter module 100 can be divided into distinct zones: a first "communication zone" (e.g., "Bluetooth range" 106), a second "authorization zone" 104, and a third "payment zone" 102. The payment zone 102 is smaller than or equal to (overlapping completely) the authorization zone 104. Put another way, the payment zone 102 is within or coextensive with the authorization zone 104. The payment zone 102 is a subset of the authorization zone 104 with a ratio of the payment zone 102 to the authorization zone 104 ranging from 0.01:1 to 1:1. It is not necessarily a fixed ratio and can vary between different payment accepting units 120, different mobile devices 150, different users, and over time. While the zones 102, 104, 106 are depicted as having a uniform shape, the zones may not necessarily be uniform (or constant over time) in that the shape can vary. For example, the shape of the Bluetooth range 106 may vary depending on environmental conditions such as obstacles in the room and payment accepting unit 120 door/wall materials.

[0055] Bluetooth Range 106 (sometimes also herein called the "communication zone"): The outermost range is the Bluetooth range 106 (shown in Figures 1-2). This is the area in which the adapter module 100 is able to broadcast its presence. In most situations, the Bluetooth range 106 is a passive range in that no actual data is exchanged between the mobile device 150 and the adapter module 100. While in the Bluetooth range 106, the mobile device 150 monitors the RSSI (Received Signal Strength Indicator).

[0056] Authorization Zone 104: The middle region is the authorization zone 104 (shown in Figures 1-2). This is a computed area based on the RSSI. As mentioned, the mobile device 150 monitors the RSSI while it is in the Bluetooth range 106. When the RSSI reaches a certain predetermined threshold based on In-Range Heuristics, the mobile device 150 can be considered to be in the authorization zone 104. In the authorization zone 104 the mobile device 150 establishes a connection to the adapter module 100 (e.g., a Bluetooth connection (Figure 5) with SSL protection (Figure 6)) and informs the adapter module 100 of its presence. After a successful handshake with the adapter module 100, the mobile device 150 registers the adapter module 100 and the adapter module 100 requests an authorization to the server 130 via the mobile devices' network connection (e.g., a Wi-Fi or cellular connection (Figure 5) with SSL protection (Figure 6)). It is important to note the mobile device 150 and the adapter module 100 have a non-exclusive relationship at this point. The adapter module 100 may collect registrations for all mobile devices 150 that are within the authorization zone 104.

[0057] An authorization occurs in preparation for when the user enters the payment zone 102 (shown in Figures 1-2). An authorization expires in a set period of time (for example, five minutes), so if the mobile device 150 is still in the authorization zone 104 at the time of expiration, the adapter module 100 submits for and receives another authorization. This will continue for a set number of times (for example, the limit may be three times to limit cases

of numerous authorizations for a mobile device that may remain in the authorization zone 104 for an extended period of time without completing a transaction). Should authorization fail (for instance if the limit had been reached) prior to the user entering the payment zone 102, the adapter module 100 will request authorization when the mobile device 150 enters the payment zone 102 (which adds a few seconds to the experience).

[0058] Payment Zone 102: As a user enters the payment zone 102, the mobile device 150 establishes exclusive control of the adapter module 100. Once established, any other user in the payment zone 102 is put into a "waiting" status.

[0059] In the payment zone 102, the payment can be triggered automatically if the payment processing system has and is in hands-free mode. In such instances, the mobile device 150 is running the application 140 in background mode and will send credit to the payment accepting unit 120 without any explicit user interaction. The user completes the transaction on the payment accepting unit 120 in much the same manner as if cash had been inserted into the payment accepting unit 120 to establish credit. After the user completes the transaction (that may include one or more purchases), details of the transaction are preferably returned to the mobile device 150 and server 130 in separate messages. The message to the server 130 is preferably encrypted with the adapter module's 100 private key (Figure 6) to ensure data integrity. As shown in Figure 7, the "private key" coded message (Encrypted VendDetails) is preferably sent via the mobile device 150. The message to the mobile device 150 may be sent solely for the purpose of closing the transaction. The transaction history and balance are updated server-side via the encrypted message sent to the server 130.

[0060] The other mode of operation is manual mode. In manual mode, the user launches the mobile device 150 and is able to swipe to send payment to the payment accepting unit 120. The user can also swipe back to cancel the payment. Like in hands-free mode, the purchase transaction is completed on the payment accepting unit 120 in the same manner as if cash were inserted into the payment accepting unit 120. The mobile device 150 is only used to send payment. Selection is made directly on the payment accepting unit 120.

[0061] Self-Calibrating Zone Threshold: A key, but optional feature, of the payment processing system is a self-calibrating payment zone RSSI threshold. Because RSSI can vary machine to machine, environment to environment, and device to device, having a fixed threshold at which payment is triggered can be problematic. The approach suggested herein is the creation of a self-calibrating threshold. When the user is interacting with the payment accepting unit 120 (such as when he makes his selection on the payment accepting unit 120), the payment accepting unit 120 notifies the adapter module 100 and the adapter module 100 logs the conditions such as RSSI, type of user mobile device 150, accelerometer data, and other information. It is at this point that it can be ascertained safely that the user is within arm's-length from the payment accepting unit 120 (by necessity the user is arm's-length because he is making some physical interaction with the payment accepting unit 120). This is the only point in the entire transaction in which it can be certain that the user is within arm's-length from the payment accepting unit 120.

[0062] Figure 4 shows a simplified set of steps involved when users enter the payment zone 102. Specifically, Figure 4 shows that credit is established 200 (this may have been done in the authorization zone 104, but if not it would be handled in the payment zone 102), that the user makes a selection using the machine 202, that the machine notifies the adapter module of the selection 204, that the adapter module (optionally) logs the RSSI 206, and that the purchase process(es) continues 208. Using the historically logged RSSI data, the adapter module 100 calculates one of several "average" RSSI using various mathematical models. This "average" could be a traditional average, a moving average, a weighted average, a median, or other similar summary function. The adapter module 100 could pre-process the historical data before running the function, such as to eliminate top and bottom data points, suspect data points, etc.

[0063] Optionally, during the handshake between the mobile device 150 and the adapter module 100, the information transmitted to the adapter module 100 may include, for example, the model of the mobile device 150. Using the received information pertaining to the mobile device models, the adapter module 100 can create multiple payment thresholds, one for each mobile device model. This allows for variances that may be inherent in different types of Bluetooth radios. An alternative to this method is for the adapter module 100 to broadcast a baseline payment zone threshold, and the mobile device 150 can use an offset from this baseline based on its specific model type. The payment zone thresholds (or baseline offsets) can be unique to specific types of mobile devices (e.g., by manufacturer, operating system, or component parts), models of mobile devices, or individual mobile devices (unique to each user).

[0064] In a typical scenario in which the payment zone threshold has been calibrated, the adapter module 100 advertises its presence along with the threshold at which it considers any mobile device 150 to be in the authorization zone 104. This is a one-way communication from adapter module 100 to mobile device 150. Once the mobile device 150 enters the authorization zone 104, there is a handshake that is established between the adapter module 100 and the mobile device 150. During this handshake, the mobile device 150 can share its model information with the adapter module 100, and the adapter module 100 can return the payment zone 102 threshold for that specific model.

[0065] Optionally, in addition to calibrating the payment zone threshold, the adapter module 100 can apply the self-calibrating model to the authorization zone 104 to calibrate the authorization zone threshold. As with the

payment zone thresholds, the authorization zone thresholds can be unique to specific types of mobile devices, models of mobile devices, or individual mobile devices. In this scenario, the adapter module 100 would broadcast multiple thresholds by device type and the mobile device 150 would determine which threshold to apply (or alternatively broadcast a baseline and the mobile device 150 uses an offset based on its device model). Even in this scenario, the authorization zone 104 is a one-way communication.

[0066] Optionally, along with the threshold that is calculated (in the payment and/or the authorization zone(s)), a safety margin can be added to minimize scenarios in which a user is within range, but the mobile-device-to-machine payment processing system does not recognize it because the threshold may not have been reached. For example, if the calculated RSSI for an iPhone™ 5 on machine 4567 is -68 db, the mobile-device-to-machine payment processing system may add a safety margin of -5 db, and establish the threshold at -73 db. So when a user's phone is communicating with the adapter module 100 at an RSSI of -73 db or better, the mobile-device-to-machine payment processing system will allow the mobile device 150 to credit the payment accepting unit 120. The safety margin can be set on the server 130 and downloaded to the adapter module 100, or set on the mobile device 150, or set on the adapter module 100 itself.

[0067] Optionally, in the payment zone threshold, the mobile device 150 can use other data to determine when to cancel the exclusive control of the payment accepting unit 120, to identify when the user is moving out of the payment zone 102. External data could include accelerometer data from the mobile device 150. Using that data, the mobile device 150 can determine whether the user is standing relatively still in front of the payment accepting unit 120, or if the user is in motion - effectively walking away from the payment accepting unit 120.

SIGNAL UNAVAILABILITY ADAPTATION

[0068] The mobile-device-to-machine payment processing system described herein uses a mobile device's 150 short-range communication technology (e.g., Bluetooth mechanisms) (shown as short-range communication capability 876 in Figure 21) and a mobile device's 150 long-range communications technology (e.g., cellular and/or Wi-Fi mechanisms) (shown as long-range communication capability 872 in Figure 21). The short-range communication capability 876 communicates with the adapter module's 100 short-range communication technology (e.g., Bluetooth mechanisms) (shown as short-range communication capability 776 in Figure 20). The long-range communication capability 872 communicates with the server's 130 long-range communications technology (e.g., cellular and/or Wi-Fi mechanisms) (shown as long-range communication capability 972 in Figure 22). The mobile device 150 (with a mobile application 140 thereon) acts as a communication bridge between the adapter module 100 (associated with a payment accepting unit 120) and the server 130. This process is described herein and works properly if there is cellular or Wi-Fi coverage within the payment zone 102.

[0069] One option if there is no cellular or Wi-Fi coverage within the payment zone 102 is to determine whether there is cellular or Wi-Fi coverage within the authorization zone 104 or the Bluetooth range 106. If there is, then the sizes of the zones 102, 104, 106 could be adapted and the timing could be adapted. For example, if the mobile devices 150 detected problems with the cellular or Wi-Fi coverage within the payment zone 102, the user could carry his mobile device 150 into the other zones (or the mobile device 150 could use short-range communication technology to communicate with other mobile devices 150 within the authorization zone 104 or the Bluetooth range 106) to determine whether the zones have cellular or Wi-Fi coverage. If they do have coverage, communication between the mobile device 150 and the server 130 can be advanced (conducted earlier when the mobile device 150 is further from the machine 120) or delayed (conducted later when the mobile device 150 is further from the machine 120). This can be thought of as changing the size or shapes of the zones 102, 104, 106. The timing would also have to be adjusted so that the authorization of funds (AuthGrant) does not expire before the user has a chance to make a purchase. It also means that balance updates to the server 130 may happen after the user has moved away from the machine 120 and has cellular or Wi-Fi coverage again.

[0070] Another option if there is no cellular or Wi-Fi coverage within any of the zones 102, 104, 106 is for the user to obtain authorization while outside of the zones in a place with cellular or Wi-Fi coverage. This may occur, for example, if a user knows that he will be going to a place with a payment accepting unit 120 equipped with an adapter module 100 (perhaps to a favorite payment accepting unit 120) that does not have (or rarely has) cellular or Wi-Fi coverage. A user may also use the mobile application 140 to query payment accepting units 120 in a given range (e.g., within 50 miles) or at a given location (e.g., at a campground or in a particular remote city) to determine whether there is cellular or Wi-Fi coverage within the zones 102, 104, 106. The user can then obtain pre-authorization from the server 130 using the mobile application 140. Again, the timing would also have to be adjusted so that the authorization of funds (AuthGrant) does not expire before the user has a chance to make a purchase. It also means that balance updates to the server 130 may happen after the user has moved away from the machine 120 and has cellular or Wi-Fi coverage again. A mobile-device-to-machine payment system having the ability to implement this option would be able to accept cashless payments without requiring any network connection near the payment accepting unit 120. In some implementations, the mobile-device-to-machine payment processing systems described herein is located in a remote location where no signal is available,

therefore, can accept cashless payments.

[0071] As an example of a situation in which there might be no cellular or Wi-Fi coverage within any of the zones 102, 104, 106 of a particular payment accepting unit 120, the user (a teenager) may be traveling to a remote location to attend summer camp where there is no cellular or Wi-Fi coverage. The camp may have several payment accepting units 120 (e.g., a machine that creates a dedicated "hot spot" that requires payment for use, vending machines, or machines for renting equipment such as bikes, kayaks, or basketballs). The camp facility might notify parents that the mobile-device-to-machine payment system is available. The parents, while at home, could obtain authorization for a particular amount (that could be doled out a certain amount per day or limited to type of machine or location) to be authorized and "loaded" into the user's mobile device 150 and specify that the authorization will not expire for a certain period or until a certain date. Thereafter, while at camp, the user could use the mobile application 140 on his mobile device 150 in a manner similar to those discussed elsewhere herein. Short-range communications may be used for communications between the adapter modules 100 (associated with the machines 120) and users' mobile devices 150.

[0072] One subtle but powerful component of the payment processing system described herein is that it requires a long-range communication capability (e.g., an Internet or cellular network connection) only in the authorization zone 104 and only for the time period required to send the AuthRequest and receive the AuthGrant. Once a valid AuthGrant is received by the mobile device 150, the long-range communication capability (e.g., an Internet or cellular network connection) is not required by either the mobile device 150 or the adapter module 100 in the payment zone 102 as long as the AuthGrant is valid (unexpired). This mechanism allows the system to seamlessly handle authenticated transactions in (temporary) offline mode, with the deferred acknowledgement and transaction messages performing the bookkeeping and cleanup when network connection is regained. The alternatives described above provide a unique way to artificially extend the authorization zone to include any location where the mobile device 150 can communicate with the server 130.

MULTIPLE USER RESOLUTION

[0073] As shown in Figure 2, in one practical scenario, multiple users are in the zones 102, 104, 106. As shown in Figure 2, users 1, 2, and 3 are in the payment zone 102 near the machine 120; users 5 and 6 are shown as positioned between the authorization zone 104 and the Bluetooth range 106; users 4 and 7 are in the Bluetooth range 106, user 10 is positioned on the edge of the Bluetooth range 106; and users 8 and 9 are positioned outside of Bluetooth range 106. In some implementations, the mobile-device-to-machine payment processing system manages and resolves issues pertaining to multiple users.

[0074] Users 4 and 7 are within the Bluetooth range 106 and the user 10 is either entering or leaving the Bluetooth range 106. Within the Bluetooth range 106 the users' mobile devices 150 are able to see the adapter module's 100 advertisement. In this zone, the mobile device 150 preferably does not initiate a connection. The adapter module 100 is preferably unaware of the users in the Bluetooth range 106. All the adapter module 100 is doing is advertising its presence to any multitude of users that may be in Bluetooth range 106.

[0075] The adapter module 100 begins to log users as the users (and their respective mobile devices 150) enter the authorization zone 104 (shown in Figure 2 as users 5 and 6). At this point, there is a non-exclusive connection initiated by the mobile device 150 to the adapter module 100. It does a handshake (e.g., to exchange information needed to obtain authorization and, optionally, to log information needed for a self-calibrating authorization zone threshold) and the mobile device 150 contacts the server 130 for an authorization (e.g., sending an AuthRequest and receiving an AuthGrant). The adapter module 100 registers all mobile devices 150 that have requested and received AuthGrants. The adapter module 100 continues communicating with any other mobile device 150 that enters the authorization zone 104. After initial contact, the adapter module 100 may provide the mobile device 150 with a deferral delay of when to check back in with the adapter module 100 allowing opportunity for other mobile devices 150 to communicate with the adapter module 100.

[0076] If there is only one user in the payment zone 102, a purchase transaction may be performed. If there are multiple users in the payment zone 102, the mobile-device-to-machine payment system must handle the situation.

[0077] One optional solution for handling the situation of the multiple users in the payment zone 102 is queuing users in the payment zone 102. Once any mobile device 150 enters the payment zone 102, it establishes exclusivity to a particular mobile device 150 (e.g., in a first-come-first-serve manner). Technically, however, the adapter module 100 is not establishing an exclusive connection to the mobile device 150. The adapter module 100 can still perform a round-robin poll and communicate with and advertise to other mobile devices 150. Instead, the adapter module 100 establishes a queue prioritized by RSSI and time (e.g., who was first and whether the authorization has expired) and it notifies (e.g., alerts) other mobile devices 150 to wait. The earliest valid (unexpired) authorization takes precedence when there is any tie in the RSSI. Otherwise, for example, the strongest average RSSI takes priority. Preferably the queue is not a static measure of the RSSI but an averaged measure over the period of time in the queue. This compensates for a scenario in which a user may be walking around in the queue and then shows up at the payment accepting unit 120 just as the previous user is finishing. If another

user was also in the payment zone 102 and stood there the entire time, but may have newer authorization, he could win out.

[0078] Anytime that the adapter module 100 cannot determine exactly which user is in the payment zone 102 in front of the payment accepting unit 120, the adapter module 100 will disable hands-free payment. The mobile device 150 will send an alert to the user and he can use swipe to pay (manual mode). All users in payment zone 102 will show "Connected" and the first to swipe payment to the payment accepting unit 120 then locks out other users.

MULTIPLE MODULE RESOLUTION

[0079] In the scenario where there are multiple modules present, determining which payment accepting unit 120 a user is in front of can be a challenge. In some implementations, the mobile-device-to-machine payment processing system described herein allows adapter modules 100 to communicate to other adapter modules 100 in range via Bluetooth. Each user receives authorization grants for specific payment accepting units 120. This means if there are multiple adapter modules 100 within the same authorization zone 104, there will be multiple authorization grants for the user. When the user enters the payment zone 102, it can be difficult to differentiate which payment accepting unit 120 the user is in front of if the payment zones 102 overlap.

[0080] To solve this problem, when the user enters the payment zone 102, the adapter modules 100 communicate with each other to determine the RSSI for the particular user (based on the signal from his mobile device 150) to triangulate which adapter module 100 (and the associated payment accepting unit 120) is closer to the user. Optionally, the intermodule communications can restrict the user to establishing an exclusive connection with only one payment accepting unit 120.

[0081] Optionally, when the user connects to a payment accepting unit 120, the mobile device 150 can send a communication to the payment accepting unit 120 for momentary display to the user on the display 122, 124 of the payment accepting unit 120. For example, the mobile device 150 can send a communication (e.g., "connected" or "Fred's Mobile Device Connected") to the payment accepting unit's display 122, 124 for a predetermined period of time (e.g., 1-3 seconds) so when the user is in payment zone 102, it is clear which payment accepting unit 120 the user is connected to prior to making a purchase (either in hands-free or manual mode).

[0082] In addition, when the user is in manual mode, the mobile device 150 can display (e.g., on the touch screen 152 as shown in Figures 10A-10D) a visual indication of the payment accepting unit 120 (e.g., a picture and/or a payment accepting unit ID of the payment accepting unit 120) for visual confirmation. If the user is in manual mode, the user can manually change the payment accepting unit 120.

DESCRIPTIVE SCENARIO

[0083] Figure 7, Figures 8A-8G, and 9A-9E (as well as other figures) can be used to understand a detailed scenario of the mobile-device-to-machine payment processing system described herein. A flow of communications and steps are loosely described below with reference to these (and other figures). It should be noted that alternative scenarios could include, for example, a modified order of the steps performed.

[0084] Prior to vending transactions, a user downloads a mobile application 140 onto his mobile device 150, creates an account, and configures a funding source via, for example, a funding source server 160. A funding source may be, for example, a debit card, a credit card, campus cards, rewards points, bank accounts, payment services (e.g., PayPal™) or other payment option or combination of payment options known or yet to be discovered. The funding sources may be traditional and/or nontraditional payment sources that are integrated into the ecosystem described herein and then used indirectly as a source of funds. Funds from the funding source are preferably held on the server 130 such that when an AuthRequest is received by the server 130, the server 130 can send an AuthGrant authorizing funds for a purchase.

[0085] The user can specify one or more "favorite" adapter module(s) 100 (that has a one-to-one relationship to the payment accepting unit 120) that he may visit regularly, such as a vending machine at school or work. Favorite adapter modules 100 appear on a prefiltered list and allow for additional rich features such as hands-free payment.

[0086] The payment accepting unit 120 may be equipped with an adapter module 100 that is constantly advertising its availability via Bluetooth (or other "signals," "communications," and/or "transmissions"). This ongoing advertising and scanning for adapter modules is shown in Figure 8A. As shown, the mobile device 150 is continuously scanning for any adapter module 100 within Bluetooth (or other "signal," "communication," and/or "transmission") range. When the user is within range of that adapter module 100, the mobile device 150 tracks and monitors the signal strength until a predetermined "authorization zone" threshold is achieved.

[0087] Figures 8B and 9A generally show that when the authorization zone threshold is reached, the mobile device 150 enters the authorization zone (block 302) and registers the adapter module 100. The mobile device 150 connects to the server 130 (block 304). The application 140 on the mobile device 150 creates a request for authorization (AuthRequest) and passes the AuthRequest to the server 130 using appropriate communication technology (e.g., GSM, CDMA, Wi-Fi, or the like) (block 306). The server 130 responds with an authorization grant (AuthGrant) encrypted with the specific adapter module's private key (block 306). This authorization token may minimally include the User identifier (ID), Apparatus ID (for the adapter module 100), authorization amount, and

expiration time. The mobile device 150 receives the AuthGrant from the server 130, and retains it until the mobile device 150 is ready to issue payment to an adapter module 100. The mobile device 150 collects all pending AuthGrants that may be one or more depending on how many adapter modules 100 are in-range. Unused AuthGrants that expire are purged from the mobile device 150 and the server 130. It is important to note that the mobile device 150 is unable to read the AuthGrant because it is encrypted with the adapter module's unique private key that is only known to server 130 and adapter module 100. This provides a preferred key element of security in the system as the adapter module 100 only trusts AuthGrants that are issued by the server 130, and the AuthGrants cannot be read or modified by the mobile device 150 or any other party in between the server and the adapter module 100. Additional mobile devices 150 may enter the authorization zone 104 (block 308).

[0088] As the user approaches a specific adapter module 100, the user enters the payment zone 102 and an event threshold is triggered based on heuristics performed by the mobile device 150. Blocks 310 and 312 show the loop steps of waiting for a mobile device 150 from the authorization zone 104 to enter the payment zone 102. If the user leaves the authorization zone 104 without entering the payment zone 102, the adapter module 100 returns to advertising its presence (block 300).

[0089] Figures 8C and 9B generally show the user entering the payment zone. The mobile device 150 verifies that it has an unexpired and valid AuthGrant. If the AuthGrant is not good, it may be requested again, repeating the Authorization Request process (block 315). If the AuthGrant is good, the mobile device 150 sends the valid AuthGrant (including the wallet balance (block 322)) to the adapter module 100 to initiate a transaction. The mobile device 150 may issue the AuthGrant automatically without specific user interaction if the hands-free mode is supported (and the device is a favorite (block 318), there is only one device in the payment zone 102 (block 318), and (optionally) there is only one user in the authorization zone 104 (block 320). If any of these factors are not present, the mobile device 150 will prompt and/or wait for the user to begin the transaction manually (block 324).

[0090] Figures 8D, 9C, and 9D generally show the transaction process. As shown in Figure 9C, the adapter module 100 runs through a series of questions to determine if there are any issues that would prevent vending including: has the user canceled in-app? (block 326), has the user walked away? (block 328), is the coin return pressed? (block 330), has more than a predetermined period of time elapsed? (block 332). If the answer to any of these questions is "yes," the transaction does not proceed. If the answers to all of these questions is "no," the user makes a selection (block 334) on the payment accepting unit 120 in the same or similar manner as compared to if cash or credit were presented to the payment accepting unit 120. If the machine 120 is able to vend (block 336), it attempts to release the product. If the vend fails (block 338) it is reported by the machine (block 340) and a credit is returned to the virtual wallet (block 342). If the vend is successful (block 338) it is reported by the machine (block 344). Put another way, after the transaction is complete, the adapter module 100 returns to the mobile device 150 the details of the transaction as well as an encrypted packet containing the vend details to be sent to the server 130 via the mobile device 150. Optionally, the adapter module 100 can pass additional information not directly related to the transaction such as payment accepting unit health, sales data, error codes, etc.

[0091] Figures 8D and 9E generally show the multi-vend function. If the machine has enabled multi-vend capabilities (block 350) and the multi-vend limit has not been reached, the process returns to the question of whether the user is in the payment zone (block 310 of Figure 9A). If the machine does not have enabled multi-vend capabilities (block 350) or the multi-vend limit has been reached, the wallet is decremented by the vend amount(s) and "change" is returned to the virtual wallet (block 354) and the process ends (block 356).

[0092] Figure 8E is a schematic flow diagram of an example login process. Figure 8F is a schematic flow diagram of an example boot-up process. Figure 8G is a schematic flow diagram of an example account check/update process.

[0093] Several of the figures are flow charts (e.g., Figures 9A-9E) illustrating methods and systems. It will be understood that each block of these flow charts, components of all or some of the blocks of these flow charts, and/or combinations of blocks in these flow charts, may be implemented by software (e.g., coding, software, computer program instructions, software programs, subprograms, or other series of computer-executable or processor-executable instructions), by hardware (e.g., processors, memory), by firmware, and/or a combination of these forms. As an example, in the case of software, computer program instructions (computer-readable program code) may be loaded onto a computer to produce a machine, such that the instructions that execute on the computer create structures for implementing the functions specified in the flow chart block or blocks. These computer program instructions may also be stored in a memory that can direct a computer to function in a particular manner, such that the instructions stored in the memory produce an article of manufacture including instruction structures that implement the function specified in the flow chart block or blocks. The computer program instructions may also be loaded onto a computer to cause a series of operational steps to be performed on or by the computer to produce a computer implemented process such that the instructions that execute on the computer provide steps for implementing the functions specified in the flow chart block or blocks. Accordingly, blocks of the flow charts support combinations of steps, structures, and/or modules for performing the specified functions. It will also be understood that each block of the

flow charts, and combinations of blocks in the flow charts, may be divided and/or joined with other blocks of the flow charts without affecting the scope of the invention. This may result, for example, in computer-readable program code being stored in whole on a single memory, or various components of computer-readable program code being stored on more than one memory.

ADDITIONAL IMPLEMENTATIONS

[0094] Figure 23 illustrates a schematic flow diagram of a process 1000 of authenticating a user to perform a transaction in the payment processing system in accordance with some implementations. In some implementations, the payment processing system includes one or more payment modules 100 (e.g., each associated with a respective payment accepting unit 120 such an automatic retailing machine for dispensing goods and/or services), one or more mobile devices 150 (e.g., each executing the application 140 for the payment processing system either as a foreground or background process), and the server 130. The server 130 manages the payment processing system and, in some cases, is associated with an entity that supplies, operates, and/or manufactures the one or more payment modules 100. For brevity, the process 1000 will be described with respect to a respective payment module 100 and a respective mobile device 150 in the payment processing system.

[0095] The payment module 100 broadcasts (1002), via a short-range communication capability (e.g., BLE), a packet of information (sometimes also herein called "advertised information"). The packet of information at least includes an authorization code and an identifier associated with the payment module 100 (module ID). In some implementations, the packet of information further includes a firmware version of the payment module 100 and one or more status flags corresponding to one or more states of the payment module 100 and/or the payment accepting unit 120. The information included in the packet broadcast by the payment module 100 is further discussed below with reference to Figure 24A.

[0096] In some implementations, the payment module 100 sends out a unique authorization code every X seconds (e.g., 100 ms, 200 ms, 500 ms, etc.). In some implementations, the unique authorization codes are randomly or pseudo-randomly generated numbers. In some implementations, the payment module 100 stores broadcasted authorization codes until a received authorization grant token matches one of the stored authorization codes. In some implementations, the payment module 100 stores broadcasted authorization codes for a predetermined amount of time (e.g., Y minutes) after which time an authorization code expires and is deleted. In some implementations, the authorization code is encrypted with a shared secret key known by the server 130 but unique to the payment module 100. In some implementations, the payment module 100 initializes a random number and then the authorization codes are sequential counts from this random number. In such implementations, the payment module 100 stores the earliest valid (unexpired) counter without a need to store every valid authorization code. In some implementations, the authentication code included in the broadcast packet of information is a hash value of the randomly or pseudo-randomly generated number or the sequential number.

[0097] The mobile device 150 receives the broadcasted packet of information, and the mobile device 150 sends (1004), via a long-range communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), an authorization request to the server 130. For example, an application 140 that is associated with the payment processing system is executing as a foreground or background process on the mobile device 150. In this example, the application 140 receives the broadcasted packet of information when the mobile device 150 is within the communication zone of the payment module 100 (i.e., BLE range) and either automatically sends the authorization request to the server 130 or sends the authorization request to the server 130 when the mobile device 150 is within the authorization zone of the payment module 100. In some implementations, the broadcasted packet of information includes a baseline authorization zone threshold (i.e., an authorization zone criterion) indicating a baseline RSSI that the mobile device 150 (or the application 140) is required to observe before being within the authorization zone of the payment module 100. In some implementations, the mobile device 150 (or the application 140) offsets the baseline authorization zone threshold based on the strength and/or reception of the short-range communication capability (e.g., BLE radio/transceiver) of the mobile device 150. In some implementations, the authorization request at least includes the authorization code which was included in the broadcasted packet of information, an identifier associated with the user of the mobile device 150 or the user account under which the user of the mobile device 150 is logged into the application 140 (user ID), and the identifier associated with the payment module 100 (module ID). In some implementations, the authentication code included in authorization request is the hash value in cleartext. The authorization request is further discussed below with reference to Figure 24B.

[0098] After receiving the authorization request, the server 130 processes (1006) the authorization request. In some implementations, the server 130 decrypts the authorization code included in the authorization request with the shared secret key corresponding to the payment module 100. In some implementations, the server 130 determines whether the user associated with the user ID in the authorization request has sufficient funds in his/her account for the payment processing system to perform a transaction at the machine 120 that is associated with the payment module 100 corresponding to the module ID in the authorization request.

[0099] The server 130 sends (1008), via a long-range communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), an authorization grant token to the mobile device

150. In some implementations, the server 130 does not send the authorization grant token if the authorization code in the authorization request cannot be decrypted with the shared secret key corresponding to the payment module 100 (e.g., the authorization code is corrupted or hacked). In some implementations, the server 130 does not send the authorization grant token if the user associated with the user ID in the authorization request does not have sufficient funds in his/her account. In some implementations, in addition to the authorization grant token, the server 130 sends a message directly to the mobile device 150 which is not encrypted with the shared secret key corresponding to the payment module 100. After receiving the message, the mobile device 150 displays an appropriate message to the user such as insufficient balance or declined authorization. In some implementations, the server 130 sends an authorization grant token for an amount equal to zero; in which case, the payment module 100 interprets this as a declined or failed authorization which can result for any number of reasons including, but not limited to, insufficient balance or credit.

[0100] The mobile device 150 receives the authorization grant token, and, subsequently, the mobile device 150 detects (1010) a trigger condition. In some implementations, the mobile device 150 (or the application 140) detects the trigger condition via the hand-free mode (e.g., upon entrance into the payment zone of the payment module 100) or manual mode (e.g., interacting with the user interface of the application 140 to initiate a transaction with the payment accepting unit associated with the payment module 100).

[0101] In some implementations, unused authorization grants (e.g., if there was no trigger condition or it expired) are canceled by the mobile device 150 by sending a cancellation message to the server 130 corresponding to the unused authorization grant. In some implementations, the server 130 denies or limits the number of authorization grants sent to the mobile device 150 until it has received transaction information or cancellation of authorization outstanding authorization grants sent to the mobile device 150.

[0102] In response to detecting the trigger condition, the mobile device 150 sends (1012), via a short-range communication capability (e.g., BLE), the authorization grant token to the payment module 100. Subsequently, the machine 120 displays credit to the user (e.g., via one of the displays 122 or 124 shown in Figure 19) and the user interacts with the input mechanisms of the machine 120 (e.g., via the buttons 126 or a touch screen display 124 shown in Figure 19) to purchase products and/or services.

[0103] Figure 24A illustrates a block diagram of a packet 1100 of information broadcast by the payment module 100 (e.g., in step 1002 of the process 1000 in Figure 23) in accordance with some implementations. In some implementations, the packet 1100 at least includes: module ID 1102 and authorization code 1104. In some implementations, the packet 110 additional includes: a firmware version 1106 and one or more status flags 1108.

[0104] In some implementations, the module ID 1102 is a unique identifier corresponding to the payment module 100 (sometimes also herein called the "adapter module 100") that broadcast the packet 1100.

[0105] In some implementations, the authorization code 1104 is a hash value in cleartext. In some implementations, the payment module 100 randomly or pseudo-randomly generates a number or determines a sequential number (*See* step 1002 of process 1000 in Figure 23) and performs a predetermined hash function (e.g., SHA-256) on the number to produce the hash value as the authorization code 1104. In some implementations, the authorization code 1104 is a unique code that is encrypted with a secret encryption key corresponding to the payment module 100. The secret encryption key is shared with the server 130, which enables the server 130 to decrypt the authorization code 1104 and encrypt the authorization grant token but not the mobile device 150. In some implementations, the encryption between server 130 and payment module 100 is accomplished by two pairs of public/private keys.

[0106] In some implementations, the firmware version information 1106 identifies a current firmware version 1112 of the payment module 100. In some implementations, the firmware version information 1106 also includes update status information 1114 indicating one or more packets received by the payment module 100 to update the firmware or one or more packets needed by the payment module 100 to update the firmware. See Figures 26A-26B and 30A-30D and the accompanying text for further discussion regarding updating the firmware of the payment module 100.

[0107] In some implementations, the one or more status flags 1108 indicate a state of the payment module 100 and/or the payment accepting unit 120 associated with the payment module 100. In some implementations, the one or more status flags 1108 indicate a state of the payment module 100 such upload information indicator 1116 indicating that that the payment module 100 has information to be uploaded to the server 130 (e.g., transaction information for one or more interrupted transactions). In some implementations, upload information indicator 1116 triggers the mobile device 150 to connect to payment module 100 immediately (e.g., if it has interrupted transaction information to be uploaded to the server 130). See Figures 25A-25B and 29A-29C and the accompanying text for further discussion regarding interrupted transactions. In some implementations, the one or more status flags 1108 indicate a state of the payment accepting unit 120 including one or more of an error indicator 1118 (e.g., indicating that a bill and/or coin acceptor of the payment accepting unit 120 is experiencing a jam, error code, or malfunction), a currency level indicator 1120 (e.g., indicating that the level of the bill and/or coin acceptor reservoir of the payment accepting unit 120 is full or empty), and/or inventory level(s) indicator 1122 (e.g., indicating that one or more products of the payment

accepting unit 120. In some implementations, the one or more status flags 1108 are error codes issued by payment accepting unit 120 over the MDB.

[0108] In some implementations, the zone criteria information 1110 specifies an authorization zone criterion 1124 (e.g., a baseline authorization zone threshold indicating a baseline RSSI that the mobile device 150 (or the application 140) is required to observe before being within the authorization zone of the payment module 100) and/or a payment zone criterion 1126 (e.g., a baseline payment zone threshold indicating a baseline RSSI that the mobile device 150 (or the application 140) is required to observe before being within the payment zone of the payment module 100). In some implementations, the baseline authorization zone threshold and the baseline payment zone threshold are default values determined by the server 130 or stored as variables by the application 140, in which case the authorization zone criterion 1124 and payment zone criterion 1126 are offsets to compensate for the strength and/or reception of the short-range communication capability (e.g., BLE radio/transceiver) of the payment module 100. Alternatively, zone criteria information 1110 includes a spread between the baseline authorization zone threshold and the baseline payment zone threshold. Thus, the mobile device 150 (or the application 140) determines the baseline authorization zone threshold and the baseline payment zone threshold based on the spread value and a default value for either the baseline authorization zone threshold or the baseline payment zone threshold. For example, the spread indicates -10 db and the default baseline payment zone threshold is -90 db; thus, the baseline authorization zone threshold is -80 db. Continuing with this example, after determining the baseline authorization zone threshold and the baseline payment zone threshold, the mobile device 150 (or the application 140) may further adjust the authorization zone threshold and/or the payment zone threshold based on the strength and/or reception of its short-range communication capability (i.e., BLE radio/transceiver).

[0109] Figure 24B is a block diagram of an authorization request 1130 sent by the mobile device 150 to the server 130 (e.g., in step 1004 of the process 1000 in Figure 23) in accordance with some implementations. In some implementations, the authorization request 1130 at least includes: a module ID 1102, a user ID 1134, and an authorization code 1104.

[0110] In some implementations, the module ID 1102 is a unique identifier corresponding to the payment module 100 that broadcast the 1100 that included the authorization code 1104.

[0111] In some implementations, the user ID 1134 is an identifier associated with the user of the mobile device 150 sending the authorization request 1130 to the server 130. In some implementations, the user ID 1134 is associated with the user account under which the user of the mobile device 150 is logged into the application 140.

[0112] In some implementations, the authorization code 1130 includes the authorization code 1104 included in the packet 1100 of information that was broadcast by the payment module 100.

[0113] Figure 24C is a block diagram of an authorization grant token 1140 sent by the server 130 to the mobile device 150 (e.g., in step 1008 of the process 1000 in Figure 23) in accordance with some implementations. In some implementations, in accordance with a determination that the authorization code 1136 included in the authorization request 1130 from the mobile device 150 is valid and that the user associated with the mobile device 150 has sufficient funds in his/her account for the payment processing system, the server 130 generates the authorization grant token 1140. In some implementations, the authorization grant token 1140 at least includes: a module ID 1102, a user ID 1134, an authorized amount 1146, (optionally) an expiration period offset 1148, and (optionally) the authorization code 1104.

[0114] In some implementations, the module ID 1102 is a unique identifier corresponding to the payment module 100 that broadcast the packet 1100 that included the authorization code 1104.

[0115] In some implementations, the user ID 1134 is an identifier associated with the user of the mobile device 150 that sent the authorization request 1130 to the server 130.

[0116] In some implementations, the authorized amount 1146 indicates a maximum amount for which the user of the mobile device 150 is authorized for a transaction using the authorization grant token 1140. For example, the authorized amount 1146 is predefined by the user of the mobile device 150 or by the server 130 based on a daily limit or based on the user's total account balance or based on a risk profile of the user correspond to the user ID 1134.

[0117] In some implementations, the expiration period 1148 offset indicates an offset to the amount of time that the payment module 100 holds the authorization grant token 1140 valid for initiation of a transaction with the machine 120 associated with the payment module 100. For example, the expiration period offset 1148 depends on the history and credit of the user of mobile device 150 or a period predefined by the user of mobile device 150.

[0118] In some implementations, the authorization grant token 1140 further includes the authorization code 1104 that was included in the authorization request 1130. In some implementations, when the authorization code 1104 is the hash value, the server 130 encrypts the authorization grant token 1140 including the hashed value with the shared secret encryption key associated with payment module 100. Subsequently, when mobile device 150 sends the authorization grant token 1140 to payment module 100 after detecting a trigger condition, the payment module 100 decrypts the authorization grant token 1140 using the secret key known only to server 130 and payment module 100 (which authenticates the message and the authorization grant), and then matches the hash value included in the decrypted authorization

grant token 1140 to previously broadcast valid (unexpired) hash values (i.e., auth codes) to determine validity of the (which was known only by payment module 100).

[0119] Figure 24D illustrates a block diagram of transaction information 1150 generated by the payment module 100 (e.g., in step 1204 of the process 1200 in Figure 25A) in accordance with some implementations. In some implementations, the transaction information 1150 includes: a transaction ID 1152 for the respective transaction, a module ID 1154, a user ID 1156, (optionally) the authorization code 1158, transaction status information 1160, the transaction amount 1162, and other information 1164.

[0120] In some implementations, the transaction ID 1152 is a unique identifier corresponding to the respective transaction. In some implementations, the transaction ID 1152 is encoded based on or associated with the time and/or date on which and the location at which the respective transaction took place.

[0121] In some implementations, the module ID 1154 is a unique identifier corresponding to the payment module 100 that performed the respective transaction.

[0122] In some implementations, the user ID 1156 is an identifier associated with the user of the mobile device 150 that initiated the respective transaction.

[0123] In some implementations, the authorization code 1158 corresponds to the original authorization code (e.g., auth code 1104, Figures 24A-24C) and/or authorization grant token (e.g., auth grant token 1140, Figure 24C) that was used to initiate the respective transaction. In some implementations, the authorization code 1156 is encrypted with a unique encryption key corresponding to the payment module 100.

[0124] In some implementations, the transaction status information 1160 includes an indication whether the respective transaction was completed, not-completed, or aborted. For example, the respective transaction is incomplete if a jam occurred at the payment accepting unit 120 and the user did not receive the product associated with the respective transaction. For example, if the user walks away from the payment accepting unit 120 after money was credited for the respective transaction, the respective transaction is aborted. In another example, if respective transaction times out after a predetermined time period because the user failed to select a product at the payment accepting unit 120, the respective transaction is aborted. In another example, if the user actuates a bill or coin return mechanism of the payment accepting unit 120, the respective transaction is aborted.

[0125] In some implementations, the transaction amount 1162 indicates the amount of the respective transaction or the amount of each of multiple transactions (e.g., in a multi-vend scenario). In some implementations, the transaction amount 1162 is encrypted with a unique encryption key corresponding to the payment module 100.

[0126] In some implementations, the other information 1164 includes other information related to the respective transaction such as the items dispensed by the payment accepting unit 120 and the type of transaction (e.g., coins, bills, credit card, manual mode, hands-free mode, etc.). In some implementations, the other information 1164 includes other information related to the payment module 100 and/or the payment accepting unit 120 associated with the payment module 100. For example, the other information 1164 includes a verification request to the server 130 in order to implement new firmware. See Figures 26A-26B and the accompanying text for further discussion of the verification request. In another example, the other information 1164 includes transaction information from one or more previous interrupted transactions. In another example, the other information 1164 includes transaction information for one or more transactions paid via bills and/or coins. In another example, the other information 1164 includes inventory information as to one or more products of the payment accepting unit 120.

[0127] Figure 25A illustrates a schematic flow diagram of a process 1200 of processing acknowledgement information in accordance with some implementations. In some implementations, the payment processing system includes one or more payment modules 100 (e.g., each associated with a respective payment accepting unit 120 such an automatic retailing machine for dispensing goods and/or services), one or more mobile devices 150 (e.g., each executing the application 140 for the payment processing system either as a foreground or background process), and the server 130. The server 130 manages the payment processing system and, in some cases, supplies, operates, and/or manufactures the one or more payment modules 100. For brevity, the process 1200 will be described with respect to a respective payment module 100 associated with a respective payment accepting unit 120 (machine 120) and a respective mobile device 150 in the payment processing system. In the process 1200, the payment module 100 receives first acknowledgment information for a first transaction via the mobile device 150 that initiated the first transaction.

[0128] The payment module 100 obtains (1202) a first notification indicating completion of a first transaction from the machine 120. For example, after the process 1000 in Figure 23, the user of the mobile device 150 selects a product to purchase from the machine 120 by interacting with one or more input mechanisms of the machine 120 (e.g., buttons 126 or a touch screen display 124 shown in Figure 19), and the machine 120 dispenses the selected product. Continuing with this example, after the product is dispensed, the transaction is complete and the payment module 100 obtains a notification from the machine of the completed transaction. In some implementations, the notification includes the amount of the transaction and (optionally) machine status information associated with the machine 120 such as inventory information as to one or more products of the payment accepting unit 120 and/or the like.

[0129] After obtaining the first notification, the payment module 100 generates (1204) first transaction informa-

tion based on the first notification, and the payment module 100 stores the first transaction information. In some implementations, the transaction information includes a transaction ID for the first transaction, a module ID corresponding to payment module 100, a user ID corresponding to the mobile device 150, transaction status information indicating that the first transaction is complete, and the transaction amount indicated by the first notification. In some implementations, the payment module 100 retains the authorization code included in the original broadcasted packet and/or the authorization grant token and includes the authorization code in the first transaction information. In some implementations, the authorization code is encrypted with a secret key corresponding to the payment module 100, which is shared with the server 130 but not the mobile device 150. In some implementations, the first transaction information further includes other information such as the machine status information included in the first notification or transaction information corresponding to previous interrupted transaction(s). See Figure 24D and the accompanying text for further discussion regarding transaction information 1150.

[0130] The payment module 100 sends (1206), via a short-range communication capability (e.g., BLE), the first transaction information to the mobile device 150.

[0131] The mobile device 150 sends (1208), via a long-range communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), the first transaction information to the server 130.

[0132] The server 130 processes (1210) the first transaction information. For example, the server 130 debits the account of the user associated with the user ID in the first transaction information in the amount indicated by the first transaction information.

[0133] The server 130 sends (1212), via a long-range communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), first acknowledgment information to the mobile device 150. In some implementations, the first acknowledgment information acknowledges that the server 130 received the first transaction information. In some implementations, the first acknowledgment information includes the user ID, the module ID, the transaction ID, and (optionally) the authorization grant included in the transaction information (e.g., auth grant 1158, Figure 24D).

[0134] After receiving the first acknowledgement information, the mobile device 150 sends (1214), via a short-range communication capability (e.g., BLE), the first acknowledgment information to the payment module 100.

[0135] After receiving the first acknowledgment information, the payment module 100 deletes (1216) the stored first transaction information.

[0136] Figure 25B illustrates a schematic flow diagram of a process 1250 of processing interrupted transactions in accordance with some implementations. In some implementations, the payment processing system includes one or more payment modules 100 (e.g., each associated with a respective payment accepting unit 120 such an automatic retailing machine for dispensing goods and/or services), one or more mobile devices 150 (e.g., each executing the application 140 for the payment processing system either as a foreground or background process), and the server 130. The server 130 manages the payment processing system and, in some cases, supplies, operates, and/or manufactures the one or more payment modules 100. For brevity, the process 1250 will be described with respect to a respective payment module 100 associated with a respective payment accepting unit 120 (machine 120) and a respective mobile device 150 in the payment processing system. In the process 1250, the payment module 100 receives first acknowledgment information for a first transaction via a second mobile device 150-2 that did not initiate the first transaction.

[0137] After receiving a first authorization request associated with a first authorization code from a first mobile device 150-1, the server 130 sends (1252), via a long-range communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), a first authorization grant token to the first mobile device 150-1 associated with a first user 1251-1.

[0138] After receiving the first authorization grant token and in response to detecting a trigger condition (e.g., via the hand-free mode or the manual mode), the first mobile device 150-1 sends (1254), via a short-range communication capability (e.g., BLE), the first authorization grant token to the payment module 100 associated with the machine 120 in order to initiate a first transaction.

[0139] The payment module 100 processes (1256) the first transaction associated with the first authorization grant token and generates first transaction information when the first transaction is completed. In some implementations, the first transaction information includes a transaction ID for the first transaction, a module ID corresponding to payment module 100, a user ID corresponding to the first mobile device 150-1, transaction status information indicating that the first transaction is complete, and the transaction amount for the first transaction. The payment module 100 stores the first transaction information with a timestamp indicating the time and date that the first transaction information was generated.

[0140] The payment module 100 sends (1258), via a short-range communication capability (e.g., BLE), the first transaction information to the first mobile device 150-1 to send to the server 130 in order to acknowledge the first transaction.

[0141] In accordance with a determination that first acknowledgement information is not received for the first transaction within a predefined time period, the payment module 100 times-out (1260) the first transaction and maintains the first transaction information. In some implementations, a transaction times-out when the connection between the mobile device and the payment module is interrupted and transaction information is not acknowledged within a predefined time period.

[0142] For example, the connection between the first mobile device 150-1 and the payment module 100 is in-

terrupted when the first user 1251-1 turns off the first mobile device 150-1, the first user 1251-1 turns the first mobile device 150-1 into airplane mode, the first user 1251-1 walks away out of the communication zone (i.e., BLE range) of the payment module 100, the first mobile device 150-1 otherwise loses its long-range communication connection, or the first mobile device 150-1 otherwise loses power. In this example, either the first user 1251-1 maliciously interrupted the connection to prevent the acknowledgement information from being received by the payment module 100 by powering down the first mobile device 150-1, or the connection was involuntarily or unintentionally interrupted by the first mobile device 150-1's battery running out or a losing cellular signal.

[0143] In some implementations, the first user 1251-1 is be blocked by the payment module 100 from performing any additional transactions until the payment module 100 receives an acknowledgement from the server 130 via any connection (e.g., from the second user 1251-2). In some implementations, unused authorization grants (e.g., if there was no trigger condition or it expired) are canceled by the first mobile device 150-1 by sending a cancellation message to the server 130 corresponding to the unused authorization grant. In some implementations, the server 130 denies or limits the number of authorization grants sent to the first mobile device 150-1 until it has received transaction information or cancellation of authorization outstanding authorization grants sent to the first mobile device 150-1. In some implementations, server 130 denies approval of, or limit the number of, additional authorization grants from user 1251-1 for transacting with a second payment module (not shown) until the server 130 receives transaction information, cancellation of authorization, or a predefined time period has expired for outstanding authorization grants sent to the first mobile device 150-1 for transacting with a first payment module. In this example, a user may be limited to only 1 authorization grant for the first payment module 100 and no more than 3 outstanding authorization grants in a predetermined number of hours regardless of the number of payment modules the user may be attempting to use.

[0144] After receiving a second authorization request associated with a second authorization code from a second mobile device 150-2 subsequent to receiving the first authorization request from the first mobile device 150-1, the server 130 sends (1262), via a long-range communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), a second authorization grant token to the second mobile device 150-2 associated with a second user 1251-2.

[0145] After receiving the second authorization grant token and in response to detecting a trigger condition (e.g., via the hand-free mode or the manual mode), the second mobile device 150-2 sends (1264), via a short-range communication capability (e.g., BLE), the second authorization grant token to the payment module 100 associated with the machine 120 in order to initiate a second transaction.

[0146] The payment module 100 processes (1266) the second transaction associated with the second authorization grant token and generates second transaction information when the second transaction is completed. In some implementations, the second transaction information includes a transaction ID for the second transaction, a module ID corresponding to payment module 100, a user ID corresponding to the second mobile device 150-2, transaction status information indicating that the second transaction is complete, and the transaction amount for the second transaction. The payment module 100 stores the second transaction information with a timestamp indicating the time and date that the second transaction information was generated.

[0147] The payment module 100 sends (1268), via a short-range communication capability (e.g., BLE), the first transaction information associated with the interrupted first transaction and the second transaction information associated with the second transaction to the second mobile device 150-1 to send to the server 130 in order to acknowledge the first and second transactions. In this way, the first transaction information associated with the previous, interrupted first transaction initiated by the first mobile device 150-1 is appended to the second transaction information for the second transaction initiated by the second mobile device 150-2.

[0148] The second mobile device 150 sends (1270), via a long-range communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), the first transaction information and the second transaction information to the server 130.

[0149] After receiving the first transaction information and the second transaction information, the server 130 processes the first transaction information and the second transaction information. For example, the server 130 debits the account of the first user 1251-1 associated with the user ID for first mobile device 150-1 in the first transaction information in the amount indicated by the first transaction information. Continuing with this example, the server 130 also debits the account of the second user 1251-2 associated with the user ID for second mobile device 150-2 in the second transaction information in the amount indicated in the second transaction information.

[0150] The server 130 sends (1272), via a long-range communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), first and second acknowledgment information to the second mobile device 150-2 acknowledging the first and second transactions. In some implementations, the first acknowledgment information includes the user ID of the first mobile device 150-1 that imitated the first transaction, the module ID of the payment module 100 that processed the first transaction, the transaction ID of the first transaction, and (optionally) the authorization code associated with the first transaction. In some implementations, the second acknowledgment information includes the user ID of the second mobile device 150-2 that imitated the second transaction, the module ID of the payment module 100 that processed the second

transaction, the transaction ID of the second transaction, and (optionally) the authorization code associated with the second transaction.

[0151] After receiving the first and second acknowledgment information, the mobile device 150 sends (1274), via a short-range communication capability (e.g., BLE), the first acknowledgment information to the payment module 100.

[0152] After receiving the first and second acknowledgment information, the payment module 100 deletes (1276) the stored first transaction information and also the stored second transaction information. In some implementations, the payment module 100 marks the first and second transaction as complete.

[0153] Figure 26A is a schematic flow diagram of a process 1300 of updating firmware of the payment module 100 in the payment processing system in accordance with some implementations. In some implementations, the payment processing system includes one or more payment modules 100 (e.g., each associated with a respective payment accepting unit 120), one or more mobile devices 150 (e.g., each executing the app 140 for the payment processing system either as a foreground or background process), and the server 130. The server 130 manages the payment processing system and, in some cases, supplies, operates, and/or manufactures the one or more payment modules 100. For brevity, the process 1300 will be described with respect to a respective payment module 100 and a respective mobile device 150 in the payment processing system.

[0154] The payment module 100 broadcasts (1302), via a short-range communication capability (e.g., BLE), a packet of information (e.g., broadcast packet 1100, Figure 24A). The packet of information at least includes a firmware version (e.g., current firmware version 1112, Figure 24A) of the payment module 100. The information included in the packet broadcasted by the payment module 100 is further discussed herein with reference to Figure 24A.

[0155] The mobile device 150 determines (1304) that the current firmware version of the payment module 100 satisfies firmware criteria (e.g., predates a firmware version stored by the mobile device 150). Various other firmware criteria are further discussed below with reference to the method 1700 in Figures 30A-30D.

[0156] In accordance with a determination that the firmware criteria are satisfied, the mobile device 150 sends (1306) firmware update information (e.g., data packets corresponding to the firmware of the mobile device 150) to the payment module 100.

[0157] The payment module 100 broadcasts (1308) update status information (e.g., update status information 1114 in Figure 24A, identifying remaining data packets needed for the firmware update) included in the advertised information to the one or more mobile devices in the payment processing system (e.g., at least including the respective mobile device 150). Although not illustrated, the process 1300 sometimes includes a second mobile device, which sends firmware update information that includes additional data packets distinct from the data packets sent by the respective mobile device 150.

[0158] When all needed data packets have been received by the payment module 100, the update status information includes a verification request, which the mobile device 150 then sends (1310) to the server 130 via a long-range communication capability (e.g., GSM).

[0159] The server 130 processes (1312) the verification request. For example, the server 130 processes the verification request by verifying that the received data packets are not corrupt, form a complete set, and correspond to a latest firmware version.

[0160] After processing the verification request, the server 130 sends (1314) to the mobile device 150 a firmware command (e.g., implement the firmware update at the payment module 100) via the long-range communication capability, which the mobile device 150 then sends (1316) to the payment module 100 via the short-range communication capability.

[0161] The payment module 100 then executes (1318) the firmware command. For example, the payment module implements the firmware update using the received data packets corresponding to a latest firmware version.

[0162] Figure 26B is a schematic flow diagram of a process 1320 of updating firmware of the payment module 100 in the payment processing system in accordance with some implementations. In some implementations, the payment processing system includes one or more payment modules 100 (e.g., each associated with a respective payment accepting unit 120), one or more mobile devices 150 (e.g., each executing the app 140 for the payment processing system either as a foreground or background process), and the server 130. The server 130 manages the payment processing system and, in some cases, supplies, operates, and/or manufactures the one or more payment modules 100. For brevity, the process 1320 will be described with respect to a respective payment module 100 and a respective mobile device 150 in the payment processing system.

[0163] The payment module 100 broadcasts (1322), via a short-range communication capability (e.g., BLE), a packet of information (e.g., broadcast packet 1100, Figure 24A). The packet of information at least includes a firmware version (e.g., current firmware version 1112, Figure 24A) of the payment module 100. The information included in the packet broadcasted by the payment module 100 is further discussed herein with reference to Figure 24A.

[0164] The mobile device 150 then sends (1324) to the server 130, via a long-range communication capability (e.g., GSM), the packet of information that at least includes the firmware version of the payment module 100.

[0165] The server 130 determines (1326) that current firmware version of the payment module 100 satisfies firmware criteria (e.g., predates a firmware version stored by the mobile device 150). Various other firmware criteria are further discussed below with reference to the method

1700 in Figures 30A-30D.

[0166] In accordance with a determination that the firmware criteria are satisfied, the server 130 sends (1328) to the mobile device 150 firmware update information (e.g., data packets corresponding to the firmware of the mobile device 150), which the mobile device 150 then sends (1330) to the payment module 100.

[0167] The payment module 100 broadcasts (1332) update status information (e.g., identification of remaining data packets needed for the firmware update) included in the advertised information to the one or more mobile devices in the payment processing system (e.g., at least including the respective mobile device 150), which the one or more mobile devices 150 then send (1334) to the server 130. When all needed data packets have been received by the payment module 100, the update status information includes a verification request.

[0168] The server 130 processes (1336) the verification request. For example, the server 130 processes the verification request by verifying that the received data packets are not corrupt, form a complete set, and correspond to a latest firmware version.

[0169] After processing the verification request, the server 130 sends (1338) to the mobile device 150 a firmware command (e.g., implement the firmware update at the payment module 100) via the long-range communication capability, which the mobile device 150 then sends (1340) to the payment module 100 via the short-range communication capability.

[0170] The payment module then executes (1342) the firmware command. For example, the payment module implements the firmware update using the received data packets corresponding to a latest firmware version.

[0171] Figure 26C is a schematic flow diagram of a process 1350 of updating firmware of the payment module 100 in the payment processing system in accordance with some implementations. In some implementations, the payment processing system includes one or more payment modules 100 (e.g., each associated with a respective payment accepting unit 120), one or more mobile devices 150 (e.g., each executing the app 140 for the payment processing system either as a foreground or background process), and the server 130. The server 130 manages the payment processing system and, in some cases, supplies, operates, and/or manufactures the one or more payment modules 100. For brevity, the process 1350 will be described with respect to a respective payment module 100 and a respective mobile device 150 in the payment processing system.

[0172] The payment module 100 broadcasts (1352), via a short-range communication capability (e.g., BLE), a packet of information (e.g., broadcast packet 1100, Figure 24A). The packet of information at least includes a firmware version (e.g., current firmware version 1112, Figure 24A) of the payment module 100. The information included in the packet broadcast by the payment module 100 is further discussed herein with reference to Figure 24A.

[0173] The mobile device 150 determines (1354) that the current firmware version of the payment module 100 satisfies firmware criteria (e.g., predates a firmware version stored by the mobile device 150). Various other firmware criteria are further discussed below with reference to the method 1700 in Figures 30A-30D. In some implementations, the mobile device 150 stores a firmware image for the payment module 100. For example, the firmware image was previously downloaded by the mobile device 150 from the server 130 as part of an update for application 140. In some implementations, the firmware image downloaded by the mobile device 150 is encrypted with a common encryption key known to all payment modules 100 in the payment processing system (as opposed to the unique encryption key corresponding to each payment module 100 in the payment processing system). In some implementations, the firmware image downloaded by the mobile device 150 is encrypted with an encryption key that is later sent as part of the firmware approval message in steps 1360 and 1362, where the firmware approval message is encrypted with a unique encryption key corresponding to the payment module 100.

[0174] In accordance with a determination that the firmware criteria are satisfied, the mobile device 150 sends (1356), via a second communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), a firmware update request (e.g., a request for permission to update the firmware of the payment module 100) to the server 130. In some embodiments, the firmware update request includes a module ID corresponding to the payment module 100, a user ID associated with a user of the mobile device 150, the current firmware version 1112 of the payment module 100, and the firmware version stored by the mobile device 150.

[0175] The server 130 processes (1358) the firmware update request. The server 130 determines whether to permit or decline the firmware update request. If the server 130 permits the firmware update request, the mobile device 150 updates the firmware version of the payment module 100 with the firmware version stored by the mobile device. For example, the server 130 declines the firmware update request if the firmware version stored by the mobile device 150 is out of date (i.e., a firmware version C, distinct from a firmware version A of the payment module 100 and a firmware version B stored by the mobile device 150, is the latest firmware image). In another example, the server 130 declines the firmware update request if the firmware version stored by the mobile device 150 (e.g., firmware version B) is determined to be faulty and/or blacklisted, even if a latest firmware (e.g., firmware version C, distinct from firmware version A and firmware version B) is not yet available.

[0176] In accordance with a determination that the firmware version stored by the mobile device 150 is approved by the server 130, the server 130 sends (1360) to the mobile device 150, via the second communication capability, a firmware update approval message (e.g.,

permission to update the firmware of the payment module 100), which the mobile device 150 then sends (1362) to the payment module 100 via the short-range communication capability. In some implementations, in accordance with a determination that the server 130 permits the firmware update request, the server 130 responds to the application 140 of the mobile device 150 with an affirmative firmware update approval message (e.g., approval to update firmware of payment module 100) to be sent to the payment module 100. In some implementations, the firmware update approval message contains one or more verification values (e.g., a list of checksum values for each 4 KB block of encrypted firmware image stored by the mobile device 150) and a hash value (e.g., SHA-256 hash of the complete, decrypted firmware image) for data packets corresponding to the approved firmware update version. Furthermore, in some implementations, the firmware update approval message includes a firmware decryption key (e.g., for decrypting received data packets corresponding to the approved firmware update version). In some implementations, the firmware update approval message is encrypted using a unique encryption key corresponding to the payment module 100.

[0177] In some implementations, if the payment module 100 is already updating its firmware when it receives the firmware update approval message (e.g., a process which may have been started by a different mobile device 150 at an earlier time), then the payment module 100 simply verifies the validity of the firmware update approval message and resumes the firmware update. However, if the payment module 100 was not already updating its firmware, it will verify the validity of the firmware update approval message, store to the memory 760 (Figure 20) (e.g., EEPROM) the one or more verification values and the hash value (e.g., list of checksums and a SHA-256 hash), and erase the firmware-update area of the memory 760 (e.g., where the one or data packets for the firmware update will be stored). In these cases, the payment module 100 subsequently receives firmware update formation (e.g., one or more data packets) from the mobile device 150 which are stored in the firmware-update area of the memory 760. Furthermore, in some implementations, if the payment module 100 already has a stored firmware update to the specified firmware version ready to be processed, but it has not yet rebooted to install it, the firmware update request is ignored.

[0178] After the server 130 approves the firmware update request, the mobile device 150 sends (1364) to the payment module 100, via the short-range communication capability, firmware update information (e.g., data packets corresponding to the firmware stored by the mobile device 150). As long as the payment module 100 is connected to the mobile device 150, the payment module 100 will identify one or more data blocks (i.e., corresponding to the approved firmware update version) that are still needed, which are sent by the mobile device 150 as the one or more data packets. In some implementations, after receiving the firmware update approval message, the payment module 100 sends to the mobile device 150 update status information identifying and requesting one or more data blocks still needed for the firmware update (e.g., a specific 256 B block/chunk of firmware). Additionally and/or alternatively, the packet of information (e.g., broadcast packet 1100, Figure 24A) broadcast by the payment module 100 includes information identifying one or more data blocks still needed by the payment module 100 for the firmware update or one or more data blocks already received by the payment module 100. After receipt of the one or more data packets, the firmware update information is stored by the payment module 100 into memory (e.g., the memory 760, Figure 20).

[0179] The payment module 100 verifies (1366) the firmware update information. In some implementations, each time a data packet is received from the mobile device 150 corresponding to a complete data block (e.g., a 4 KB block) of the firmware update, the payment module 100 will compare a generated verification value (e.g., a checksum of the 4 KB block) against a corresponding verification value for the block that was included in the update approval message (e.g., a checksum from the list included in the firmware update approval message). If the verification values do not match, the corresponding data block is erased, and the update process resumes from that point (e.g., the particular 4 KB block that did not pass verification).

[0180] After verifying the firmware update information, the payment module 100 executes (1368) the firmware update information. After all data blocks have been received by the payment module 100, and their verification values (e.g., checksums) have been successfully verified, in some implementations, the payment module 100 sets an internal flag indicating that it should reboot itself when it determines it is a safe time to do so. In some implementations, a safe time for rebooting is based on the current time of day (e.g., 2:00 AM), or observed activity of the payment accepting unit 120 (e.g., when no user has connected in the past 10 minutes).

[0181] In some implementations, when the payment module 100 decides to reboot, it sets an install-firmware flag in memory (e.g., EEPROM) and resets itself. Upon reboot of the payment module 100, a bootloader observes the set install-firmware flag and executes an associated firmware installation handler. In some implementations, the firmware installation handler double checks all of the block checksums of the firmware update information (e.g., a firmware update image). If the checksums do not match, an error has occurred (e.g., corrupted data) and the update is aborted, with the currently installed firmware then booting up. If the checksums do match, however, the bootloader erases the currently installed firmware and then decrypts the firmware update information (e.g., a firmware update image) into the installed firmware area of memory (e.g., memory 760, Figure 20).

[0182] After the firmware update information has been decrypted, the bootloader computes a hash value (e.g.,

a SHA-256 hash) of the firmware update information (e.g., of a firmware update image) and compares it to the hash value received from the server 130 that was included in the update approval message. If the hash values do not match, an error has occurred, causing the bootloader to erase the installed firmware and re-install a default (i.e., gold master) firmware image, as that is the only image available to install at that point. Finally, the bootloader loads and runs the installed firmware (either the updated firmware version or the gold master).

[0183] Figures 27A-27C illustrate a flowchart diagram of a method 1400 of payment processing in accordance with some implementations. In some implementations, the method 1400 is performed by a device with one or more processors, memory, and two or more communication capabilities. For example, in some implementations, the method 1400 is performed by the mobile device 150 (Figures 5 and 21) or a component thereof (e.g., application 140). In some implementations, the method 1400 is governed by instructions that are stored in a non-transitory computer readable storage medium (e.g., the memory 860, Figure 21) and the instructions are executed by one or more processors (e.g., the processing unit 840, Figure 21) of the device. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders).

[0184] The device obtains (1402), from a payment module, advertised information via a first communication capability, where the advertised information at least includes an authorization code. In some implementations, the payment module 100 broadcasts/advertises a packet of information (i.e., the advertised information such as the packet 1100, Figure 24A) via one or more short-range communication protocols such as BLE, NFC, and/or the like (i.e., a non-persistent communication channel). As such, the payment module 100 is not tied up in handshakes with each mobile device 150 within its communication zone. In some implementations, the application 140 associated with the payment processing system, which is executed on the mobile device 150 (e.g., a mobile phone), receives the packet when the mobile device 150 is within the communication zone (i.e., BLE range) of the payment module 100.

[0185] In some implementations, the advertised information is a packet with a module identifier (ID) associated with the payment module 100, an authorization code, the payment module 100's current firmware version, and a plurality of status flags associated with a state of the payment accepting unit 120 and/or the payment module 100. For example, Figure 24A illustrates the packet 1100 of information that is broadcast by the payment module 100. In some implementations, the authorization code is a cleartext hash value. In some implementations, the authorization code is encrypted with a unique encryption key corresponding to the payment module 100. In some implementations, the packet also includes customized or baseline thresholds for the authorization and payment zones (e.g., RSSI values such as -80 db and -90 db, respectively). In some implementations, the packet also includes a request (e.g., a status flag) for mobile device 150 to connect to it immediately so as to upload information to the server 130 (e.g., transaction information for one or more interrupted transaction). In some implementations, the payment module 100 broadcasts the advertised information every X second with a unique authorization code.

[0186] In some implementations, the first communication capability corresponds (1404) to a short-range communication protocol. For example, the first communication capability of the mobile device 150 is a radio/transceiver means for communicating via one or more short-range communication protocols such as BLE, NFC, and/or the like (i.e., a non-persistent communication channel).

[0187] The device sends (1406), to a server, at least the authorization code from the advertised information via a second communication capability distinct from the first communication capability. In some implementations, the mobile device 150 sends an authorization request to the server 130 that at least includes the authorization code from the obtained advertised information, the user ID corresponding to the user of the mobile device 150, and the module ID corresponding to the payment module 100. For example, see authorization request 1130 in Figure 24B.

[0188] In some implementations, the second communication capability corresponds (1408) to a long-range communication protocol. For example, the second communication capability of the mobile device 150 is a radio/transceiver means for communicating via one or more long-range communication protocols such as Wi-Fi, CDMA, GSM, and/or the like (i.e., a non-persistent communication channel).

[0189] In some implementations, the advertised information further includes (1410) an authorization zone threshold criterion, and the device sends at least the authorization request code comprises sending, to the server, at least the authorization request code via the second communication capability in accordance with a determination that the authorization zone threshold criterion is satisfied. In some implementations, the advertised information includes a baseline authorization zone threshold (i.e., an authorization zone criterion) indicating a baseline RSSI that the mobile device 150 (or the application 140) is required to observe before being within the authorization zone of the payment module 100. In some implementations, the mobile device 150 (or the application 140) offsets the baseline authorization zone threshold based on the strength and/or reception of the short-range communication capability (e.g., BLE radio/transceiver) of the mobile device 150. In some implementations, the mobile device 150 forwards the authorization code to the server 130 when the authorization zone criterion is satisfied (i.e., the mobile device 150 observes an RSSI equal to or exceeding the baseline authorization zone threshold). For example, baseline authorization zone threshold

for a payment module associated with module ID 0xA23 is -70 db. Continuing with this example, the mobile device 150 (or the application 140) offsets the baseline authorization zone threshold by -5 db because the mobile device 150's BLE radio/transceiver is weak. Continuing with this example, when the mobile device 150 observes an RSSI equal to or exceeding -75 db from payment module 100 associated with module ID 0xA23, the mobile device 150 forwards the authorization code to the server 130.

[0190] In some implementations, the advertised information further includes status information indicating one or more states of at least one of the payment module and the payment accepting unit, and the device sends (1412), to the server, the status information from the advertised information via the second communication capability. Figure 24A, for example, shows the packet 1100 with one or more status flags 1108. For example, the one or more status flags 1108 included in the packet 1100 are encoded with a predetermined code known by the server 130. In some implementations, the status information indicates that the payment module 100 has information to be uploaded to the server (e.g., transaction information for one or more interrupted transactions). In some implementations, the status information indicates information for the attention of the payment accepting machine 120's operator. For example, when the payment accepting unit 120 is a vending machine, the status information indicates that a particular item is low or out of stock. In another example, the status information indicates that the payment accepting unit 120 is experiencing a bill and/or coin jam. In another example, the status information indicates that the payment accepting unit 120's bill and/or coin reservoir is empty, nearly empty, full, or nearly full.

[0191] In response to sending at least the authorization code, the device obtains (1414), from the server, authorization information via the second communication capability, where the authorization information at least includes an authorization grant token. Figure 24B, for example, shows the authorization grant token 1140. In some implementations, the mobile device 150 receives the authorization grant token when the authorization code is valid and the first user has sufficient funds in his/her account for the payment processing system to perform a transaction at the payment accepting unit 120. In some implementations, the authorization grant token or a portion thereof is encrypted with the encryption key corresponding to the payment module 100. In some implementations, the authorization grant token includes an authorized amount, an expiration offset period, a user ID associated with the user of the mobile device 150, and a module ID associated with the payment module 100. For example, the expiration offset period depends on the first user's history and credit or a period predefined by the first user. For example, the authorized amount is predefined by the first user, based on a daily limit, based on the first user's total balance, or based on a risk profile associated with the user identified by the user ID. In some implementations, the authorization grant token or a portion thereof (e.g., the authorized amount or the auth code) is encrypted with an encryption key corresponding to the payment module 100 identified by the module ID.

[0192] In some implementations, the authorization request code is (1416) encrypted with a shared secret key corresponding to the payment module, and at least a portion of the authorization grant token is encrypted with the shared secret key corresponding to the payment module. For example, at least the authorized amount or the authorization code included in the authorization grant token is encrypted with the shared secret key. In some implementations, the shared secret key is known by the payment module 100 and the server 130. For example, the server 130 manages transactions for a plurality of payment modules and the server 130 stores a table of encryption keys for each of the payment modules. In this example, the server 130 selects an encryption key that corresponds to the respective payment module 100 and encrypts the authorized amount with the selected encryption key. In some implementations, the shared secret key is one of a public or private key in an asymmetrical cryptography scheme. Thus, in the above example, the mobile device 150 is an un-trusted party in the payment processing system; thus, the mobile device 150 cannot decrypt the authorization code or at least a portion of the authorization grant token.

[0193] After obtaining the authorization information, the device detects (1418) a trigger condition to perform a first transaction with a payment accepting unit (e.g., an automatic retailing machine such as a vending machine for dispensing goods and/or services) associated with the payment module. In the hands-free mode, the trigger condition is detected when the mobile device 150 enters the payment zone of the payment module 100 which occurs upon satisfaction of a payment zone criterion. In the manual mode, trigger condition is detected when the user of the mobile device 150 interacts with the user interface of the application 140 for the payment processing system while the application 140 is executed in as a foreground process on the mobile device 150.

[0194] In some implementations, the advertised information further includes (1420) a payment zone threshold criterion, and the device the trigger condition by: determining whether the payment zone threshold criterion is satisfied; and, in accordance with a determination that the payment zone threshold criterion is satisfied, detecting the trigger condition. In some implementations, the advertised information includes payment zone threshold criterion indicating a baseline RSSI that the mobile device 150 (or the application 140) is required to observe before being within the payment zone of payment module 100. In some implementations, the payment zone threshold criterion is a default RSSI value (e.g., -80 db) and the advertised information includes an offset (e.g., -5 db) to account for the strength and/or reception quality of the short-range radio/transceiver (e.g., BLE) of the payment module 100. In some implementations, the trigger condition is detected when the mobile device 150 enters the

payment zone of the payment module 100 which occurs upon satisfaction of a payment zone criterion. For example, when the RSSI observed by the mobile device 150 from the payment module 100 exceeds a predetermined payment zone threshold the payment zone threshold criterion is satisfied. In some implementations, the mobile device 150 provides an indication on the user interface of the application 140 for the payment processing system indicating whether the user is within the payment zone of payment module 100 and/or how close he/she is to the payment zone of payment module 100.

[0195] In some implementations, the device detects (1422) the trigger condition by: detecting a user input from a user of the device; and, in response to detecting the user input, detecting the trigger condition to perform a transaction with the payment accepting unit. For example, while the application 140 associated with the payment processing system is executed as a foreground process on the mobile device 150, the user of the mobile device interacts with the user interface of the application 140 to initiate a transaction with the payment accepting unit 120. In this example, the user performs a touch gesture with the touch screen of the mobile device 150, vocally commands the application 150 to initiate the transaction, or the like. Continuing with this example, after detecting the user interaction, the mobile device 150 (or the application 140) sends the payment module 100 the authorization grant token and the user is credited with the amount authorized in the authorization grant token in order to select goods and/or services provided by payment accepting unit 120 for purchase with the credit.

[0196] In some implementations, the authorization information further includes (1424) an expiration period for the authorization grant token, and the device sends, to the payment module, the authorization grant token via the first communication capability in response to detecting the trigger condition and in accordance with a determination that the expiration period has not elapsed. In some implementations, after detecting the trigger condition, the mobile device (or the application 140) determines whether an expiration period indicated by the authorization grant token has elapsed before sending the authorization grant token to the payment module 100. In some implementations, after determining that an expired authorization grant token is expired, the mobile device (or the application 140) determines automatically deletes the expired authorization grant token and requests a replacement authorization grant token by sending, to the server 130, the authorization request code included in current advertised information broadcasted by the payment module 100.

[0197] In response to detecting the trigger condition, the device sends (1426), to the payment module, the authorization grant token via the first communication capability. Continuing with the example in operation 1422, after detecting the user interaction, the mobile device 150 (or the application 140) sends the payment module 100 the authorization grant token and the user is credited with the amount authorized in the authorization grant token in order to select goods and/or services provided by payment accepting unit 120 for purchase with the credit.

[0198] For example, when the payment module 100 broadcasts the packet of information, if authorization code 12345 was issued in the packet (e.g., a new authorization code is issued every 100 ms), and a user uses that code to make a payment (when it comes back to the payment module 100 in the authorization grant token), the payment module 100 knows that authorization code 12345 has been used. Continuing with this example, if another subsequent user attempts to make a payment using the same authorization code 12345, the payment module 100 does not allow the subsequent user to use authorization code 12345 in order to prevent replay attacks. Additionally, in some implementations, the advertised authorization code expires after M minutes (e.g., 3, 5, 10, etc. minutes). In some implementations, the authorization code is a unique randomly or pseudo-randomly generated number that is stored by the payment module for M minutes after the authorization code is advertised, at which time it expires. In some implementations, the advertised authorization codes are unique incremental numbers that are advertised every X seconds. In this embodiment, the payment module 100 determines whether an authorization code in an authorization grant token is valid by identifying a current advertised authorization code and determining whether the advertised authorization is newer than the oldest valid authorization code based on the current advertised authorization code, the advertisement frequency (e.g., every X seconds), and the expiration period (e.g., M minutes).

[0199] In some implementations, after sending the authorization grant token, the device obtains (1428), from the payment module, first transaction information indicating a status of the first transaction with the payment accepting unit the first communication capability, and the device sends, to the server, the first transaction information corresponding to the status of the first transaction the second communication capability. In some implementations, the first transaction information indicates the status of the transaction initiated with an authorization grant token such as a complete, incomplete, or aborted transaction. For example, the first transaction is incomplete when the payment accepting unit 120 experiences a malfunction (e.g., a vending mechanism jams and the user of the mobile device 150 fails to receive a selected product) or the first transaction times-out by the user of the mobile device 150 waiting Z seconds without selecting goods and/or services from the payment accepting unit 120. For example, the first transaction is aborted when the user of the mobile device 150 actuates the coin return of the payment accepting unit 120 or walks away from the payment accepting unit 120 without selecting goods and/or service. In some implementations, the first transaction information includes the amount of the first transaction, current inventory state of products in payment accepting unit 120, other machine status informa-

tion, and the like.

[0200] It should be understood that the particular order in which the operations in Figures 27A-27C have been described is merely exemplary and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein. Additionally, it should be noted that details of other processes described herein with respect to other methods described herein (e.g., the method 1500 in Figures 28A-28B, the method 1600 in Figures 29A-29C, the and method 1700 in Figures 30A-30D) are also applicable in an analogous manner to the method 1400 described above with respect to Figures 27A-27C.

[0201] Figures 28A-28B illustrate a flowchart diagram of a method 1500 of transmitting machine status information in accordance with some implementations. In some implementations, the method 1500 is performed by a device with one or more processors, memory, and two or more communication capabilities. For example, in some implementations, the method 1500 is performed by the mobile device 150 (Figures 5 and 21) or a component thereof (e.g., application 140). In some implementations, the method 1500 is governed by instructions that are stored in a non-transitory computer readable storage medium (e.g., the memory 860, Figure 21) and the instructions are executed by one or more processors (e.g., the processing unit 840, Figure 21) of the device. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders).

[0202] The device obtains (1502), from a payment module, advertised information via a first communication capability, where the advertised information at least includes status information indicating one or more states of at least one of a payment module and a payment accepting unit associated with the payment module. For example, in some implementations, the payment module 100 broadcasts the packet 1100 (Figure 24A) which includes the one or more status flags 1108.

[0203] In some implementations, the first communication capability corresponds (1504) to a short-range communication protocol. As described above, short-range communication protocols include BLE, NFC, and/or other protocols utilizing non-persistent communication channels.

[0204] In some implementations, the status information indicates (1506) that the payment module is storing one or more interrupted transactions. As described in greater detail below with respect to Figures 29A-29C, in some implementations, interrupted transactions (sometimes referred to as "incomplete transactions") arise from the loss of a connection (e.g., the mobile device 150 has no cellular reception) or power. In some implementations, for example, the status flags 1108 (e.g., in packet 1100, Figure 24A) include upload information indicator 1116, which indicates that the payment module is storing one or more interrupted transactions, and/or also includes transaction information (e.g., transaction information 1150, Figure 24D) corresponding to the one or more incomplete transactions (e.g., an amount of the first transaction, a user ID, etc.). In some implementations, upload information indicator 1116 triggers the mobile device 150 to connect to payment module 100 immediately (e.g., if it has interrupted transaction information to be uploaded to the server 130). Alternatively, as described in greater detail below, the transaction information 1150 is generated and sent separately from the status flags 1108.

[0205] In some implementations, the status information indicates (1508) that the payment accepting unit requires servicing. For example, the status flags 1108 (e.g., in packet 1100, Figure 24A) include the bill/coin jam indicator 1118, which indicates that a blockage is detected in the payment feeding mechanism (e.g., bill or coin jam). Furthermore, in some implementations, the status flags 1108 (e.g., in packet 1100, Figure 24A) include the full bill/coin reservoir indicator 1120, which indicates that currency stored in the payment accepting unit requires collection by an operator of the machine. In some implementations, the status information indicates that the payment accepting unit requires servicing after a predefined period of time has elapsed since a prior servicing. In an example, the payment module 100 is configured to send status information indicating that the payment accepting unit 120 requires servicing after one month has elapsed since a last servicing.

[0206] In some implementations, the status information indicates (1510) a count of at least one product in the payment accepting unit. For example, the status flags 1108 (e.g., in packet 1100, Figure 24A) includes the inventory levels indicator 1122, which indicates that the remaining inventory of an item (e.g., an inventory levels indicator 1122 having a value of 1 indicates one corresponding item remaining for a particular product).

[0207] In some implementations, the status information is (1512) encoded with a predefined code. In some implementations, the status information is encrypted and/or encoded with a predefined code and/or key. For example, the status flags 1108 (e.g., in packet 1100, Figure 24A) include 4 Bytes of information which is encoded according to a predefined encoding scheme known by the server 130 which indicates a plurality of states of the payment module and/or the payment accepting unit 120 associated with the payment module 100.

[0208] In some implementations, the advertised information further includes (1514) an authorization code for authorizing a user of the device to perform a cashless transaction with the payment accepting unit. Authorization codes are described in greater detail above with respect to Figures 24C and 27A-27C and the accompanying text.

[0209] The device sends (1516), to a server, at least the status information from the advertised information via a second communication capability distinct from the first communication capability. For example, in step 1004 of method 1000 in Figure 23, the mobile device 150 sends

an authorization request to the server 130 that includes the authorization code included in the broadcasted packet, the user ID associated with the mobile device 150, the module ID associated with the payment module 100, and also the status information.

[0210] In some implementations, the second communication capability corresponds (1518) to a long-range communication protocol. For example, in some implementations, the long-range communication protocol is one of GSM, Wi-Fi, CDMA, LTE, and/or the like.

[0211] In some implementations, after sending the status information to the server, the device receives (1520) a request, from the server, via the second communication capability to obtain one or more interrupted transactions from the payment module; obtains, from the payment module, transaction information via the first communication capability, where the transaction information corresponds to the one or more interrupted transactions performed by one or more previous users at the payment accepting unit; and sends, to the server, the transaction information via the second communication capability. In some implementations, in response to the status flags indicating one or more interrupted transactions, the server 130 requests that the mobile device 150 connect to the payment module 100 to upload the one or more interrupted transactions. This may occur even when the user of the mobile device 150 does not initiate a transaction with the payment module 100. In some implementations, the mobile device 150 obtains the transaction information upon entering an authorization zone (e.g., the authorization zone 104). See Figures 29A-29C and the accompanying text for further discussion of interrupted transactions. For example, interrupted transactions arise from the loss of a network connection (e.g., the mobile device 150 has no cellular reception) or power.

[0212] It should be understood that the particular order in which the operations in Figures 28A-28B have been described is merely exemplary and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein. Additionally, it should be noted that details of other processes described herein with respect to other methods described herein (e.g., the method 1400 in Figures 27A-27C, the method 1600 in Figures 29A-29C, and the method 1700 in Figures 30A-30D) are also applicable in an analogous manner to the method 1500 described above with respect to Figures 28A-28B.

[0213] Figures 29A-29C illustrate a flowchart diagram of a method 1600 of pay payment processing acknowledgment in accordance with some implementations. In some implementations, the method 1600 is performed by a payment module with one or more processors, memory, and one or more first communication capabilities, which is coupled with a payment accepting unit (e.g., the payment accepting unit 120 (sometimes also herein called "machine 120") (Figures 5 and 19) such as a vending machine or kiosk for dispensing goods and/or services). For example, in some implementations, the method 1600 is performed by the adapter module 100, (Figures 5 and 20). In some implementations, the method 1600 is governed by instructions that are stored in a non-transitory computer readable storage medium (e.g., the memory 760, Figure 20) and the instructions are executed by one or more processors (e.g., the processing unit 750, Figure 20) of the payment module. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders).

[0214] The payment module obtains (1602), from the payment accepting unit, a first notification indicating completion of a first transaction performed by a first user of a first device at the payment accepting unit and an amount of the first transaction. For example, in step 1202 of the process 1200 in Figure 25A, the payment module 100 obtains a first notification from the payment accepting unit 120 after a first transaction is completed at the payment accepting unit 120.

[0215] In response to receiving the notification, the payment module (1604): generates first transaction information based at least in part on the first notification; stores the generated first transaction information; and sends the generated first transaction information to the first device via one of the one or more first communication capabilities. In some implementations, the payment module 100 generates the transaction information 1150 (Figure 24D) which includes the transaction ID 1152 (e.g., which sequentially increases after each completed transaction), the module ID 1154 (e.g., a unique ID for the payment module 100), the user ID 1156 (e.g., a unique user ID of the mobile device 150 such as a MAC address), the authorization grant 1158, the transaction status 1160 (e.g., complete, incomplete, or aborted), the transaction amount 1162 (e.g., $1.00), and/or other information 1164. The other information 1164 includes, in some implementations, information included in the packet 1100 (Figure 24A), such as one or more status flags 1108 indicating a state of the payment accepting unit, and/or other information pertaining to the payment accepting unit, the first device, the first transaction, and/or the first user. In some implementations, the first transaction information is also stored until reception of an acknowledgement from the first device (e.g., the mobile device 150). The transaction information 1150, for example, is stored in the memory 760 (Figure 20) of the payment module 100.

[0216] In some implementations, the one or more first communication capabilities correspond to a short-range communication protocol. As described above, short-range communication protocols include BLE, NFC, and/or other protocols utilizing non-persistent communication channels.

[0217] In some implementations, the first device forwards the first transaction information to a server (e.g., the server 130) via a second communication capability (e.g., a long-range communication protocol such as CD-

MA, GSM, Wi-Fi, or the like), and the server 130 debits the account of the first user of the first device based on the amount of the first transaction, which is indicated in the first transaction information. In some implementations, the server 130 sends encrypted acknowledgment information via the second communication capability to the first device, and the first device forwards the encrypted acknowledgment information to the payment module via the first communication capability.

[0218] After sending the first transaction information to the first device and in accordance with a determination that first acknowledgement information is received from the first device within a predetermined time period, the payment module deletes (1606) the stored first transaction information generated for the first transaction performed by the first user of the first device. In some implementations, the payment module 100 marks the first and second transaction as complete (in addition to or instead of deleting the first and second transaction information). For example, when the predetermined time period is 30 seconds, the payment module 100 deletes the first transaction information stored in the memory 760 (Figure 20) if the first acknowledgment information is received within 30 seconds of sending the first transaction information. For example, the payment module 100 determines whether the first acknowledgement information is received within a predetermined time period by comparing a timestamp of the stored first transaction information and the current time when (or if) the first acknowledgement is received.

[0219] In some implementations, the payment module encrypts (1608) the generated first transaction information, and the first acknowledgement information is encrypted. For example, the first transaction information is encrypted with a key corresponding to the payment module 100, and the first acknowledgement information is encrypted with a key selected by the server 130 that corresponds to the payment module 100. In this example, the keys are distinct, the same, or mutually known.

[0220] After sending the first transaction information to the first device and in accordance with a determination that the first acknowledgement information is not received from the first device within the predetermined time period, the payment module maintains (1610) the stored first transaction information generated for the first transaction performed by the first user of the first device. In one example, an acknowledgement is not received because the first device (e.g., the mobile device 150) loses power. In another example, the first device loses its long-range communication connection to the server 130, and is therefore unable to forward the first transaction information to a server for debiting the first user's account, or receiving an acknowledgement from the server 130. In another example, the user of the first device maliciously severs the long-range communication connection to interrupt the transaction information from being sent to the server 130, or to interrupt the acknowledgement information from being received by the payment module 100.

In some implementations, if the payment module 100 does not receive the acknowledgment information within the predetermined time period, or if the acknowledgment information cannot be decrypted (e.g., it has been fraudulently modified or accessed), the payment module 100 maintains the first transaction information (e.g., keeps transactions information 1150 stored in the memory 760, Figure 20) and attempts to send the first transaction information to the server 130 via another device (e.g., a second mobile device 150). For example, as discussed in greater detail below, the payment module leverages a subsequent second transaction involving a second device, and the first transaction information is sent to the second device with second transaction information that corresponds to a second transaction initiated by the user of the second device.

[0221] In some implementations, in accordance with the determination that the first acknowledgement information is not received from the first device within a predetermined time period, the payment module disables (1612) usage rights for the first user at the payment accepting unit. For example, the first user or user ID associated with the first device is suspended from performing cashless transactions, and further authorization grant tokens received from the first user or user ID are ignored by the payment module 100. Thus, for example, the first user cannot initiate another transaction cashless transaction with the payment module 100. In some implementations, the server 130 and/or payment module 100 records a history of incomplete transactions. In some implementations, the server 130 and/or payment module 100 blacklists the user only after a predefined number of incomplete transactions (e.g., 20 incomplete transactions), accounting for incomplete transactions that arise from non-malicious actions, such as a loss of cellular connection or power.

[0222] In some implementations, in accordance with the determination that the first acknowledgement information is not received from the first device within the predetermined time period, the payment module broadcasts (1614) an information packet via one of the one or more first communication capabilities, where the information packet includes one or more status flags indicating one or more unacknowledged first transactions including the first transaction performed by the first user of the first device. For example, the payment module 100 broadcasts packets (e.g., the packet 1100, Figure 24A) which include the status flags 1108 that indicate (e.g., upload information indicator 1116) that the payment module 100 has information that needs to be uploaded to the server 130 (e.g., transaction information for the interrupted/unacknowledged first transaction). Alternatively, in some implementations, the transaction information 1150 (Figure 24D) corresponding to an incomplete transaction is appended to the advertised information instead of merely setting the upload information indicator 1116 in the broadcast advertised information.

[0223] In some implementations, after determining that

the first acknowledgement information is not received from the first device within the predetermined time period, the payment module obtains (1616), from the payment accepting unit, a second notification indicating completion of a second first transaction performed by a second user of a second device at the payment accepting unit and an amount of the first transaction. In response to receiving the second notification, the payment module 100 generates second transaction information based at least in part on the second notification, stores the generated second transaction information, and sends the generated first transaction information and the generated second transaction information to the second device via one of the one or more first communication capabilities. Thus, the payment module 100 leverages the subsequent second transaction by sending the first transaction information with the second transaction information. In some implementations, when the second user enters an authorization zone (e.g., authorization zone 104, Figure 1), transaction information corresponding to the first user's incomplete first transaction is transmitted to the second device.

[0224] In some implementations, in accordance with a determination that second acknowledgement information is received from the second device within the predetermined time period, the payment modules deletes (1618) the stored first transaction information generated for the first transaction performed by the first user of the first device and the stored second transaction information generated for the second transaction performed by the second user of the second device. For example, in step 1272 of the process 1250 in Figure 25B, after receiving the first transaction information and the second transaction information, the server 130 sends acknowledgement information to the payment module 100 via the second device 150-2, which acknowledges reception of the first transaction information and the second transaction information. Continuing with this example, in step 1276 of the process 1250 in Figure 25B, after receiving the acknowledgement information, the payment module 100 deletes the first transaction information and the second transaction information.

[0225] In some implementations, in accordance with a determination that the second acknowledgement information is not received from the second device within a predetermined time period, the payment module maintains (1620) the stored first transaction information generated for the first transaction performed by the first user of the first device and the stored second transaction information generated for the second transaction performed by the second user of the second device. In some further implementations, the payment module 100 leverages a subsequent transaction involving a third device, and both the first and second transaction information are sent to the third device with third transaction information that corresponds to a third transaction initiated by the user of the third device.

[0226] It should be understood that the particular order in which the operations in Figures 29A-29C have been described is merely exemplary and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein. Additionally, it should be noted that details of other processes described herein with respect to other methods described herein (e.g., the method 1400 in Figures 27A-27C, the method 1500 in Figures 28A-28B, and the method 1700 in Figures 30A-30D) are also applicable in an analogous manner to the method 1600 described above with respect to Figures 29A-29C.

[0227] Figures 30A-30D illustrate a flowchart diagram of a method 1700 of updating firmware of the payment module in the payment processing system in accordance with some embodiments. In some implementations, the method 1700 is performed by a device (e.g., the mobile device 150) with one or more processors, memory, and two or more communication capabilities. In some implementations, the method 1700 is governed by instructions that are stored in a non-transitory computer readable storage medium (e.g., the memory 860, Figure 21) and the instructions are executed by one or more processors of the mobile device 150 (e.g., the processing unit 850, Figure 21). Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders).

[0228] As noted above, in some circumstances, a payment module (e.g., the payment module 100, Figure 26A) in a payment processing system cannot establish a direct communications channel to a server (e.g., the server 130, Figure 26A), and is therefore unable to directly receive firmware updates from the server. In these cases, as described below, if the firmware of the payment module is determined to satisfy certain criteria (e.g., firmware version is out-of-date), one or more devices (e.g., the mobile device 150) will send data packets to the payment module for updating the payment module's firmware. The device serves as a communications bridge between the payment module and the server, whereby the device obtains a verification request from the payment module, which the device then sends to the server for processing (e.g., the server 130 verifies that the data packets are non-corrupted and complete). After processing the verification request, the server sends a firmware command to the device, which the device then sends to the payment module for execution.

[0229] A device (e.g., the mobile device 150) obtains (1702), from a payment module (e.g., the payment module 100), advertised information via a first communication capability, where the advertised information at least includes a current firmware version of the payment module. In some implementations, the current firmware version corresponds to a timestamp (e.g., February 5, 2014), while in other implementations, the current firmware version is denoted by a version number (e.g., v1.4). Advertised information is described in greater detail herein with respect to Figure 24A.

[0230] In some implementations, the first communication capability corresponds (1704) to a short-range communication protocol. As described above, short-range communication protocols include BLE, NFC, and/or other protocols utilizing non-persistent communication channels.

[0231] The device determines (1708) that the current firmware version of the payment module satisfies one or more predefined firmware criteria.

[0232] In some implementations, the current firmware version of the payment module is compared (1710) with a firmware version stored by the device, and the one or more predefined firmware criteria are satisfied (1712) if the current firmware version of the payment module does not match the firmware version stored by the device. In some implementations, the device obtains an indication (e.g., from the server 130, Figure 26A) that the firmware version it stores is the latest firmware version. In some implementations, the latest firmware version is determined from a timestamp associated with the firmware. Alternatively, in some implementations, it is presumed that the firmware version stored by the device is the latest version.

[0233] In some implementations, the predefined firmware criteria are satisfied if the current firmware version of the payment module predates the firmware version stored by the device (e.g., the firmware of payment module 100 has a timestamp of February 5, 2014, compared to the firmware of mobile device 150 which has a timestamp of April 4, 2014), or has a version number less than the firmware version stored by the device (e.g., firmware v1.4 of the payment module 100 compared to firmware v1.5 of the mobile device 150, where the firmware version numbers are assigned in monotonically ascending order by the server 130). In other implementations, the predefined firmware criteria are satisfied if the current firmware version of the payment module is newer than the firmware version stored by the device. This arises, for example, if a firmware rollback procedure is initiated, where the newer firmware version of the payment module is overwritten with an older firmware version of the device.

[0234] Alternatively, in some implementations, the device receives, from a server, the determination that the current firmware version of the payment module satisfies one or more predefined firmware criteria. In these implementations, for example, the device sends the current firmware version of the payment module to the server (e.g., the server 130, Figure 26A) via a second communication capability (e.g., GSM), where the server determines (e.g., by comparing the current firmware version of the payment module and a latest version of the firmware) if the current firmware version satisfies predefined firmware criteria (as described in greater detail above).

[0235] In some implementations, prior to sending firmware update information and in accordance with a determination that the current firmware version of the payment module does not match the firmware version stored by the first device (1714), the device sends (1716), to a server, a firmware update request so as to update the firmware of the payment module via a second communication capability. In some implementations, the second communication capability corresponds (1718) to a long-range communication protocol. For example, in some implementations, the long-range communication protocol is one of GSM, Wi-Fi, CDMA, LTE, and/or the like.

[0236] In some implementations, in response to sending the firmware update request, the device receives (1720) from the server, a firmware update approval message, and in response to receiving the firmware update approval message, the device sends (1724), to the payment module, the firmware update approval message. In some implementations, as described in greater detail with respect to Figure 26C, the server (e.g., server 130) permits or declines the firmware update request for any of a number of reasons (e.g., firmware stored by the mobile device 150 is out of date).

[0237] Furthermore, in some implementations, firmware update approval message includes (1722) a verification value for each of the one or more data packets and a hash value for the firmware update information. A more in-depth discussion is provided in the corresponding description for Figure 26C, with respect to the ways in which the payment module 100 uses the verification value and hash value for verifying and executing the firmware update information (e.g., one or more data packets corresponding to a latest firmware version).

[0238] In some implementations, sending the firmware update information via the first communication capability includes (1726), in response to receiving the firmware update approval message from the server, sending, to the payment module, the firmware update information via the first communication capability.

[0239] In some implementations, the device obtains (1728) update status information from the payment module, wherein the update status information indicates remaining packets for updating the current firmware version of the payment module. As described in greater detail with respect to Figure 26C, in some implementations, the update status information indicating remaining packets (e.g., packets 50-100) for updating the current firmware version is sent by the payment module 100 after receiving a firmware update approval message, while in other implementations, the update status information is included in broadcasted packet 1100 that is broadcast by the payment module 100. Alternatively and/or additionally, the update status information indicates one or more data packets received for updating the current firmware version of the payment module to the most recent firmware version. Furthermore, in some implementations, the update status information identifies the firmware version to which the remaining and/or received data packets correspond. Furthermore, in some implementations, sending the firmware update information via

the first communication capability is based on the update status information.

[0240] In accordance with a determination that the current firmware version of the payment module satisfies one or more predefined firmware criteria, the device sends (1730), to the payment module, firmware update information via the first communication capability, where the firmware update information includes one or more data packets for updating the current firmware version of the payment module. In some implementations, the firmware update information is stored by the device, and was included in a latest update to the application 140 associated with the payment processing system. In some implementations, the firmware update information is obtained from a server (e.g., the server 130, Figure 26A) via a second communication capability (e.g., GSM), and sent to the payment module via the first communication capability.

[0241] In some implementations, advertised information further includes (1706) an authorization zone threshold criterion, and the device determines (1732) that the authorization zone threshold criterion is satisfied. For example, in some implementations, the device starts transmitting data packets to update the payment module firmware upon entering the authorization zone (e.g., the authorization zone 140, Figure 1) of the payment module. In some implementations, once the device has started transmitting data packets, if the device later leaves the authorization zone, the device continues to transmit data packets as long as the device remains with the communication zone of the payment module (e.g., BLE range), or, alternatively, the device ceases transmission of data packets.

[0242] In some implementations, the device obtains (1734) additional advertised information, where the additional advertised information at least includes update status information. In some implementations, the additional advertised information further includes a new authorization code and/or status flags. In some implementations, the update status information identifies (1738) one or more remaining data packets (e.g., packets 50-100) for updating the current firmware version of the payment module to the most recent firmware version. Alternatively and/or additionally, the update status information identifies (1740) one or more data packets received for updating the current firmware version of the payment module to the most recent firmware version. Furthermore, in some implementations, the update status information identifies the firmware version to which the remaining and/or received data packets correspond. Optionally, the update status information is included in transaction information (e.g., the transaction information 1150, after completing a transaction).

[0243] In some implementations, the update status information includes (1736) a verification request. For example, a verification request is generated and included in update status information when the payment module 100 has received all data packets necessary for completing the firmware update. A verification request is generally associated with a request to implement the received data packets in order to update the firmware of the payment module 100. In some implementations, a verification request is a request for a server (e.g., the server 130, Figure 26A) to determine if any received data packets are corrupted, if the received data packets form a complete set sufficient to initiate a firmware update, and/or if the received data packets correspond to a latest firmware version. In some embodiments, a verification request is a checksum performed by the payment module on the received data packets for the firmware update according to a predefined checksum algorithm. Furthermore, in some implementations, the verification request is sent to the server with transaction status information (e.g., transaction information 1150, Figure 24D) after a user completes a transaction (e.g., the step 1260 in process 1250 of Figure 25B).

[0244] In some implementations in which the device obtains additional advertised information including a verification request, the device sends (1742), to a server, at least the current firmware version and a verification request via a second communication capability. In some implementations, the second communication capability corresponds (1744) to a long-range communication protocol (e.g., GSM, Wi-Fi, CDMA, LTE, and/or the like). In some implementations, the verification request is sent by the payment module directly to the server via a secure communications channel (e.g., an encrypted channel).

[0245] Furthermore, in some implementations, the device obtains (1746), from the server, a firmware command via the second communication capability, and sends (1748), to the payment module, the firmware command via the first communication capability. The server processes the verification request prior to issuing a firmware command to the device to send to the payment module. As described above, in some implementations, the server determines if any received data packets are corrupt (e.g., by using a checksum), if the received data packets form a complete set sufficient to initiate a firmware update, and/or if the received data packets correspond to a latest firmware version. In some implementations, unless some or all of these aforementioned conditions (e.g., corrupted data packets, complete set, etc.) are not satisfied, the server issues an approval code and/or a firmware command to initiate an update of the payment module's firmware. In some embodiments, the server determines whether a checksum included in the verification request matches a checksum value determined by the server for the firmware update indicated by the verification request (e.g., a version number). For example, if the checksum included in the verification request does not match the server's checksum, the server issues a firmware command to not implement the firmware update and to delete the data packets corresponding to the firmware update associated with the verification request was sent. In this example, the checksums may not match if one or more of the data packets

for the firmware update are corrupted or have been altered.

[0246] In some implementations, the firmware command is a rollback command (e.g., ignore firmware update and keep current firmware version of the payment module 100), a delete command (e.g., deleting either all or a portion of the data packets for the firmware update), or an initialization command (e.g., initializing the firmware update in the payment module 100). If the server determines, in some implementations, that the received data packets do not correspond to a latest firmware version (e.g., the firmware update information stored by the device corresponds to firmware v1.4, compared to a latest firmware v1.5), the server will send, to the device or directly to the payment module, firmware update information including one or more data packets corresponding to a latest firmware version. This may occur, for example, if the mobile device 150 itself is not storing the latest firmware. In some implementations, the firmware command is encrypted with an encryption key that corresponds to the payment module (e.g., a shared secret key or a public key in an asymmetric cryptography scheme).

[0247] In some implementations, a second device with one or more processors, memory, and two or more communication capabilities, obtain (1750), from the payment module, advertised information via the first communication capability, where the advertised information at least includes a current firmware version of the payment module and the update status information. The second device determines (1752) whether the current firmware version of the payment module predates a most recent firmware version. In accordance with a determination that the current firmware version of the payment module satisfies one or more predefined firmware criteria, the second device sends (1754), to the payment module, firmware update information via the first communication capability, where the firmware update information one or more additional data packets for updating the current firmware version based at least in part on the update status information. Thus, in some implementations, multiple devices send to the payment module portions of a complete set of data packets needed for a firmware update, where the data packets sent by one device are distinct from the data packets sent by another device. In one example, when a firmware update includes data packets 1 through 100, a first device (e.g. the mobile device 150) sends data packets 1 through 50 to the payment module 100, and a second device (a different mobile device 150, not shown) sends data packets 50 through 100.

[0248] It should be understood that the particular order in which the operations in Figures 30A-30D have been described is merely exemplary and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein. Additionally, it should be noted that details of other processes described herein with respect to other methods described herein (e.g., the method 1400 in Figures 27A-27C, the method 1500 in Figures 28A-28B, and the method 1600 in Figures 29A-29C) are also applicable in an analogous manner to the method 1600 described above with respect to Figures 30A-30D.

[0249] Figure 31A illustrates a schematic flow diagram of a process 1800 for providing a representation of a machine event at a mobile device in accordance with some implementations. In some implementations, the payment processing system includes one or more payment modules 100 (e.g., each associated with a respective payment accepting unit 120 such an automatic retailing machine for dispensing goods and/or services), one or more mobile devices 150 (e.g., each executing the application 140 for the payment processing system either as a foreground or background process), and the server 130. The server 130 manages the payment processing system and, in some cases, supplies, operates, and/or manufactures the one or more payment modules 100. For brevity, the process 1800 will be described with respect to a respective payment module 100 associated with a respective payment accepting unit 120 (sometimes also herein called the "machine 120") and a respective mobile device 150 in the payment processing system.

[0250] In some implementations, the process 1800 occurs after the mobile device 150 sends the AuthGrant in Figure 8C. In some implementations, the process 1800 occurs after the mobile device 150 sends the authorization grant to the payment module 100 in operation 1012 of process 1000 in Figure 23.

[0251] The payment module 100 obtains (1802) an indication corresponding to an event at the machine 120. For example, after the process 1000 in Figure 23, the user of the mobile device 150 selects a product to purchase from the machine 120 by interacting with one or more input mechanisms of the machine 120 (e.g., buttons 126 or a touch screen display 124 shown in Figure 19), and the machine 120 dispenses the selected product. Continuing with this example, after the product is dispensed, the transaction is complete and the payment module 100 obtains an indication from the machine of the completed transaction. In some implementations, the indication includes the amount of the transaction and (optionally) machine status information associated with the machine 120 such as inventory information as to one or more products of the payment accepting unit 120 and/or the like. In some implementations, the indication includes status information indicating that the transaction was aborted (e.g., via actuation of a coin return mechanism at the machine 120) or that there was an error with the transaction (e.g., a vending jam or other malfunction with the machine 120).

[0252] After obtaining the indication corresponding to completion of the first transaction, the payment module 100 generates (1804) a notification corresponding to the event at the machine 120.

[0253] The payment module 100 sends (1806), via a short-range communication capability (e.g., BLE), the

notification to the mobile device 150. In some embodiments, in addition to the notification corresponding to the event at machine 120, the payment module 100 sends a promotion or advertisement to the mobile device 150 that is targeted to the user of the mobile device 150 based on the transaction or the user ID included in the AuthGrant or authorization grant token that initiated the transaction. In some embodiments, in addition to the notification corresponding to the event at machine 120, the payment module 100 sends a pseudo randomly selected promotion or advertisement to the mobile device 150 that is selected from a set of promotions or advertisements stored by the payment module 100. For example, the promotion is a coupon for a free soda following the purchase of ten sodas from the machine 120 by the user of the mobile device 150. For example, the promotion is a random 50% off coupon or free soda coupon. For example, the transaction corresponds to a vended soda and the advertisement corresponds to a new soda from the same company that produces the vended soda.

[0254] The mobile device 150 provides (1808) a representation of the notification. For example, in Figure 32A, the mobile device 150 displays user interface 1902 on touch screen 152 with a message 1906 that indicates that the first transaction is complete. For example, in Figure 32C, the mobile device 150 displays user interface 1920 on touch screen 152 with a message 1922 that indicates that the transaction was aborted. For example, in Figure 32D, the mobile device 150 displays user interface 1930 on touch screen 152 with a message 1932 that indicates that there was an error with the transaction. For example, the mobile device 150 also displays a representation of the promotion of advertisement on the user interface for the application 140.

[0255] Figure 31B illustrates a schematic flow diagram of a process 1850 for processing acknowledgement information in accordance with some implementations. In some implementations, the payment processing system includes one or more payment modules 100 (e.g., each associated with a respective payment accepting unit 120 such an automatic retailing machine for dispensing goods and/or services), one or more mobile devices 150 (e.g., each executing the application 140 for the payment processing system either as a foreground or background process), and the server 130. The server 130 manages the payment processing system and, in some cases, supplies, operates, and/or manufactures the one or more payment modules 100. For brevity, the process 1850 will be described with respect to a respective payment module 100 associated with a respective payment accepting unit 120 (machine 120) and a respective mobile device 150 in the payment processing system.

[0256] In some implementations, the process 1850 occurs after the mobile device 150 sends the AuthGrant in Figure 8C. In some implementations, the process 1850 occurs after the mobile device 150 sends the authorization grant to the payment module 100 in operation 1012 of process 1000 in Figure 23.

[0257] The payment module 100 obtains (1852) an indication corresponding to completion of a first transaction from the machine 120. For example, after the process 1000 in Figure 23, the user of the mobile device 150 selects a product to purchase from the machine 120 by interacting with one or more input mechanisms of the machine 120 (e.g., buttons 126 or a touch screen display 124 shown in Figure 19), and the machine 120 dispenses the selected product. Continuing with this example, after the product is dispensed, the transaction is complete and the payment module 100 obtains an indication from the machine of the completed transaction. In some implementations, the indication includes the amount of the transaction and (optionally) machine status information associated with the machine 120 such as inventory information as to one or more products of the payment accepting unit 120 and/or the like.

[0258] After obtaining the indication corresponding to completion of the first transaction, the payment module 100 generates (1854) a first notification with first transaction information based on the indication, and the payment module 100 stores the first transaction information. In some implementations, the first transaction information includes a transaction ID for the first transaction, a module ID corresponding to payment module 100, a user ID corresponding to the mobile device 150, transaction status information indicating that the first transaction is complete, and the transaction amount indicated by the indication. In some implementations, the payment module 100 retains the authorization code included in the original broadcasted packet and/or the authorization grant token and includes the authorization code in the first transaction information. In some implementations, the authorization code is encrypted with a secret key corresponding to the payment module 100, which is shared with the server 130 but not the mobile device 150. In some implementations, the first transaction information further includes other information such as the machine status information included in the first notification or transaction information corresponding to previous interrupted transaction(s). See Figure 24D and the accompanying text for further discussion regarding transaction information 1150.

[0259] The payment module 100 sends (1856), via a short-range communication capability (e.g., BLE), the first notification with first transaction information to the mobile device 150. In some embodiments, in addition to first transaction information corresponding to completion of the first transaction at machine 120, the first notification includes a promotion or advertisement to the mobile device 150 that is targeted to the user of the mobile device 150 based on the transaction or the user ID included in the AuthGrant or authorization grant token that initiated the transaction. In some embodiments, in addition to first transaction information corresponding to completion of the first transaction at machine 120, the first notification includes a pseudo randomly selected promotion or advertisement to the mobile device 150 that is selected from

a set of promotions or advertisements stored by the payment module 100. For example, the promotion is a coupon for a free soda following the purchase of ten sodas from the machine 120 by the user of the mobile device 150. For example, the promotion is a random 50% off coupon or free soda coupon. For example, the transaction corresponds to a vended soda and the advertisement corresponds to a new soda from the same company that produces the vended soda.

[0260] The mobile device 150 provides (1858) a representation of the first notification. For example, in Figure 32A, the mobile device 150 displays user interface 1902 on touch screen 152 with a message 1906 that indicates that the first transaction is complete. For example, the mobile device 150 also displays a representation of the promotion of advertisement on the user interface for the application 140.

[0261] The mobile device 150 sends (1860), via a long-range communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), the first transaction information to the server 130.

[0262] The server 130 processes (1862) the first transaction information. For example, the server 130 debits the account of the user associated with the user ID in the first transaction information in the amount indicated by the first transaction information.

[0263] The server 130 sends (1864), via a long-range communication capability (e.g., GSM, CDMA, Wi-Fi, or the like), first acknowledgment information to the mobile device 150. In some implementations, the first acknowledgment information acknowledges that the server 130 received the first transaction information. In some implementations, the first acknowledgment information includes the user ID, the module ID, the transaction ID, and (optionally) the authorization grant included in the transaction information (e.g., auth grant 1158, Figure 24D).

[0264] After receiving the first acknowledgement information, the mobile device 150 sends (1866), via a short-range communication capability (e.g., BLE), the first acknowledgment information to the payment module 100.

[0265] After receiving the first acknowledgment information, the payment module 100 deletes (1868) the stored first transaction information.

[0266] Attention is now directed towards implementations of user interfaces and associated processes that may be implemented on the mobile device 150 with zero or more speakers, zero or more microphones, and a display. For example, the display is a touch screen (sometimes also herein called a "touch screen display") enabled to receive one or more contacts and display information (e.g., media content, websites and web pages thereof, user interface for the application 140, and/or user interfaces for applications). Figures 32A-32D illustrate example user interfaces for providing a representation of a machine event at a mobile device in accordance with some implementations.

[0267] Figures 32A-32D show user interfaces displayed on mobile device 150 (e.g., a mobile phone); however, one skilled in the art will appreciate that the user interfaces shown in Figures 32A-32D may be implemented on other similar computing devices. The user interfaces in Figures 32-32D are used to illustrate the processes described herein, including the process described with respect to Figures 31A-31B and 33A-33B.

[0268] For example, a user of the mobile device 150 approaches a machine 120 (e.g., vending machine 78x928 as shown in Figures 10A-10D) and executes application 140 on the mobile device 150 so as to perform an electronic transaction with the machine 120. For example, with reference to Figures 10C-10D, the user of the mobile device 150 initiates a transaction with the machine 120 (e.g., vending machine 78x928) by performing a swipe gesture at a location corresponding to the representation of the dollar bill (e.g., a substantially vertical swipe gesture from a location corresponding to the representation of the dollar bill to the top edge of the mobile device 150).

[0269] Figure 32A illustrates the mobile device 150 displaying a user interface 1902 of the application 140 on touch screen 152 after the user of the mobile device 150 initiates and performs a transaction with the machine 120. In Figure 32A, the user interface 1902 includes prepaid balance 1904 which indicates that $1.00 has been deducted from the prepaid balance after performing a transaction with the machine 120 as compared to the prepaid balance in Figure 10C-10D (i.e., $9.00 in Figures 10C-10D and $8.00 in Figure 32A). In Figure 32A, the user interface 1902 also includes a message 1906 indicating that the transaction with the machine 120 is complete.

[0270] Figure 32B illustrates the mobile device 150 displaying a user interface 1910 of the application 140 on touch screen 152 after the user of the mobile device 150 initiates a transaction with the machine 120 and an error with the transaction occurs or the transaction is aborted. In Figure 32B, the user interface 1910 shows the representation of the dollar bill sliding onto the touch screen 152 (e.g., in a substantially top to bottom manner). In Figure 32B, the interface 1910 includes prepaid balance 1912 which indicates that no money has been deducted from the prepaid balance after performing a transaction with the machine 120 as compared to the prepaid balance in Figure 10C-10D (i.e., $9.00 in Figures 10C-10D and $9.00 in Figure 32B).

[0271] Figure 32C illustrates the mobile device 150 displaying a user interface 1920 of the application 140 on touch screen 152 after the representation of the dollar bill slides onto the touch screen 152 in Figure 32B due to the transaction being aborted. For example, the user aborts the transaction by actuating a coin return mechanism of the machine 120. In another example, the user aborts the transaction by selection an abort affordance on the interface of the application 140 (not shown). In Figure 32C, the user interface 1920 includes a message 1922 indicating that the transaction with the machine 120

was aborted and that the user's account was not debited for the aborted transaction.

[0272] Figure 32D illustrates the mobile device 150 displaying a user interface 1930 of the application 140 on touch screen 152 after the representation of the dollar bill slides onto the touch screen 152 in Figure 32B due to the occurrence of an error with the transaction. For example, a malfunction with the machine 120 (e.g., a vending jam or stuck item) causes the error to occur. In Figure 32D, the user interface 1930 is associated with the application 140 executed on the mobile device 150. In Figure 32D, the user interface 1930 includes a message 1932 indicating that an error occurred during the transaction with the machine 120 and that the user's account was not debited for the transaction.

[0273] Figures 33A-33B illustrate a flowchart diagram of a method 2000 of presenting representations of payment accepting unit events in accordance with some implementations. In some implementations, the method 2000 is performed by a device with one or more processors, memory, one or more output devices, and two or more communication capabilities. For example, in some implementations, the method 2000 is performed by the mobile device 150 (Figures 5 and 21) or a component thereof (e.g., the application 140). In some implementations, the method 2000 is governed by instructions that are stored in a non-transitory computer readable storage medium (e.g., the memory 860, Figure 21) and the instructions are executed by one or more processors (e.g., the processing unit 850, Figure 21) of the device. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders).

[0274] After sending a request to a payment module via a first communication capability transaction to initiate a transaction with a payment accepting unit (e.g., an offline-payment operated machine such as a vending machine or kiosk) associated with the payment module, the mobile device obtains (2002) a notification from the payment module via the first communication capability, where the notification indicates an event at the payment accepting unit associated with the payment module. In some implementations, method 2000 occurs after the mobile device 150 sends the AuthGrant in Figure 8C. In some implementations, method 2000 occurs after the mobile device 150 sends the authorization grant to the payment module 100 in operation 1012 of process 1000 in Figure 23. Operation 1806 of Figure 31A, for example, shows the mobile device 150 receiving a notification sent by the payment module 100 (e.g., the adapter module 100, Figures 5 and 20) sent via the first communication capability (e.g., a short-range communication technology/protocol such as BLE). The notification indicates an event at the payment accepting unit (e.g., the payment accepting unit 120, Figures 5 and 19) (sometimes also herein called "machine 120") associated with the payment module 100.

[0275] In some implementations, the first communication capability corresponds (2004) to a short-range communication protocol. As described above, the short-range communication protocols include BLE, NFC, and/or other protocols utilizing non-persistent communication channels.

[0276] In response to obtaining the notification, the mobile device provides (2006) a representation of the notification to a user of the mobile device via the one or more output devices of the mobile device. For example, in Figure 32A, the mobile device 150 displays user interface 1902 on touch screen 152 with a message 1906 that indicates that the first transaction is complete. For example, in Figure 32C, the mobile device 150 displays user interface 1920 on touch screen 152 with a message 1922 that indicates that the transaction was aborted. For example, in Figure 32D, the mobile device 150 displays user interface 1930 on touch screen 152 with a message 1932 that indicates that there was an error with the transaction.

[0277] In some implementations, the one or more output devices of the mobile device include (2008) at least one of: a display, one or more speakers, one or more LEDs, and a vibration mechanism. For example, the mobile device 150 includes one or more of a display (e.g., the touch screen 152, Figures 10A-10D), one or more speakers, one or more LEDs, and a vibration mechanism.

[0278] In some implementations, the representation of the notification is at least one of (2010): a message displayed on the display of the mobile device; a banner notification displayed on a display of the mobile device; a vibration alert from the vibration mechanism of the mobile device; an aural alert from the one or more speakers of the mobile device; and a visual alert from the one or more LEDs of the mobile device. For example, in Figures 32B-32D, the representation of the notification includes messages 1906, 1922, and 1932 displayed on the touch screen 152 of the mobile device 150. In another example, the representation of the notification is a predefined sequence of vibrations provided by the vibration mechanism of the mobile device 150. In another example, the representation of the notification is a predefined sequence of tones provided by the one or more speakers of the mobile device 150. In another example, the representation of the notification is a predefined sequence of blinking LEDs of the mobile device 150.

[0279] In some implementations, the notification indicates (2012) abortion of a transaction initiated by the user of the mobile device. In Figure 32C, for example, the user interface 1920 includes the message 1922 indicating that the transaction has been aborted. For example, the user aborts the transaction by actuating a coin return mechanism of the machine 120. In another example, the user aborts the transaction by selection an abort affordance on the interface of the application 140 (not shown).

[0280] In some implementations, the notification indicates (2014) completion of a transaction between the user of the mobile device and the payment accepting unit. In Figure 32A, for example, the user interface 1902 includes the message 1906 indicating that completion of

the transaction with the machine 120 initiated by the user of the mobile device 150.

[0281] In some implementations, the notification indicating completion of the transaction at least includes (2016) an amount of the completed transaction. In Figure 32A, for example, the user interface 1902 includes prepaid balance 1904 which indicates that $1.00 has been deducted from the prepaid balance after performing a transaction with the machine 120 as compared to the prepaid balance in Figure 10C-10D (i.e., $9.00 in Figures 10C-10D and $8.00 in Figure 32A).

[0282] In some implementations, the mobile device sends (2018) at least a portion of the notification to a server via a second communication capability distinct from the first communication capability. Operation 1860 of Figure 31B, for example, shows the mobile device 150 sending first transaction information to the server 130 for a completed transaction via the second communication capability (e.g., a long-range communication protocols such as Wi-Fi, CDMA, GSM, and/or the like). For example, the first transaction information at least includes the amount of the first completed transaction.

[0283] In some implementations, the first communication capability corresponds (2020) to a short-range communication protocol and the second communication capability corresponds to a long-range communication protocol. For example, the first communication capability of the mobile device 150 is a radio/transceiver means for communicating via one or more short-range communication protocols such as BLE, NFC, and/or the like (i.e., a non-persistent communication channel). For example, the second communication capability of the mobile device 150 is a radio/transceiver means for communicating via one or more long-range communication protocols such as Wi-Fi, CDMA, GSM, and/or the like.

[0284] In some implementations, the notification indicates (2022) failure of a transaction initiated by the user of the mobile device or a malfunction associated with the payment accepting unit. In Figure 32D, for example, the user interface 1930 includes the message 1932 indicating that there was an error with the transaction. For example, the transaction fails due to a vending jam or other malfunction. In another example, the payment accepting unit experiences a malfunction due to an open door or the like. In some implementations, at least a portion of the failure/malfunction notification is sent to the sever 130 and an alert is subsequently sent to the operator of the payment accepting unit (e.g., the machine 120) by the server 130.

[0285] It should be understood that the particular order in which the operations in Figures 33A-33B have been described is merely for example purposes and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein. Additionally, it should be noted that details of other processes described herein with respect to other methods described herein are also applicable in an analogous manner to the method 2000 described above with respect to Figures 33A-33B.

[0286] Figure 34A illustrates a block diagram of an offline-payment operated machine 2100 in accordance with some implementations. For example, the offline-payment operated machine 2100 (e.g., a form of the machine 120) is an electro-mechanical machine capable of accepting currency (e.g., coins), which is not connected to any networks (e.g., telephone, cellular, or Wi-Fi). For example, the offline-payment operated machine 2100 is a washer or dryer at a laundromat, a parking meter, a car wash payment kiosk, or other offline-payment operated machine that dispenses goods and/or services.

[0287] In Figure 34A, the offline-payment operated machine 2100 includes a microswitch 2102, a control unit 2106, a power supply 2108, a transistor 2110, and an operation unit 2112. The components of the offline-payment operated machine 2100 in Figure 34A are examples and one of skill in the art will appreciate that various other components may be included in or excluded from the offline-payment operated machine 2100.

[0288] In Figure 34A, the microswitch 2102 is a leveraged microswitch with lever 2104. For example, the microswitch 2102 is a CHERRY BRAND™ microswitch with a normally open terminal ("NO"), a normally closed terminal ("NC"), and a common terminal. For example, the lever 2104 is incorporated into a coin slot of the offline-payment operated machine 2100 and is depressed whenever a coin slides down the coin slot into a coin reservoir of the offline-payment operated machine 2100 (not shown). For example, when the lever 2104 is depressed and the microswitch 2102 is wired in the NO configuration as shown in Figure 34A, the switch is closed. Continuing with this example, when the switch is closed, control unit 2106 receives a pulse (i.e., a payment acceptance signal) from the common terminal of the microswitch 2102 indicating depression of the lever 2104 from the reception of a US quarter (i.e., $0.25) or coin of another value.

[0289] In some implementations, when the control unit 2106 receives a preset sequence of payment acceptance signals indicative of a preset number of coins being received by the microswitch 2102, the control unit 2106 initiates the operation of the offline-payment operated machine 2100. For example, after receiving the preset sequence of payment acceptance signals (e.g., three pulses indicating reception of three US quarters), the control unit 2106 initiates operation of the offline-payment operated machine 2100 by applying current to the gate of the transistor 2110 which allows current to flow from the power supply 2108 to operation unit 2112. For example, the operation unit 2112 is a motor of a dryer which begins spinning once current flows from the power supply 2108.

[0290] In Figure 34A, payment module 2120 (e.g., a form of the adapter module 100, Figures 5 and 20) is configured to be installed in the offline-payment operated

machine 2100 so as to retrofit the offline-payment operated machine 2100 to be able to accept electronic payments. In some implementations, the payment module 2120 includes all or some of the components included adapter module 100 in Figure 20 such as processing unit 750, memory 760, a security unit 755, and a communications unit 770. In some implementations, the payment module 2120 also includes a first interface module 2122, a second interface module 2124, and a lead 2136 for drawing power from power supply 2108 of the offline-payment operated machine 2100.

[0291] In Figure 34A, the first interface module 2122 is configured to sample payment acceptance signals from the microswitch 2102 (e.g., a coin receiving switch) via lead 2132 of the offline-payment operated machine 2100. For example, the payment acceptance signals are indicative of a coin being received by the microswitch 2102 which depress lever 2104. In Figure 34A, the second interface module 2124 is configured to sample control signals from the control unit 2106 of the offline-payment operated machine 2100 via lead 2134 that initiate an operation of the offline-payment operated machine (e.g., the application of current to the gate of the transistor 2110) in response to receiving a preset sequence of payment acceptance signals from the microswitch 2102 (e.g., the coin receiving switch) indicative of the preset number of coins.

[0292] Figure 34B illustrates signals sampled by the payment module 2120 in accordance with some implementations. In Figure 34B, sample 2150 represents a preset sequence of payment acceptance signals sampled by the first interface module 2122 via lead 2132 that are sent from the microswitch 2102 to the control unit 2106. For example, the preset sequence of payment acceptance signals indicative of the preset number of coins include pulses (i.e., payment acceptance signals) 2152, 2154, 2156, and 2158. For example, the leading edges of pulses 2152, 2154, 2156, and 2158 at times 2182, 2184, 2186, and 2188 indicate reception of a coin by microswitch 2102 which causes the switch to close when wired in the NO configuration as shown in Figure 34A. In Figure 34B, sample 2170 represents a control signal sampled by the second interface module 2124 via lead 2134 that is sent from the control unit 2106 to transistor 2110. In Figure 34B, the sample 2170 includes a pulse 2172 that is sent from the control unit 2106 to transistor 2110 at time 2190 after receiving the preset sequence of payment acceptance signals from the microswitch 2102 (i.e., pulses 2152, 2154, 2156, and 2158).

[0293] Figures 35A-35B illustrate a flowchart diagram of a method of retrofitting an offline-payment operated machine to accept electronic payments in accordance with some implementations. In some implementations, the method 2200 is performed by a payment module with one or more processors and memory. In some implementations, the payment module also includes a short-range communication capability corresponding to a short-range communication protocol (e.g., a non-persistent communication channel such as BLE, NFC, and/or the like), where the short-range communication capability is configured to communicate with one or more mobile devices, where each of the one or more mobile devices is configured with a complimentary short-range communication capability and a long-range communication capability corresponding to a long-range communication protocol (e.g., Wi-Fi, CDMA, GSM, and/or the like).

[0294] In some implementations, the payment module is coupled with an offline-payment operated machine (e.g., the payment accepting unit 120, Figures 5 and 19 (sometimes also herein called "machine 120"), or the offline-payment operated machine 2100, Figure 34A) such as dryer or washer in a laundromat, a parking meter, a car wash payment kiosk, or the like. In some implementations, the offline-payment operated machine includes a coin receiving switch (e.g., the microswitch 2102, Figure 34A) and a control unit (e.g., the control unit 2106, Figure 34A). In some implementations, the payment module further includes: (A) a first interface module (e.g., the first interface module 2122, Figure 34A) configured to sample payment acceptance signals from the coin receiving switch of the offline-payment operated machine, where the signals are indicative of a coin being received by the coin receiving switch; and (B) a second interface module (e.g., the second interface module 2124, Figure 34A) configured to sample control signals from the control unit of the offline-payment operated machine that initiate an operation of the offline-payment operated machine in response to receiving a preset sequence of payment acceptance signals from the coin receiving switch indicative of the preset number of coins. By sampling and storing these signals, the payment module 2120 is able to simulate operation of a respective coin receiving switch in response to receiving the correct/preset number of coins so as to trigger operation of the offline-payment operated machine in response to completion of an electronic payment.

[0295] For example, in some implementations, the method 2200 is performed by the adapter module 100 (Figures 5 and 20) or payment module 2120 (Figure 34A). In some implementations, the method 2200 is governed by instructions that are stored in a non-transitory computer readable storage medium (e.g., the memory 760, Figure 20) and the instructions are executed by one or more processors (e.g., the processing unit 750, Figure 20) of the payment module. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders).

[0296] In some implementations, the payment module detects (2202), via the first interface module, a preset sequence of payment acceptance signals from the coin receiving switch that causes the control unit to initiate the operation of the offline-payment operated machine, where the preset sequence of payment acceptance signals are indicative of a preset number of coins received by the coin receiving switch. For example, with reference to Figures 34A-34B, the first interface module 2122 of

the payment module 2120 samples payment acceptance signals via lead 2132 from the microswitch 2102 to the control unit 2106. For example, each of the payment acceptance signals is indicative of reception of a coin by the microswitch 2102. Continuing with this example, the second interface module 2124 of the payment module 2120 samples control signals via lead 2134 from the control unit 2106 to the transistor 2110. The payment module 2120 detects a preset sequence of payment acceptance signals from the microswitch 2102 that causes the control unit 2106 to apply a current to the gate of the transistor 2110 (e.g., the control signals). For example, the preset sequence of payment acceptance signals is indicative of a preset number of coins received by the microswitch 2102 to cause operation of the offline-payment operated machine 2100. For example, the application of current to the gate of the transistor 2110 allows current to flow from the power supply 2108 to the operation unit 2112 so that the operation. For example, the operation unit 2112 is a motor of a dryer which begins spinning once current flows from the power supply 2108.

[0297] In some implementations, the payment module determines (2204) the predefined signal sequence to emulate the preset sequence of payment acceptance signals from the coin receiving switch. In some implementations, after detecting the preset sequence of payment acceptance signals that causes the control unit 2106 to initiate the operation of the offline-payment operated machine 2100, the payment module 2120 determines a predefined signal sequence to emulate the preset sequence of payment acceptance signals. In some implementations, the money value associated with each pulse in the preset sequence of payment acceptance signals from the microswitch 2102, indicative of the preset number of coins to initiate the operation of the offline-payment operated machine 2100, is a default currency (e.g., USD) and amount (e.g., $0.25) set in the firmware of the payment module 2120. In some implementations, the money value associated with the each pulse in the preset sequence of payment acceptance signals from the microswitch 2102, indicative of the preset number of coins to initiate the operation of the offline-payment operated machine 2100, is set by the server 130 and can be changed remotely by using the mobile device 150 as a communications bridge to send information indicating the value of a pulse from the server 130 to the mobile device 150 via the second communication capability (e.g., GSM, CDMA, or Wi-Fi) and forwarding the information from the mobile device to the payment module 2120 via the first communication capability (e.g., BLE). For instance, in most cases, each pulse is US $0.25.

[0298] In some implementations, determining the predefined signal sequence includes (2206) at least one of: identifying a count of pulses in the present sequence of payment acceptance signals; identifying amplitude of pulses in the present sequence of payment acceptance signals; identifying shape of pulses in the present sequence of payment acceptance signals; and identifying an interval between pulses. In some implementations, after detecting the preset sequence of payment acceptance signals (e.g., the sample 2150, Figure 34B), the payment module 2120 determines a predefined signal sequence to emulate the preset sequence of payment acceptance signals by identifying a count of pulses in the preset sequence of payment acceptance signals, an interval between pulses in the preset sequence of payment acceptance signals, the shape of pulses in the preset sequence of payment acceptance signals, and an amplitude of pulses in the preset sequence of payment acceptance signals.

[0299] The payment module receives (2208) a request via the short-range communication capability from a respective mobile device to perform an operation of the offline-payment operated machine. For example, with reference to Figure 8C, the payment module 2120 (Figure 34A) receives the AuthGrant from the mobile device 150 via the short-range communication capability (e.g., BLE) indicating that the user of the mobile device 150 wishes to perform the operation of the offline-payment operated machine 2100 (Figure 34A). For example with reference to operation 1012 in Figure 23, the payment module 2120 (Figure 34A) receives an authorization grant token from the mobile device 150 via the short-range communication capability (e.g., BLE) indicating that the user of the mobile device 150 wishes to perform the operation of the offline-payment operated machine 2100 (Figure 34A).

[0300] The payment module validates (2210) the request. Validation of the request indicates (2212) that the respective mobile device is authorized to initiate payment for the operation by a remote server via the long-range communication capability. In some implementations, the payment module 2120 validates the request from the mobile device 150 by determining whether the AuthGrant or the authorization grant token includes a valid authorization code.

[0301] In accordance with a determination that the request is valid, the payment module causes (2220) the payment operated machine to perform the operation by issuing a predefined signal sequence to the control unit, where the predefined signal sequence emulates a signal sequence that would be issued by the coin receiving switch in response to receiving a preset number of coins. For example, with reference to Figure 34B, the payment module 2120 issues a predefined signal sequence with first interface module 2122 to the control unit 2106 that emulates sample 2150 in Figure 34B. Continuing with this example, in response to receiving the predefined signal sequence from the payment module 2120 control unit 2106 causes initiation of the operation of the offline-payment operated machine 2100 by applying current to the gate of the transistor 2110 which allows current to flow from the power supply 2108 to operation unit 2112. In some implementations, the control unit 2106 causes initiation of the operation by setting a timer to an amount of time corresponding to the preset number of coins

whereby current flows to the gate of the transistor 2110 for the set amount of time. For example, the preset number of coins is a number of a coins required to run the offline-payment operated machine 2100 by for a default amount of time and subsequent coins may be added to extend the amount of time that the offline-payment operated machine 2100 by will run. In some implementations, the preset number of coins is a number of a coins required to cause the offline-payment operated machine 2100 to dispense a purchased item, such as laundry detergent.

[0302] Alternatively, in some implementations, in accordance with a determination that the request is valid, the offline-payment operated machine 2100 displays credit to the user (e.g., via one of the displays 122 or 124 shown in Figure 19) and the user interacts with the input mechanisms of the offline-payment operated machine 2100 (e.g., via the buttons 126 or a touch screen display 124 shown in Figure 19) to perform the operation of the machine. For example, if the offline-payment operated machine 2100 is a dryer, the user of the mobile device 150 selects the appropriate spin cycle via input mechanisms of the dryer, and when the user of the mobile device 150 selects a start/run input mechanism of the dryer, control unit 1506 of the dryer causes initiation of the operation of the dryer (e.g., starting a motor that corresponds to operation unit 2112 in Figure 34A).

[0303] In some implementations, prior to sending the operation information and after causing the offline-payment operated machine to perform the operation by issuing the predefined signal sequence to the control unit, the payment module obtains (2216) a notification from the offline-payment operated machine indicating initiation of the operation of the offline-payment operated machine and the preset number of coins. For example, after issuing the preset signal sequence to control unit 2106, the payment module 2120 (Figure 34A) obtains a notification indicating that the control unit 2106 sent control signals to initiate operation of the offline-payment operated machine 2100 in response to receiving the predefined signal sequence. For example, the notification is obtained by the second interface module 2124 (e.g., the sample 2170, Figure 34B) sampling controls signals sent by control unit 2106 (e.g., application of current to the gate of the transistor 2110 which allows current to flow from the power supply 2108 to operation unit 2112).

[0304] In response to receiving the notification, the payment module (2218): generates the operation information based at least in part on the notification; and stores the generated operation information in the memory. For example, after obtaining the notification, the payment module 2120 (Figure 34A) generates operation information corresponding to performance of the operation and the preset number of coins associated with the predefined signal sequence (e.g., the amount required to initiate operation of the offline-payment operated machine 2100) and stores the operation information in memory local to the payment module 2120 (e.g., the memory 760, Figure 20).

[0305] In some implementations, the payment module sends (2220) operation information corresponding to the operation to the respective mobile device via the short-range communication capability. For example, after operation 2218, the payment module 2120 (Figure 34A) sends the operation information to the mobile device 150 via the first communication capability of the mobile device 150 such as a radio/transceiver means for communicating via one or more short-range communication protocols such as BLE, NFC, and/or the like (i.e., a non-persistent communication channel)

[0306] It should be understood that the particular order in which the operations in Figures 35A-35B have been described is merely for example purposes and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein. Additionally, it should be noted that details of other processes described herein with respect to other methods described herein (e.g., the method 2300 in Figure 36) are also applicable in an analogous manner to the method 2200 described above with respect to Figures 35A-35B.

[0307] Figure 36 illustrates a flowchart diagram of a method 2300 of enabling a payment operated machine to accept electronic payments in accordance with some implementations. In some implementations, the method 2300 is performed by an offline-payment operated machine (e.g., the payment accepting unit 120, Figures 5 and 19 (sometimes also herein called "machine 120"), or the offline-payment operated machine 2100, Figure 34A) such as dryer or washer in a laundromat, a parking meter, a car wash payment kiosk, or the like.

[0308] In some implementations, the offline-payment operated machine includes a control unit (e.g., the control unit 2106, Figure 34A), memory, and a coin receiving switch (e.g., the microswitch 2102, Figure 34A). In some implementations, the offline-payment operated machine also includes a short-range communication capability corresponding to a short-range communication protocol (e.g., a non-persistent communication channel such as BLE, NFC, and/or the like), where the short-range communication capability is configured to communicate with one or more mobile devices, where each of the one or more mobile devices is configured with a complimentary short-range communication capability and a long-range communication capability corresponding to a long-range communication protocol (e.g., Wi-Fi, CDMA, GSM, and/or the like). For example, in some implementations, the method 2300 is performed by the machine 120, (Figures 5 and 19). In some implementations, the method 2300 is governed by instructions that are stored in a non-transitory computer readable storage medium and the instructions are executed by the control unit of the offline-payment operated machine.

[0309] The offline-payment operated machine receives (2302) a request via a short-range communication

capability from a respective mobile device to perform an operation of the offline-payment operated machine. For example, with reference to Figure 8C, the payment module 2120 (Figure 34A) receives the AuthGrant from the mobile device 150 via the short-range communication capability (e.g., BLE) indicating that the user of the mobile device 150 wishes to perform the operation of the offline-payment operated machine 2100 (Figure 34A). For example with reference to operation 1012 in Figure 23, the payment module 2120 (Figure 34A) receives an authorization grant token from the mobile device 150 via the short-range communication capability (e.g., BLE) indicating that the user of the mobile device 150 wishes to perform the operation of the offline-payment operated machine 2100 (Figure 34A).

[0310] The offline-payment operated machine validates (2304) the request. Validation of the request indicates (2306) that the respective mobile device is authorized to initiate payment for the operation by a remote server via the long-range communication capability. In some implementations, the payment module 2120 validates the request from the mobile device 150 by determining whether the AuthGrant or the authorization grant token includes a valid authorization code.

[0311] In accordance with a determination that the request is valid, the offline-payment operated machine performs (2308) the operation by issuing a predefined signal sequence to the control unit, where the predefined signal sequence emulates a preset number of coins received by the coin receiving switch. For example, in accordance with a determination that the request is valid, the offline-payment operated machine or a component thereof issues a predefined signal sequence to the control unit 2106 that emulates sample 2150 in Figure 34B. Continuing with this example, in response to receiving the predefined signal sequence from the payment module 2120, control unit 2106 causes initiation of the operation of the offline-payment operated machine 2100 by applying current to the gate of the transistor 2110 which allows current to flow from the power supply 2108 to operation unit 2112. In another example, in accordance with a determination that the request is valid, the control unit 2106 causes initiation of the operation of the offline-payment operated machine 2100 by applying current to the gate of the transistor 2110 which allows current to flow from the power supply 2108 to operation unit 2112.

[0312] It should be understood that the particular order in which the operations in Figure 36 have been described is merely for example purposes and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein. Additionally, it should be noted that details of other processes described herein with respect to other methods described herein (e.g., the method 2200 in Figures 35A-35B) are also applicable in an analogous manner to the method 2300 described above with respect to Figure 36.

[0313] Figure 37 is a block diagram of a device 2400 for retrofitting the payment accepting unit 120 (sometimes also herein called "machine 120") to accommodate a plurality of payment peripherals 2430 in accordance with some implementations. The device 2400 is similar to and adapted from adapter module 100 (sometimes also herein called "payment module 100") as shown in Figure 20 in that the device 2400 connects to a multi-drop bus (MDB) of payment accepting unit 120 and, optionally, provides the payment processing functionalities discussed in Figures 7, 8A-8G, 9A-9E, and 23 (e.g., via the internal payment peripheral 2440).

[0314] In some implementations, during normal operation, the payment accepting unit 120 includes a multi-drop bus (MDB) connecting a payment accepting unit controller 2460 of the payment accepting unit 120 with payment peripherals (e.g., other payment peripheral(s) 2450, 2455 including coin acceptors, bill acceptors, cashless payment devices such as a payment card reader, and/or the like). In some implementations, the device 2400 is connected in-line to the MDB as shown in Figures 17 and 18. In some implementations, the MDB protocol or the payment accepting unit 120 is configured to support a limited number of payment peripherals or does not support particular payment peripherals. For example, in some circumstances, the payment accepting unit 120 supports a maximum of two cashless payment devices, or the payment accepting unit 120 only supports a bill acceptor and a coin acceptor but not cashless payment devices or other payment peripherals. The device 2400 expands the number of payment peripherals connected to the payment accepting unit 120 beyond this limited number and enables support for a plurality of payment peripherals, which may or may not be compliant with the payment accepting unit 120 and/or the MDB protocol.

[0315] In Figure 37, the device 2400 is configured to perform as a virtual payment peripheral of the payment accepting unit 120 and to perform as a virtual payment accepting unit for the one or more payment peripherals 2430. As such, in some implementations, the payment accepting unit controller 2460 views the device 2400 as another payment peripheral connected to the MDB, where the device 2400 sends signals to the payment accepting unit controller 2460 in a manner as if originated by the device 2400 that is functioning as a singular virtual payment peripheral. Moreover, in some implementations, the one or more payment peripherals 2430 view the device 2400 as the payment accepting unit controller 2460, where signals are sent to the one or more payment peripherals 2430 in a manner as if originated by the payment accepting unit 120. To accomplish this, the device 2400 manages and hosts the one or more payment peripherals 2430. Additionally, the device 2400 translates addresses and modifies the communications as necessary to ensure the payment accepting unit 120 understands the traffic that is coming through to it as a singular virtual payment peripheral.

[0316] In Figure 37, the device 2400 includes a slave

interface 2402 (e.g., the male adapter 720, Figure 20) and an additional interface 2404 (e.g., the female adapter 730, Figure 20) for connecting the device 2400 to the MDB. In some implementations, the device 2400 includes a pass-through channel to enable signals from the payment accepting unit controller 2460 to reach other payment peripheral(s) 2455 and to enable signals from other the payment peripheral(s) 2455 to reach the payment accepting unit controller 2460. In Figure 37, the device 2400 also includes a device controller 2410 with a processing unit 2412 (e.g., including one or more processors, cores, microcontrollers, microprocessors, or the like) and memory 2414 storing one or more programs for execution by the processing unit 2412. In some implementations, the one or more programs cause the device 2430 to perform as a virtual payment peripheral of the payment accepting unit 120 and to perform as a virtual payment accepting unit for the one or more payment peripherals 2430. In Figure 37, the device 2400 also includes one or more host interfaces 2420 (e.g., MDB ports or non-MDB ports) for connecting the device 2400 with one or more payment peripherals 2430 (e.g., payment peripherals 2430-A to 2430-N).

[0317] In some implementations, device 2400 optionally includes internal payment peripheral 2440 with hardware, software, firmware, or a combination thereof for providing the payment processing functionalities discussed in Figures 7, 8A-8G, 9A-9E, and 23 (e.g., including the security unit 755 and the communications unit 770 shown in Figure 20).

[0318] Figure 38 illustrates a schematic flow diagram of a payment peripheral registration process 2500 in accordance with some implementations. As a result of process 2500, the device 2400 is registered as a slave (e.g., a payment peripheral) to the payment accepting unit 120, and the one or more payment peripherals 2430 are registered as slaves to the device 2400, for example, in accordance with MDB protocol.

[0319] In some implementations, the payment accepting unit 120 (i.e., the payment accepting unit controller 2460, Figure 37) polls (2502) the device 2400.

[0320] In some implementations, in response to the poll command, the device 2400 sends (2504) a reset signal to the payment accepting unit 120. For example, the device 2400 sends the reset signal to the payment accepting unit 120 if it has not yet been registered as a slave (e.g., a payment peripheral). In another example, the device 2400 sends the reset signal to re-register itself as a slave. In some implementations, the device 2400 identifies itself as a coin acceptor, a bill acceptor, or a cashless payment device to the payment accepting unit 120 via the reset signal.

[0321] In some implementations, in response to the reset signal, the payment accepting unit 120 sends (2506) a setup signal to the device 2400. In some implementations, the setup signal includes an address assigned to the device 2400.

[0322] In some implementations, after receiving and processing the setup signal, the device 2400 sends (2508) an acknowledgement to the payment accepting unit 120 confirming registration as a slave.

[0323] In some implementations, the device 2400 polls (2512) the payment peripheral 2430-A.

[0324] In some implementations, in response to the poll command, the payment peripheral 2430-A sends (2514) a reset signal to the device 2400. For example, the payment peripheral 2430-A sends the reset signal to the device 2400 if it has not yet been registered as a slave (e.g., a payment peripheral) to the device 2400. In another example, the payment peripheral 2430-A sends the reset signal to re-register itself as a slave. In some implementations, the payment peripheral 2430-A identifies itself as a coin acceptor, a bill acceptor, or a cashless payment device to the device 2400 via the reset signal.

[0325] In some implementations, in response to the reset signal, the device 2400 sends (2516) a setup signal to the payment peripheral 2430-A. In some implementations, the setup signal includes an address assigned to the payment peripheral 2430-A.

[0326] In some implementations, after receiving and processing the setup signal, the payment peripheral 2430-A sends (2518) an acknowledgement to device 2400 confirming registration as a slave.

[0327] In some implementations, the device 2400 polls (2522) the payment peripheral 2430-N.

[0328] In some implementations, in response to the poll command, the payment peripheral 2430-N sends (2524) a reset signal to the device 2400. For example, the payment peripheral 2430-N sends the reset signal to the device 2400 if it has not yet been registered as a slave (e.g., a payment peripheral). In another example, the payment peripheral 2430-N sends the reset signal to re-register itself as a slave. In some implementations, the payment peripheral 2430-N identifies itself as a coin acceptor, a bill acceptor, or a cashless payment device to the device 2400 via the reset signal.

[0329] In some implementations, in response to the reset signal, the device 2400 sends (2526) a setup signal to the payment peripheral 2430-N. In some implementations, the setup signal includes an address assigned to the payment peripheral 2430-N.

[0330] In some implementations, after receiving and processing the setup signal, the payment peripheral 2430-N sends (2528) an acknowledgement to the device 2400 confirming registration as a slave.

[0331] Figures 39A-39B illustrate a schematic flow diagram of a payment process 2600 in accordance with some implementations. In some implementations, device 2400 has already been registered as a slave (i.e., a payment peripheral) to payment accepting unit 120 and payment peripherals 2430-A, 2430-N have already been registered as slaves to device 2400 according to process 2500 in Figure 38.

[0332] In some implementations, the payment accepting unit 120 polls the device 2400, along with other payment peripherals connected to the MDB and registered

as slaves (e.g., other payment peripherals 2450, 2455 (Figure 37)), according to a predetermined time period (e.g., 5 ms). For example, the predetermined time period is assigned by the MDB protocol or specification (e.g., versions 1.0 to 3.0 or higher), which is incorporated herein by reference in its entirety. In response to the poll commands, all slave devices (e.g., at least including the device 2400) respond with an acknowledgment (e.g., indicating that it is still present on the MDB) or with another signal (e.g., indicating another state). In some implementations, in response to a command from payment accepting unit 120, the device 2400 immediately responds to the command and asynchronously relays the command to at least one of the one or more payment peripherals 2430.

[0333] In some implementations, in a manner similar to the payment accepting unit 120, the device 2400 also polls all of the one or more payment peripherals 2430 according to the predetermined time period (e.g., 5 ms). For example, the device 2400 polls all of the one or more payment peripherals 2430 whenever it is polled by the payment accepting unit 120.

[0334] In some implementations, the payment accepting unit 120 (i.e., the payment accepting unit controller 2460, Figure 37) polls (2602) the device 2400.

[0335] In response to the polling command in operation 2602, the device 2400 sends (2604) an acknowledgment to the payment accepting unit 120.

[0336] In response to or independent of the polling command in operation 2602, the device 2400 also polls (2606) the payment peripheral 2430-N. In response to the polling command in operation 2606, the payment peripheral 2430-N sends (2608) an acknowledgment to the device 2400.

[0337] In response to or independent of the polling command in operation 2602, the device 2400 also polls (2610) the payment peripheral 2430-A. In response to the polling command in operation 2610, the payment peripheral 2430-A sends (2612) a request to begin a payment session. For example, the request to begin the payment session is sent in response to a user inserting payment (e.g., a bill(s) or coin(s)) into the payment peripheral 2430-A prior to the polling command in operation 2610.

[0338] In response to the request to begin the payment session, the device 2400 sends (2614) an acknowledgment to the payment peripheral 2430-A.

[0339] In response to the request to begin the payment session, the device 2400 also sends a disable command to the payment peripheral 2430-N so as to disable the payment peripheral 2430-N while processing the payment session for the payment peripheral 2430-A. In response to the disable command, the payment peripheral 2430-N sends (2618) an acknowledgment to the device 2400.

[0340] The payment accepting unit 120 (i.e., the payment accepting unit controller 2460, Figure 37) polls (2620) the device 2400.

[0341] In response to the polling command in operation 2620, the device 2400 sends (2622) a request to begin a payment session to the payment accepting unit 120. For example, the request to begin the payment session mirrors the request to begin the payment session received from the payment peripheral 2430-A.

[0342] In response to the request to begin the payment session in operation 2622, the payment accepting unit 120 sends (2624) an acknowledgement to the device 2400 and also sends (2626) a vend request to the device 2400. In process 2600, vending of a service or product is taken as a non-limiting example.

[0343] In response to receiving the vend request, the device 2400 sends (2628) an acknowledgment to the payment accepting unit 120.

[0344] In some implementations, the payment accepting unit 120 polls (2630) the device 2400 N times prior to sending the vend approved signal in operation 2640. In some implementations, the device 2400 responds to the N polling command with acknowledgments indicating that the device 2400 is still present and processing the vend request.

[0345] In response to receiving the vend request, the device 2400 also relays (2630) the vend request to the payment peripheral 2430-A.

[0346] In response to the vend request, the payment peripheral 2430-A sends (2632) an acknowledgement to the device 2400.

[0347] Subsequently, the device 2400 polls (2634) the payment peripheral 2430-A. In response to the polling command in operation 2634, the payment peripheral 2430-A sends (2636) a vend approved signal to the device 2400. For example, the vend approved signal indicates that the payment inserted by the user was not-refunded and was used to purchase a service or product.

[0348] In response to receiving the vend approved signal, the device 2400 sends (2638) an acknowledgment to the payment peripheral 2430-A and also relays (2640) the vend approved signal to the payment accepting unit 120.

[0349] In response to receiving the vend approved signal, the payment accepting unit 120 sends (2642) an acknowledgment to the device 2400 and also sends (2644) a request to the device 2400 to indicate whether the vend was a success or a failure.

[0350] In response to receiving the request in operation 2644, the device 2400 sends (2646) a response to the payment accepting unit 120 indicating that the vend was a success or a failure and also relays (2650) the request to the payment peripheral 2430-A to indicate whether the vend was a success or a failure.

[0351] In response to the request in operation 2650, the payment peripheral 2430-A sends (2652) an acknowledgement to the device 2400.

[0352] In response to receiving the response in operation 2646, the payment accepting unit 120 sends (2648) an acknowledgement to the device 2400 and also sends (2654) a command to end the payment session to the device 2400.

[0353] In response to receiving the command to end the payment session, the device 2400 sends (2656) an acknowledgment to the payment accepting unit 120 and relays (2658) the command to end the payment session to the payment peripheral 2430-A.

[0354] In response to the command to end the payment session, the payment peripheral 2430-A sends (2660) an acknowledgment to the device 2400.

[0355] After receiving the acknowledgment from the payment peripheral 2430-A, the device 2400 sends (2662) an enable command to the payment peripheral 2430-N so as to enable the payment peripheral 2430-N after completion of the payment session for the payment peripheral 2430-A. In response to the enable command, the payment peripheral 2430-N sends (2664) an acknowledgment to the device 2400.

[0356] Figures 40A-40D illustrate a flowchart diagram of a method 2700 retrofitting a payment accepting unit to accommodate a plurality of payment peripherals in accordance with some implementations. In some implementations, the method 2700 is performed by a device with one or more processors, memory, a slave interface configured to couple the device with the payment accepting unit via a multi-drop bus (MDB), and one or more host interfaces configured to couple the device with one or more payment peripherals, (e.g., a coin acceptor, a bill acceptor, a cashless payment device such as a payment card reader, and the like) where a respective payment peripheral is decoupled from an MDB interface of the payment accepting unit and coupled with a respective one of the one or more host interfaces, and where the one or more payment peripherals are configured to communicate via MDB protocol. For example, in some implementations, the method 2700 is performed by the device 2400 (Figure 37) or a component thereof (e.g., device controller 2410). In some implementations, the method 2700 is governed by instructions that are stored in a non-transitory computer readable storage medium (e.g., the memory 2414, Figure 37) and the instructions are executed by one or more processors (e.g., the processing unit 2412, Figure 37) of the device. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders).

[0357] The device performs (2702) as a virtual payment peripheral for the payment accepting unit by registering the device as a slave to the payment accepting unit, and the device performs as a virtual payment accepting unit for the one or more payment peripherals by registering the one or more payment peripherals as a slaves to the device using the MDB protocol. In some implementations, the MDB protocol supports a limited number of payment peripherals. Device 2400 expands the number of payment peripherals connected to the payment accepting unit 120 beyond this limited number by emulating the payment accepting unit 120 to the one or more payment peripherals 2430 coupled with the one or more host interfaces 2420 and emulating a payment peripheral to the payment accepting unit 120. As such, in some implementations, the payment accepting unit 120 (i.e., the payment accepting unit controller 2460) views the device 2400 as another payment peripheral connected to the MDB, where the device 2400 sends signals to the payment accepting unit controller 2460 in a manner as if originated by the device 2400 that is functioning as a singular virtual payment peripheral. Moreover, in some implementations, the one or more payment peripherals 2430 view the device 2400 as the payment accepting unit controller 2460, where signals are sent to the one or more payment peripherals 2430 in a manner as if originated by the payment accepting unit controller 2460.

[0358] In some implementations, registering the device as a slave to the payment accepting unit further comprises (2716): identifying the device to the payment accepting unit as a cashless payment peripheral; and accepting registration of the device with the payment accepting unit as a cashless payment peripheral. For example, the device 2400 identifies itself to the payment accepting unit 120 as a cashless payment device (e.g., a payment card reader) when sending the reset signal to the payment accepting unit 120 in operation 2504 (Figure 38).

[0359] In some implementations, registering the device as a slave to the payment accepting unit further comprises (2718): identifying the device to the payment accepting unit as a coin acceptor peripheral; and accepting registration of the device with the payment accepting unit as a coin acceptor peripheral. For example, the device 2400 identifies itself to the payment accepting unit 120 as a coin acceptor when sending the reset signal to the payment accepting unit 120 in operation 2504 (Figure 38).

[0360] In some implementations, registering the device as a slave to the payment accepting unit further comprises (2720): identifying the device to the payment accepting unit as a bill acceptor peripheral; and accepting registration of the device with the payment accepting unit as a bill acceptor peripheral. For example, the device 2400 identifies itself to the payment accepting unit 120 as a bill acceptor/validator when sending the reset signal to the payment accepting unit 120 in operation 2504 (Figure 38).

[0361] The device receives (2704) a command from the payment accepting unit via the slave interface, where signals from the payment accepting unit are sent in a manner as if sent to a singular payment peripheral. For example, with reference to process 2600, the payment accepting unit 120 sends a command to the device 2400 to end the payment session in operation 2654 (Figure 39B).

[0362] In response to receiving the command from the payment accepting unit, the device (2706): sends an acknowledgement to the command from the payment accepting unit via the slave interface, where signals are sent to the payment accepting unit in a manner as if originated by the device that is functioning as a singular virtual payment peripheral; and relays the command to the

respective payment peripheral via the respective one of the one or more host interfaces corresponding to the respective payment peripheral, where the device sends signals to and receives signals from the payment accepting unit asynchronous of the device sending signals to and receiving signals from the one or more payment peripherals. Continuing with the example above, with reference to process 2600, in response to receiving the command to end the payment session, the device 2400 sends an acknowledgment to the payment accepting unit 120 in operation 2656 (Figure 39B)in a manner as if originated by the device that is functioning as a singular virtual payment peripheral. Continuing with this example, in response to receiving the command to end the payment session, the device 2400 also asynchronously relays the command to end the payment session to the payment peripheral 2430-A in operation 2658 (Figure 39B). As such, the command is relayed to the payment peripheral 2430-A asynchronous of sending the acknowledgment to the payment accepting unit 120.

[0363] In some implementations, in response to relaying the command, the device receives (2708) via the respective one of the one or more host interfaces corresponding to the respective payment peripheral a response from the respective payment peripheral. For Continuing with the example above, with reference to process 2600, in response to the relayed complete session command, the payment peripheral 2430-A sends an acknowledgment to the device 2400 in operation 2660 (Figure 39B).

[0364] In some implementations, in response to receiving the response from the respective payment peripheral, the device: sends an acknowledgement to the respective payment peripheral, where signals are sent to the one or more payment peripherals in a manner as if originated by the payment accepting unit; and relays the response to the payment accepting unit via the slave interface, where the device sends signals to and receives signals from the payment accepting unit asynchronous of the device sending signals to and receiving signals from the one or more payment peripherals. In some implementations, in response to receiving the response from the respective payment peripheral, the device forgoes the above steps.

[0365] In some implementations, the device receives (2710) a command from respective payment peripheral via the respective one of the one or more host interfaces corresponding to the respective payment peripheral, where signals from the one or more payment peripherals are sent in a manner as if sent to the payment accepting unit, and, in response to receiving the command from the respective payment peripheral, the device: sending an acknowledgement to the command from the respective payment peripheral, where signals are sent to the one or more payment peripherals in a manner as if originated by the payment accepting unit; and relaying the command to the payment accepting unit via the slave interface, where the device sends signals to and receives signals from the payment accepting unit asynchronous of the device sending signals to and receiving signals from the one or more payment peripherals. For example, with reference to process 2600, when polled in operation 2634 (Figure 39B), the payment peripheral 2430-A sends a vend approved signal to the device 2400 in a manner as if sent to the payment accepting unit 120 in operation 2636. Continuing with this example, in response to receiving the vend approved signal, the device 2400 sends an acknowledgement to the payment peripheral 2430-A in a manner as if originated by the payment accepting unit 120 in operation 2638 (Figure 39B). Continuing with this example, in response to receiving the vend approved signal, the device 2400 also asynchronously relays the vend approved signal to the payment accepting unit 120 in operation 2640 (Figure 39B). As such, the command is relayed to the payment accepting unit 120 asynchronous of sending the acknowledgment to the payment peripheral 2430-A.

[0366] In some implementations, the device further includes an internal payment peripheral including a short-range communication capability corresponding to a short-range communication protocol, where the short-range communication capability is configured to communicate with one or more mobile devices, and where each of the one or more mobile devices is configured with a complimentary short-range communication capability and a long-range communication capability corresponding to a long-range communication protocol. For example, with reference to Figure 37, the device 2400 includes the internal payment peripheral 2440 which includes hardware, software, firmware, or a combination thereof for providing the payment processing functionalities discussed in Figures 7, 8A-8G, 9A-9E, and 23 (e.g., the security unit 755 and the communications unit 770 as shown in Figure 20). For example, the respective mobile device corresponds to mobile device 150 (Figure 21) with long-range communication capability 872 and short-range communication capability 876.

[0367] In some implementations, the device receives (2712) a transaction request via the short-range communication capability from a respective mobile device to perform a transaction with the payment accepting unit, validates the transaction request, where validation of the transaction request indicates that the respective mobile device is authorized to initiate payment for the transaction by a remote server via the long-range communication capability, and, in accordance with a determination that the transaction request is valid, causing the payment accepting unit to perform the requested transaction by, issuing a signal to perform the transaction to the payment accepting unit via the slave interface. In some implementations, the device 2400 or a component thereof (e.g., internal payment peripheral 2440, Figure 37) receives a transaction request via the short-range communication capability (e.g., BLE, NFC, or the like) from the respective mobile device 150 (Figures 7, 8A-8G, 9A-9E, and 21), and the device 2400 or a component thereof (e.g., inter-

nal payment peripheral 2440, Figure 37; or the device controller 2410, Figure 37) validates the transaction request from the respective mobile device 150 by determining whether an AuthGrant or authorization grant token associated with the transaction request includes a valid authorization code. In some implementations, in accordance with a determination that the transaction request is associated with a valid authorization code, the device 2400 or a component thereof (e.g., internal payment peripheral 2440, Figure 37; or the device controller 2410, Figure 37) issues a command to the payment accepting unit 120 to perform the requested transaction by via the slave interface 2402 in a manner as if originated by the device 2400 that is functioning as a singular virtual payment peripheral.

[0368] In some implementations, in accordance with a determination that a command received from the respective one of the one or more payment peripherals corresponds to a transaction, the device stores (2714) transaction information at least including an amount of the transaction in associated with an identifier for the respective one of the one or more payment peripherals; after storing the transaction information, sends the transaction information to the respective mobile device via the short-range communication capability; and issues a command to the respective mobile device to send the transaction information to the remote server via the long-range communication capability. In some implementations, the device 2400 or a component thereof (e.g., the internal payment peripheral 2440, Figure 37; or the device controller 2410, Figure 37) monitors commands and signals from the one or more payment peripherals 2430 that are relayed to the payment accepting unit 120 and, in accordance with a determination that the command and signals are associated with transactions, stores transaction information such as the transaction amount and the respective payment peripheral 2430 associated with the transaction. For example, the device 2400 stores transaction information for each of the one or more payment peripherals 2430 in a table that associates the transaction information with a payment peripheral type (e.g., bill acceptor, coin acceptor, payment card reader, etc.). In some implementations, the device 2400 or a component thereof (e.g., the internal payment peripheral 2440, Figure 37; or the device controller 2410, Figure 37) sends the table of transaction information or a portion thereof to the respective mobile device 150 that sent the transaction request (or another mobile device 150 that performs a transaction with the device 2400) via the short-range communication capability. In some implementations, the device 2400 or a component thereof (e.g., the internal payment peripheral 2440, Figure 37; or the device controller 2410, Figure 37) commands the respective mobile device 150 to send the table of transaction information or the portion thereof to the server 130 via its long-range communication capability. As such, the device 2400 uses the respective mobile device 150 as a communication bridge to the server 130.

[0369] In some implementations, the device 2400 or a component thereof (e.g., the internal payment peripheral 2440, Figure 37; or the device controller 2410, Figure 37) also monitors the commands and signals from the one or more payment peripherals 2430 that are relayed to the payment accepting unit 120 and, in accordance with a determination that the command and signals are associated with error codes (e.g., a coin jam, low coin or bill count, etc.) and other information associated with the operation of the one or more payment peripherals 2430, stores corresponding operation information. In some implementations, the device 2400 also sends the operation information along with the table the table of transaction information or the portion thereof to the server 130.

[0370] In some implementations, the payment accepting unit further includes one or more other payment peripherals coupled with the MDB, a respective payment peripheral of the one or more other payment peripherals is one of a bill acceptor, coin acceptor, or payment card reader, and where the device further includes an additional interface configured to couple the device with the one or more other payment peripherals of the payment accepting unit. For example, with reference to Figure 37, the other payment peripheral(s) 2450 (e.g., acceptors, coin acceptors, payment card readers, etc.) are connected to the MDB before the device 2400 (e.g., prior to the slave interface 2402) and the other payment peripheral(s) 2455 (e.g., acceptors, coin acceptors, payment card readers, etc.) are connected to the MDB after the device 2400 (e.g., after the additional interface 2404).

[0371] In some implementations, the device further includes a pass-through channel, the pass-through channel is configured to pass-through signals from the one or more other payment peripherals to the payment accepting unit. For example, with reference to Figure 37, the device 2400 includes a pass-through channel to enable signals from the payment accepting unit controller 2460 to reach the other payment peripheral(s) 2455 and to enable signals from other the payment peripheral(s) 2455 to reach the payment accepting unit controller 2460.

[0372] It should be understood that the particular order in which the operations in Figures 40A-40D have been described is merely exemplary and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein. Additionally, it should be noted that details of other processes described herein with respect to other methods described herein are also applicable in an analogous manner to the method 2700 described above with respect to Figures 40A-40D.

MISCELLANEOUS

[0373] It should be noted that relative terms are meant to help in the understanding of the technology and are not meant to limit the scope of the invention. Similarly, unless specifically stated otherwise, the terms used for

labels (e.g., "first," "second," and "third") are meant solely for purposes of designation and not for order or limitation. The term "short" in the phrase "short-range" (in addition to having technology specific meanings) is relative to the term "long" in the phrase "long-range."

[0374] The terms "may," "might," "can," and "could" are used to indicate alternatives and optional features and only should be construed as a limitation if specifically included in the claims.

[0375] It should be noted that, unless otherwise specified, the term "or" is used in its nonexclusive form (e.g., "A or B" includes A, B, A and B, or any combination thereof, but it would not have to include all of these possibilities). It should be noted that, unless otherwise specified, "and/or" is used similarly (e.g., "A and/or B" includes A, B, A and B, or any combination thereof, but it would not have to include all of these possibilities). It should be noted that, unless otherwise specified, the terms "includes" and "has" mean "comprises" (e.g., a device that includes, has, or comprises A and B contains A and B, but optionally may contain C or additional components other than A and B). It should be noted that, unless otherwise specified, the singular forms "a," "an," and "the" refer to one or more than one, unless the context clearly dictates otherwise.

[0376] It is to be understood that the inventions, examples, and implementations described herein are not limited to particularly exemplified materials, methods, and/or structures. It is to be understood that the inventions, examples, and implementations described herein are to be considered preferred inventions, examples, and implementations whether specifically identified as such or not.

[0377] The terms and expressions that have been employed in the foregoing specification are used as terms of description and not of limitation, and are not intended to exclude equivalents of the features shown and described. While the above is a complete description of selected implementations of the present invention, it is possible to practice the invention using various alternatives, modifications, adaptations, variations, and/or combinations and their equivalents. It will be appreciated by those of ordinary skill in the art that any arrangement that is calculated to achieve the same purpose may be substituted for the specific embodiment shown. It is also to be understood that the following claims are intended to cover all of the generic and specific features of the invention herein described and all statements of the scope of the invention that, as a matter of language, might be said to fall therebetween.

[0378] The following clauses, which form part of the description, provide general expressions of the disclosure herein:

A1. A mobile-device-to-machine payment system for facilitating a cashless transaction for purchase of at least one product or service by a user from a payment accepting unit having input mechanisms, the user having a mobile device having both short-range communication technology and long-range communication technology, the payment accepting unit capable of dispensing at least one product or service, said system comprising:

(a) an adapter module associated with the payment accepting unit, said adapter having short-range communication technology for communicating with the short-range communication technology of the mobile device;

(b) a server having long-range communication technology for communicating with the long-range communication technology of the mobile device;

(c) said adapter module for sending an authorization request for funds to the mobile device using short-range communication technology, the mobile device forwarding said authorization request for funds to said server using long-range communication technology; and

(d) said server for sending an authorization grant for funds to the mobile device using long-range communication technology, the mobile device forwarding said authorization grant for funds to said adapter module using short-range communication technology;

(e) wherein the payment accepting unit dispenses the at least one product or service in response to receiving user input to the payment accepting unit input mechanism if said adapter module has received said authorization grant.

A2. The system ofclause A1, said adapter module having security technology and said server having security technology, said authorization request being secured by said adapter module security technology to create a secured authorization request, said authorization grant being secured by said server security technology to create a secured authorization grant, and said secured authorization request and said secured authorization grant being undecipherable to the mobile device.

A3. The system of clause A1, said adapter module and said server sharing a unique private key, said adapter module having encryption/decryption technology and said server having encryption/decryption technology, said authorization request being encrypted by said adapter module encryption/decryption technology using said unique private key to create an encrypted authorization request, said encrypted authorization request being decrypted by said server encryption/decryption technology using said unique private key, said authorization grant being encrypted by said server encryption/decryption technology using said unique private key to create an encrypted authorization grant, said encrypted au-

thorization grant being decrypted by said adapter module encryption/decryption technology using said unique private key, and said encrypted authorization request and said encrypted authorization grant being undecipherable to the mobile device.

A4. The system of any ofclauses A1-A3,further comprising:

(a) said adapter module surrounded by two zones, a payment zone and an authorization zone, wherein said payment zone is within said authorization zone;
(b) said adapter module sending said authorization request when the mobile device is within said authorization zone; and
(c) the mobile device forwarding said authorization grant for funds to said adapter module when the mobile device is within said payment zone.

A5. The system of any ofclauses A1-A3,further comprising:

(a) said adapter module surrounded by three zones, a payment zone, an authorization zone, and a communication zone, wherein said payment zone is within said authorization zone and said authorization zone is within said communication zone;
(b) the mobile device receiving advertising broadcast signals from said adapter module within said communication zone;
(c) said adapter module sending said authorization request when the mobile device is within said authorization zone; and
(d) the mobile device forwarding said authorization grant for funds to said adapter module when the mobile device is within said payment zone.

A6. The system of any of clauses A1-A3, having a hands-free mode in which the payment accepting unit dispenses the at least one product or service without the user interacting with the mobile device.

A7. The system of any of clauses A1-A3, having a hands-free mode in which the payment accepting unit dispenses the at least one product or service without the user interacting with the mobile device, said system further comprising:

(a) a display of the payment accepting unit for displaying funds available based on information from said authorization grant; and
(b) input mechanisms of the payment accepting unit for receiving user selection input when the user interacts with the input mechanisms to select the at least one product or service to be dispensed.

A8. The system of any of clauses A1-A3, wherein said adapter module is an in-line dongle for in-line insertion within a multi-drop bus of the payment accepting unit.

A9. The system of any of clauses A1-A3:

(a) the payment accepting unit having a multi-drop bus to a payment receiving mechanism, the multi-drop bus having a male adapter and a female adapter;
(b) said adapter module having a male adapter and a female adapter; and
(c) said adapter module insertable in serial with the multi-drop bus by connecting said male adapter of said adapter module to the female adapter of the multi-drop bus and by connecting said female adapter of said adapter module to the male adapter of the multi-drop bus.

A10. A mobile-device-to-machine payment system for facilitating a cashless transaction for purchase of at least one product or service by a user from a payment accepting unit having input mechanisms, the user having a mobile device having both short-range communication technology and long-range communication technology, the payment accepting unit capable of dispensing at least one product or service, said system comprising:

(a) an adapter module associated with the payment accepting unit, said adapter having short-range communication technology for communicating with the short-range communication technology of the mobile device, said adapter module having security technology, and said adapter module surrounded by two zones, a payment zone and an authorization zone, wherein said payment zone is within said authorization zone;
(b) a server having long-range communication technology for communicating with the long-range communication technology of the mobile device, said server having security technology;
(c) said adapter module for sending a secured authorization request for funds secured by said adapter module security technology to the mobile device using short-range communication technology when the mobile device is within said authorization zone, the mobile device forwarding said secured authorization request for funds to said server using long-range communication technology; and
(d) said server for sending a secured authorization grant for funds secured by said server security technology to the mobile device using long-range communication technology, the mobile device forwarding said secured authorization grant for funds to said adapter module using

short-range communication technology when the mobile device is within said payment zone;
(e) wherein the payment accepting unit dispenses the at least one product or service in response to receiving user input to the payment accepting unit input mechanism if said adapter module has received said secured authorization grant.

A11. The system of clause A10, said secured authorization grant being undecipherable to the mobile device.

A12. The system of clause A10, said adapter module and said server sharing a unique private key, said adapter module security technology being encryption/decryption technology and said server security technology being encryption/decryption technology, said secured authorization request being encrypted by said adapter module encryption/decryption technology using said unique private key to create an encrypted secured authorization request, said encrypted secured authorization request being decrypted by said server encryption/decryption technology using said unique private key, said secured authorization grant being encrypted by said server encryption/decryption technology using said unique private key to create an encrypted secured authorization grant, said encrypted secured authorization grant being decrypted by said adapter module encryption/decryption technology using said unique private key, and said encrypted secured authorization request and said encrypted secured authorization grant being undecipherable to the mobile device.

A13. The system of any of clauses A10-A12, further comprising:

(a) said adapter module being surrounded by three zones, said
payment zone, said authorization zone, and a communication zone, wherein said authorization zone is within said communication zone; and
(b) the mobile device receiving advertising broadcast signals from said adapter module within said communication zone.

A14. The system of any of clauses A10-A12, having a hands-free mode in which the payment accepting unit dispenses the at least one product or service without the user interacting with the mobile device.

A15. The system of any of clauses A10-A12, having a hands-free mode in which the payment accepting unit dispenses the at least one product or service without the user interacting with the mobile device, said system further comprising:

(a) a display of the payment accepting unit for displaying funds available based on information from said secured authorization grant; and
(b) input mechanisms of the payment accepting unit for receiving user selection input when the user interacts with the input mechanisms to select the at least one product or service to be dispensed.

A16. The system of any of clauses A10-A12, wherein said adapter module is an in-line dongle for in-line insertion within a multi-drop bus of the payment accepting unit.

A17. The system of any of clauses A10-A12:

(a) the payment accepting unit having a multi-drop bus to a payment receiving mechanism, the multi-drop bus having a male adapter and a female adapter;
(b) said adapter module having a male adapter and a female adapter;
(c) said adapter module insertable in serial with the multi-drop bus by connecting said male adapter of said adapter module to the female adapter of the multi-drop bus and by connecting said female adapter of said adapter module to the male adapter of the multi-drop bus.

A18. A method for using a mobile-device-to-machine payment system for facilitating a cashless transaction for purchase of at least one product or service by a user from a payment accepting unit having input mechanisms, the user having a mobile device having both short-range communication technology and long-range communication technology, the payment accepting unit capable of dispensing at least one product or service, said method comprising the steps of:

(a) sending an authorization request for funds to the mobile device using short-range communication technology of an adapter module associated with the payment accepting unit;
(b) receiving said authorization request for funds from said short-range communication technology of said adapter module at the short-range communication technology of the mobile device;
(c) forwarding said authorization request for funds to a server using the long-range communication technology of the mobile device;
(d) receiving said authorization request for funds from the long-range communication technology of the mobile device at long-range communication technology of said server;
(e) sending an authorization grant for funds to the mobile device using said long-range communication technology of said server;

(f) receiving said authorization grant for funds from long-range communication technology of said server at the long-range communication technology of the mobile device;
(g) forwarding said authorization grant for funds to said adapter module using the short-range communication technology of the mobile device;
(h) receiving said authorization grant for funds from the short-range communication technology of the mobile device at short-range communication technology of said adapter module; and
(i) dispensing the at least one product or service from the payment accepting unit in response to receiving user input to the payment accepting unit input mechanism if said adapter module has received said authorization grant.

A19. The method of clause A18, further comprising: securing said authorization request using security technology associated with said adapter module to create a secured authorization request, securing said authorization grant using security technology associated with said server to create a secured authorization grant, and said secured authorization request and said secured authorization grant being undecipherable to the mobile device.

A20. The method of clause A18, further comprising: sharing a unique private key between said adapter module and said server, encrypting using said unique private key said authorization request using encryption/decryption technology associated with said adapter module to create an encrypted authorization request, decrypting using said unique private key said encrypted authorization request using encryption/decryption technology associated with said server, encrypting using said unique private key said authorization grant using said encryption/decryption technology associated with said server to create an encrypted authorization grant, decrypting using said unique private key said encrypted authorization grant using encryption/decryption technology associated with said adapter module, and said encrypted authorization request and said encrypted authorization grant being undecipherable to the mobile device.

A21. The method of any of clauses A18-A20, further comprising:

(a) surrounding said adapter module with two zones, a payment zone and an authorization zone, wherein said payment zone is within said authorization zone;
(b) said adapter module sending said authorization request when the mobile device is within said authorization zone; and
(c) the mobile device forwarding said authorization grant for funds to said adapter module when the mobile device is within said payment zone.

A22. The method of any ofclauses A18-A20, further comprising:

(a) surrounding said adapter module with three zones, a payment zone, an authorization zone, and a communication zone, wherein said payment zone is within said authorization zone and said authorization zone is within said communication zone;
(b) the mobile device receiving advertising broadcast signals from said adapter module within said communication zone;
(c) said adapter module sending said authorization request when the mobile device is within said authorization zone; and
(d) the mobile device forwarding said authorization grant for funds to said adapter module when the mobile device is within said payment zone.

A23. The method of any of clauses A18-A20, having a hands-free mode in which the payment accepting unit dispenses the at least one product or service without the user interacting with the mobile device.

A24. The method of any of clauses A18-A20, having a hands-free mode in which the payment accepting unit dispenses the at least one product or service without the user interacting with the mobile device, said method further comprising the steps of:

(a) displaying funds available on a display of the payment accepting unit, said funds available being based on information from said authorization grant; and
(b) receiving user selection input when the user interacts with input mechanisms of the payment accepting unit to select the at least one product or service to be dispensed.

A25. The method of any of clauses A18-A20, further comprising the step of inserting said adapter module as an in-line dongle for in-line insertion within a multi-drop bus of the payment accepting unit.

A26. The method of any of clauses A18-A20, the payment accepting unit having a multi-drop bus to a payment receiving mechanism, the multi-drop bus having a male adapter and a female adapter, and said adapter module having a male adapter and a female adapter, said method further comprising the step of inserting said adapter module in serial with the multi-drop bus by connecting said male adapter of said adapter module to the female adapter of the multi-drop bus and by connecting said female adapter of said adapter module to the male adapter of the multi-drop bus.

Claims

1. A method (2200) of retrofitting an offline-payment operated machine (2100) to accept electronic payments, the method being performed by a payment module (2120) including one or more processors, memory, and a short-range communication capability configured to communicate with one or more mobile devices, the payment module (2120) being coupled with the offline-payment operated machine (2100) that includes a coin receiving switch (2102) and a control unit (2106), the payment module (2120) further including an interface (2122) configured to emulate payment acceptance signals from the coin receiving switch (2102) of the offline-payment operated machine (2100), where the payment acceptance signals are indicative of coins being received by the coin receiving switch (2102), the method comprising:

   broadcasting, via the short-range communication capability, a packet of information (1100) including an identifier associated with the payment module (2120);
   receiving (2208) via the short-range communication capability and from a respective mobile device (150) of the one or more mobile devices a request to perform an operation of the offline-payment operated machine, wherein the request includes an authorization grant token (1140);
   validating (2210) the request by determining whether the authorization grant token (1140) includes a valid authorization code, wherein the validation indicates that the respective mobile device (150) is authorized to initiate payment for the operation by a remote server via a long-range communication capability of the respective mobile device (150); and
   in accordance with a determination that the request is valid, causing the payment operated machine to perform the operation by issuing a predefined signal sequence to the control unit (2106) via the interface, wherein the predefined signal sequence emulates a signal sequence that would be issued by the coin receiving switch in response to receiving the preset number of coins.

2. The method of claim 1, wherein the payment module is an adapter module (100) and the offline-payment operated machine (2100) is a payment accepting unit (120) comprising a payment receiving mechanism, wherein the payment receiving mechanism of the payment accepting unit (120) is a coin acceptor.

3. The method of claim 2, wherein the adapter module (100) is inserted as an in-line dongle for in-line insertion within a multi-drop bus of the payment accepting unit (120).

4. The method of claim 2, wherein the adapter module (100) includes a pass-through channel, wherein the pass-through channel is configured to enable signals from a payment accepting unit controller (2460) of the offline-payment operated machine (2100) to reach one or more other payment peripherals (2455) and to enable signals from one or more other payment peripherals to reach the payment accepting unit controller (2460).

5. The method of claim 1, further comprising:

   after issuing the preset signal sequence to the control unit (2106), obtaining by a second interface of the payment module (2124) a notification indicating that the control unit sent control signals to initiate operation of the offline-payment operated machine in response to receiving the predefined signal sequence; and
   in response (2218) to obtaining the notification:

      generating operation information based at least in part on the notification; and
      storing the generated operation information in the memory.

6. The method of claim 5, further comprising:
   sending (2220) the operation information corresponding to the operation to the respective mobile device (150) via the short-range communication technology.

7. The method of claim 1, wherein the packet of information (1100) includes a payment zone criterion (1126) that the mobile device (150) is required to observe before being within a payment zone of the payment module (2120).

8. The method of claim 7, wherein the request is sent in response to the mobile device (150) entering the payment zone of the payment module (2120) which occurs upon satisfaction of the payment zone criterion, wherein the payment zone criterion (1126) is adjusted by the respective mobile device (150) based on the strength and/or reception of its short-range communication capability.

9. The method of claim 8, wherein the payment zone criterion (1126) corresponds to a baseline payment zone threshold indicating a baseline received signal strength that the respective mobile device (150) is required to observe before being within the payment zone of the payment module (2120).

10. The method of claim 9, wherein the payment module

(2120) receives from the respective mobile device (150) a model type of the respective mobile device (150), wherein the baseline payment zone is subject to an offset based on the model type.

11. The method of claim 2, wherein the payment accepting unit (120) has a multi-drop bus to a payment receiving mechanism, the multi-drop bus having a male adapter and a female adapter, and said adapter module having a male adapter and a female adapter, wherein said adapter module (100) has been inserted in serial with the multi-drop bus by connecting said male adapter of said adapter module to the female adapter of the multi-drop bus and by connecting said female adapter of said adapter module to the male adapter of the multi-drop bus.

12. The method of claim 1, wherein the offline-payment operated machine 2100 is a dryer or washer at a laundromat, a parking meter, a car wash payment kiosk, or a vending machine.

13. A payment module (2120) for retrofitting an offline-payment operated machine (2100) to accept electronic payments, wherein the offline-payment operated machine (2100) at least includes a control unit (2106) and a coin receiving switch (2102), the payment module (2120) comprising:

an interface (2122) configured to emulate payment acceptance signals from the coin receiving switch (2102) of the offline-payment operated machine (2100), where the payment acceptance signals are indicative of coins being received by the coin receiving switch (2102);
a short-range communication technology configured to communicate with one or more mobile devices;
one or more processors; and
memory (760) storing one or more programs to be executed by the one or more processors, the one or more programs comprising instructions for: performing the method of any one of claims 1-12.

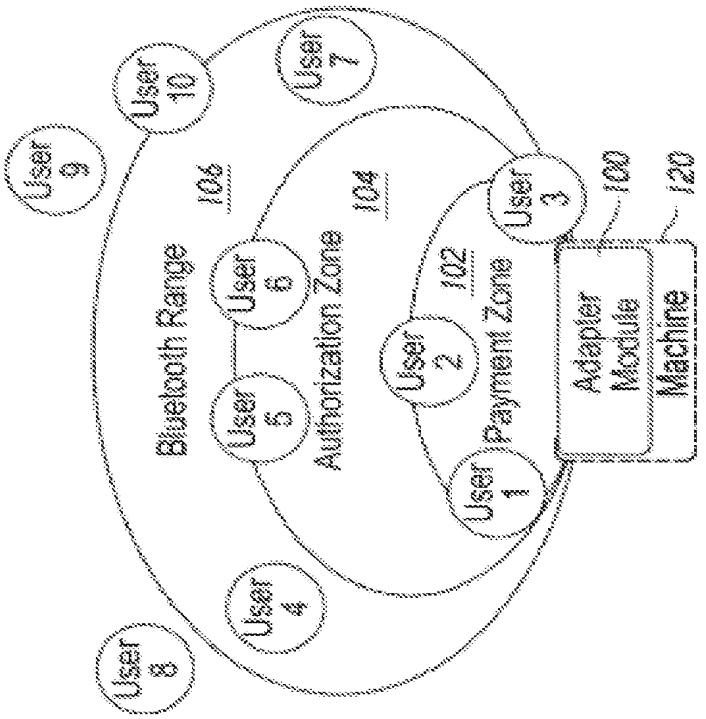


Figure 2

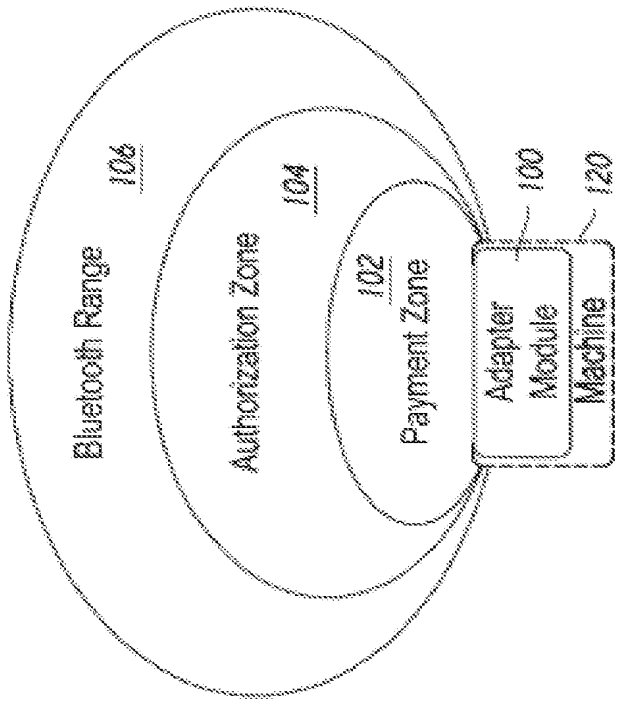


Figure 1

Tab	Favorite?	Alert	View to User
All	Yes	No	User can make Hands-free Credit with the connected vending machine
All	No	Yes	User needs to launch Mobile Device and then swipe to make transaction manually
Favorite	Yes	No	Hands-free transction will be available to the user via vending machine
Favorite	No	No	User is not alerted for the vending machine which is not a favorite machine. Hands-free mode will not work, manual swipe for transaction required by user.
Either All or Favorite	Yes	Yes	BUT Hands-free Credit is not available (disabled by module, expired AuthGrant, insufficient balance, or other issue), then user will get an alert so that user can swipe credit manually.

Figure 3

Figure 4

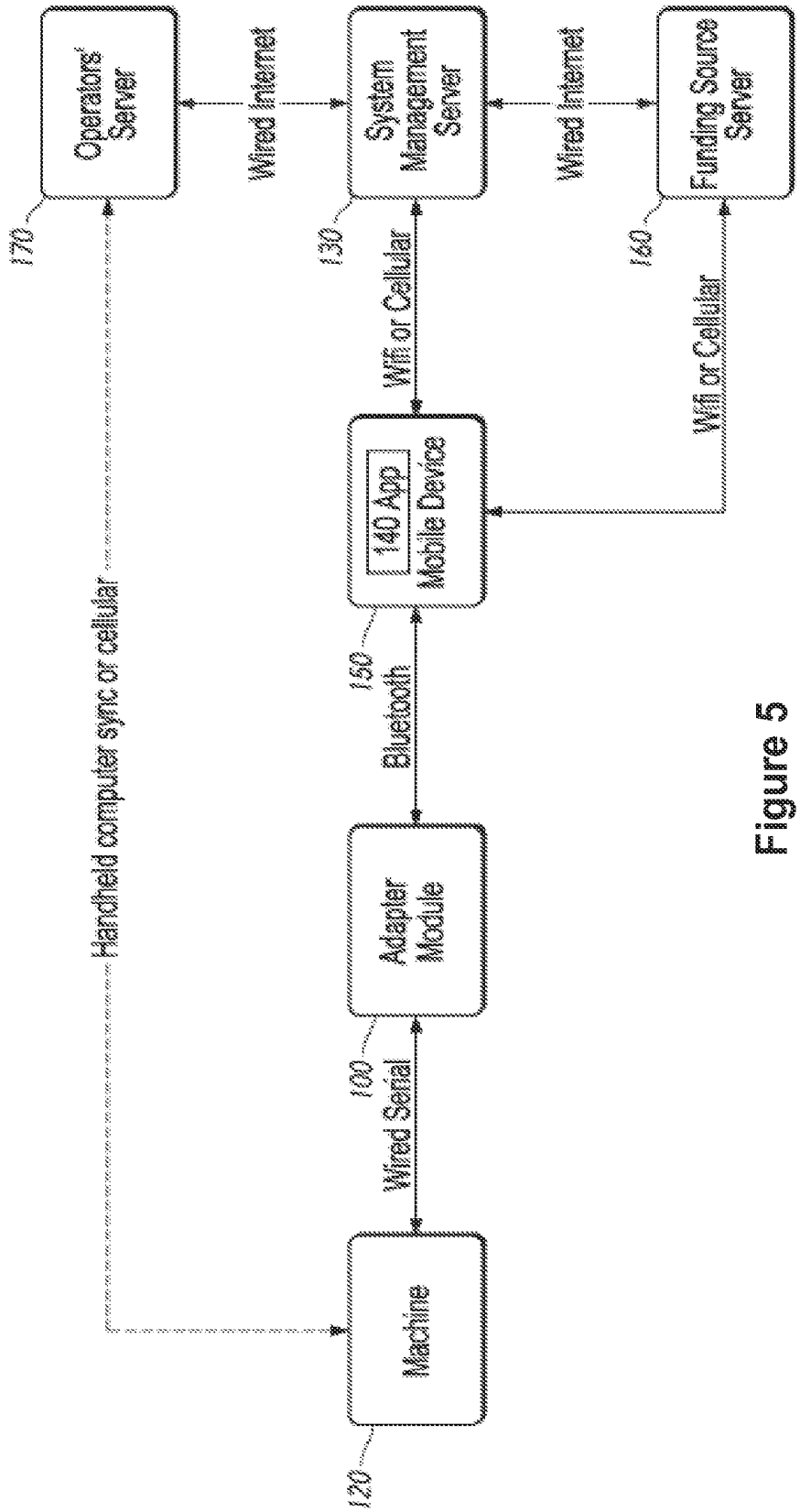

Operators' Server
170
Wired Internet
System Management Server
130
Wired Internet
Funding Source Server
160
Handheld computer sync or cellular
Wifi or Cellular
Wifi or Cellular
140 App
Mobile Device
150
Bluetooth
Adapter Module
100
Wired Serial
Machine
120

Figure 5

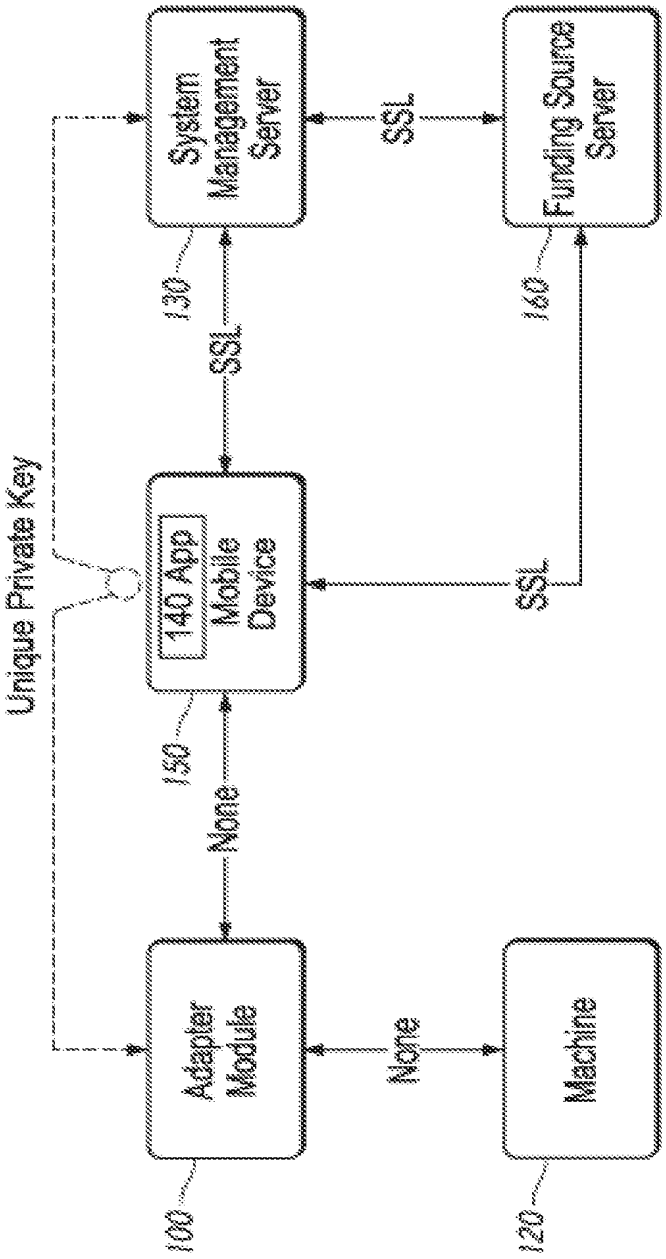

System Management Server
Funding Source Server
SSL
SSL
130
160
Unique Private Key
140 App
Mobile Device
SSL
150
None
Adapter Module
None
Machine
100
120

Figure 6

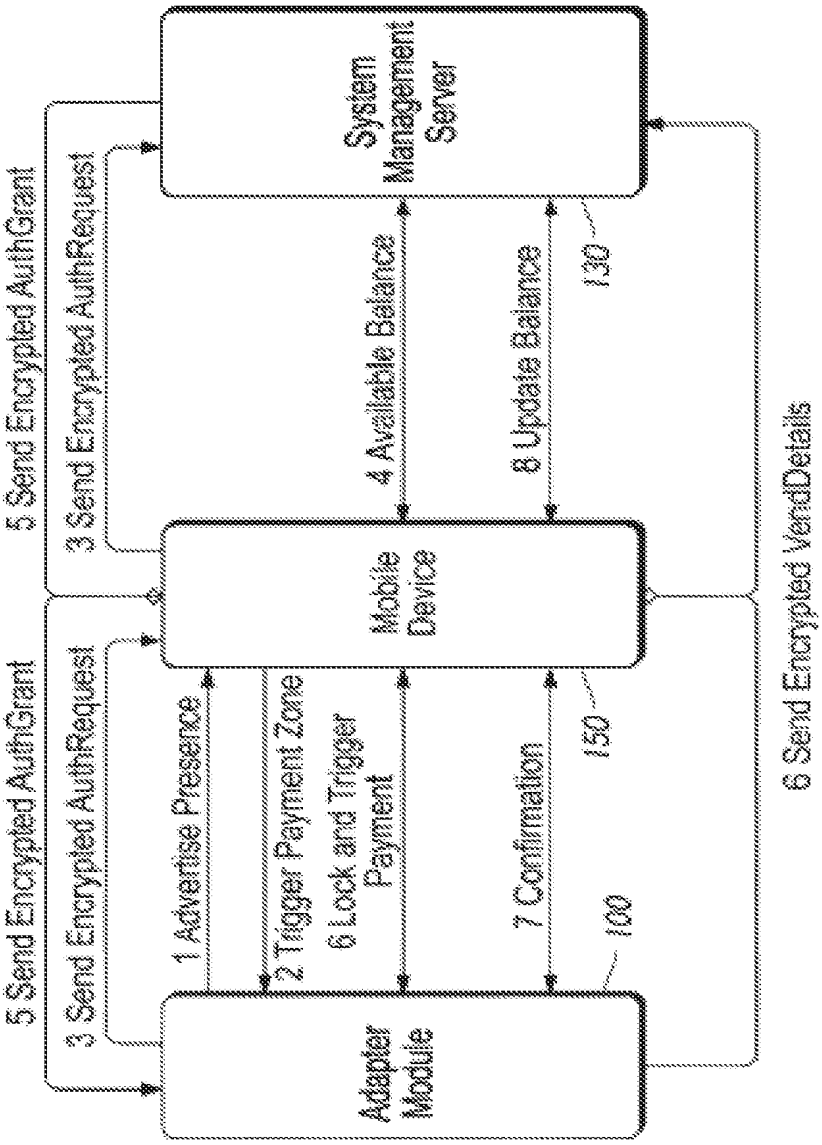

System Management Server
Mobile Device
Adapter Module
5 Send Encrypted AuthGrant
3 Send Encrypted AuthRequest
5 Send Encrypted AuthGrant
3 Send Encrypted AuthRequest
1 Advertise Presence
2 Trigger Payment Zone
6 Lock and Trigger Payment
7 Confirmation
4 Available Balance
8 Update Balance
6 Send Encrypted VendDetails
100
150
130

Figure 7

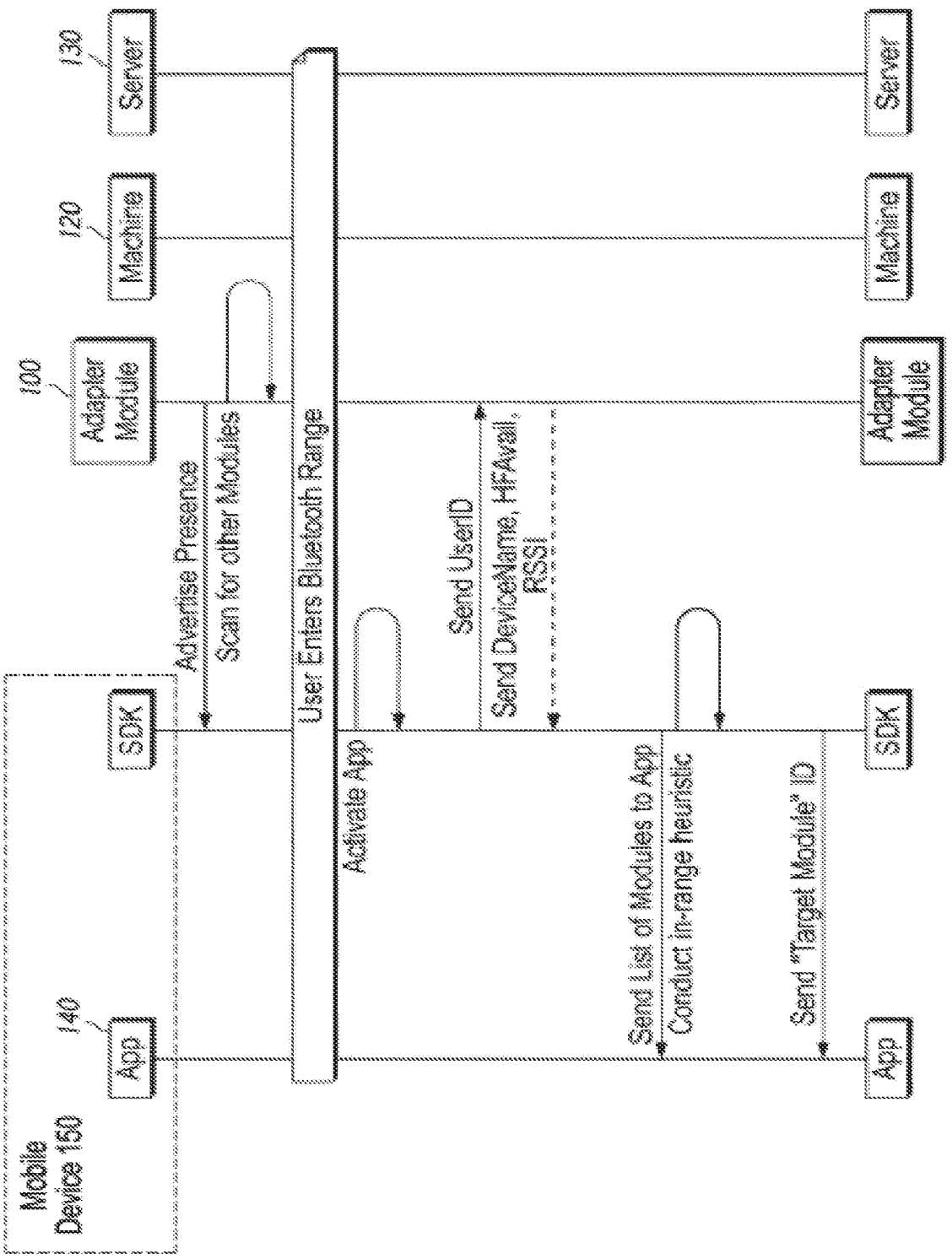

Mobile Device 150
140
App
SDK
100
Adapter Module
120
Machine
130
Server
Advertise Presence
Scan for other Modules
User Enters Bluetooth Range
Activate App
Send UserID
Send DeviceName, HFAvail, RSSI
Send List of Modules to App
Conduct in-range heuristic
Send "Target Module" ID
SDK
App
Adapter Module
Machine
Server

Figure 8A

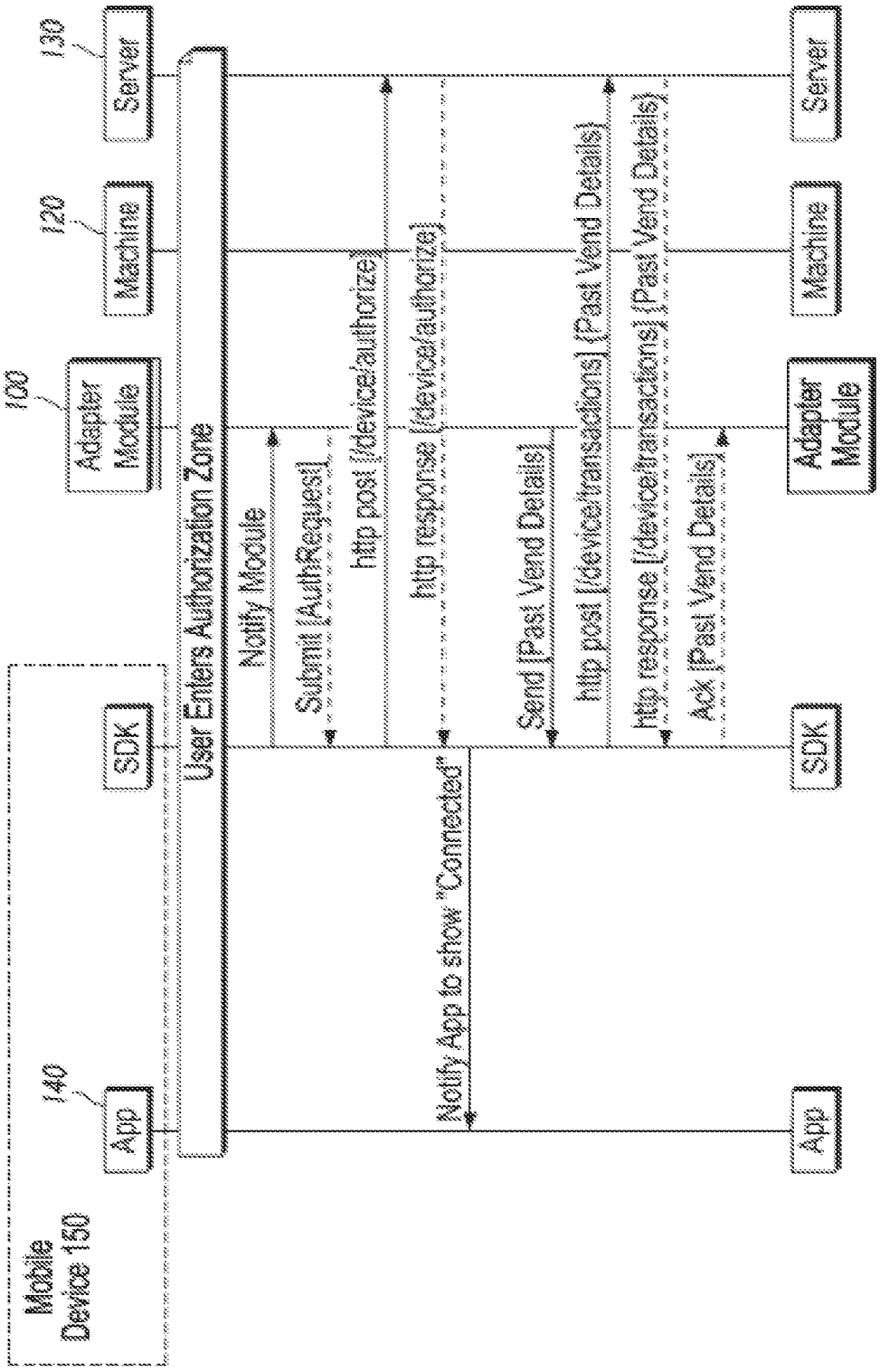

Mobile Device 150
140
App
SDK
100
Adapter Module
120
Machine
130
Server
User Enters Authorization Zone
Notify Module
Submit [AuthRequest]
http post [/device/authorize]
http response [/device/authorize]
Notify App to show "Connected"
Send [Past Vend Details]
http post [/devicetransactions] [Past Vend Details]
http response [/devicetransactions] [Past Vend Details]
Ack [Past Vend Details]
App
SDK
Adapter Module
Machine
Server

Figure 8B

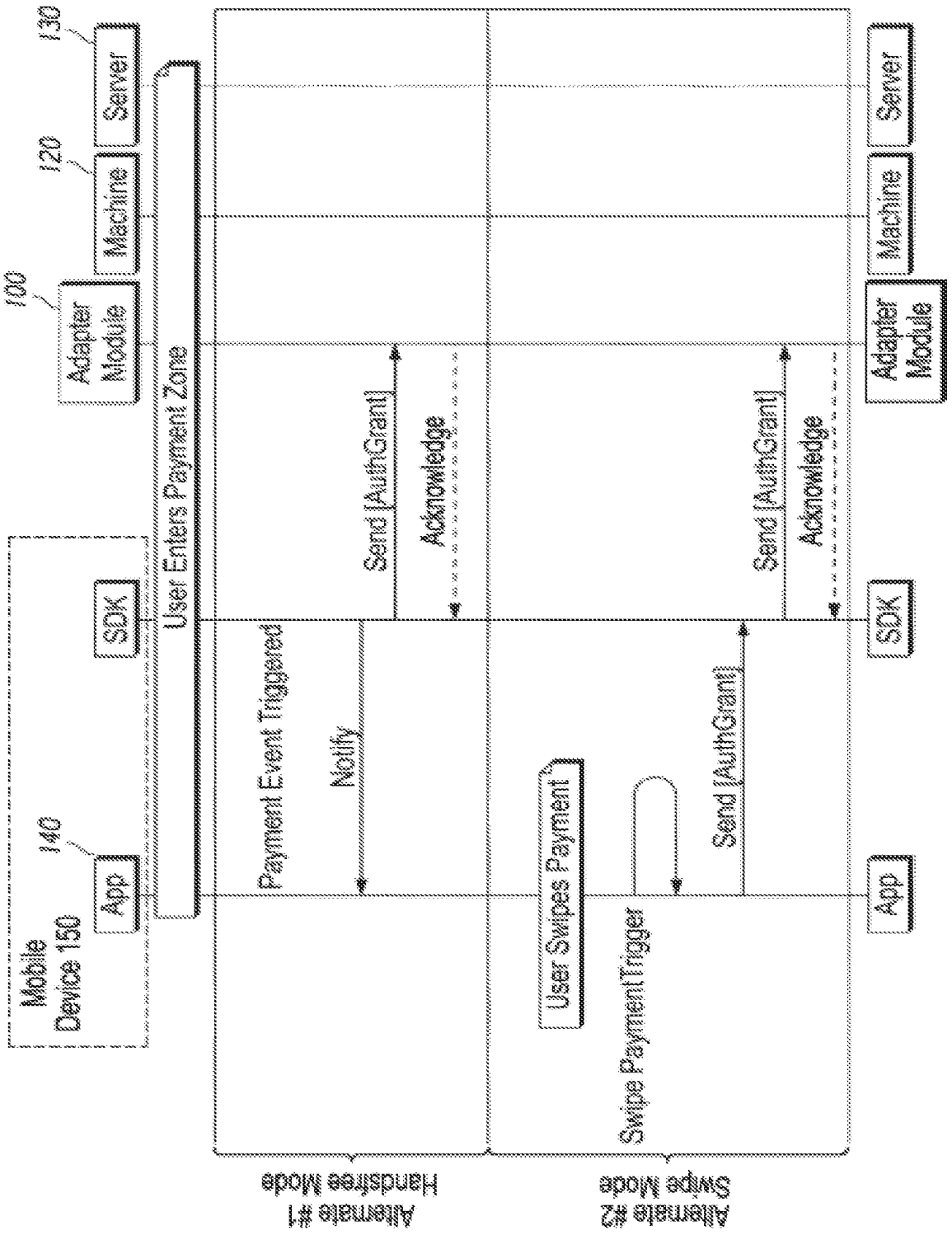

Mobile Device 150
140
App
SDK
100
Adapter Module
120
Machine
130
Server
User Enters Payment Zone
Payment Event Triggered
Notify
Send [AuthGrant]
Acknowledge
User Swipes Payment
Swipe Payment Trigger
Send [AuthGrant]
Send [AuthGrant]
Acknowledge
Alternate #1 Handsfree Mode
Alternate #2 Swipe Mode

Figure 8C

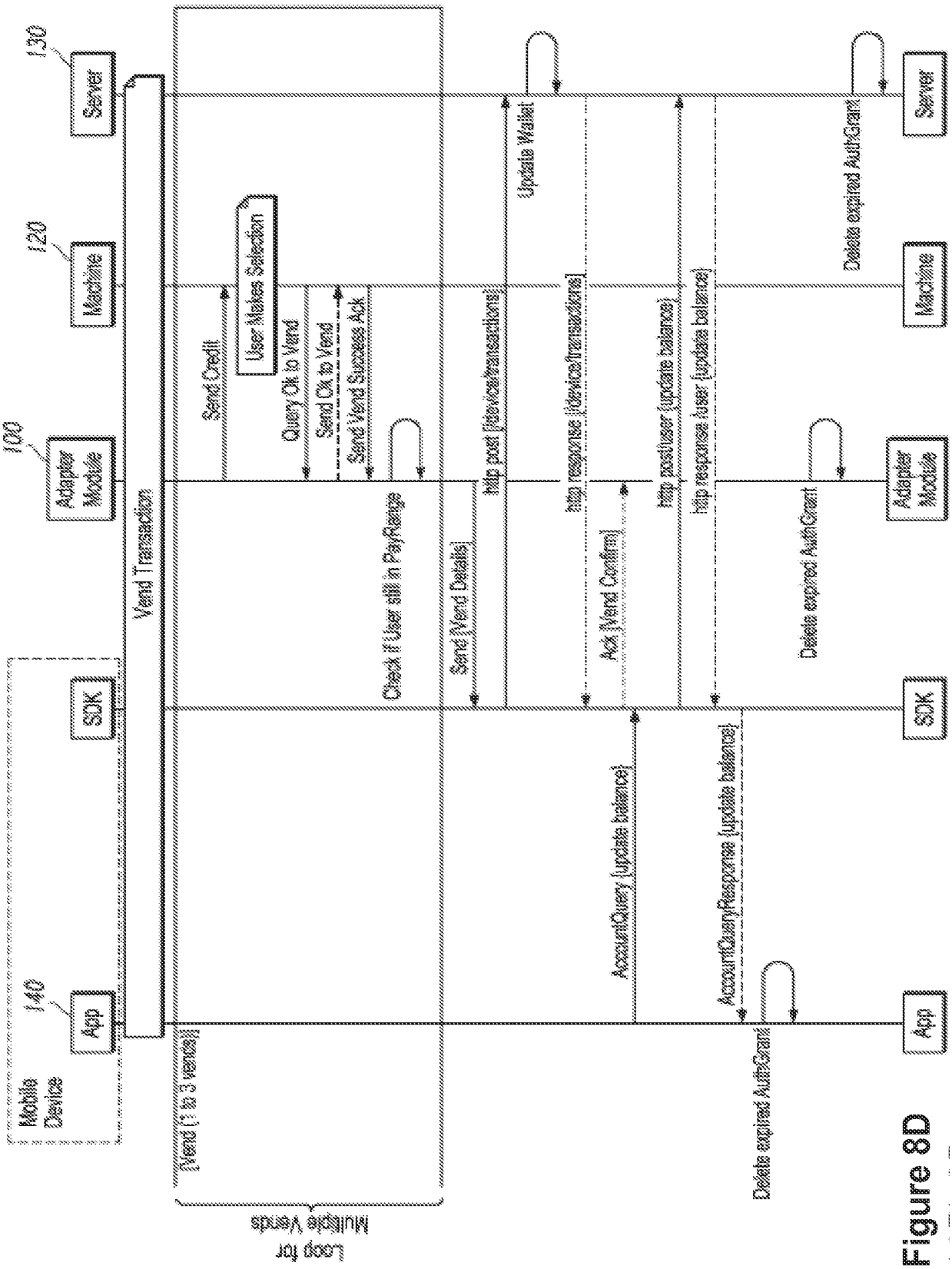

Mobile Device
140
App
SDK
100
Adapter Module
120
Machine
130
Server
Vend Transaction
Send Credit
User Makes Selection
Query Ok to Vend
Send Ok to Vend
Send Vend Success Ack
Check if User still in PayRange
Send [Vend Details]
http post [/device/transactions]
Update Wallet
http response [/device/transactions]
Ack [Vend Confirm]
http positive response [update balance]
http response /user [update balance]
Delete expired AuthGrant
Delete expired AuthGrant
AccountQuery [update balance]
AccountQueryResponse [update balance]
Delete expired AuthGrant
[Vend (1 to 3 vends)]
Loop for Multiple Vends

Figure 8D

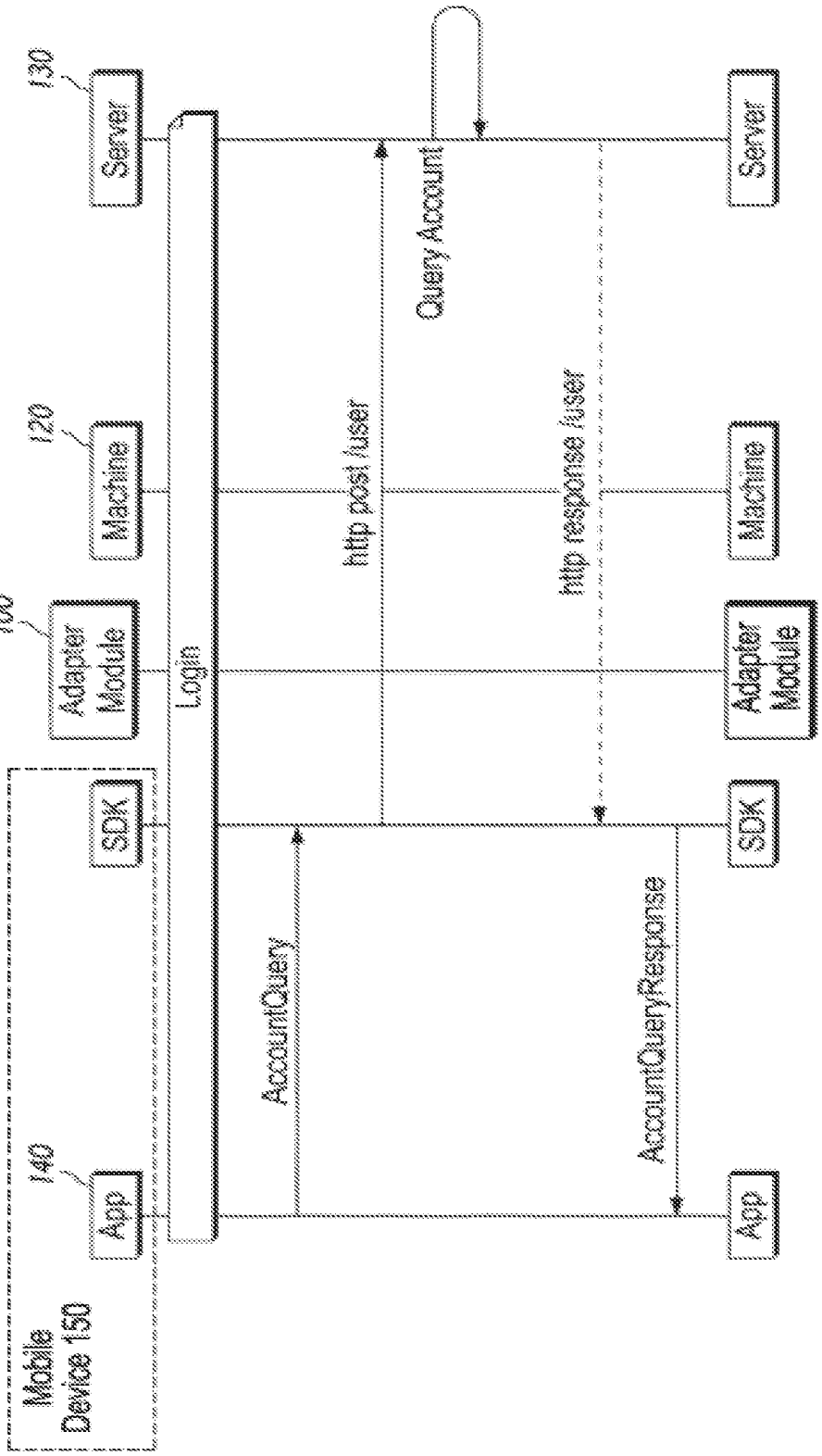

Mobile Device 150
140
App
SDK
100
Adapter Module
120
Machine
130
Server
Login
AccountQuery
http post /user
Query Account
http response /user
AccountQueryResponse
SDK
Adapter Module
Machine
Server
App

Figure 8E

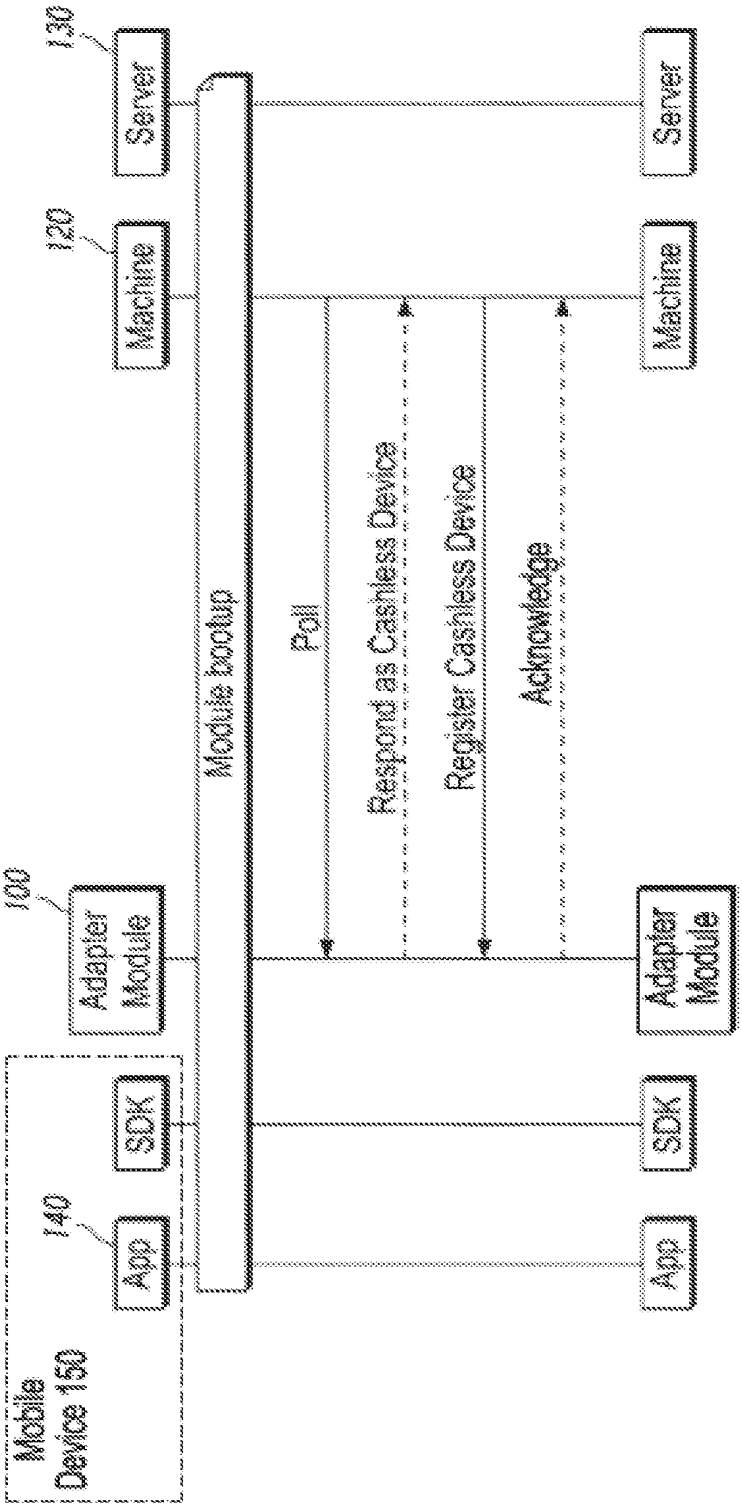

Mobile Device 150
App
140
SDK
Adapter Module
100
Machine
120
Server
130
Module bootup
Poll
Respond as Cashless Device
Register Cashless Device
Acknowledge
App
SDK
Adapter Module
Machine
Server

Figure 8F

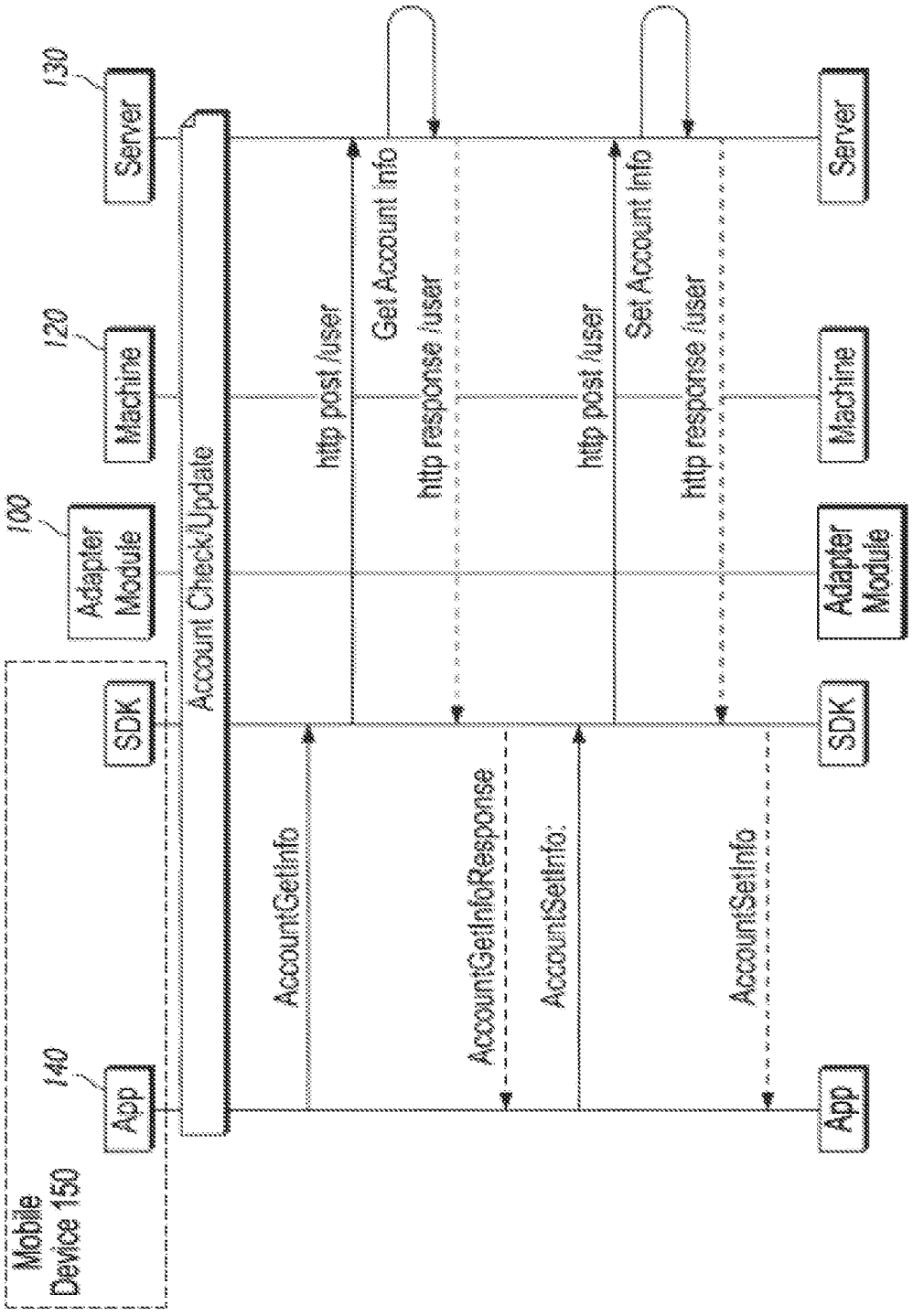

Mobile Device 150
140
App
SDK
100
Adapter Module
120
Machine
130
Server
Account Check/Update
AccountGetInfo
http post /user
Get Account Info
http response /user
AccountGetInfoResponse
AccountSetInfo
http post /user
Set Account Info
http response /user
AccountSetInfo
SDK
Adapter Module
Machine
Server
App

Figure 8G

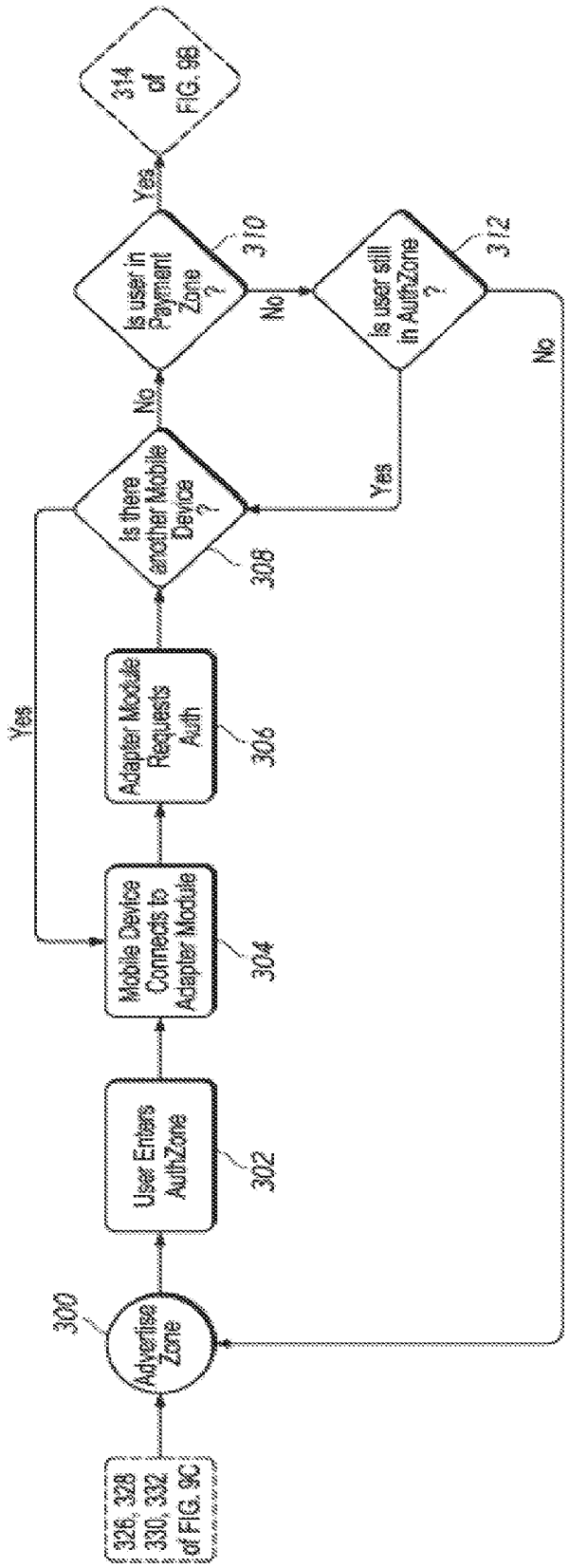

326, 328
330, 332
of FIG. 9C
300
Advertise Zone
User Enters AuthZone
302
Mobile Device Connects to Adapter Module
304
Adapter Module Requests Auth
306
Is there another Mobile Device ?
308
Yes
No
Is user in Payment Zone ?
310
Yes
No
Is user still in AuthZone ?
312
Yes
No
314 of FIG. 9B

Figure 9A

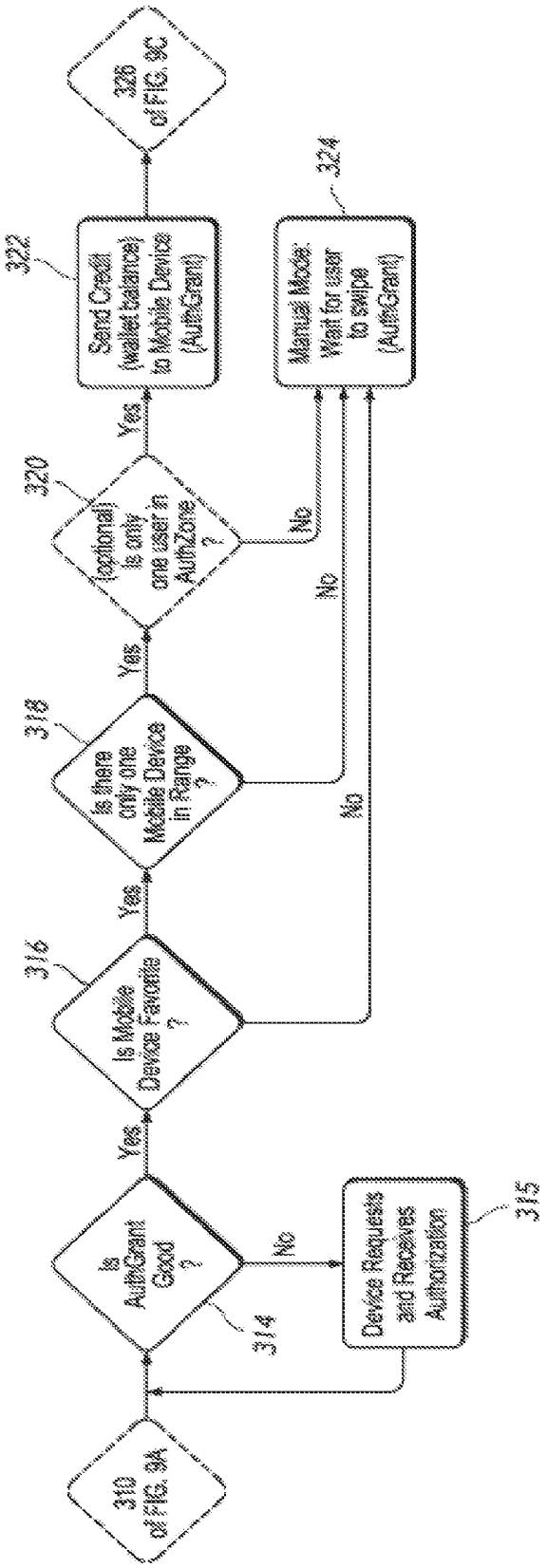
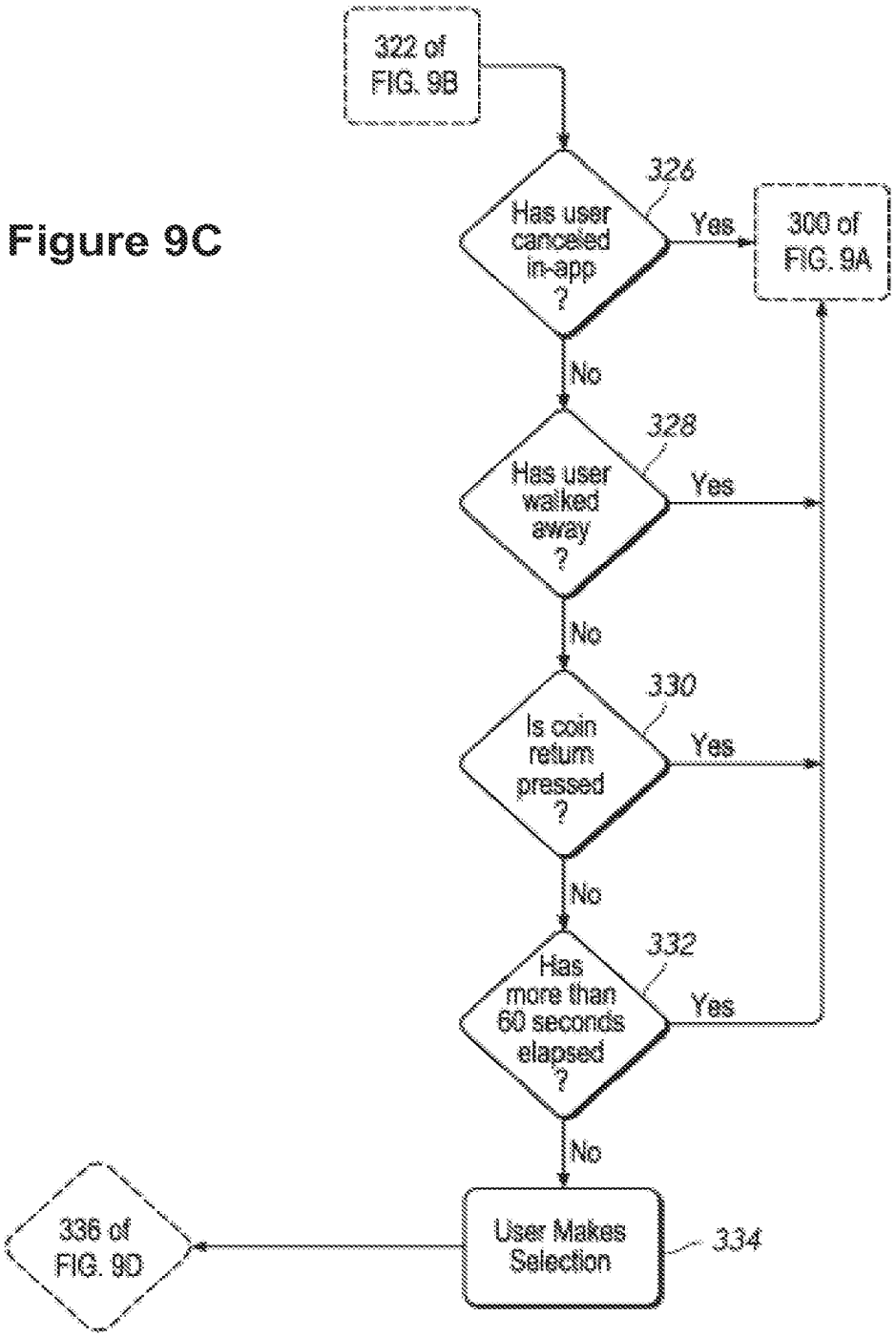

310 of FIG. 9A
Is AuthGrant Good ?
314
No
Device Requests and Receives Authorization
315
Yes
316
Is Mobile Device Favorite ?
No
Yes
318
Is there only one Mobile Device in Range ?
No
Yes
320
(optional) Is only one user in AuthZone ?
No
Yes
322
Send Credit (wallet balance) to Mobile Device (AuthGrant)
324
Manual Mode: Wait for user to swipe (AuthGrant)
326 of FIG. 9C

Figure 9B

Figure 9C


322 of FIG. 9B
326
Has user canceled in-app ?
Yes
300 of FIG. 9A
No
328
Has user walked away ?
Yes
No
330
Is coin return pressed ?
Yes
No
332
Has more than 60 seconds elapsed ?
Yes
No
User Makes Selection
334
336 of FIG. 9D

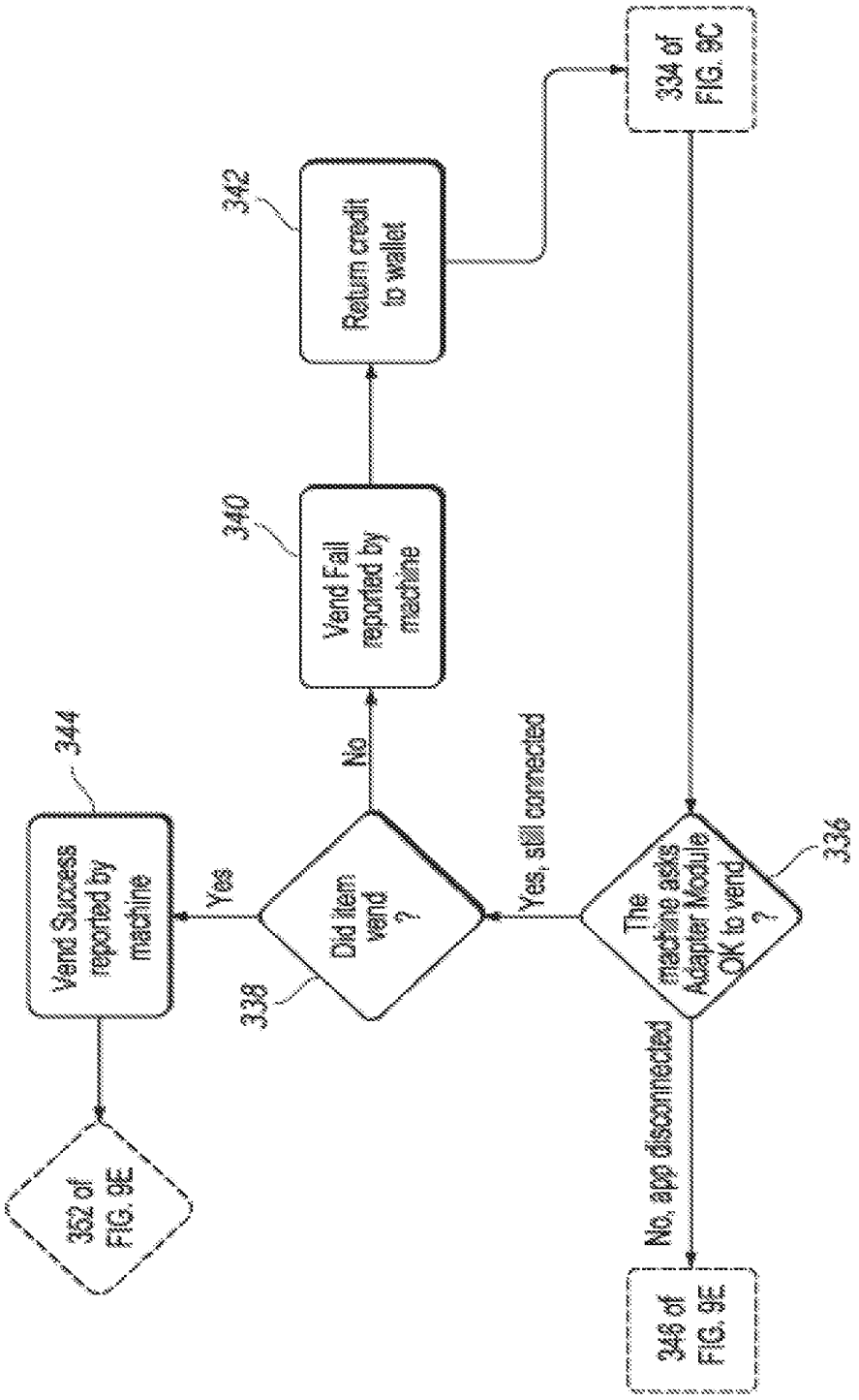

352 of FIG. 9E
Vend Success reported by machine
344
Yes
Did item vend ?
338
No
Vend Fail reported by machine
340
Return credit to wallet
342
334 of FIG. 9C
Yes, still connected
The machine asks Adapter Module OK to vend ?
336
No, app disconnected
346 of FIG. 9E

Figure 9D

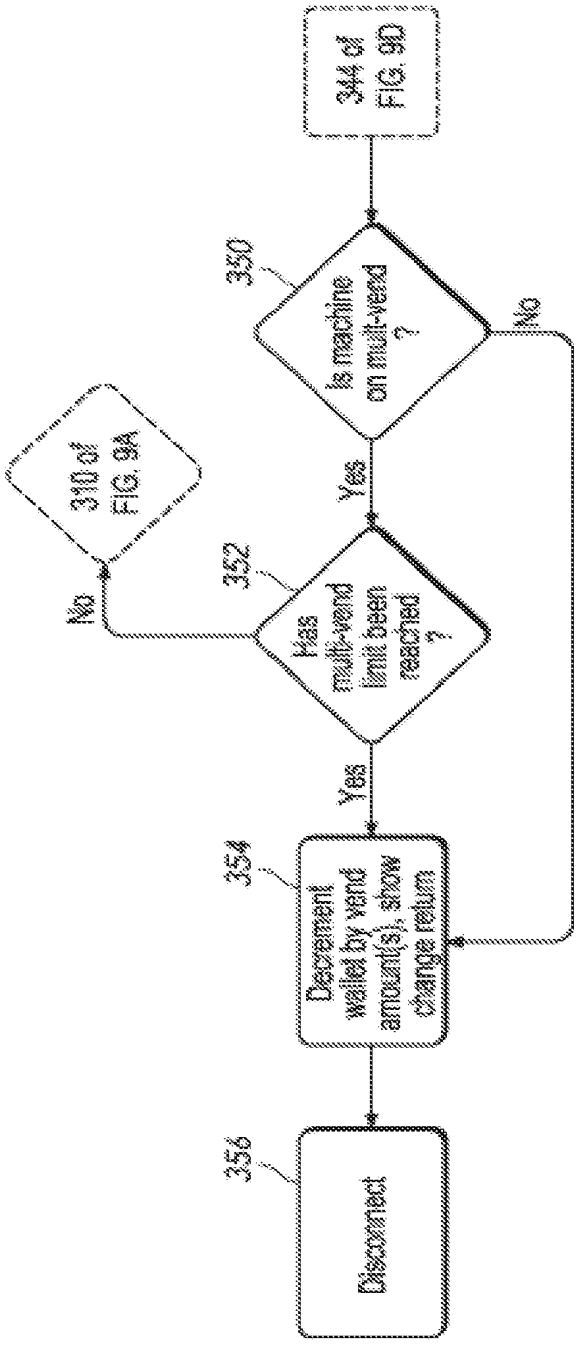

344 of FIG. 9D
350
Is machine on multi-vend ?
No
Yes
310 of FIG. 9A
No
352
Has multi-vend limit been reached ?
Yes
354
Decrement valled by vend amount(s), show change return
356
Disconnect

Figure 9E

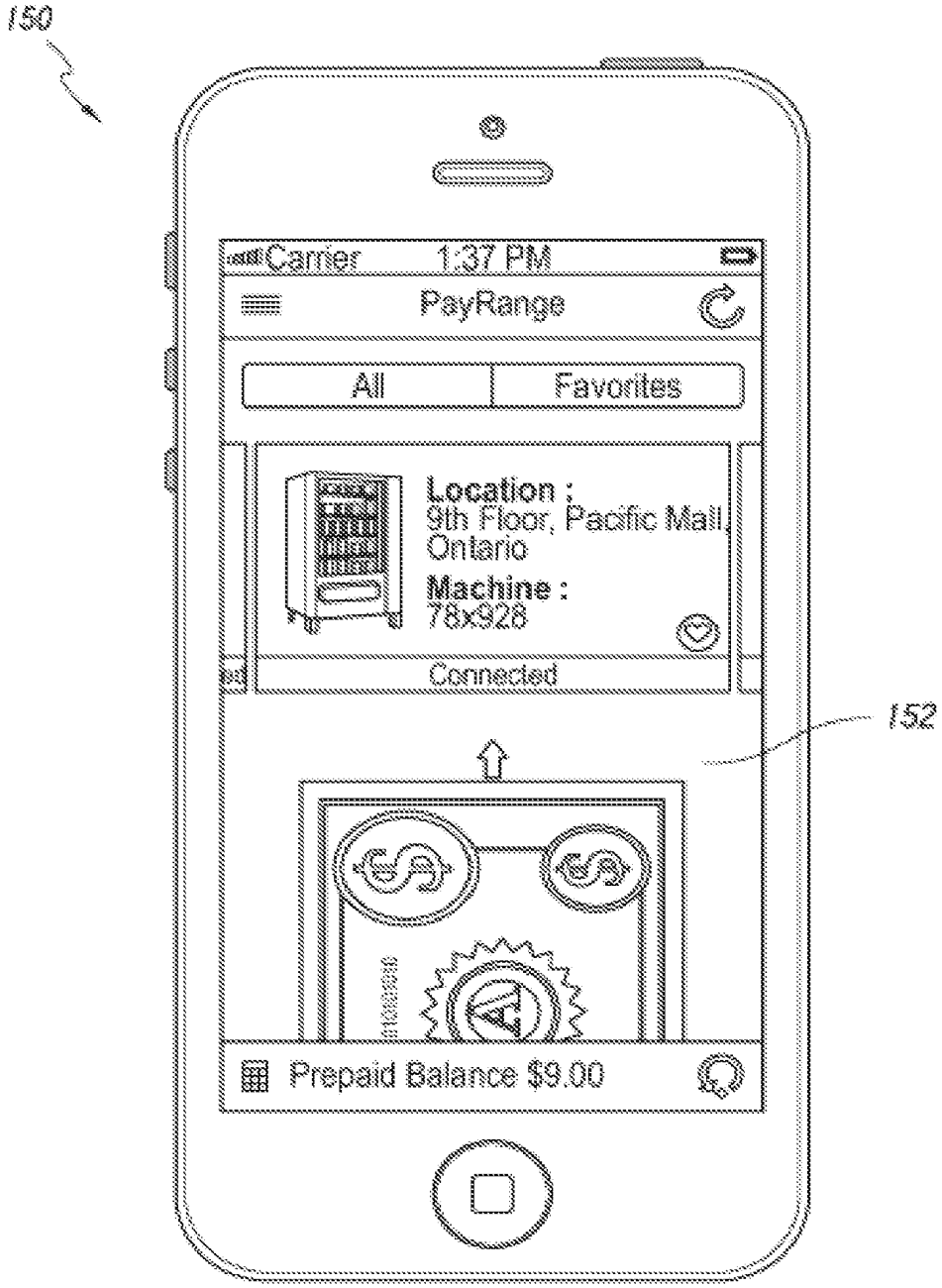

150
Carrier
1:37 PM
PayRange
All
Favorites
Location :
9th Floor, Pacific Mall,
Ontario
Machine :
78x928
Connected
152
Prepaid Balance $9.00

Figure 10A

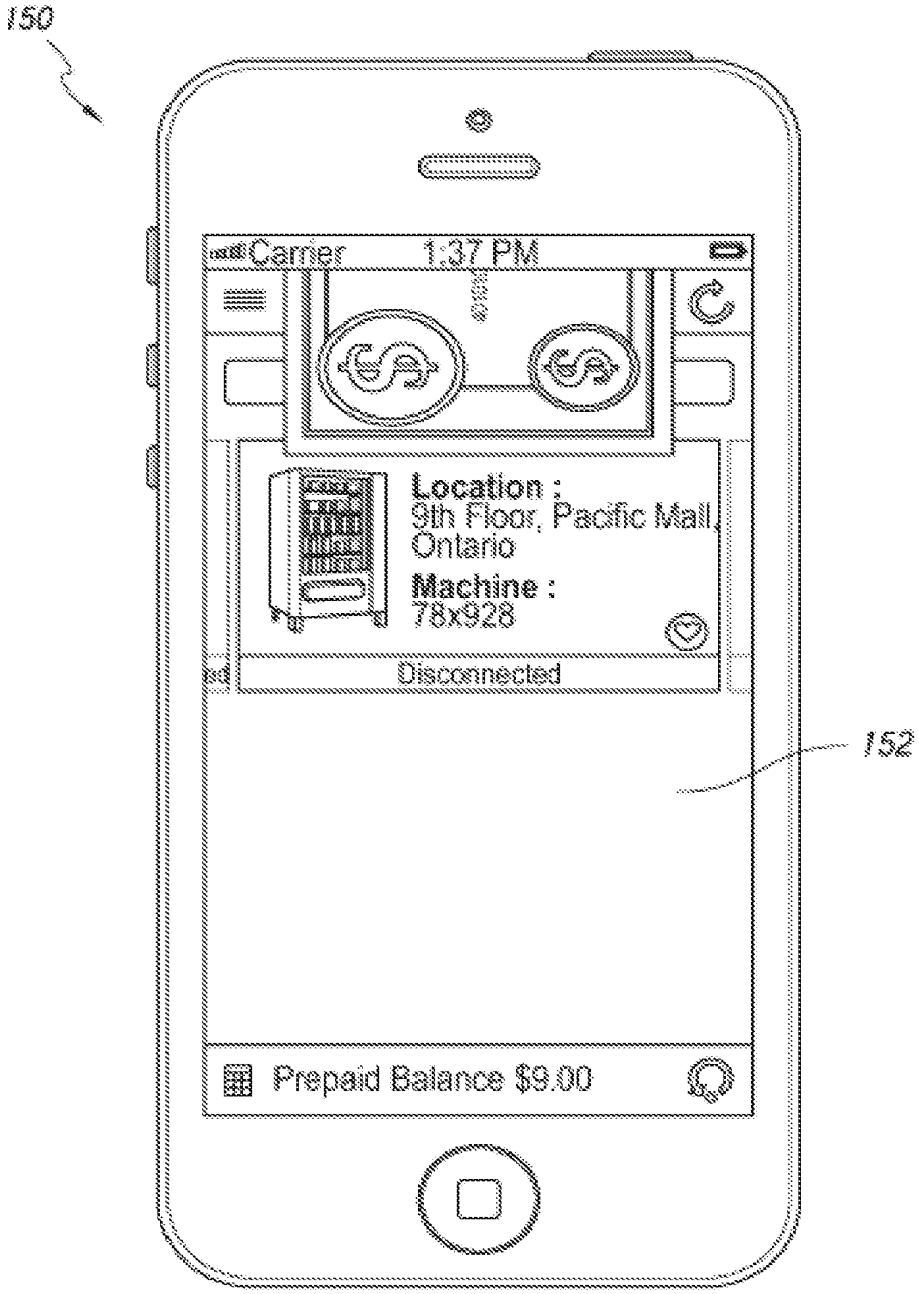

150
Carrier
1:37 PM
Location :
9th Floor, Pacific Mall,
Ontario
Machine :
78x928
Disconnected
152
Prepaid Balance $9.00

Figure 10B

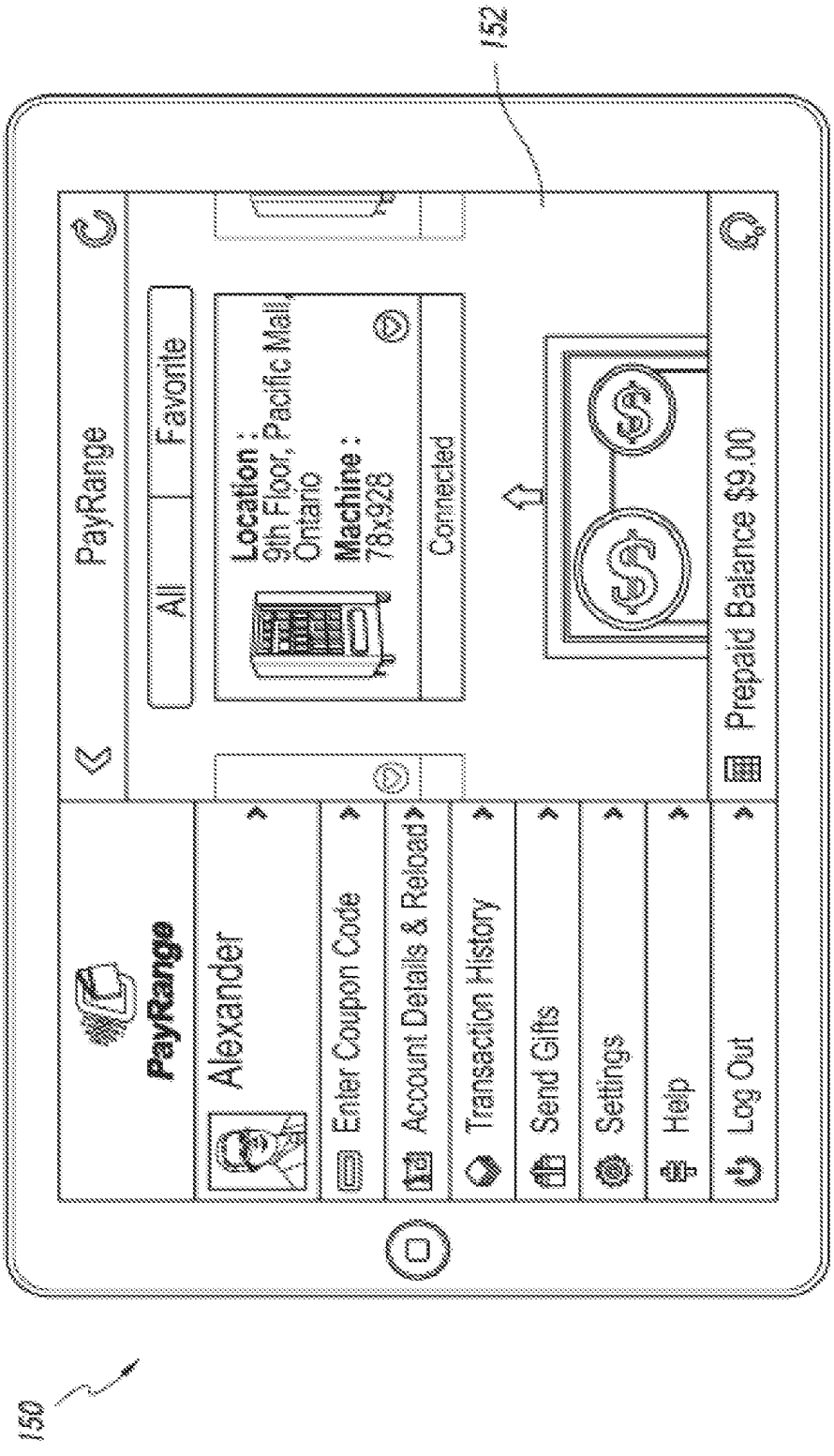

150
152
PayRange
Favorite
All
Location :
9th Floor, Pacific Mall
Ontario
Machine :
78x928
Connected
Prepaid Balance $9.00
PayRange
Alexander
Enter Coupon Code
Account Details & Reload
Transaction History
Send Gifts
Settings
Help
Log Out

Figure 10C

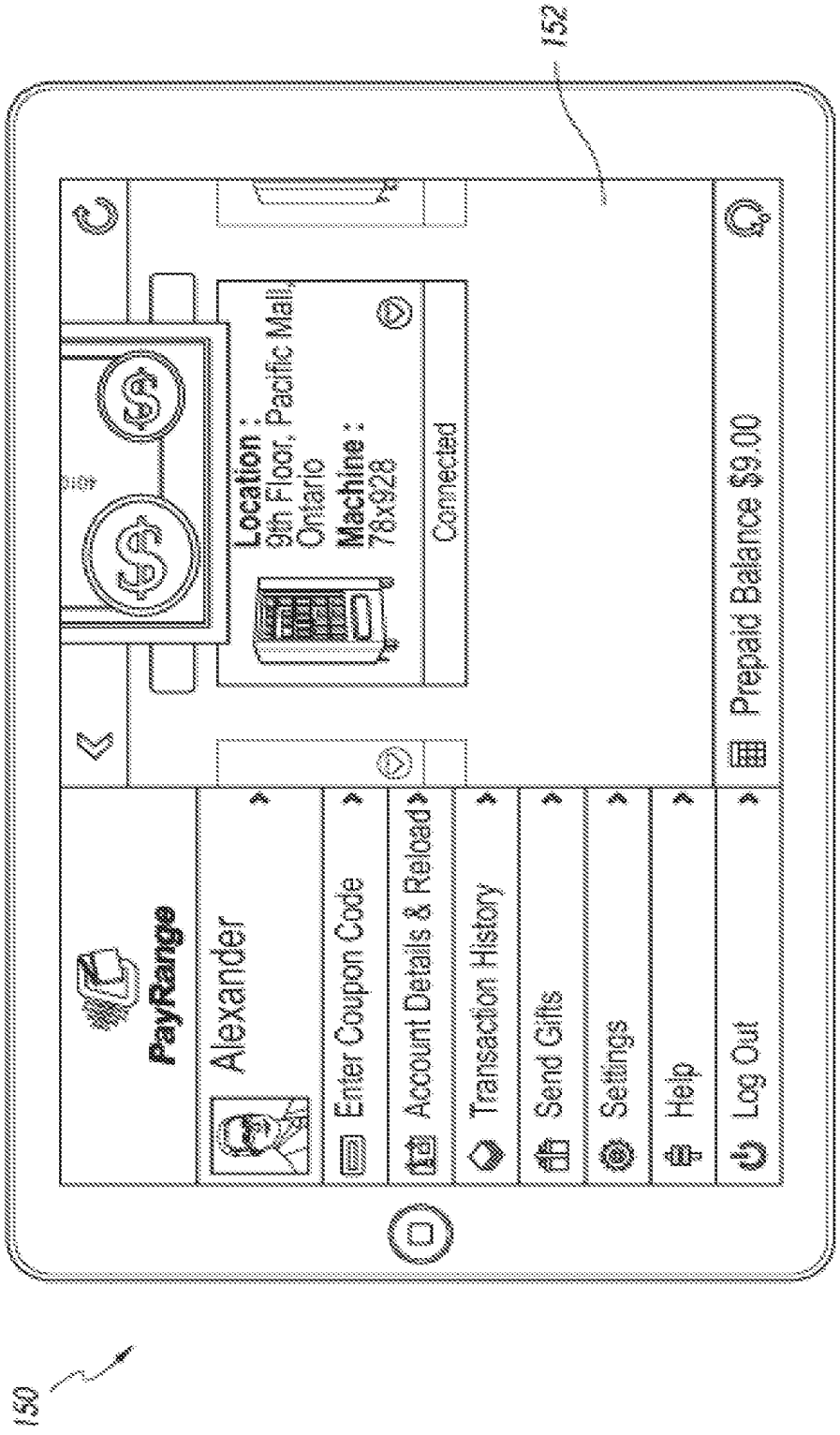

150
152
PayRange
Alexander
Enter Coupon Code
Account Details & Reload
Transaction History
Send Gifts
Settings
Help
Log Out
Location :
9th Floor, Pacific Mall,
Ontario
Machine :
78x928
Connected
Prepaid Balance $9.00

Figure 10D

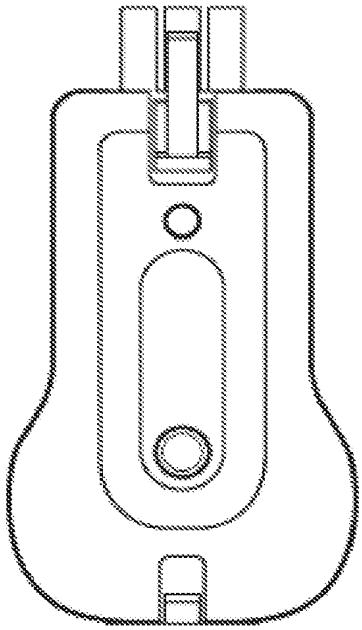


Figure 12

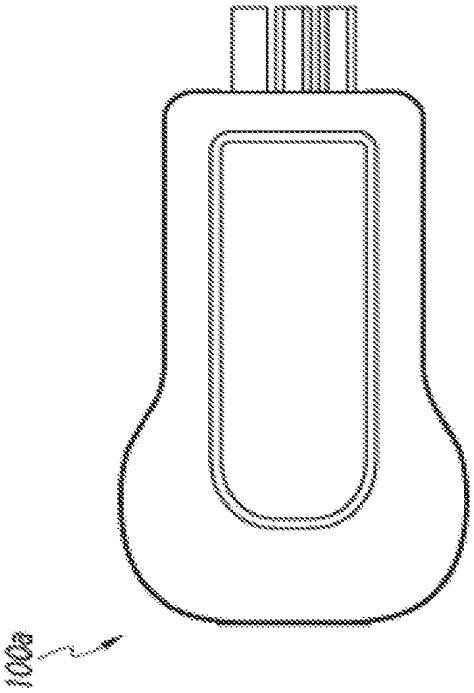


Figure 13

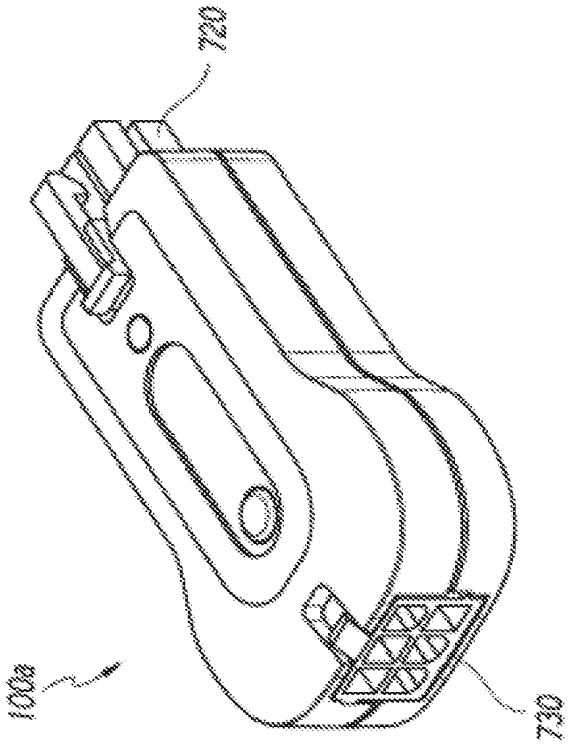


Figure 11

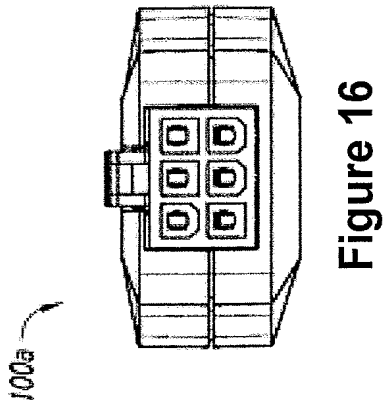


Figure 16

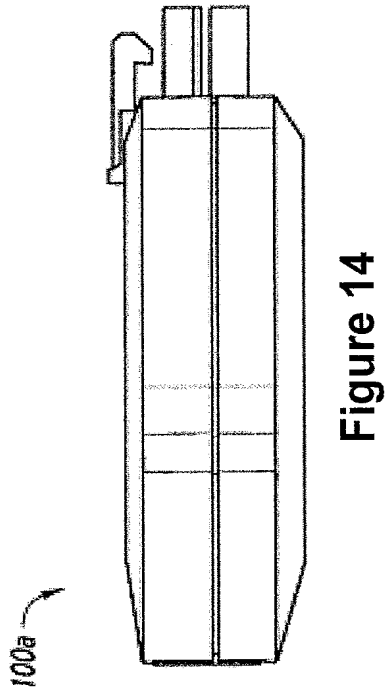


Figure 14

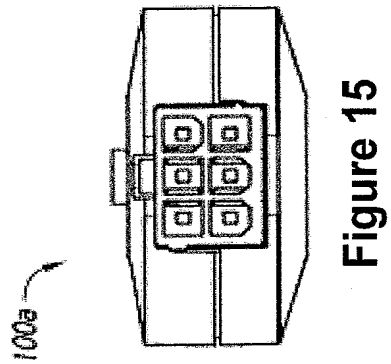


Figure 15

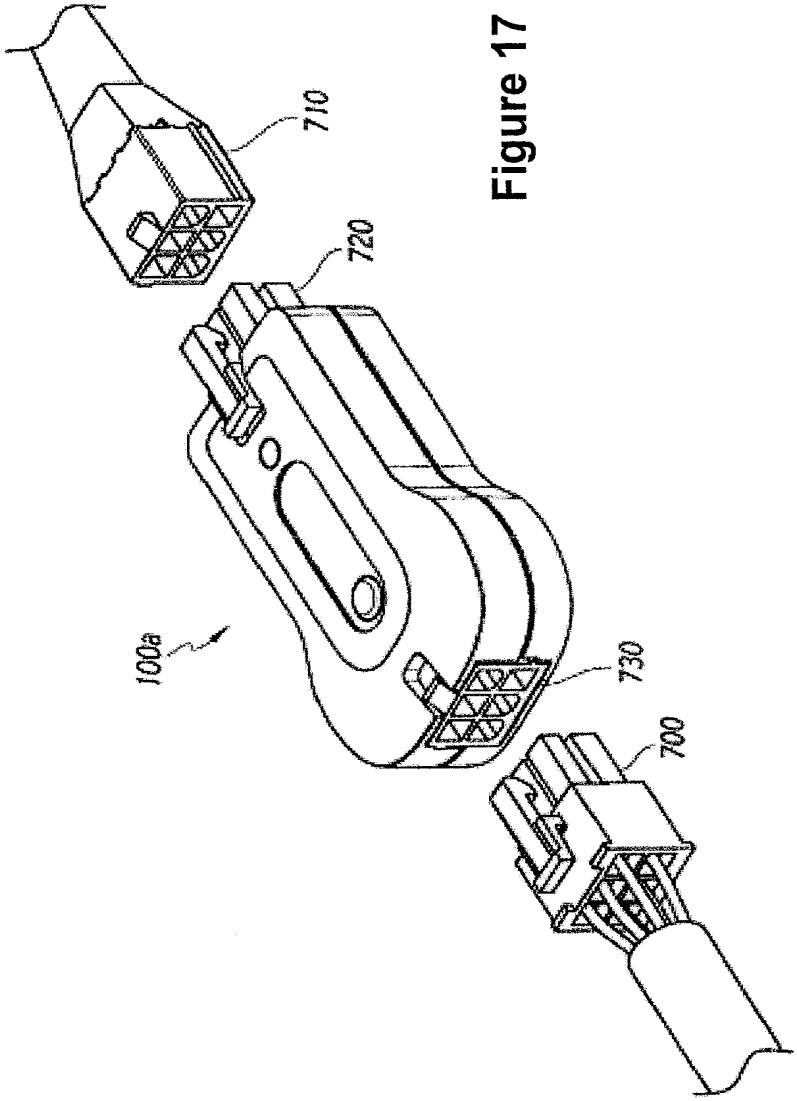


Figure 17

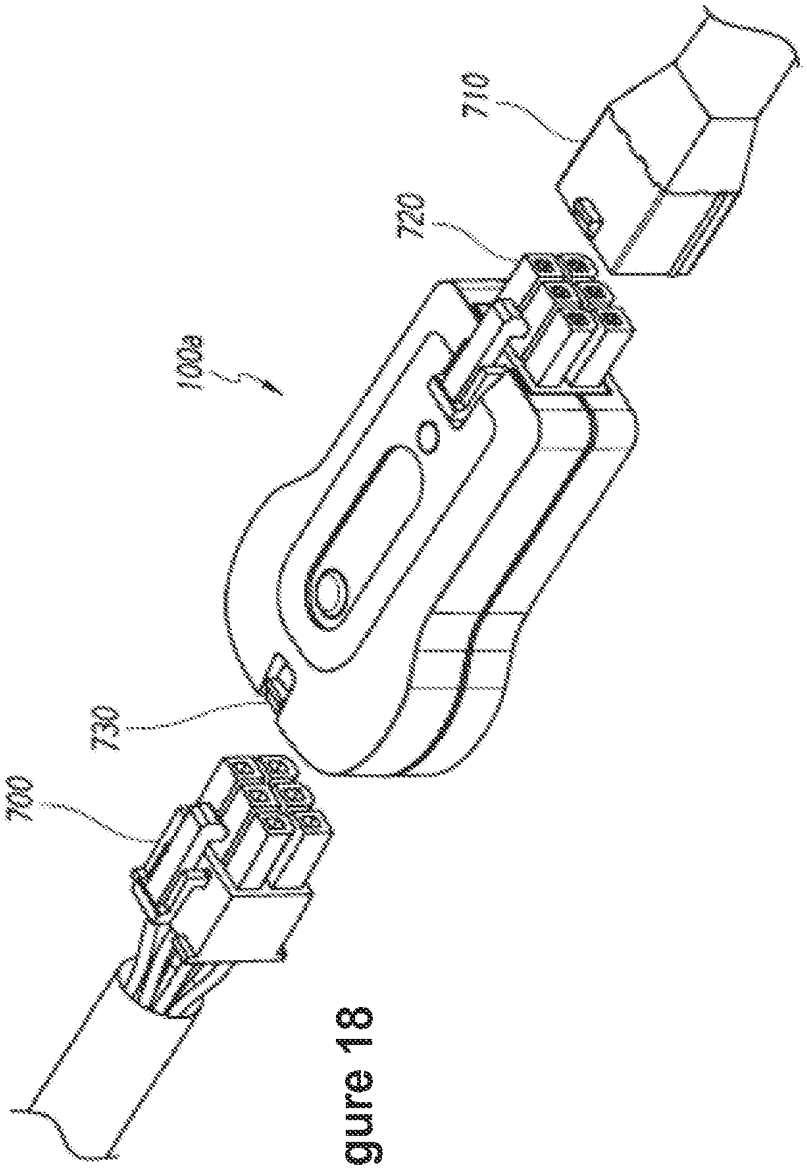


Figure 18

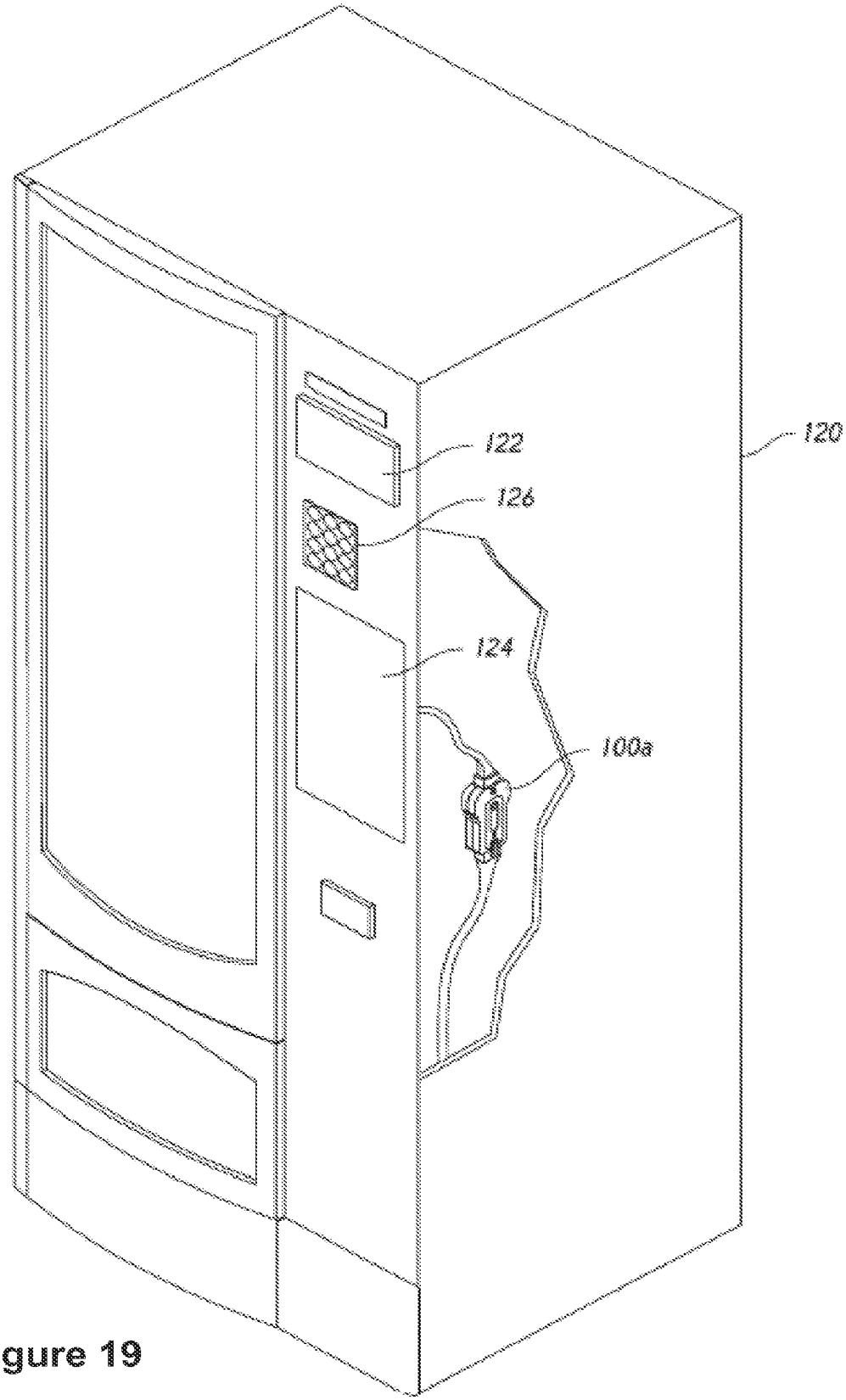


Figure 19

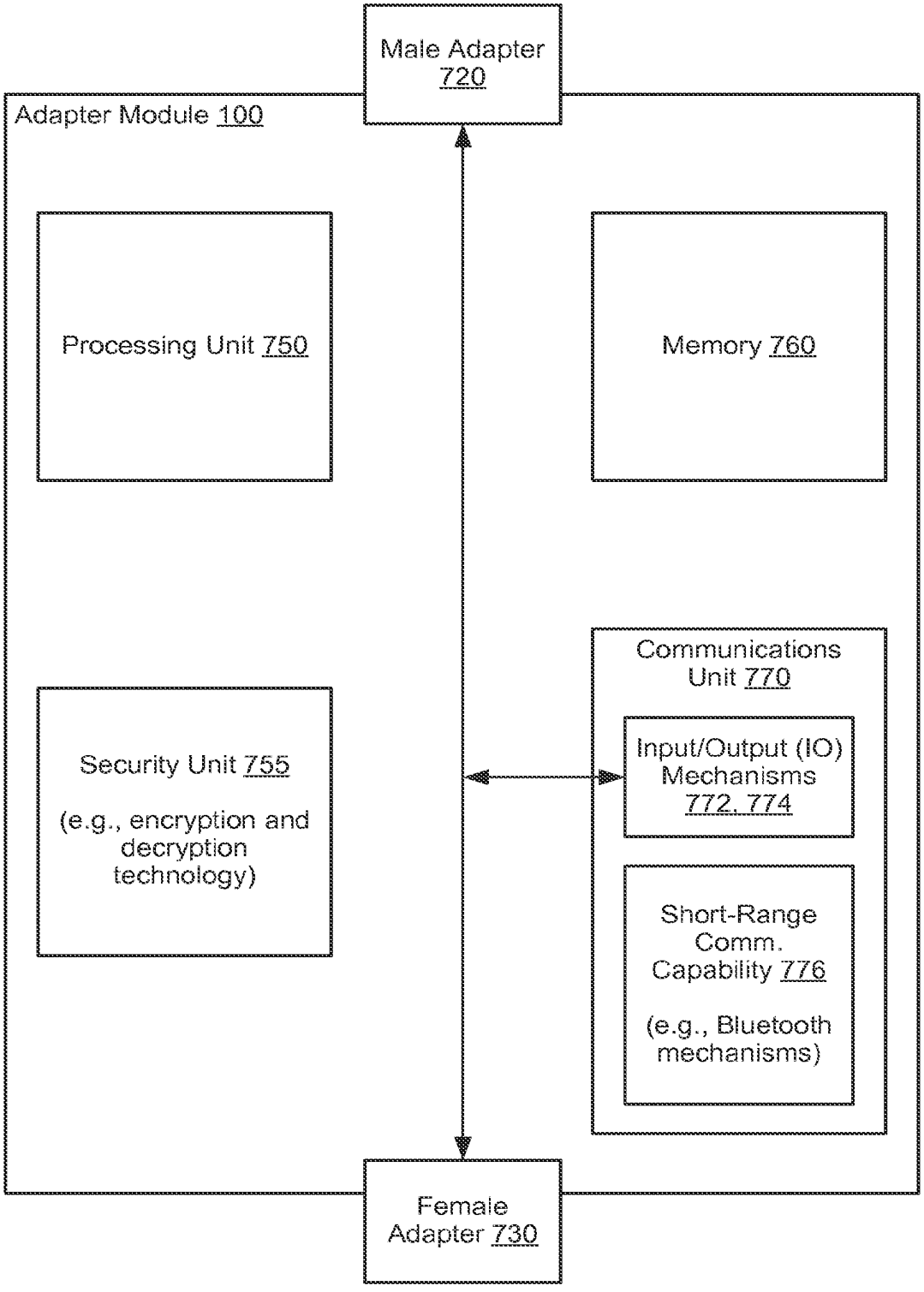

Male Adapter 720
Adapter Module 100
Processing Unit 750
Memory 760
Communications Unit 770
Input/Output (IO) Mechanisms 772, 774
Security Unit 755
(e.g., encryption and decryption technology)
Short-Range Comm. Capability 776
(e.g., Bluetooth mechanisms)
Female Adapter 730

Figure 20

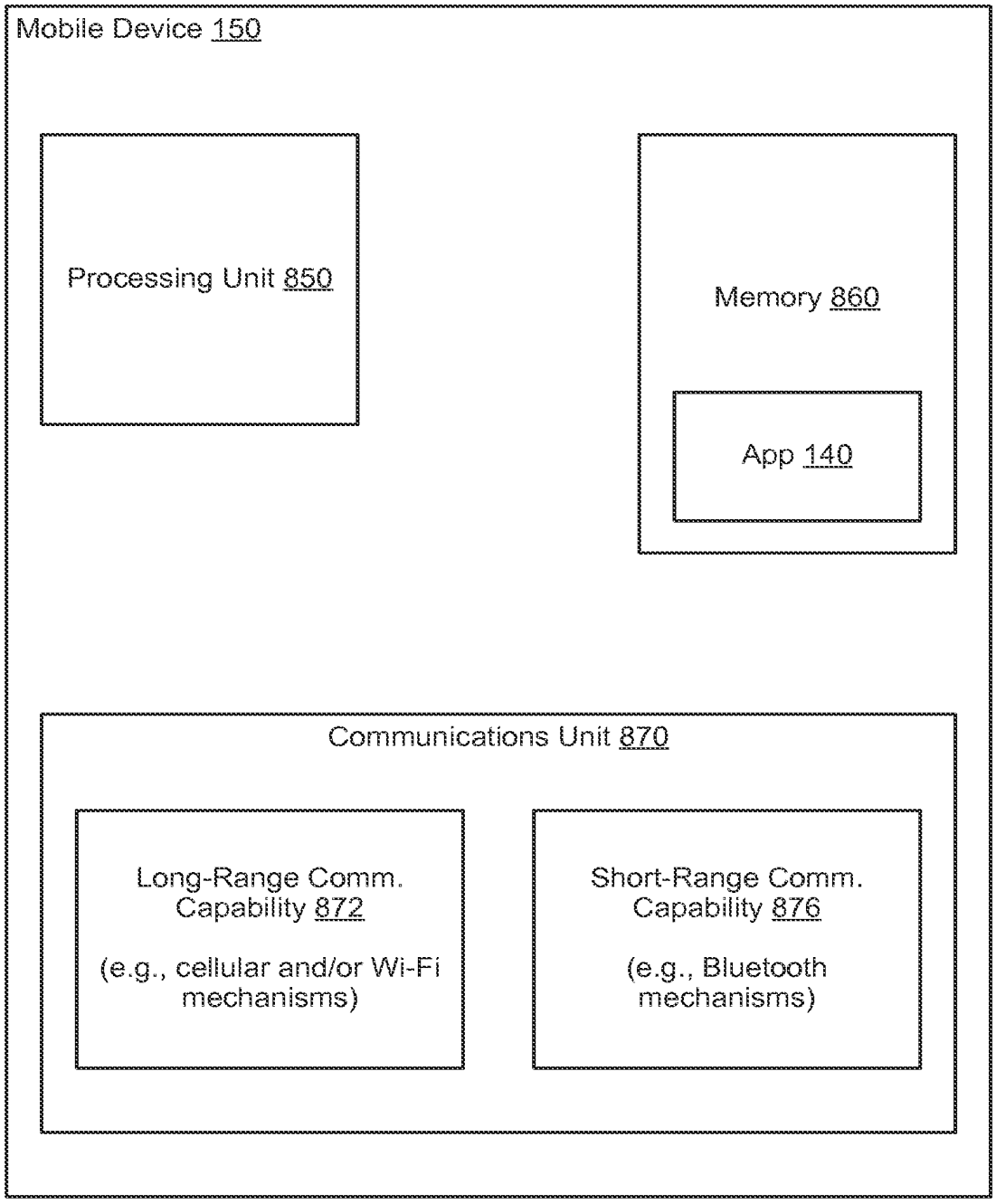

Mobile Device 150
Processing Unit 850
Memory 860
App 140
Communications Unit 870
Long-Range Comm. Capability 872
(e.g., cellular and/or Wi-Fi mechanisms)
Short-Range Comm. Capability 876
(e.g., Bluetooth mechanisms)

Figure 21

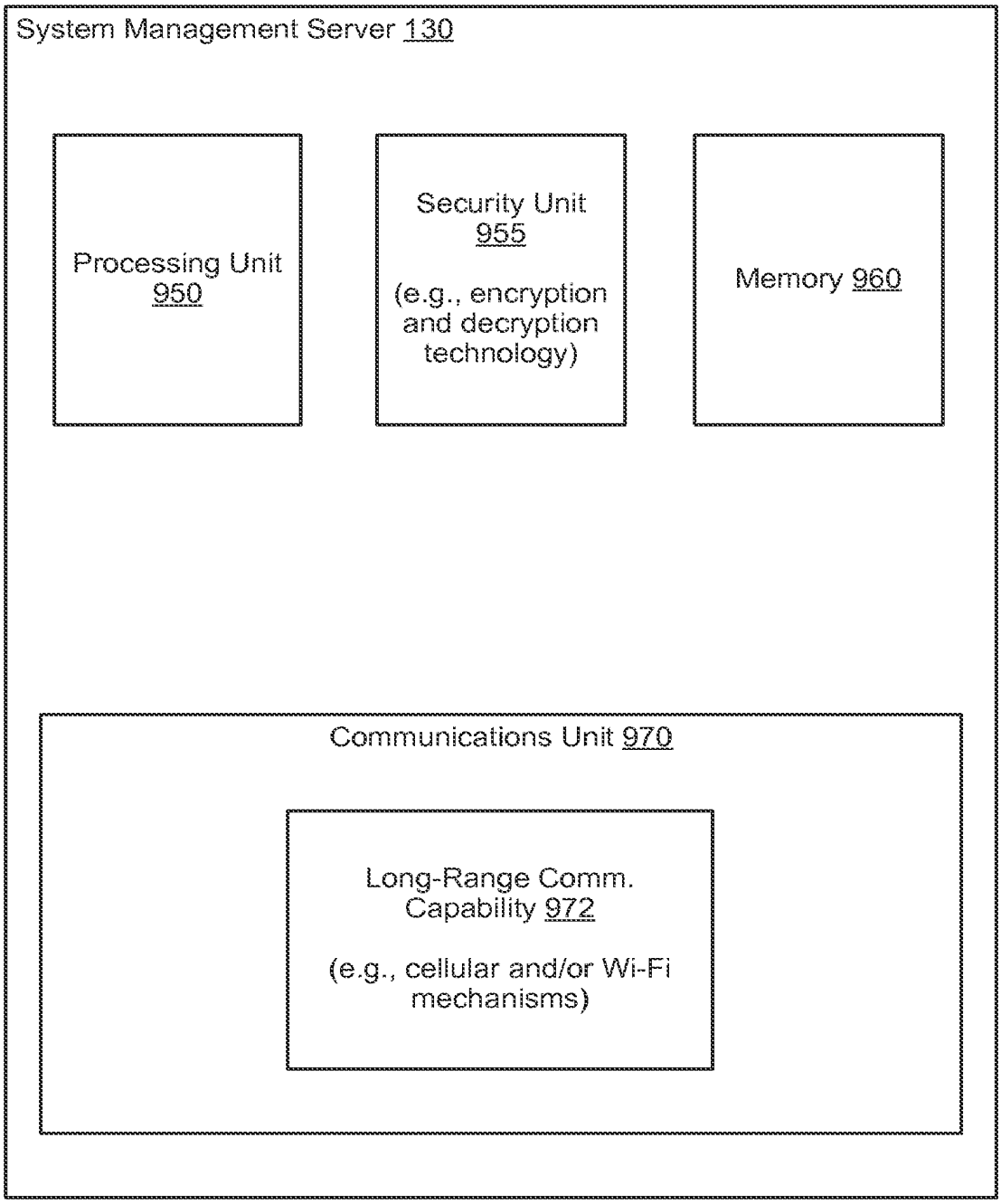

System Management Server 130
Processing Unit 950
Security Unit 955
(e.g., encryption and decryption technology)
Memory 960
Communications Unit 970
Long-Range Comm. Capability 972
(e.g., cellular and/or Wi-Fi mechanisms)

Figure 22

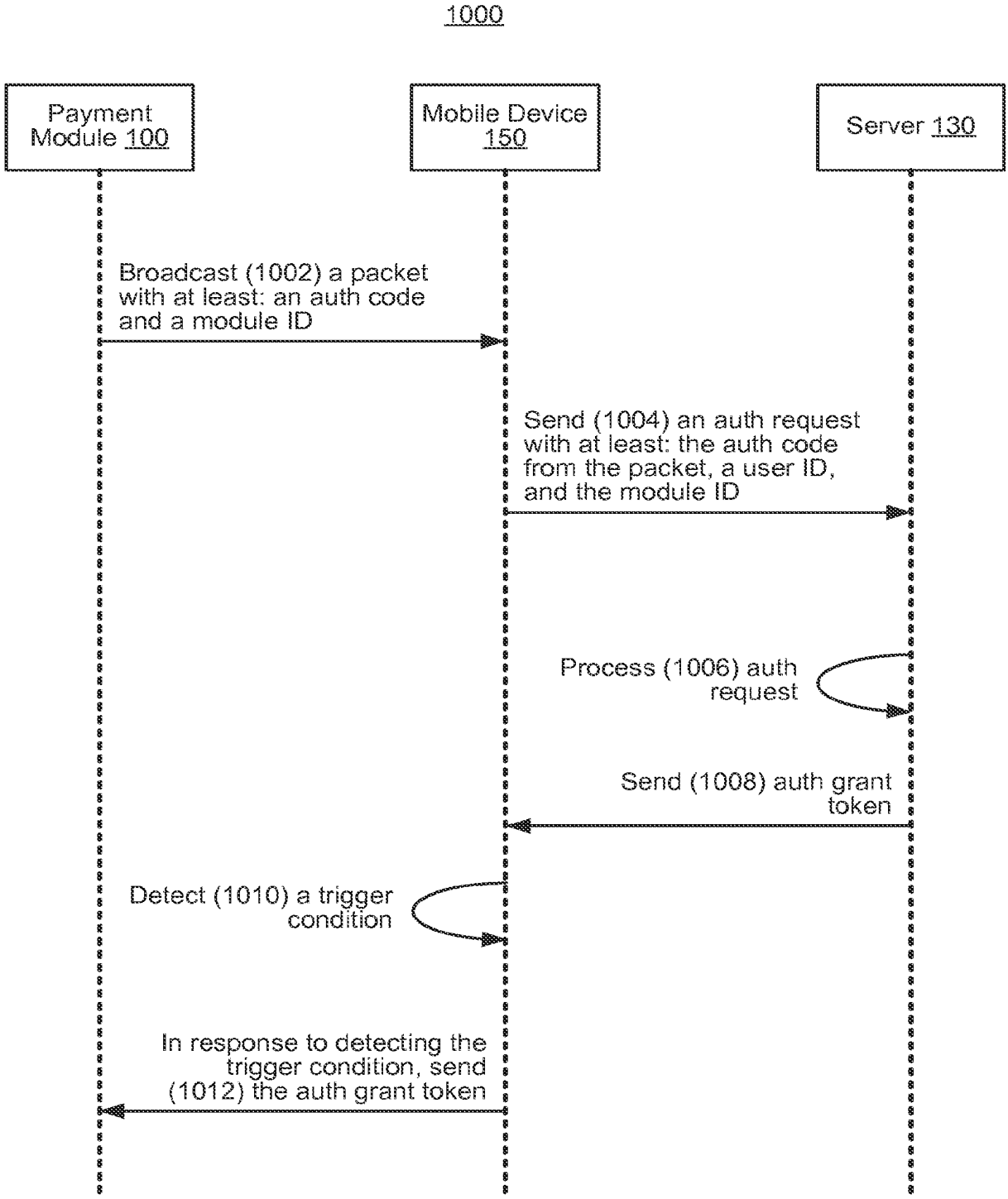

1000
Payment Module 100
Mobile Device 150
Server 130
Broadcast (1002) a packet with at least: an auth code and a module ID
Send (1004) an auth request with at least: the auth code from the packet, a user ID, and the module ID
Process (1006) auth request
Send (1008) auth grant token
Detect (1010) a trigger condition
In response to detecting the trigger condition, send (1012) the auth grant token

Figure 23

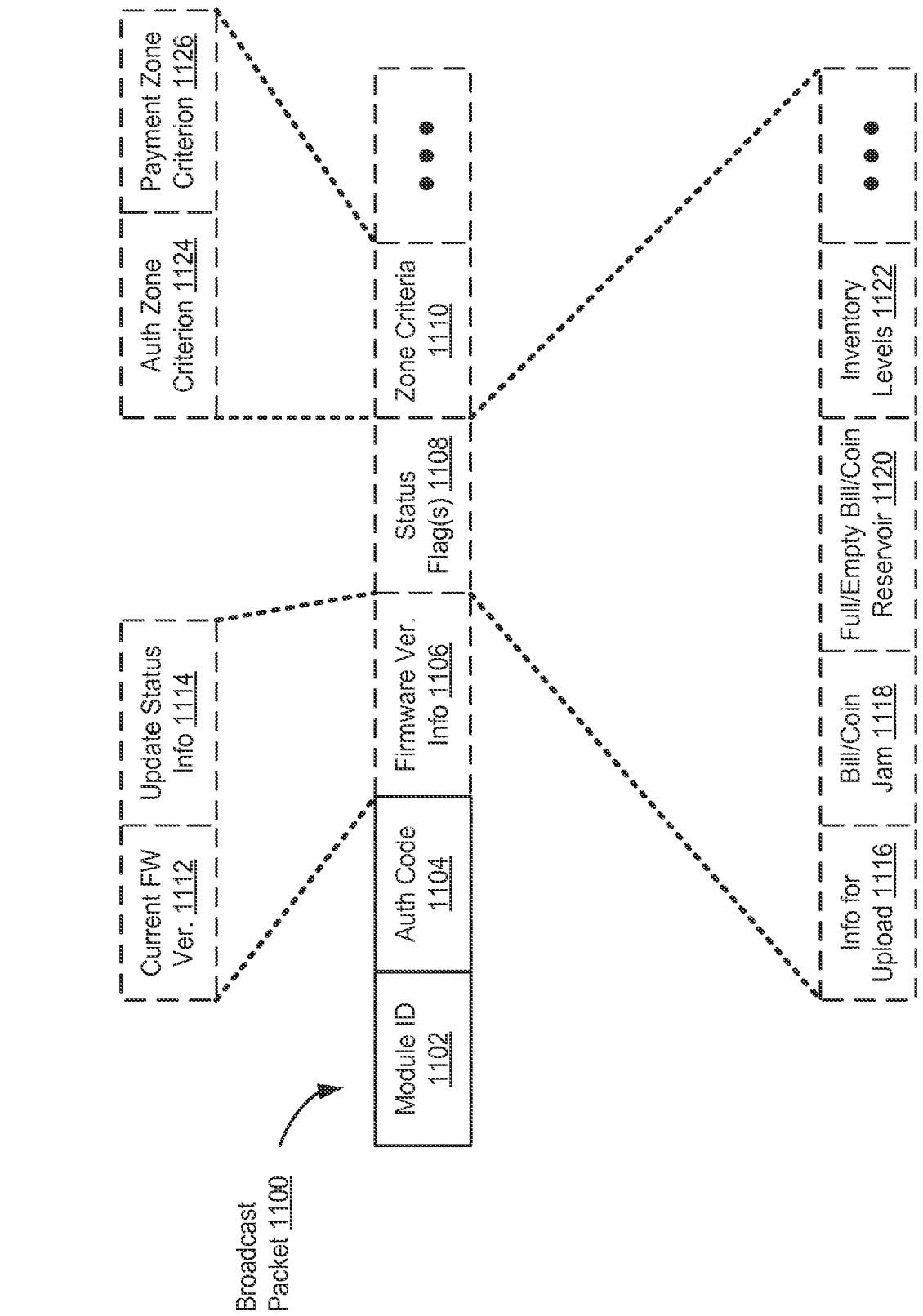

Broadcast Packet 1100
Module ID 1102
Auth Code 1104
Firmware Ver. Info 1106
Status Flag(s) 1108
Zone Criteria 1110
Current FW Ver. 1112
Update Status Info 1114
Auth Zone Criterion 1124
Payment Zone Criterion 1126
Info for Upload 1116
Bill/Coin Jam 1118
Full/Empty Bill/Coin Reservoir 1120
Inventory Levels 1122

Figure 24A

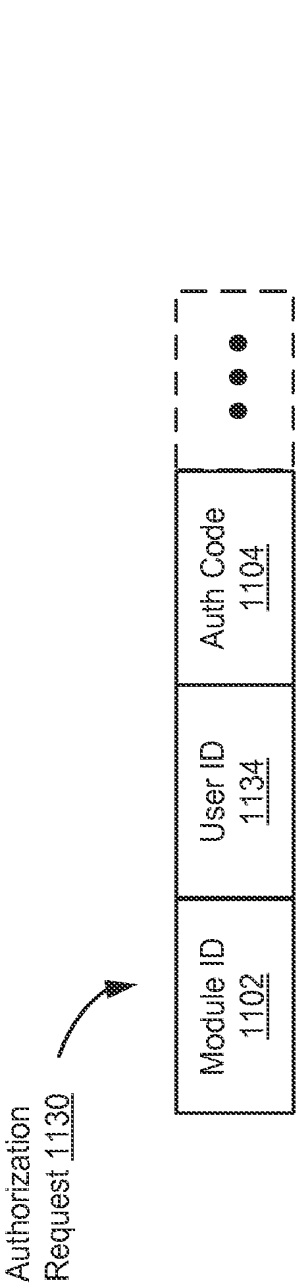


Figure 24B

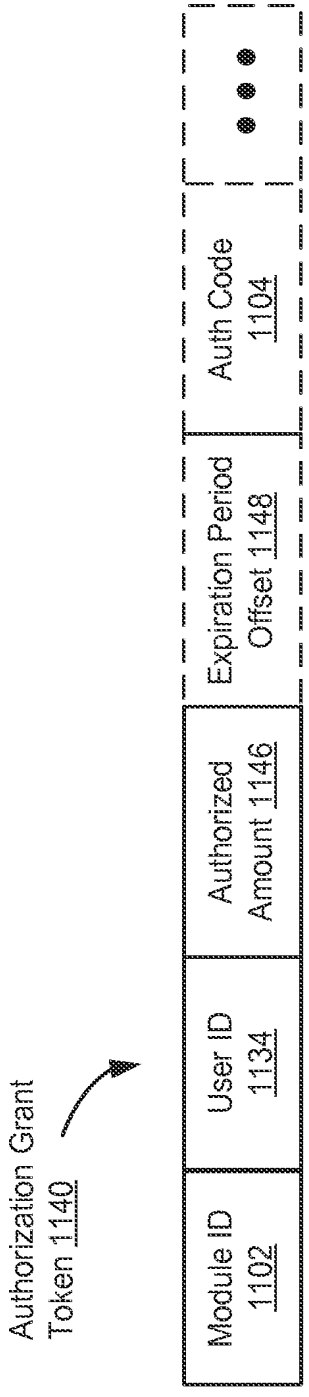


Figure 24C

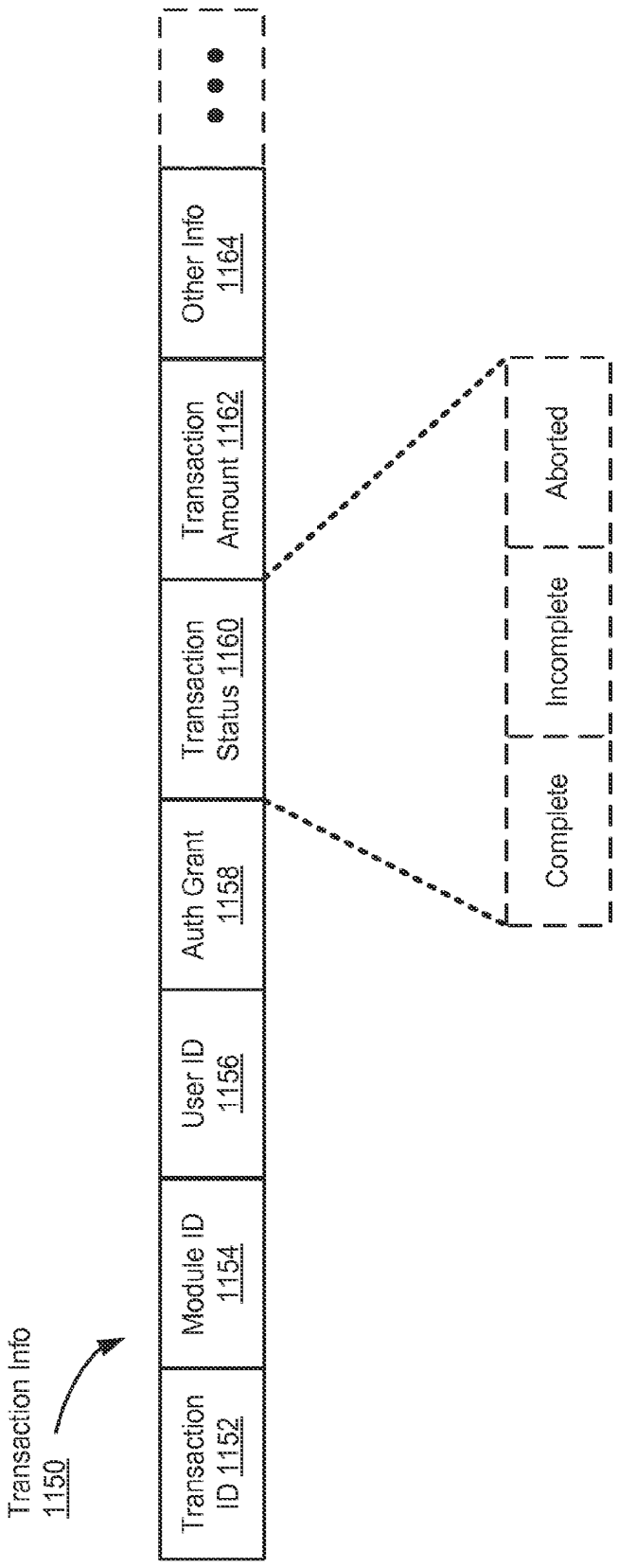

Transaction Info 1150
Transaction ID 1152
Module ID 1154
User ID 1156
Auth Grant 1158
Transaction Status 1160
Transaction Amount 1162
Other Info 1164
Complete
Incomplete
Aborted

Figure 24D

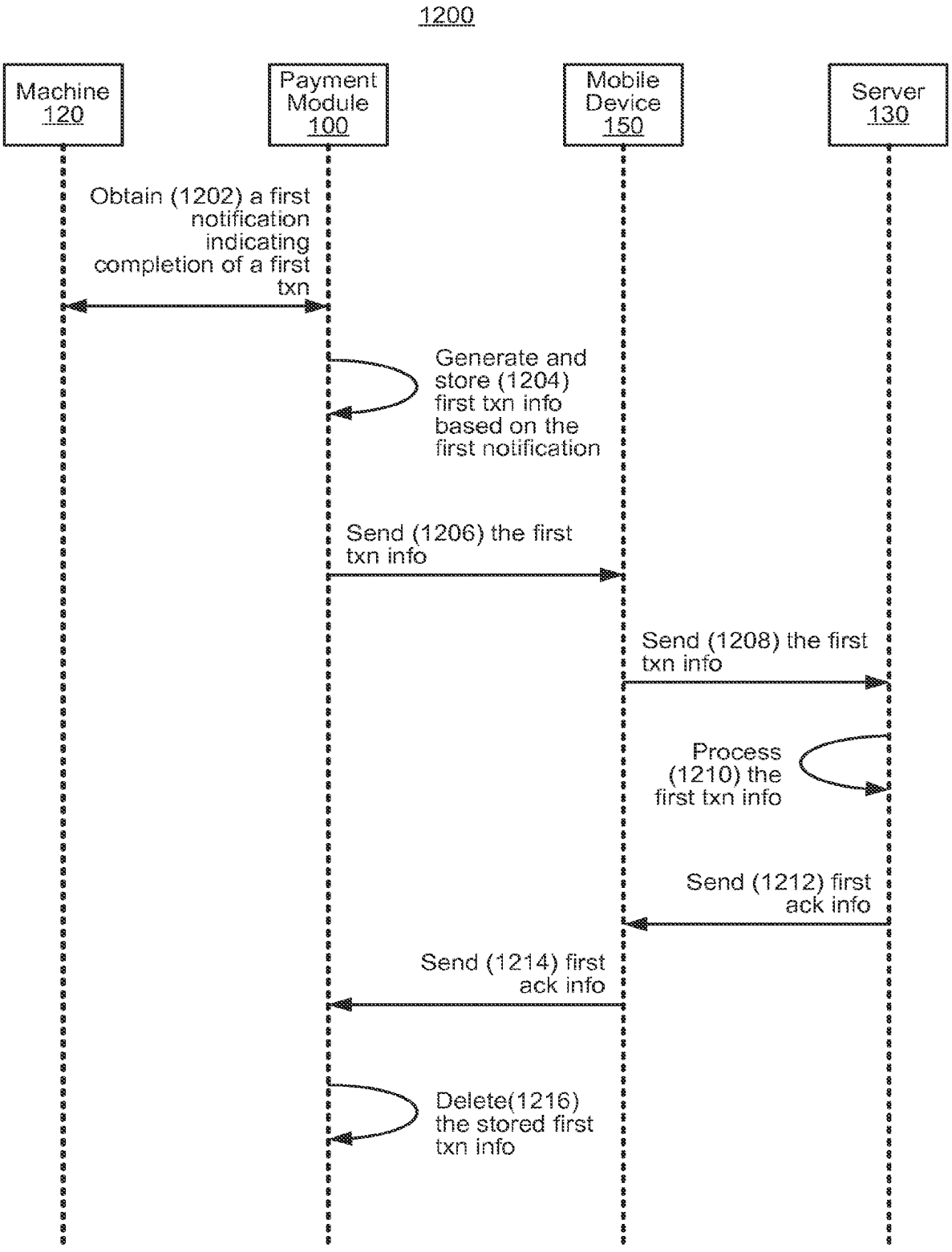

1200
Machine 120
Payment Module 100
Mobile Device 150
Server 130
Obtain (1202) a first notification indicating completion of a first txn
Generate and store (1204) first txn info based on the first notification
Send (1206) the first txn info
Send (1208) the first txn info
Process (1210) the first txn info
Send (1212) first ack info
Send (1214) first ack info
Delete(1216) the stored first txn info

Figure 25A

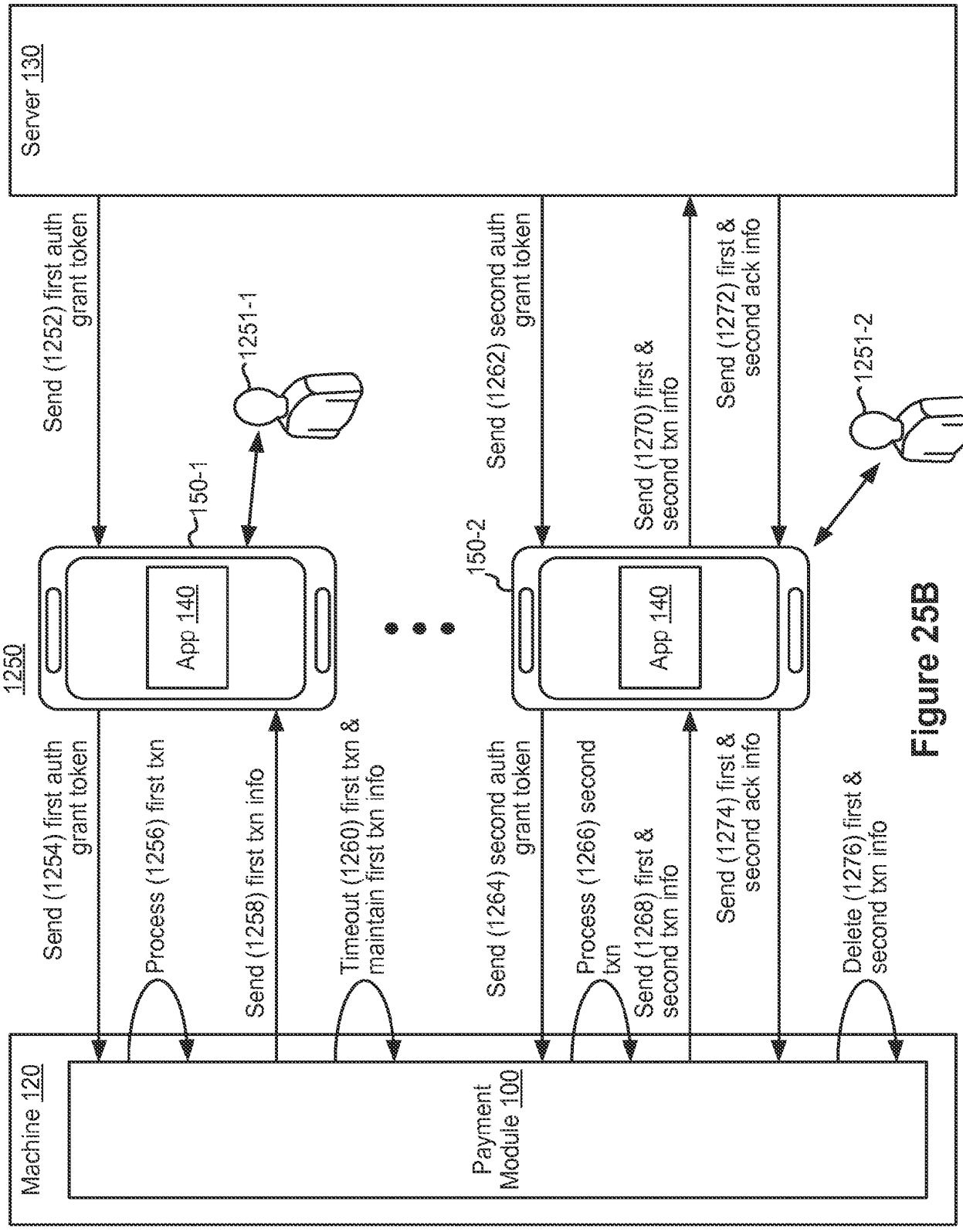

Server 130
Send (1252) first auth grant token
Send (1262) second auth grant token
Send (1270) first & second txn info
Send (1272) first & second ack info
1251-1
1251-2
150-1
150-2
1250
App 140
App 140
Send (1254) first auth grant token
Process (1256) first txn
Send (1258) first txn info
Timeout (1260) first txn & maintain first txn info
Send (1264) second auth grant token
Process (1266) second txn
Send (1268) first & second txn info
Send (1274) first & second ack info
Delete (1276) first & second txn info
Machine 120
Payment Module 100

Figure 25B

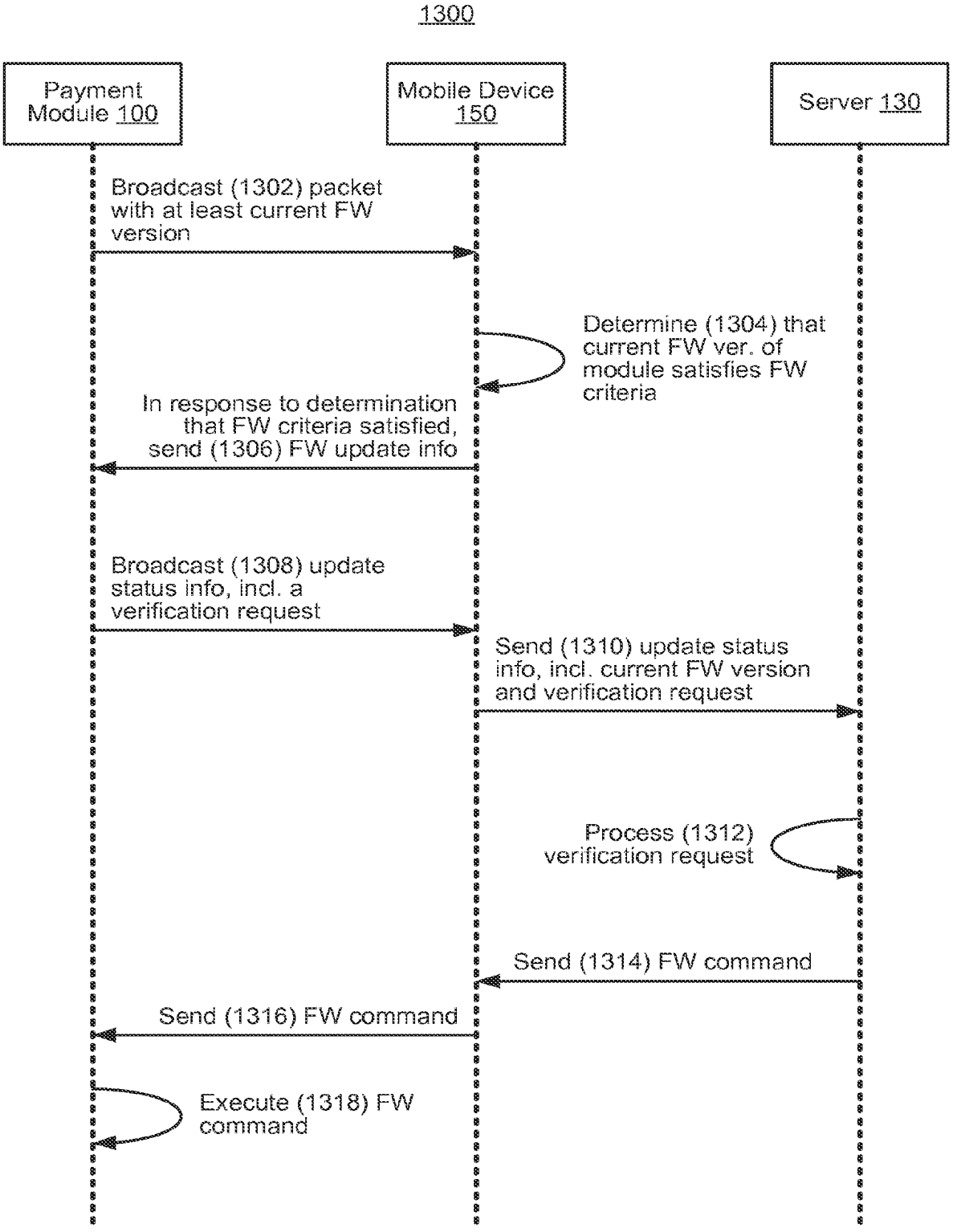

1300
Payment Module 100
Mobile Device 150
Server 130
Broadcast (1302) packet with at least current FW version
Determine (1304) that current FW ver. of module satisfies FW criteria
In response to determination that FW criteria satisfied, send (1306) FW update info
Broadcast (1308) update status info, incl. a verification request
Send (1310) update status info, incl. current FW version and verification request
Process (1312) verification request
Send (1314) FW command
Send (1316) FW command
Execute (1318) FW command

Figure 26A

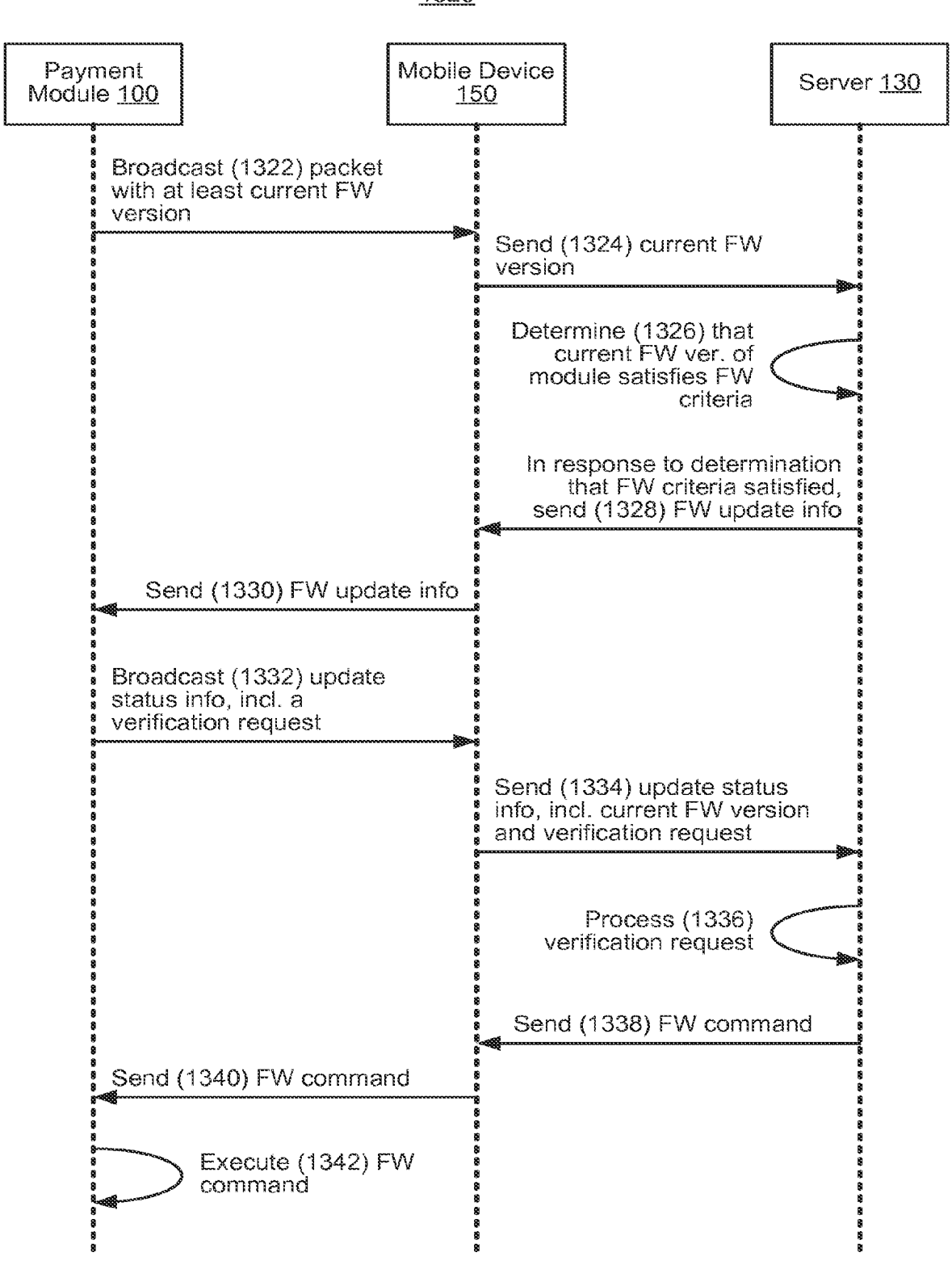

1320
Payment Module 100
Mobile Device 150
Server 130
Broadcast (1322) packet with at least current FW version
Send (1324) current FW version
Determine (1326) that current FW ver. of module satisfies FW criteria
In response to determination that FW criteria satisfied, send (1328) FW update info
Send (1330) FW update info
Broadcast (1332) update status info, incl. a verification request
Send (1334) update status info, incl. current FW version and verification request
Process (1336) verification request
Send (1338) FW command
Send (1340) FW command
Execute (1342) FW command

Figure 26B

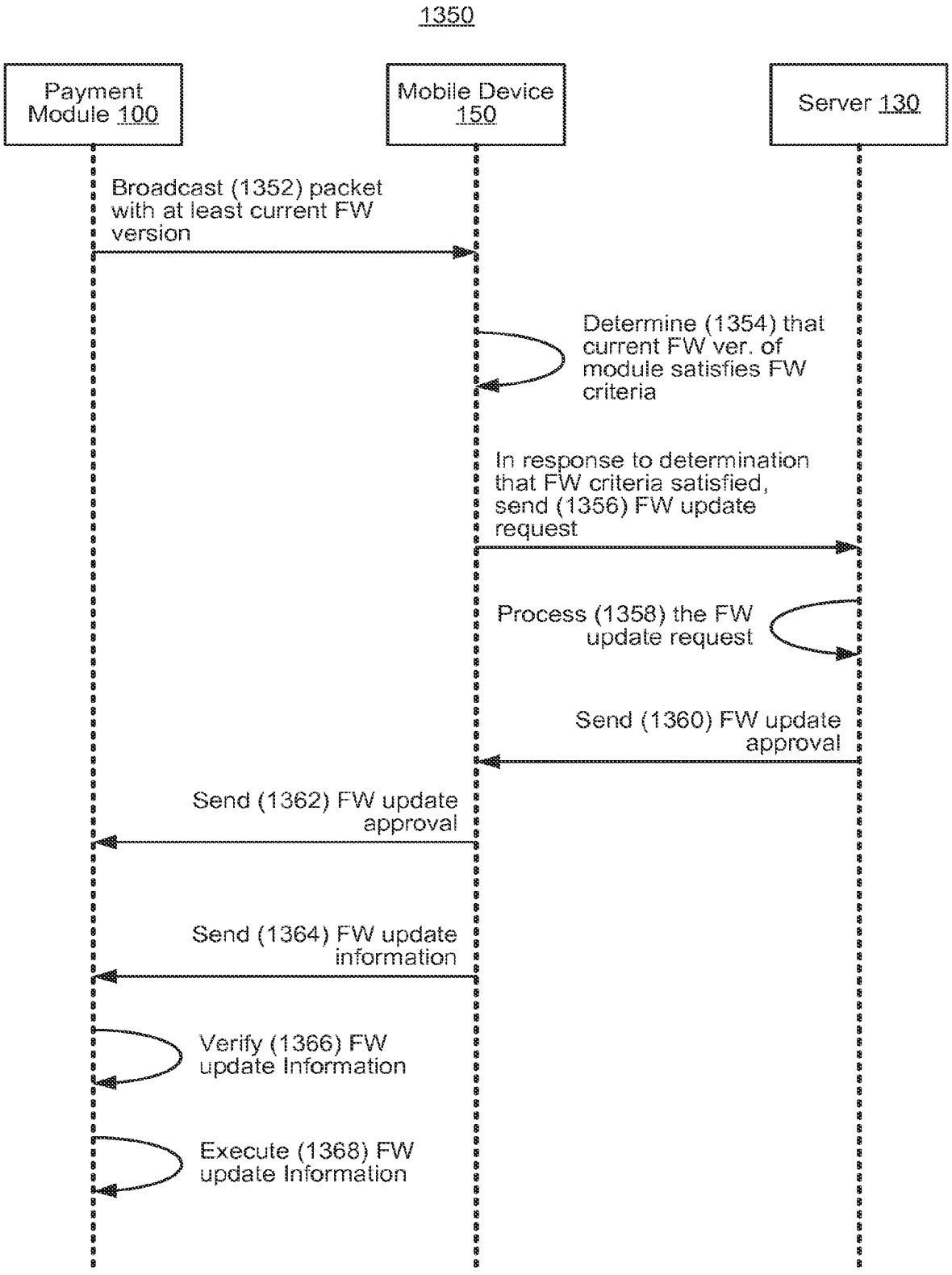

1350
Payment Module 100
Mobile Device 150
Server 130
Broadcast (1352) packet with at least current FW version
Determine (1354) that current FW ver. of module satisfies FW criteria
In response to determination that FW criteria satisfied, send (1356) FW update request
Process (1358) the FW update request
Send (1360) FW update approval
Send (1362) FW update approval
Send (1364) FW update information
Verify (1366) FW update Information
Execute (1368) FW update Information

Figure 26C

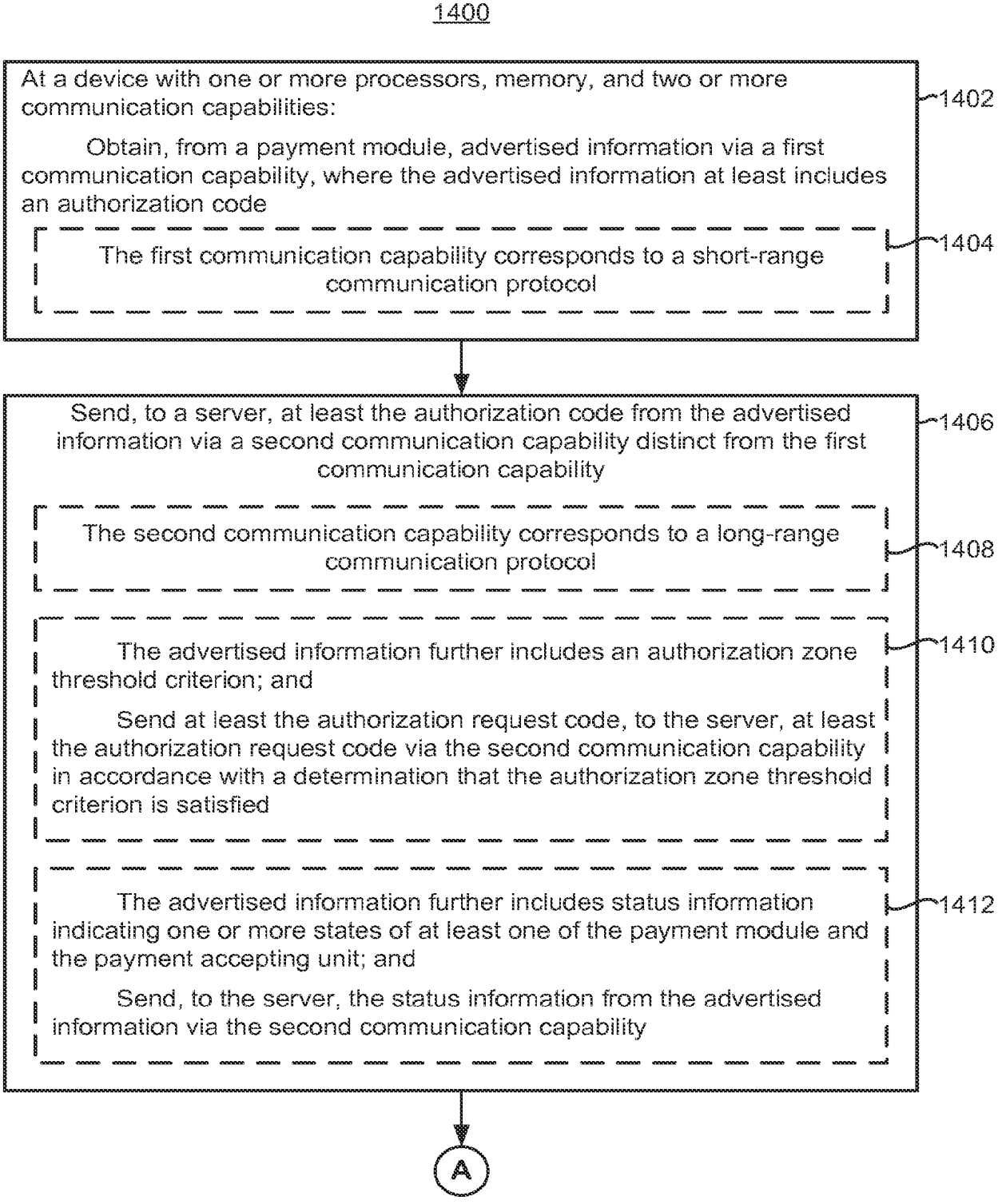

1400
At a device with one or more processors, memory, and two or more communication capabilities:
Obtain, from a payment module, advertised information via a first communication capability, where the advertised information at least includes an authorization code
1402
The first communication capability corresponds to a short-range communication protocol
1404
Send, to a server, at least the authorization code from the advertised information via a second communication capability distinct from the first communication capability
1406
The second communication capability corresponds to a long-range communication protocol
1408
The advertised information further includes an authorization zone threshold criterion; and
Send at least the authorization request code, to the server, at least the authorization request code via the second communication capability in accordance with a determination that the authorization zone threshold criterion is satisfied
1410
The advertised information further includes status information indicating one or more states of at least one of the payment module and the payment accepting unit; and
Send, to the server, the status information from the advertised information via the second communication capability
1412
A

Figure 27A

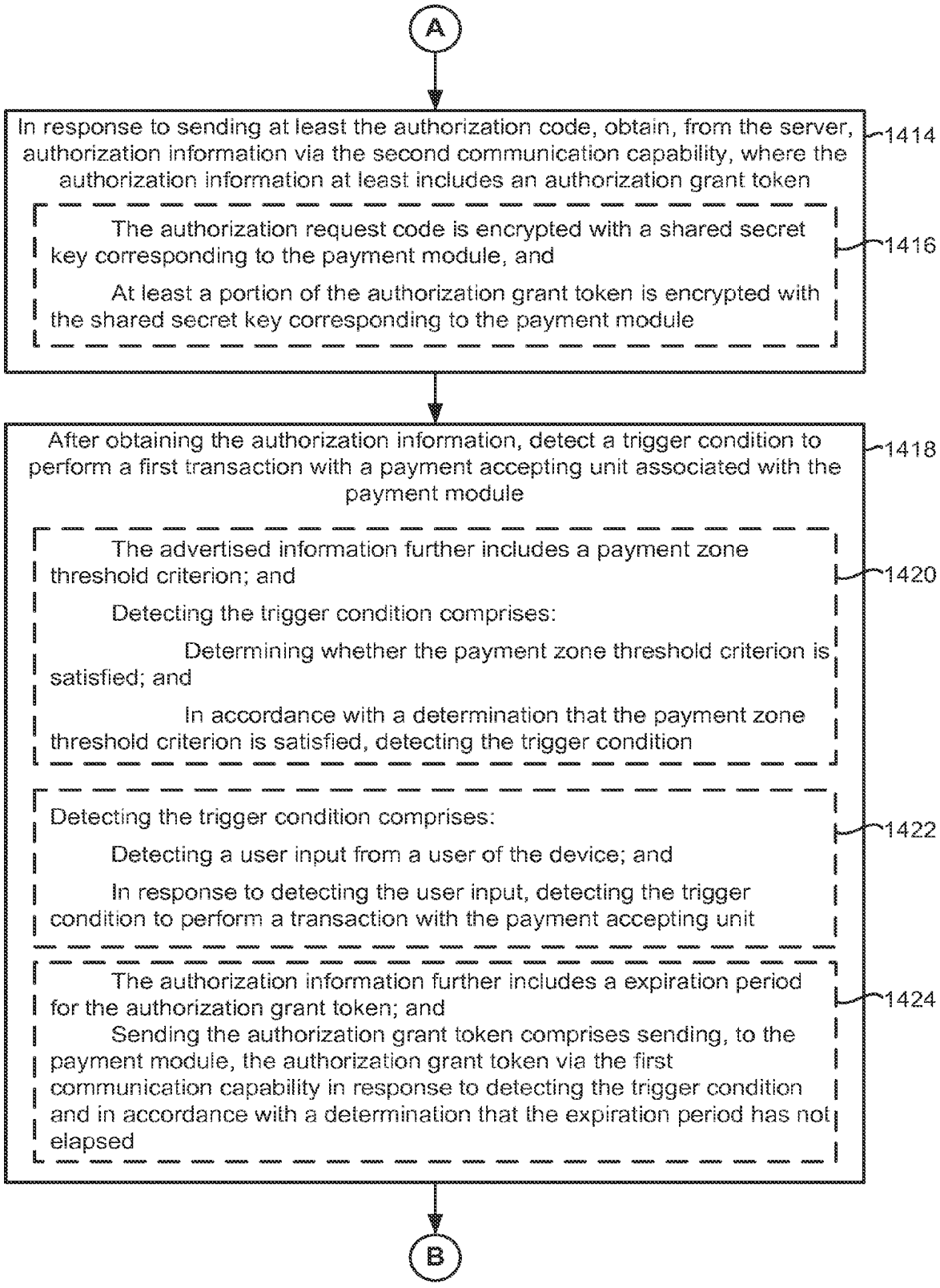

A
In response to sending at least the authorization code, obtain, from the server, authorization information via the second communication capability, where the authorization information at least includes an authorization grant token
1414
The authorization request code is encrypted with a shared secret key corresponding to the payment module, and
At least a portion of the authorization grant token is encrypted with the shared secret key corresponding to the payment module
1416
After obtaining the authorization information, detect a trigger condition to perform a first transaction with a payment accepting unit associated with the payment module
1418
The advertised information further includes a payment zone threshold criterion; and
Detecting the trigger condition comprises:
Determining whether the payment zone threshold criterion is satisfied; and
In accordance with a determination that the payment zone threshold criterion is satisfied, detecting the trigger condition
1420
Detecting the trigger condition comprises:
Detecting a user input from a user of the device; and
In response to detecting the user input, detecting the trigger condition to perform a transaction with the payment accepting unit
1422
The authorization information further includes a expiration period for the authorization grant token; and
Sending the authorization grant token comprises sending, to the payment module, the authorization grant token via the first communication capability in response to detecting the trigger condition and in accordance with a determination that the expiration period has not elapsed
1424
B

Figure 27B

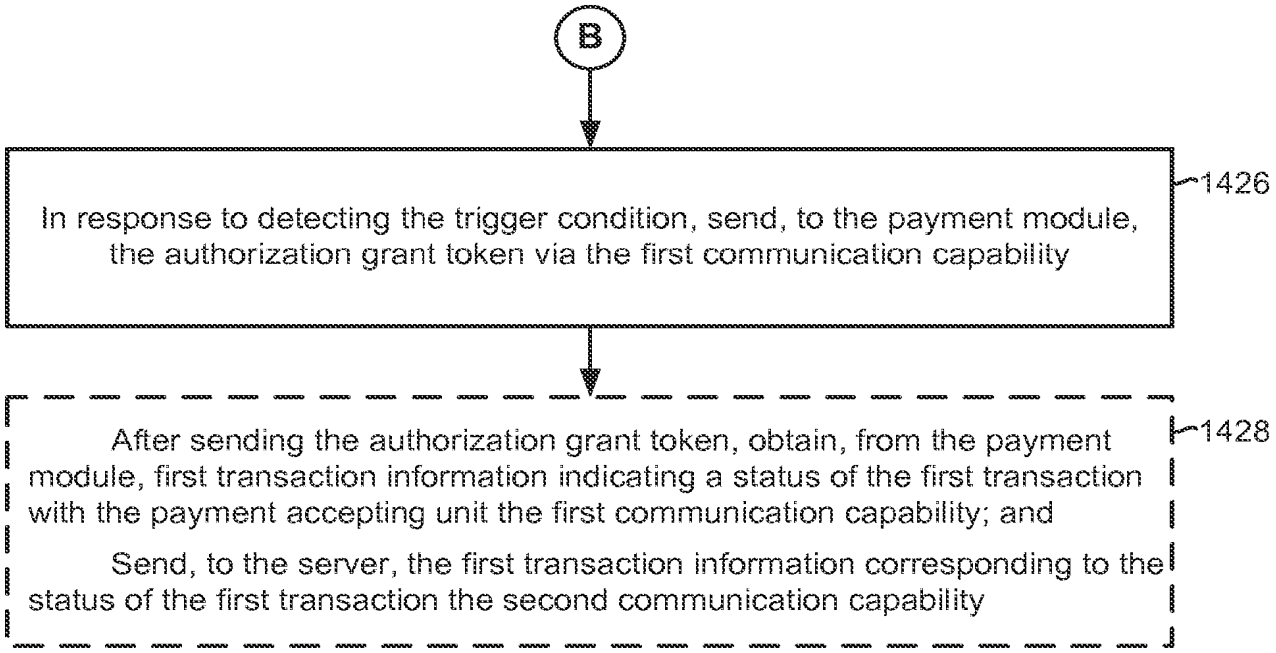

B
In response to detecting the trigger condition, send, to the payment module, the authorization grant token via the first communication capability
1426
After sending the authorization grant token, obtain, from the payment module, first transaction information indicating a status of the first transaction with the payment accepting unit the first communication capability; and
Send, to the server, the first transaction information corresponding to the status of the first transaction the second communication capability
1428

Figure 27C

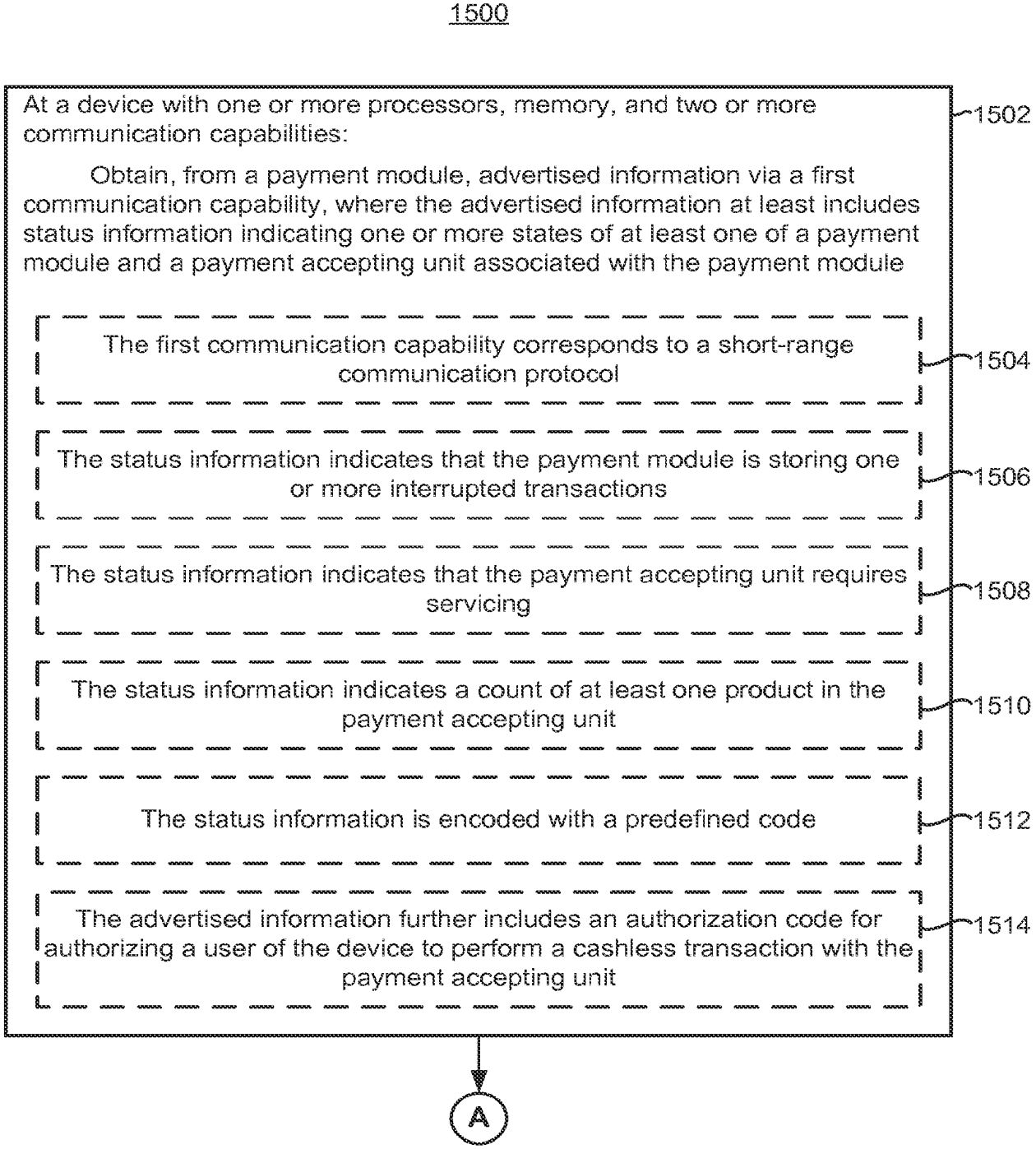

1500
At a device with one or more processors, memory, and two or more communication capabilities:
Obtain, from a payment module, advertised information via a first communication capability, where the advertised information at least includes status information indicating one or more states of at least one of a payment module and a payment accepting unit associated with the payment module
1502
The first communication capability corresponds to a short-range communication protocol
1504
The status information indicates that the payment module is storing one or more interrupted transactions
1506
The status information indicates that the payment accepting unit requires servicing
1508
The status information indicates a count of at least one product in the payment accepting unit
1510
The status information is encoded with a predefined code
1512
The advertised information further includes an authorization code for authorizing a user of the device to perform a cashless transaction with the payment accepting unit
1514
A

Figure 28A

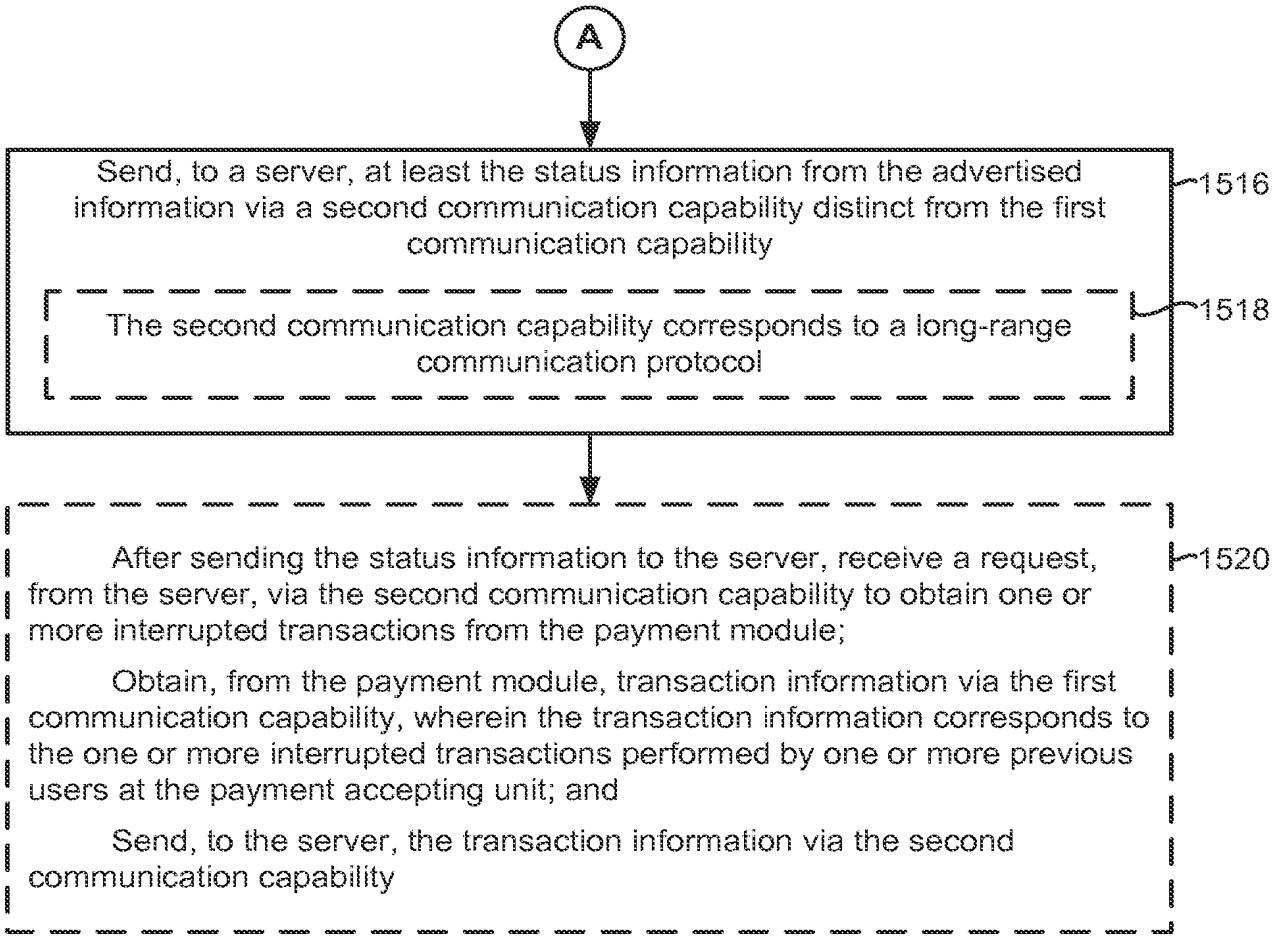

A
Send, to a server, at least the status information from the advertised information via a second communication capability distinct from the first communication capability
1516
The second communication capability corresponds to a long-range communication protocol
1518
After sending the status information to the server, receive a request, from the server, via the second communication capability to obtain one or more interrupted transactions from the payment module;
1520
Obtain, from the payment module, transaction information via the first communication capability, wherein the transaction information corresponds to the one or more interrupted transactions performed by one or more previous users at the payment accepting unit; and
Send, to the server, the transaction information via the second communication capability

Figure 28B

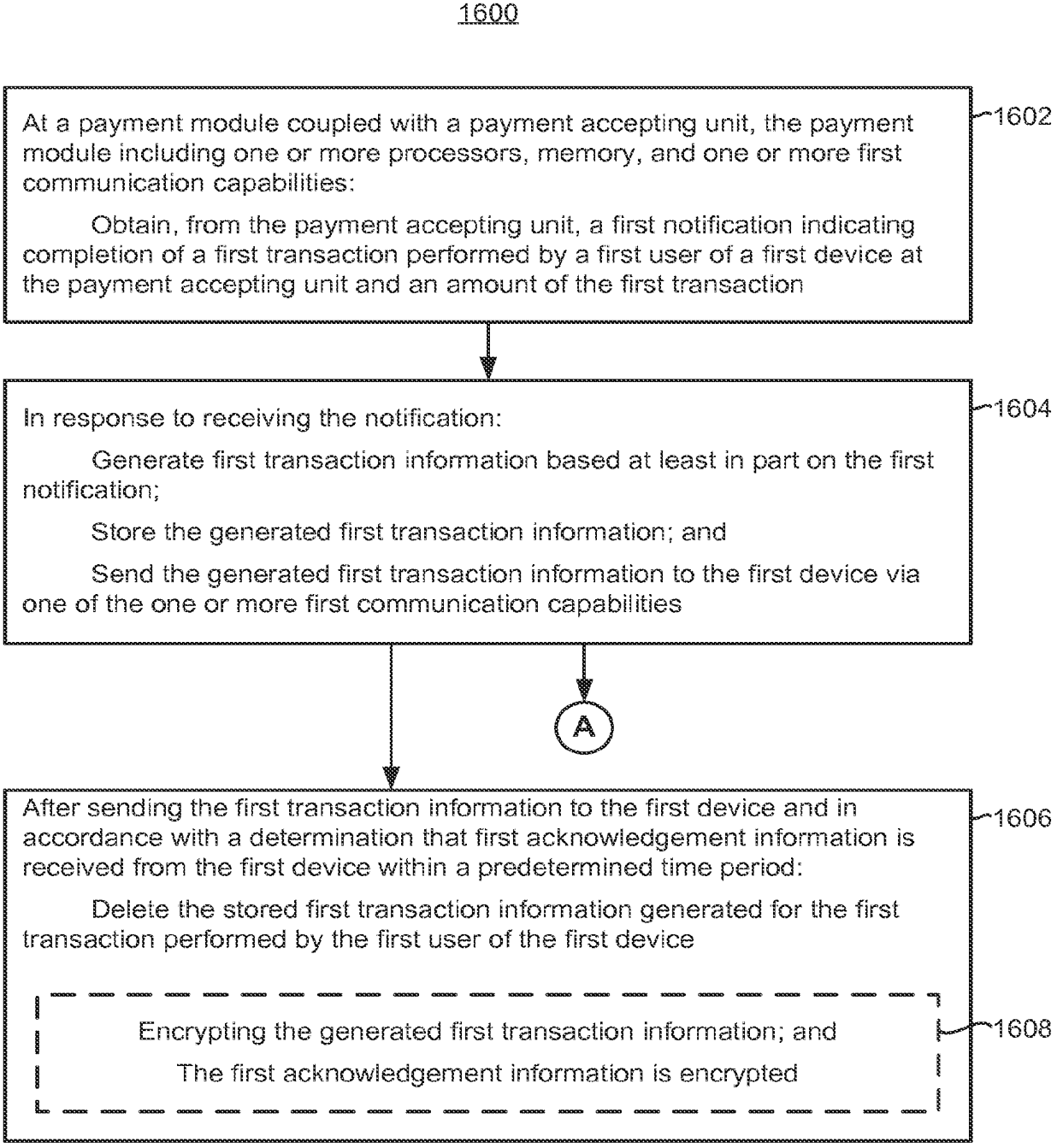

1600
At a payment module coupled with a payment accepting unit, the payment module including one or more processors, memory, and one or more first communication capabilities:
Obtain, from the payment accepting unit, a first notification indicating completion of a first transaction performed by a first user of a first device at the payment accepting unit and an amount of the first transaction
1602
In response to receiving the notification:
Generate first transaction information based at least in part on the first notification;
Store the generated first transaction information; and
Send the generated first transaction information to the first device via one of the one or more first communication capabilities
1604
A
After sending the first transaction information to the first device and in accordance with a determination that first acknowledgement information is received from the first device within a predetermined time period:
Delete the stored first transaction information generated for the first transaction performed by the first user of the first device
1606
Encrypting the generated first transaction information; and
The first acknowledgement information is encrypted
1608

Figure 29A

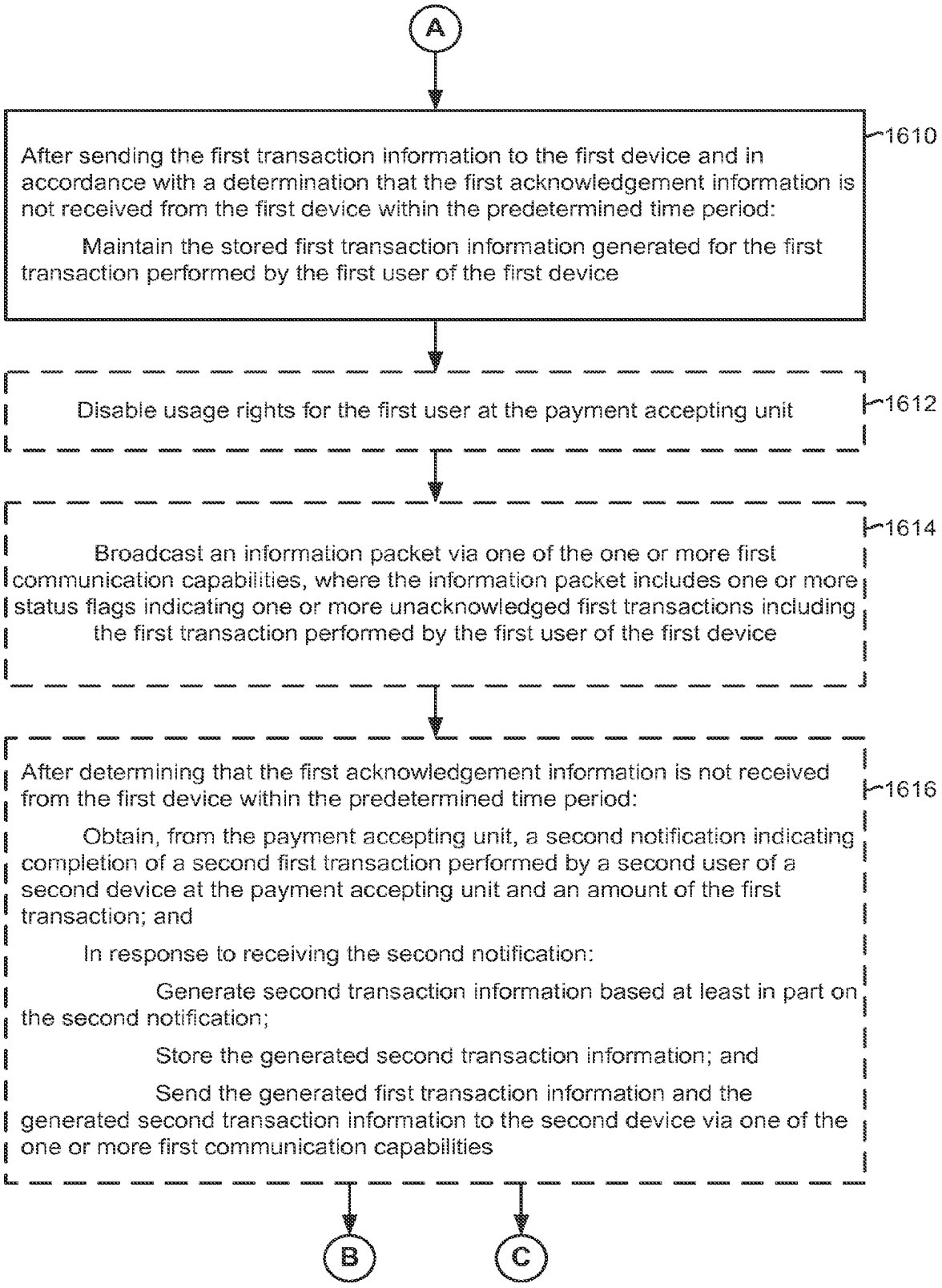

A
After sending the first transaction information to the first device and in accordance with a determination that the first acknowledgement information is not received from the first device within the predetermined time period:
Maintain the stored first transaction information generated for the first transaction performed by the first user of the first device
1610
Disable usage rights for the first user at the payment accepting unit
1612
Broadcast an information packet via one of the one or more first communication capabilities, where the information packet includes one or more status flags indicating one or more unacknowledged first transactions including the first transaction performed by the first user of the first device
1614
After determining that the first acknowledgement information is not received from the first device within the predetermined time period:
Obtain, from the payment accepting unit, a second notification indicating completion of a second first transaction performed by a second user of a second device at the payment accepting unit and an amount of the first transaction; and
In response to receiving the second notification:
Generate second transaction information based at least in part on the second notification;
Store the generated second transaction information; and
Send the generated first transaction information and the generated second transaction information to the second device via one of the one or more first communication capabilities
1616
B
C

Figure 29B

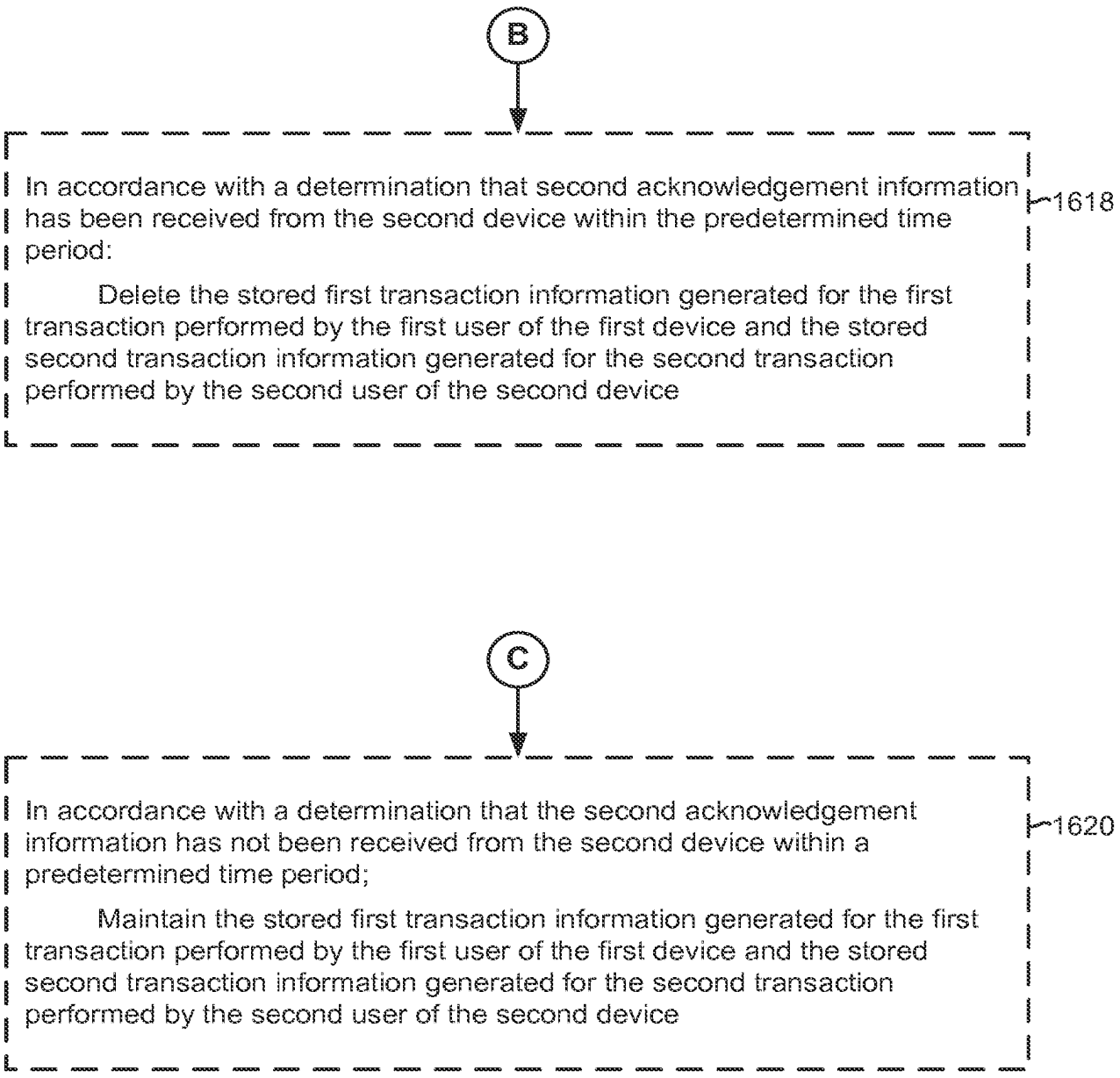

B
In accordance with a determination that second acknowledgement information has been received from the second device within the predetermined time period:
Delete the stored first transaction information generated for the first transaction performed by the first user of the first device and the stored second transaction information generated for the second transaction performed by the second user of the second device
1618
C
In accordance with a determination that the second acknowledgement information has not been received from the second device within a predetermined time period;
Maintain the stored first transaction information generated for the first transaction performed by the first user of the first device and the stored second transaction information generated for the second transaction performed by the second user of the second device
1620

Figure 29C

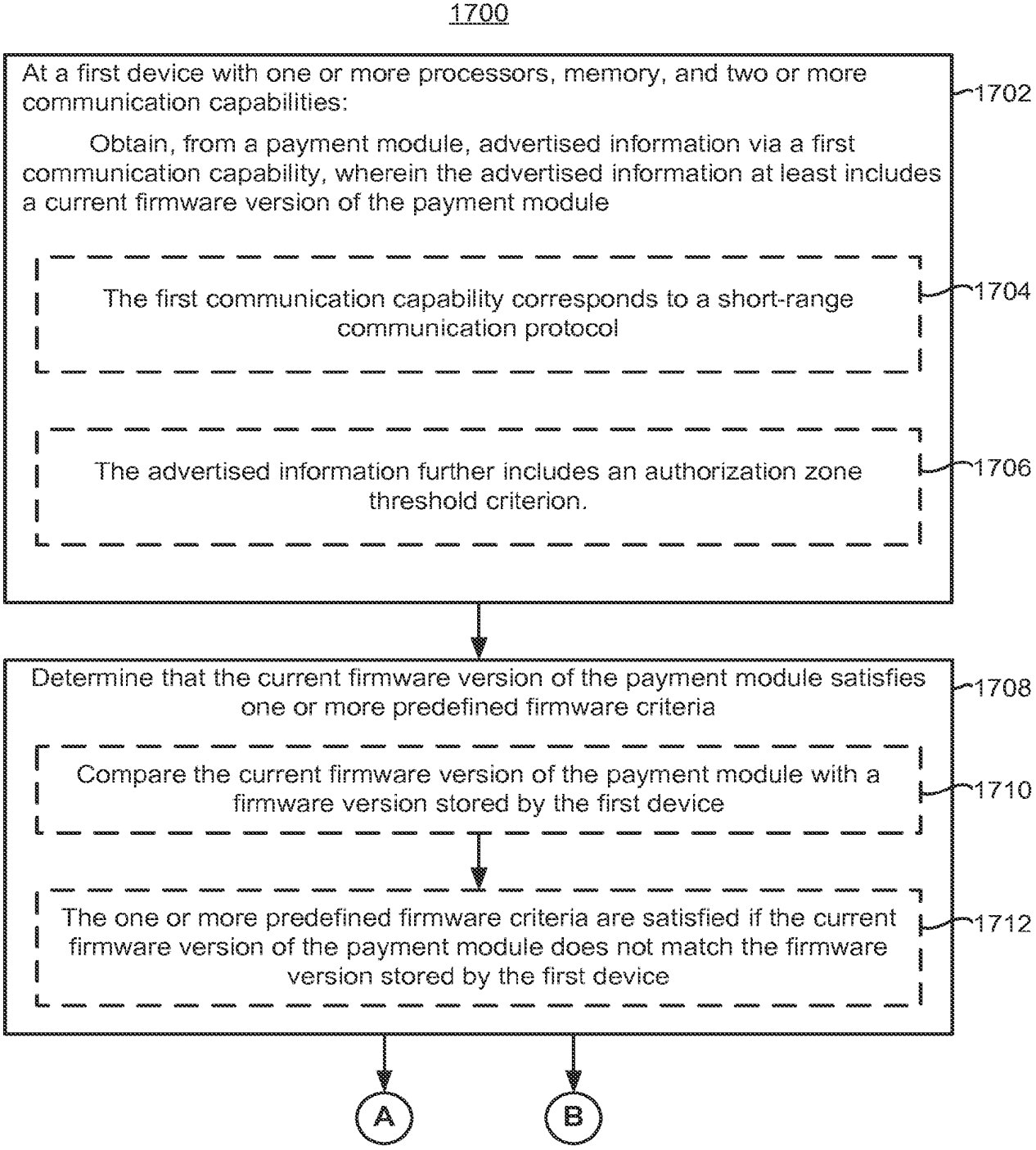

1700
At a first device with one or more processors, memory, and two or more communication capabilities:
Obtain, from a payment module, advertised information via a first communication capability, wherein the advertised information at least includes a current firmware version of the payment module
1702
The first communication capability corresponds to a short-range communication protocol
1704
The advertised information further includes an authorization zone threshold criterion.
1706
Determine that the current firmware version of the payment module satisfies one or more predefined firmware criteria
1708
Compare the current firmware version of the payment module with a firmware version stored by the first device
1710
The one or more predefined firmware criteria are satisfied if the current firmware version of the payment module does not match the firmware version stored by the first device
1712
A
B

Figure 30A

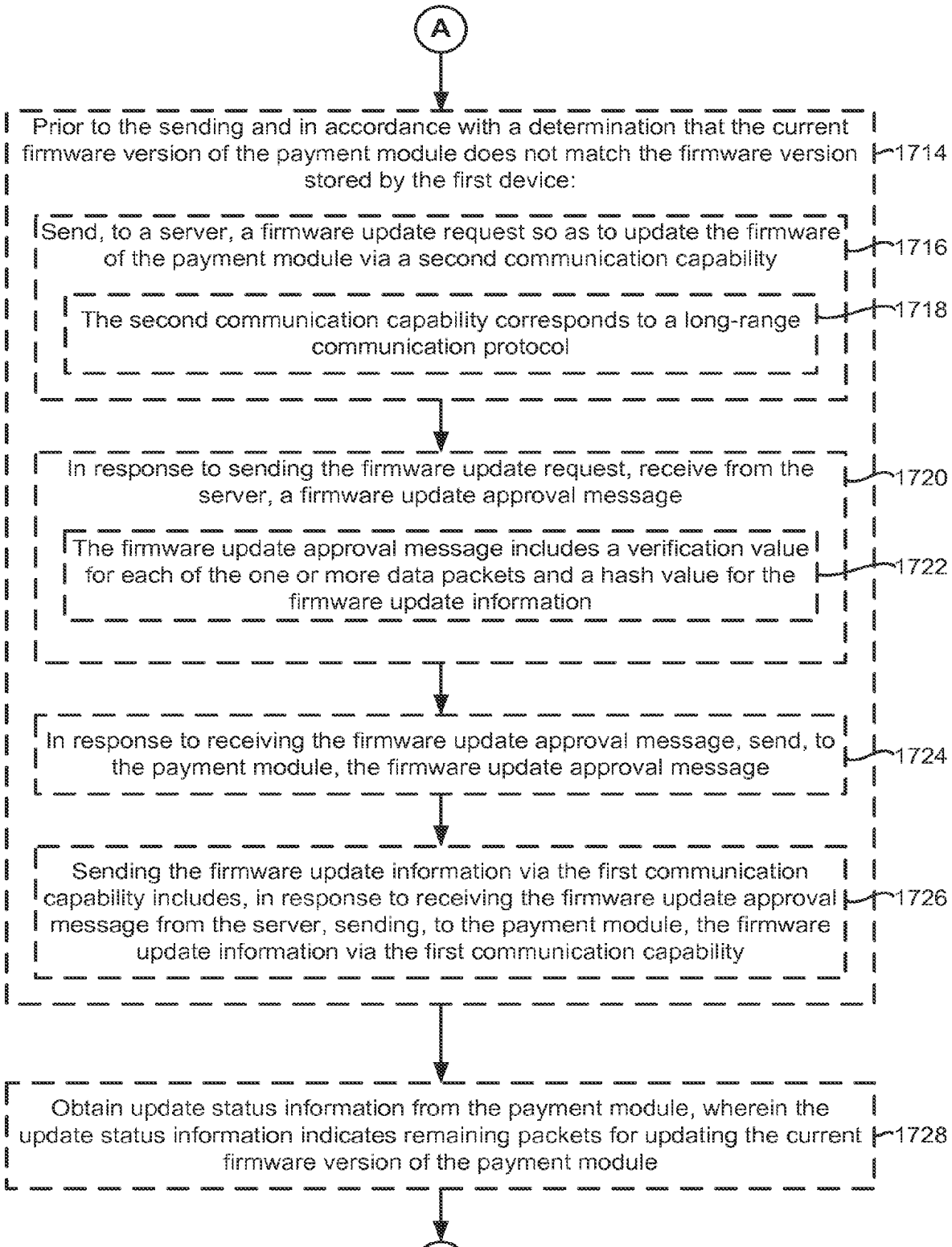

A
Prior to the sending and in accordance with a determination that the current firmware version of the payment module does not match the firmware version stored by the first device:
1714
Send, to a server, a firmware update request so as to update the firmware of the payment module via a second communication capability
1716
The second communication capability corresponds to a long-range communication protocol
1718
In response to sending the firmware update request, receive from the server, a firmware update approval message
1720
The firmware update approval message includes a verification value for each of the one or more data packets and a hash value for the firmware update information
1722
In response to receiving the firmware update approval message, send, to the payment module, the firmware update approval message
1724
Sending the firmware update information via the first communication capability includes, in response to receiving the firmware update approval message from the server, sending, to the payment module, the firmware update information via the first communication capability
1726
Obtain update status information from the payment module, wherein the update status information indicates remaining packets for updating the current firmware version of the payment module
1728
C

Figure 30B

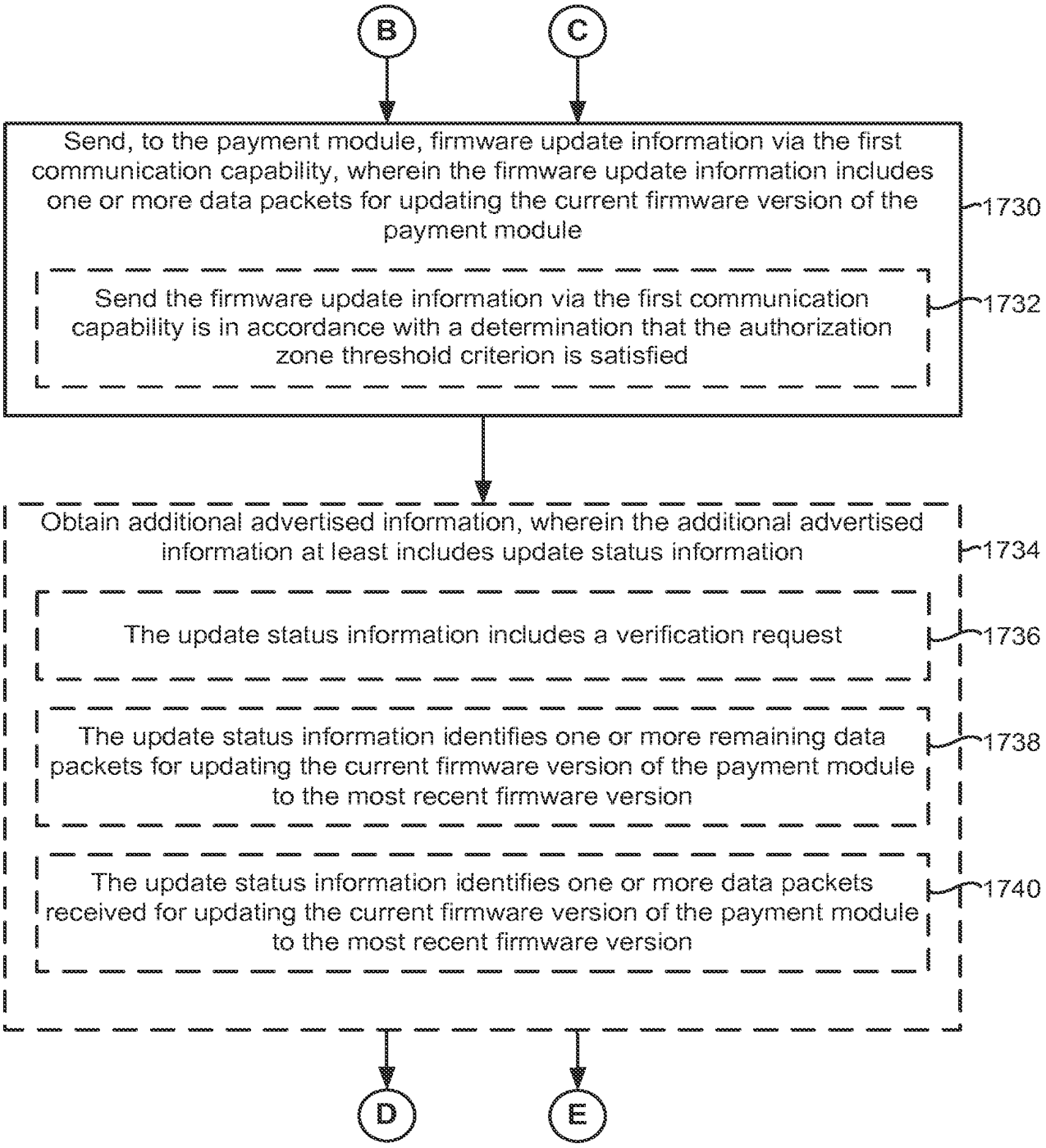

B
C
Send, to the payment module, firmware update information via the first communication capability, wherein the firmware update information includes one or more data packets for updating the current firmware version of the payment module
1730
Send the firmware update information via the first communication capability is in accordance with a determination that the authorization zone threshold criterion is satisfied
1732
Obtain additional advertised information, wherein the additional advertised information at least includes update status information
1734
The update status information includes a verification request
1736
The update status information identifies one or more remaining data packets for updating the current firmware version of the payment module to the most recent firmware version
1738
The update status information identifies one or more data packets received for updating the current firmware version of the payment module to the most recent firmware version
1740
D
E

Figure 30C

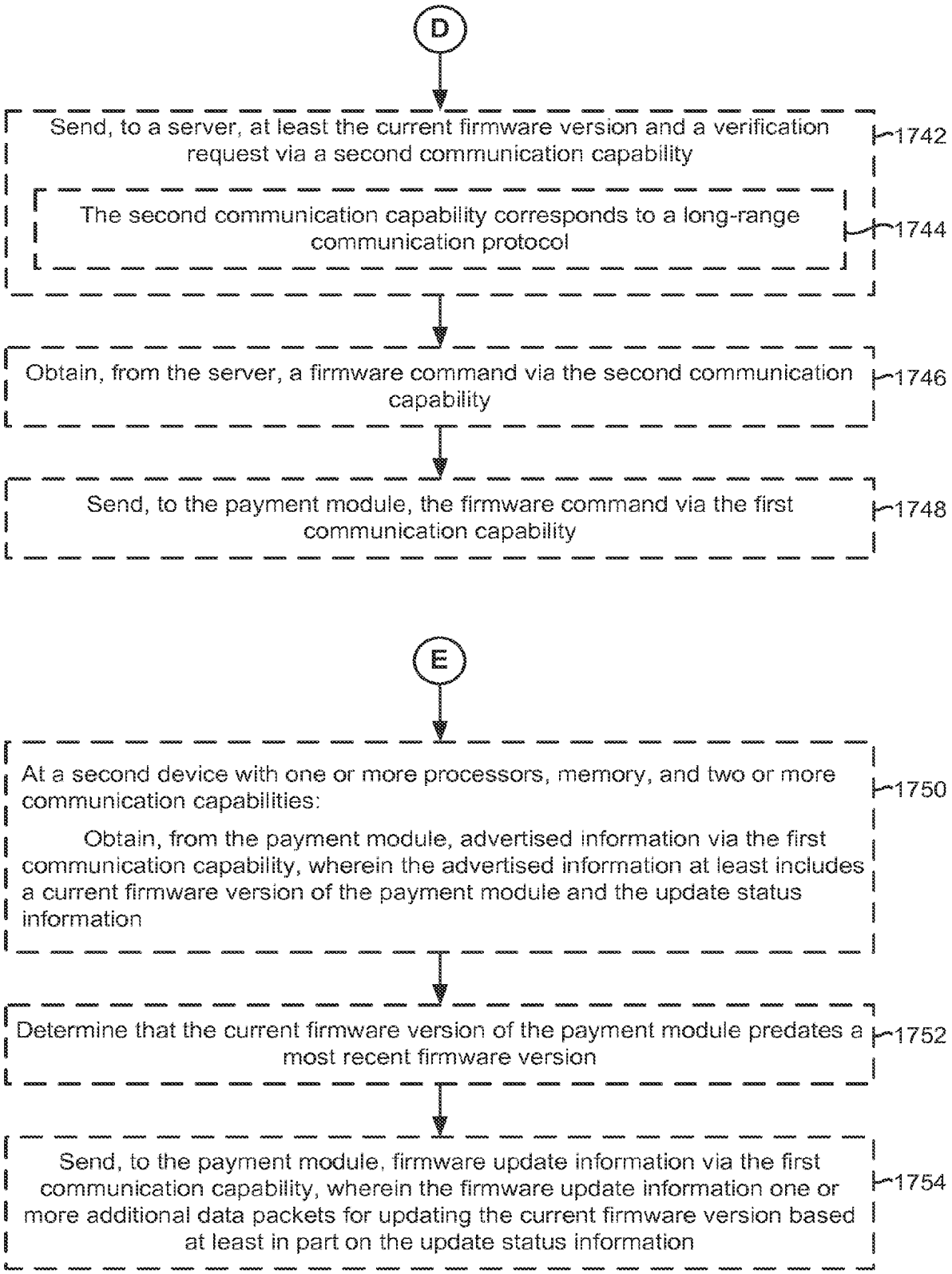

D
Send, to a server, at least the current firmware version and a verification request via a second communication capability
1742
The second communication capability corresponds to a long-range communication protocol
1744
Obtain, from the server, a firmware command via the second communication capability
1746
Send, to the payment module, the firmware command via the first communication capability
1748
E
At a second device with one or more processors, memory, and two or more communication capabilities:
1750
Obtain, from the payment module, advertised information via the first communication capability, wherein the advertised information at least includes a current firmware version of the payment module and the update status information
Determine that the current firmware version of the payment module predates a most recent firmware version
1752
Send, to the payment module, firmware update information via the first communication capability, wherein the firmware update information one or more additional data packets for updating the current firmware version based at least in part on the update status information
1754

Figure 30D

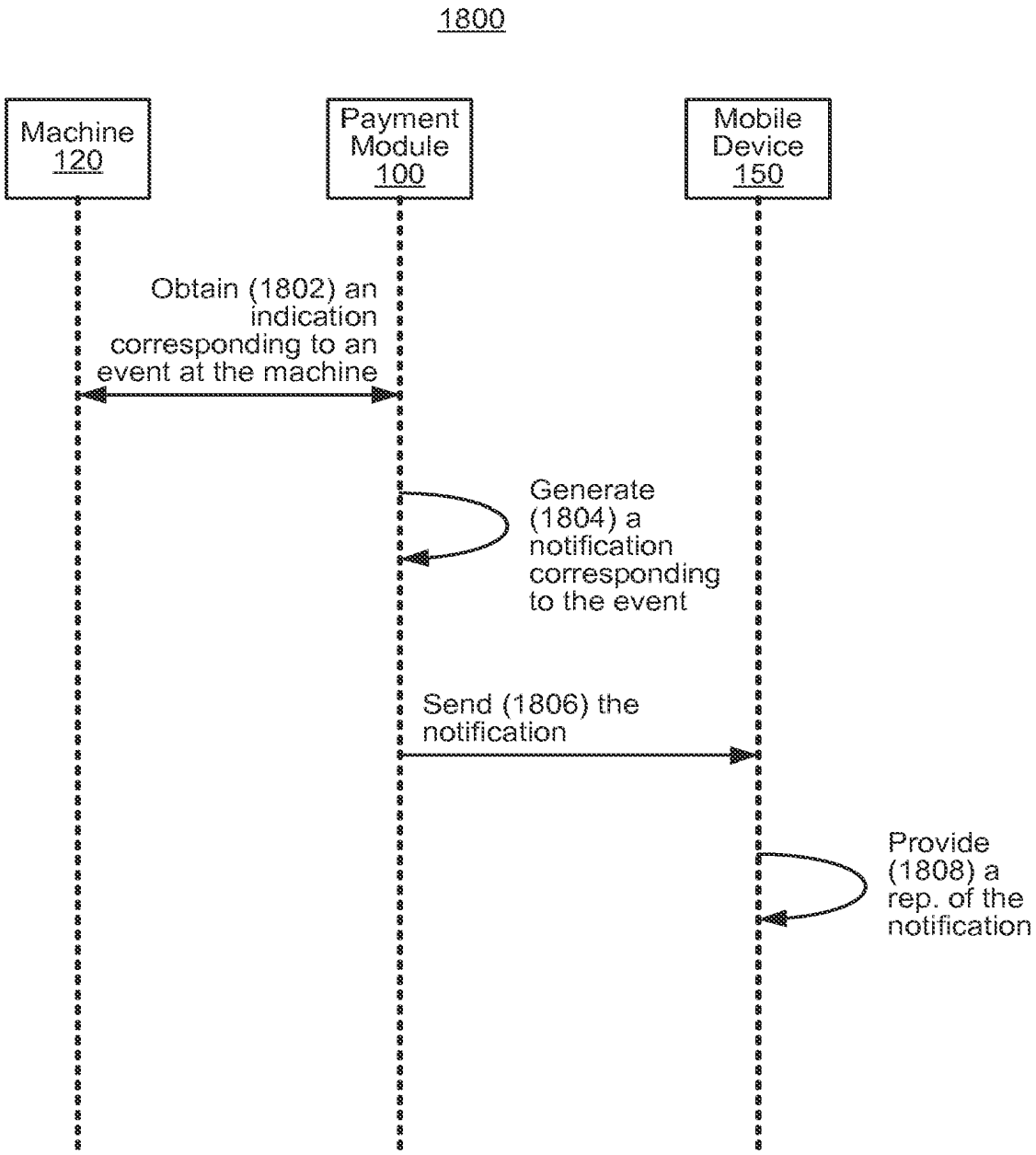

1800
Machine 120
Payment Module 100
Mobile Device 150
Obtain (1802) an indication corresponding to an event at the machine
Generate (1804) a notification corresponding to the event
Send (1806) the notification
Provide (1808) a rep. of the notification

Figure 31A

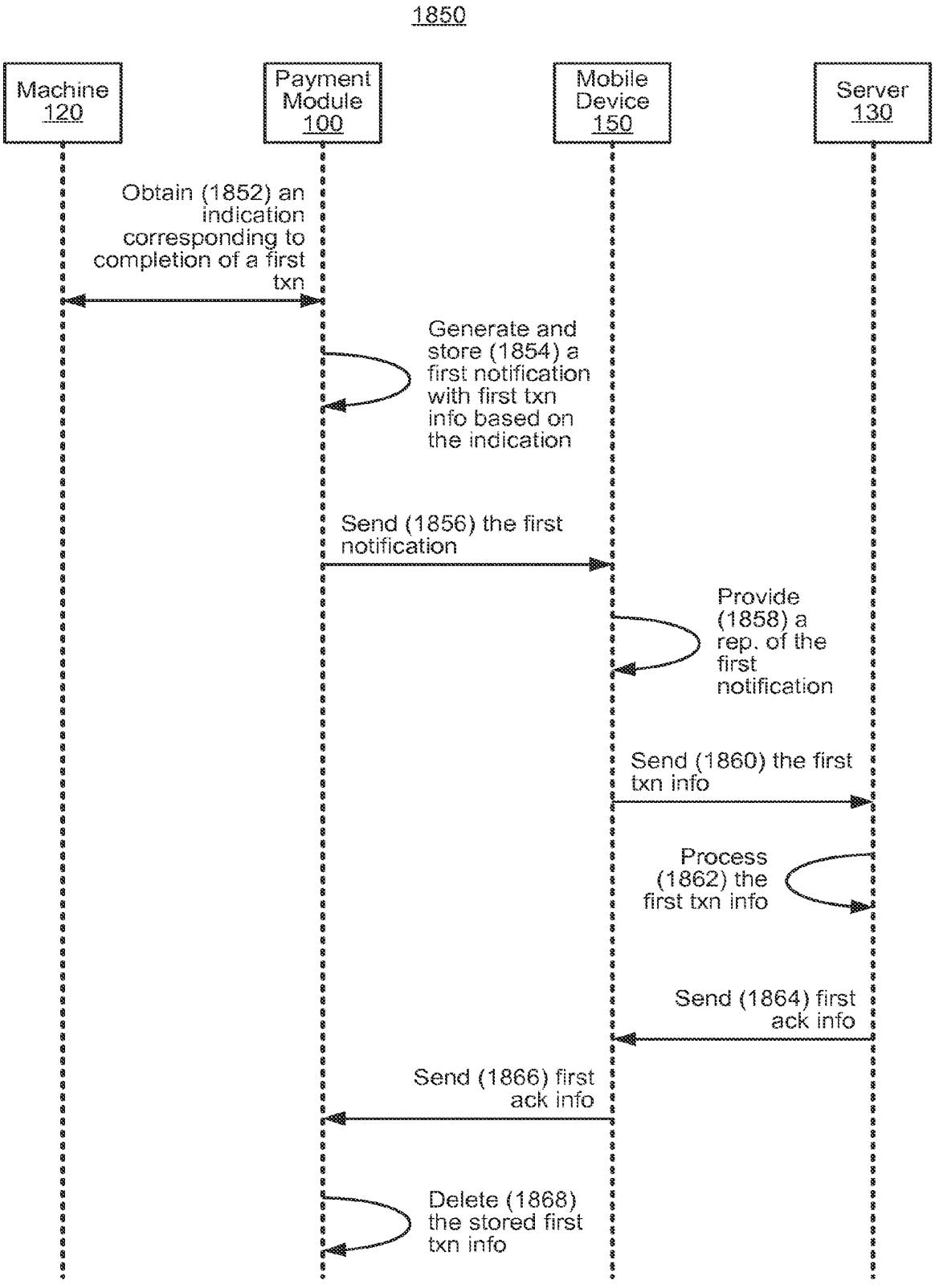

1850
Machine 120
Payment Module 100
Mobile Device 150
Server 130
Obtain (1852) an indication corresponding to completion of a first txn
Generate and store (1854) a first notification with first txn info based on the indication
Send (1856) the first notification
Provide (1858) a rep. of the first notification
Send (1860) the first txn info
Process (1862) the first txn info
Send (1864) first ack info
Send (1866) first ack info
Delete (1868) the stored first txn info

Figure 31B

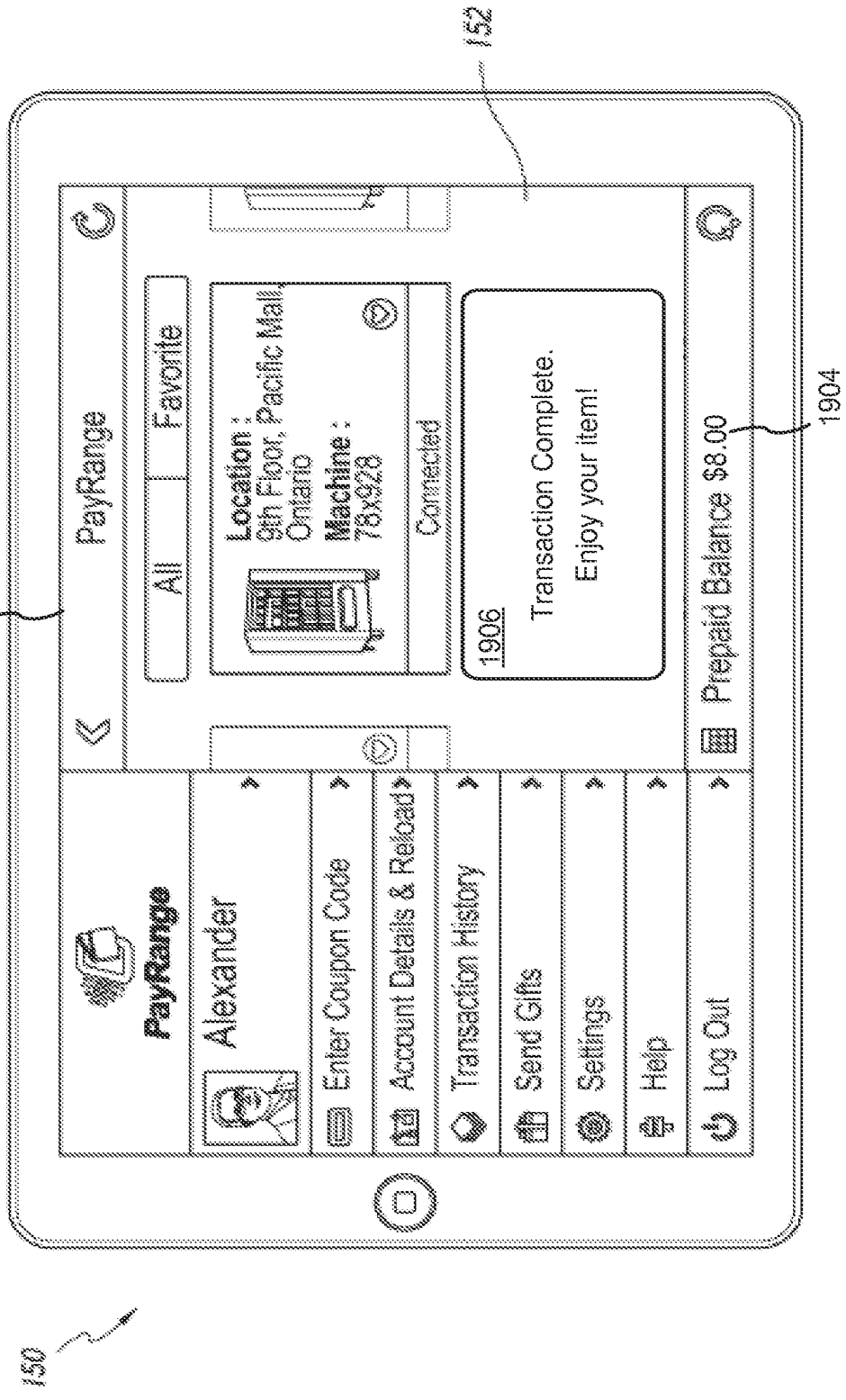

150
152
1902
1904
1906
PayRange
Alexander
Enter Coupon Code
Account Details & Reload
Transaction History
Send Gifts
Settings
Help
Log Out
PayRange
All
Favorite
Location :
9th Floor, Pacific Mall,
Ontario
Machine :
78x928
Connected
Transaction Complete.
Enjoy your item!
Prepaid Balance $8.00

Figure 32A

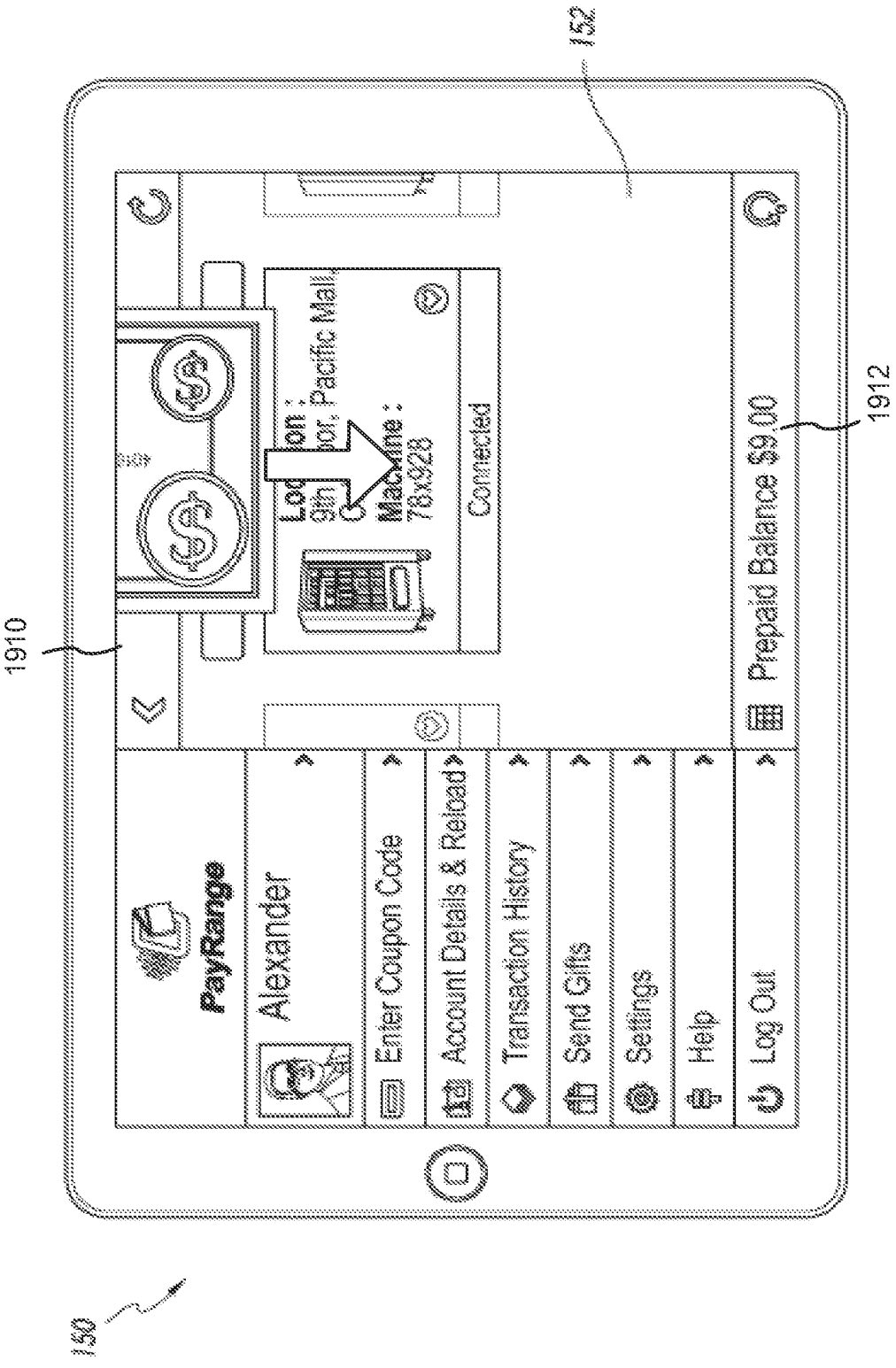

150
1910
152
1912
PayRange
Alexander
Enter Coupon Code
Account Details & Reload
Transaction History
Send Gifts
Settings
Help
Log Out
Prepaid Balance $9.00
Connected
Log
9th
Pacific Mall
Machine :
78x928

Figure 32B

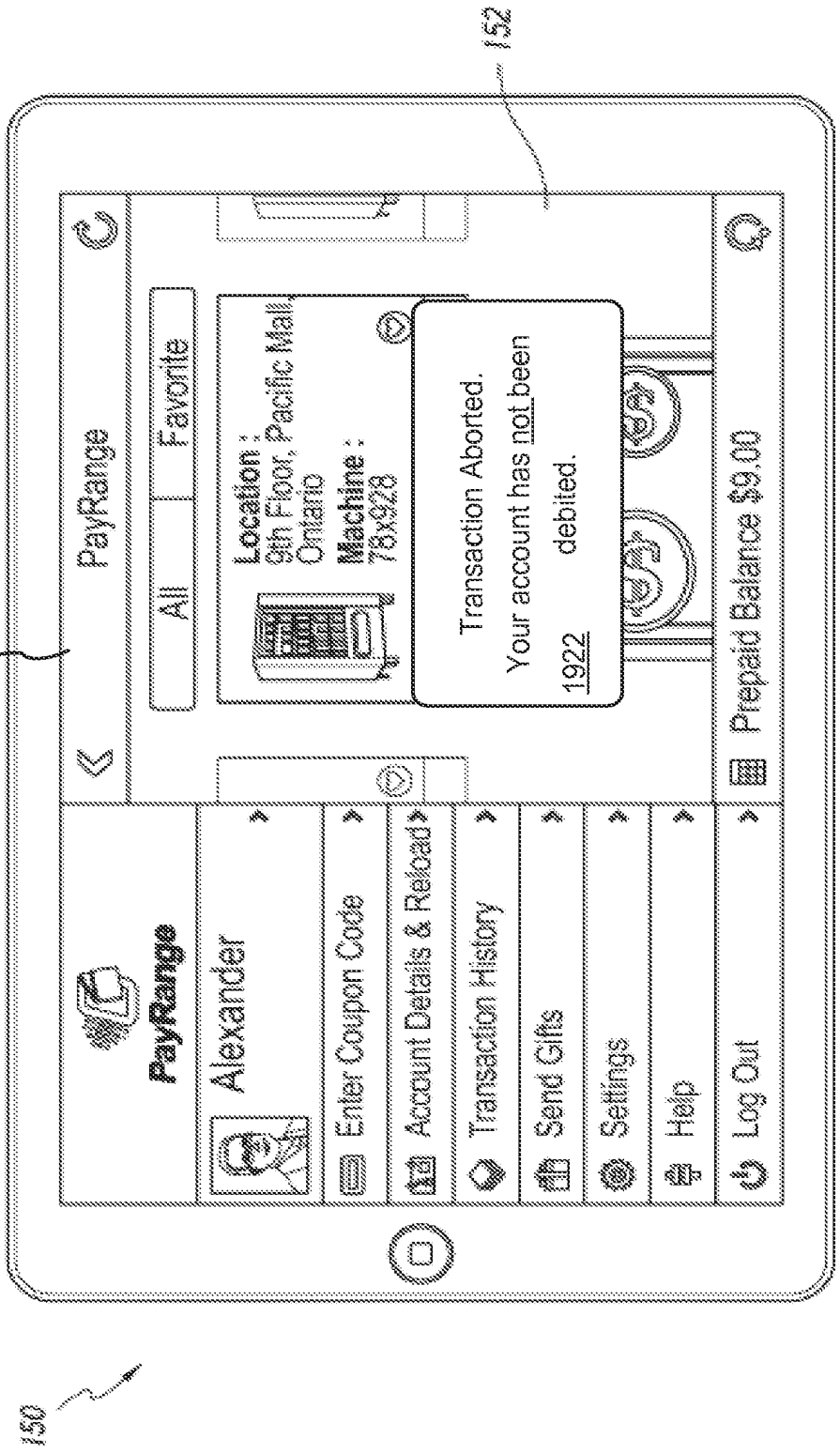

150
1920
152
PayRange
Alexander
Enter Coupon Code
Account Details & Reload
Transaction History
Send Gifts
Settings
Help
Log Out
PayRange
All
Favorite
Location :
5th Floor, Pacific Mall,
Ontario
Machine :
78x928
Transaction Aborted.
Your account has not been
debited.
1922
Prepaid Balance $9.00

Figure 32C

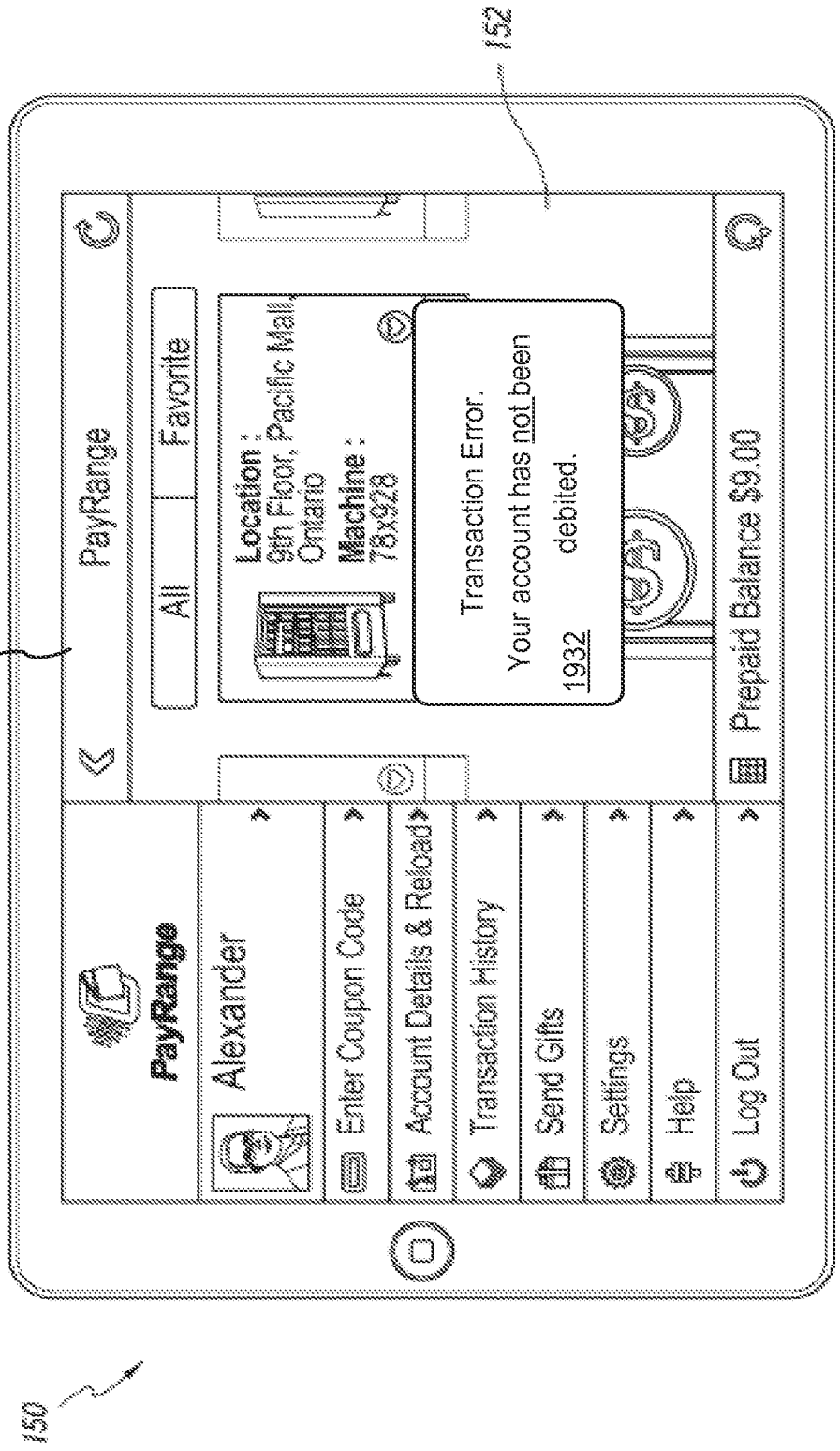

150
1930
152
PayRange
Alexander
Enter Coupon Code
Account Details & Reload
Transaction History
Send Gifts
Settings
Help
Log Out
PayRange
All
Favorite
Location :
5th Floor, Pacific Mall,
Ontario
Machine :
78x928
Transaction Error.
Your account has not been
debited.
1932
Prepaid Balance $9.00

Figure 32D

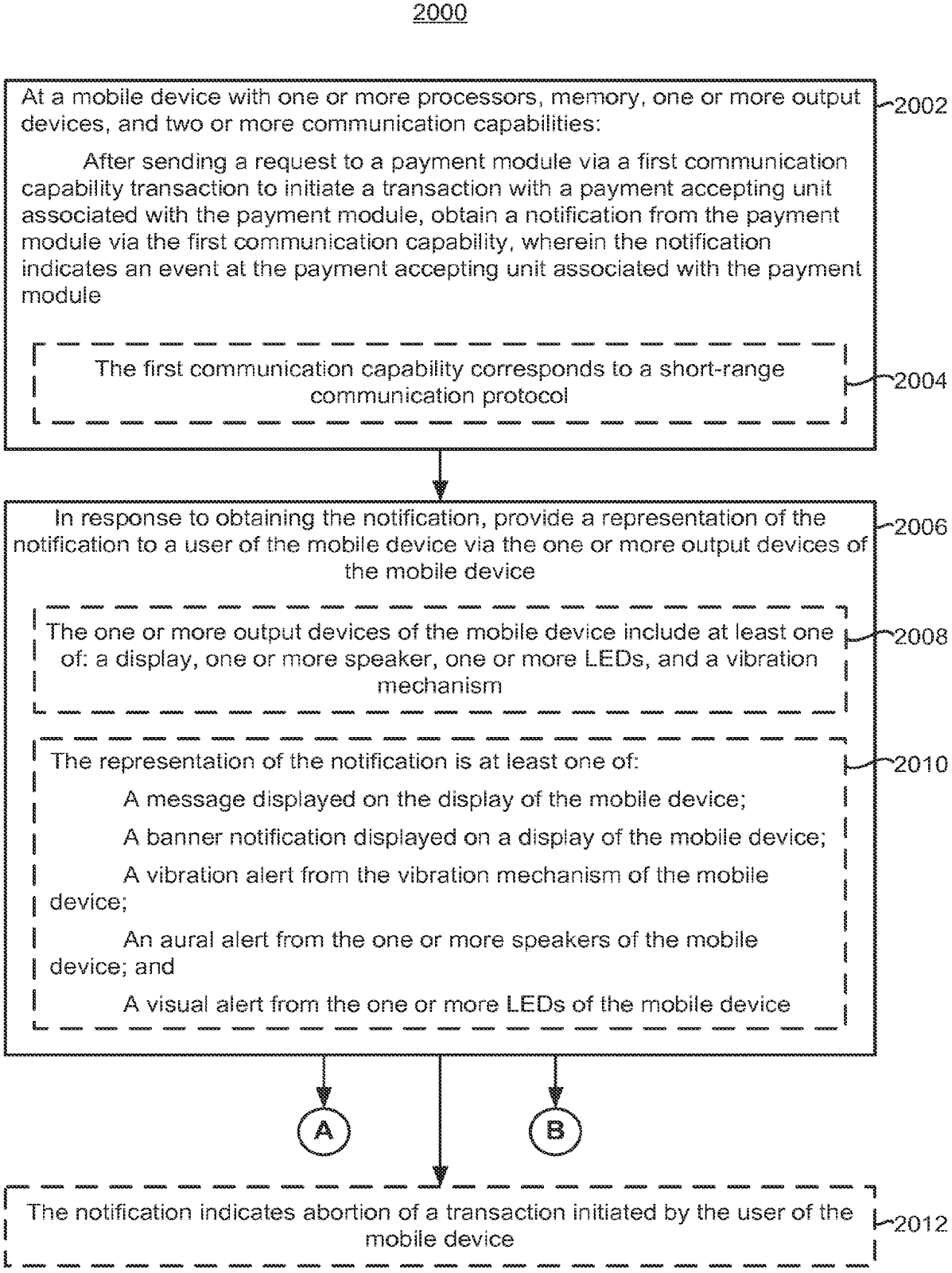

2000
At a mobile device with one or more processors, memory, one or more output devices, and two or more communication capabilities:
After sending a request to a payment module via a first communication capability transaction to initiate a transaction with a payment accepting unit associated with the payment module, obtain a notification from the payment module via the first communication capability, wherein the notification indicates an event at the payment accepting unit associated with the payment module
2002
The first communication capability corresponds to a short-range communication protocol
2004
In response to obtaining the notification, provide a representation of the notification to a user of the mobile device via the one or more output devices of the mobile device
2006
The one or more output devices of the mobile device include at least one of: a display, one or more speaker, one or more LEDs, and a vibration mechanism
2008
The representation of the notification is at least one of:
A message displayed on the display of the mobile device;
A banner notification displayed on a display of the mobile device;
A vibration alert from the vibration mechanism of the mobile device;
An aural alert from the one or more speakers of the mobile device; and
A visual alert from the one or more LEDs of the mobile device
2010
A
B
The notification indicates abortion of a transaction initiated by the user of the mobile device
2012

Figure 33A

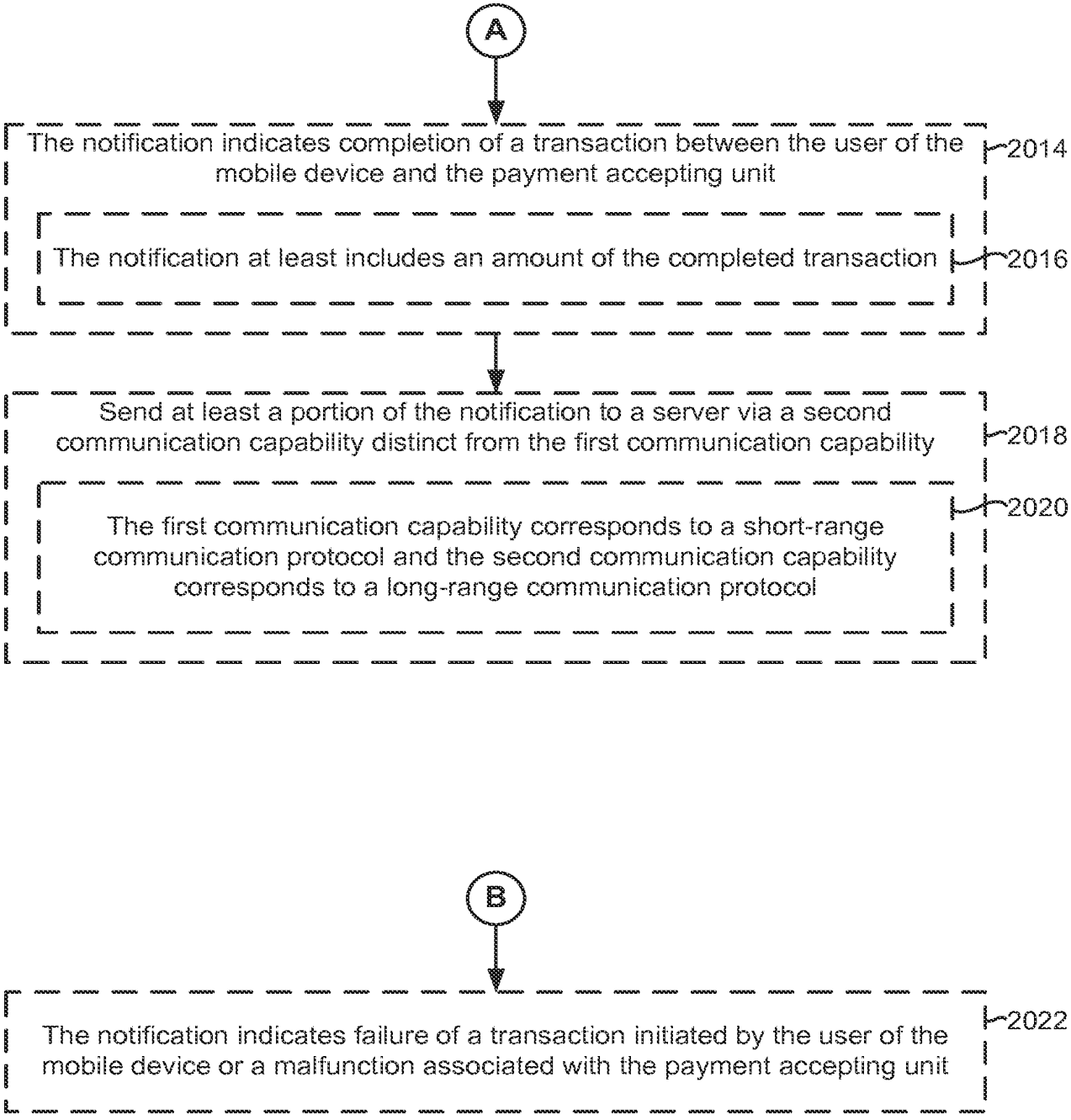

A
The notification indicates completion of a transaction between the user of the mobile device and the payment accepting unit
2014
The notification at least includes an amount of the completed transaction
2016
Send at least a portion of the notification to a server via a second communication capability distinct from the first communication capability
2018
The first communication capability corresponds to a short-range communication protocol and the second communication capability corresponds to a long-range communication protocol
2020
B
The notification indicates failure of a transaction initiated by the user of the mobile device or a malfunction associated with the payment accepting unit
2022

Figure 33B

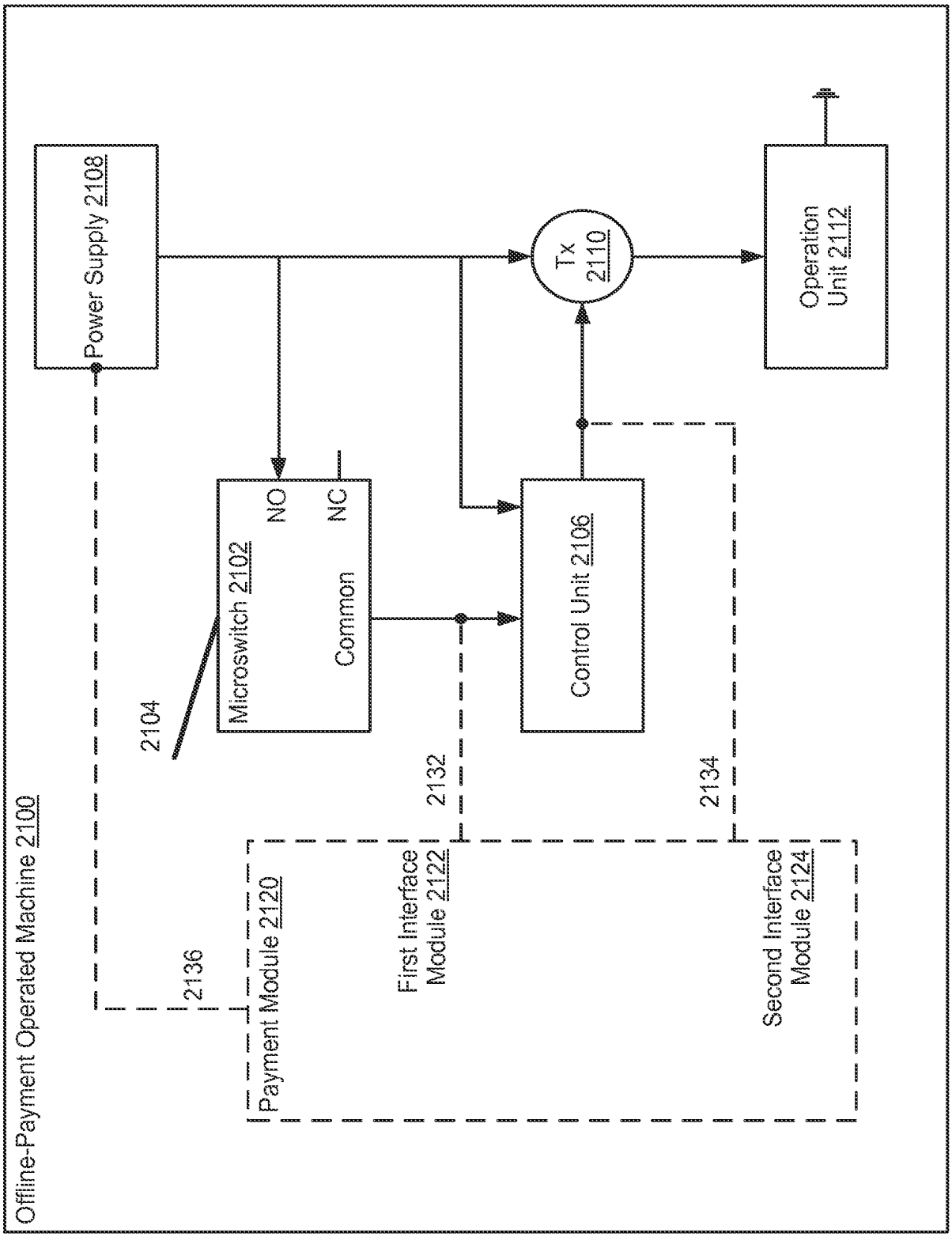

Offline-Payment Operated Machine 2100
Power Supply 2108
Tx 2110
Operation Unit 2112
2104
Microswitch 2102
NO
NC
Common
Control Unit 2106
2136
Payment Module 2120
2132
First Interface Module 2122
2134
Second Interface Module 2124

Figure 34A

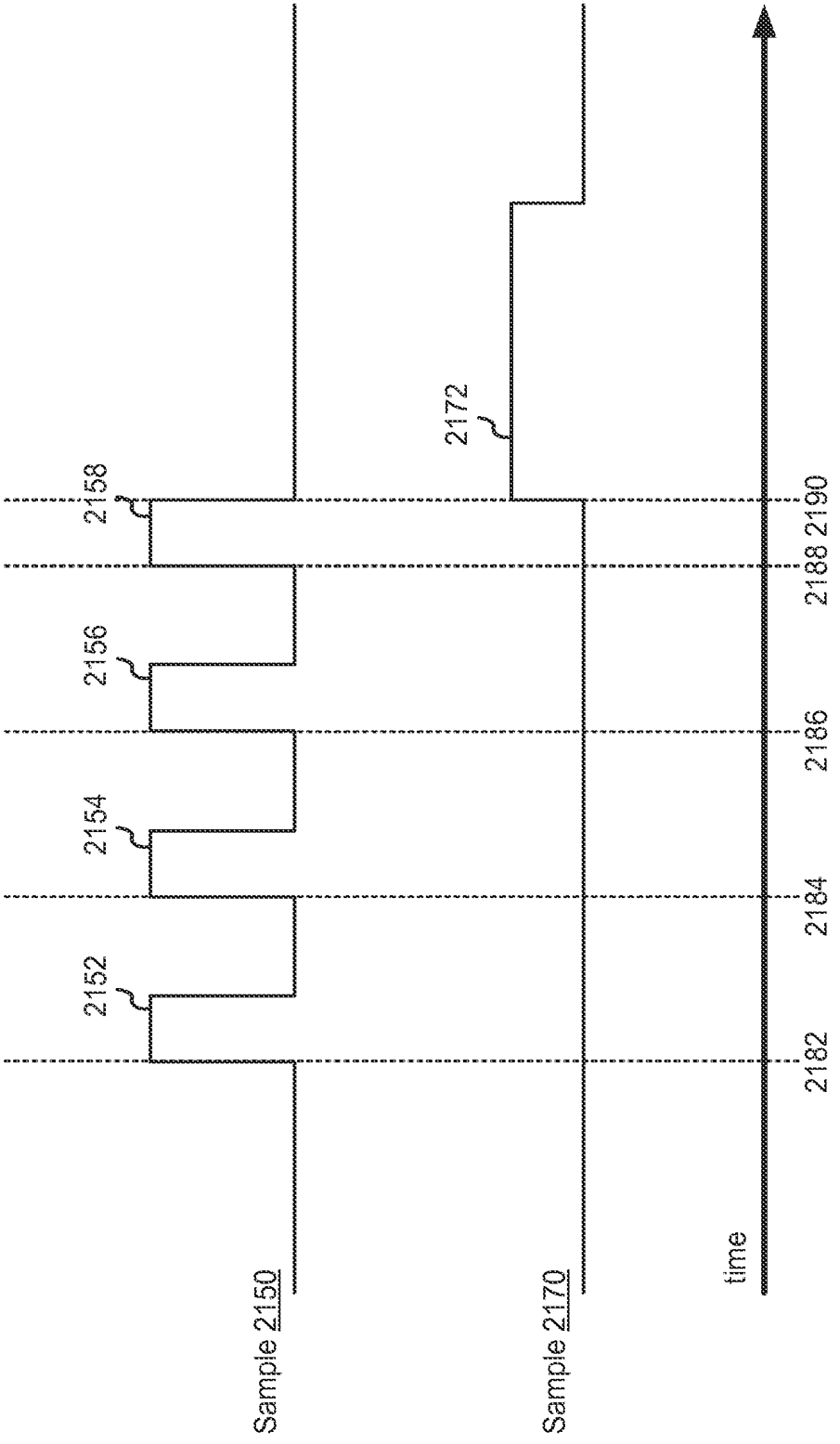

2158
2156
2154
2152
2172
2182
2184
2186
2188 2190
time
Sample 2150
Sample 2170

Figure 34B

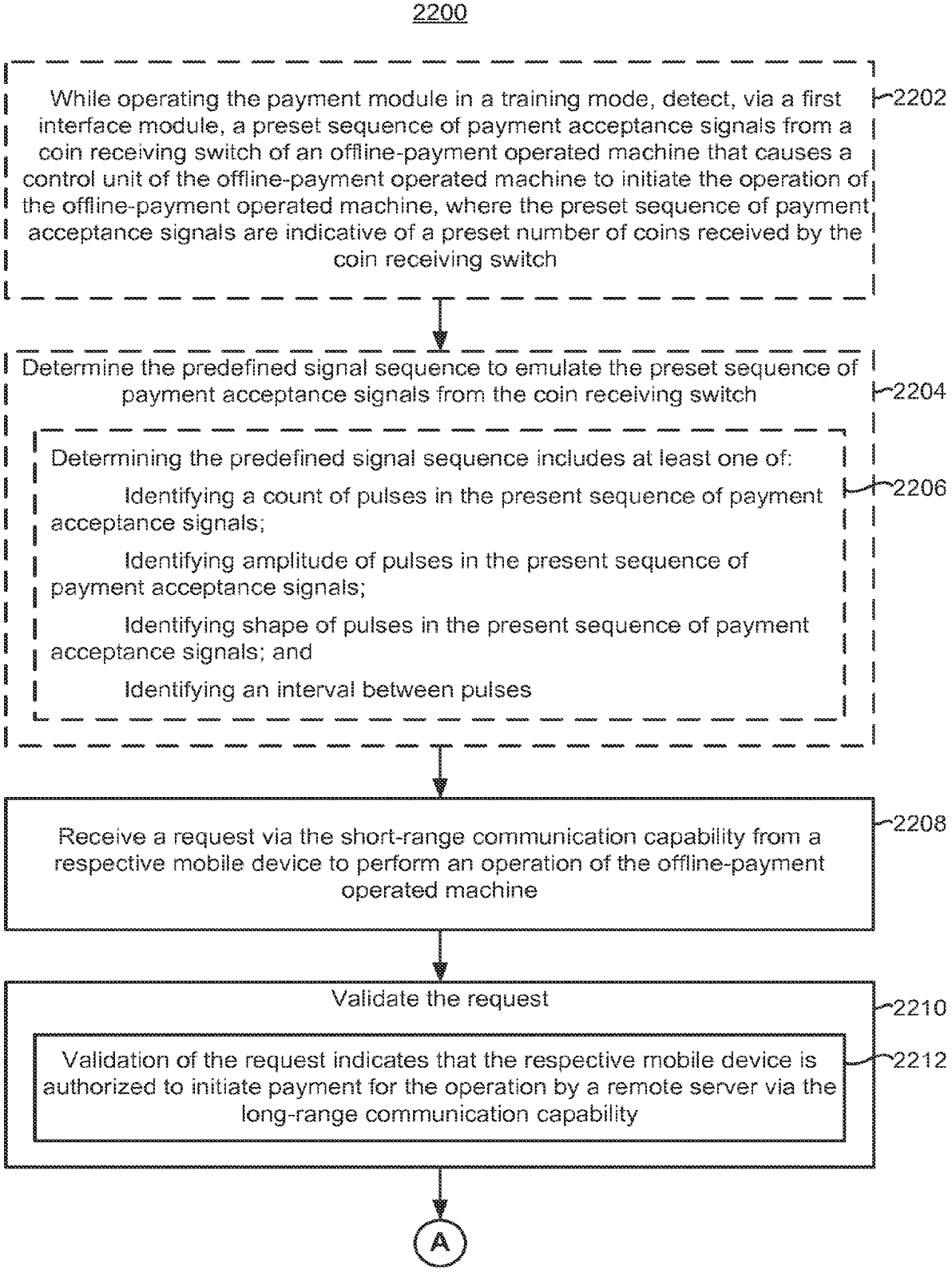

2200
While operating the payment module in a training mode, detect, via a first interface module, a preset sequence of payment acceptance signals from a coin receiving switch of an offline-payment operated machine that causes a control unit of the offline-payment operated machine to initiate the operation of the offline-payment operated machine, where the preset sequence of payment acceptance signals are indicative of a preset number of coins received by the coin receiving switch
2202
Determine the predefined signal sequence to emulate the preset sequence of payment acceptance signals from the coin receiving switch
2204
Determining the predefined signal sequence includes at least one of:
Identifying a count of pulses in the present sequence of payment acceptance signals;
Identifying amplitude of pulses in the present sequence of payment acceptance signals;
Identifying shape of pulses in the present sequence of payment acceptance signals; and
Identifying an interval between pulses
2206
Receive a request via the short-range communication capability from a respective mobile device to perform an operation of the offline-payment operated machine
2208
Validate the request
2210
Validation of the request indicates that the respective mobile device is authorized to initiate payment for the operation by a remote server via the long-range communication capability
2212
A

Figure 35A

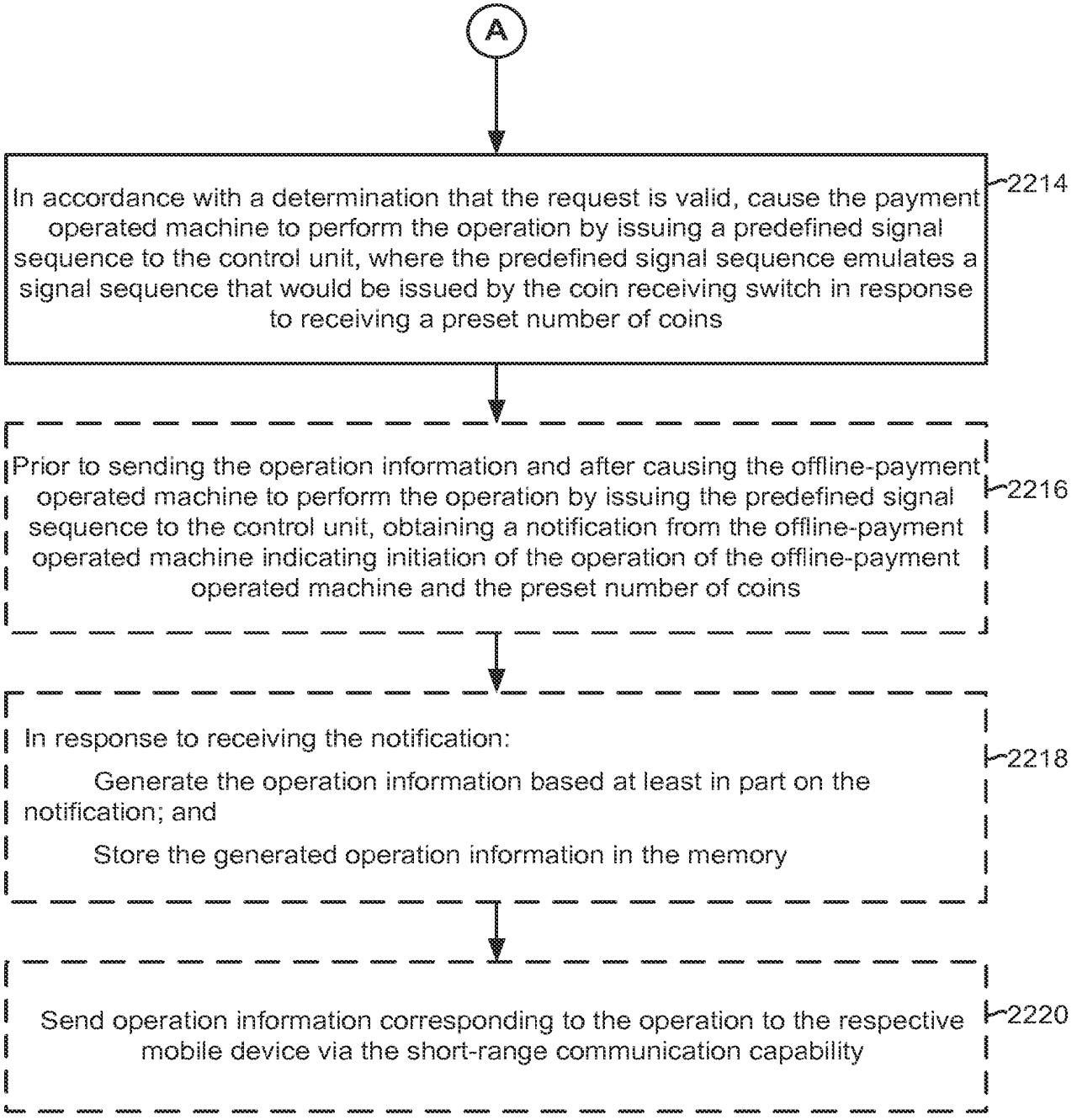

A
In accordance with a determination that the request is valid, cause the payment operated machine to perform the operation by issuing a predefined signal sequence to the control unit, where the predefined signal sequence emulates a signal sequence that would be issued by the coin receiving switch in response to receiving a preset number of coins
2214
Prior to sending the operation information and after causing the offline-payment operated machine to perform the operation by issuing the predefined signal sequence to the control unit, obtaining a notification from the offline-payment operated machine indicating initiation of the operation of the offline-payment operated machine and the preset number of coins
2216
In response to receiving the notification:
Generate the operation information based at least in part on the notification; and
Store the generated operation information in the memory
2218
Send operation information corresponding to the operation to the respective mobile device via the short-range communication capability
2220

Figure 35B

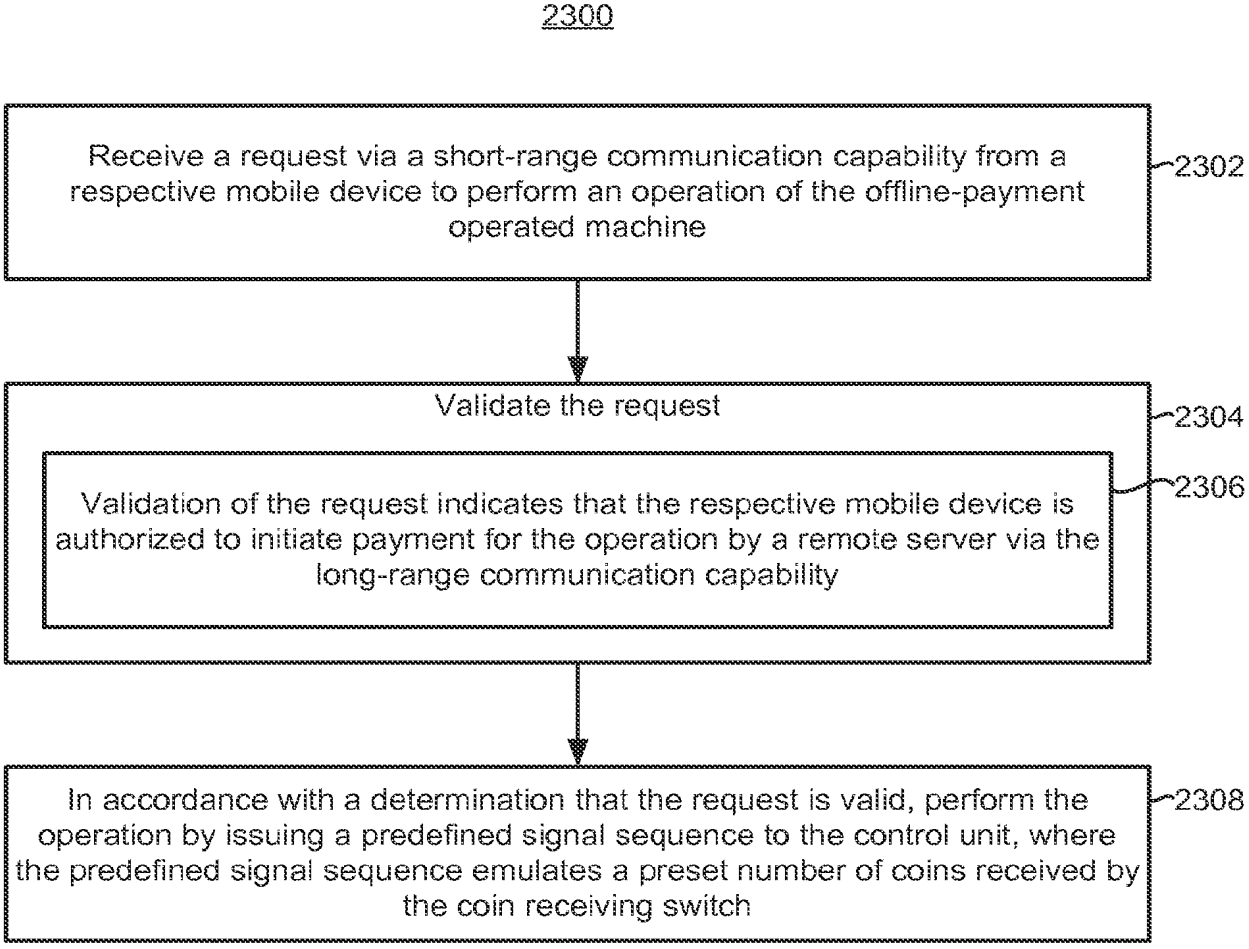

2300
Receive a request via a short-range communication capability from a respective mobile device to perform an operation of the offline-payment operated machine
2302
Validate the request
2304
Validation of the request indicates that the respective mobile device is authorized to initiate payment for the operation by a remote server via the long-range communication capability
2306
In accordance with a determination that the request is valid, perform the operation by issuing a predefined signal sequence to the control unit, where the predefined signal sequence emulates a preset number of coins received by the coin receiving switch
2308

Figure 36

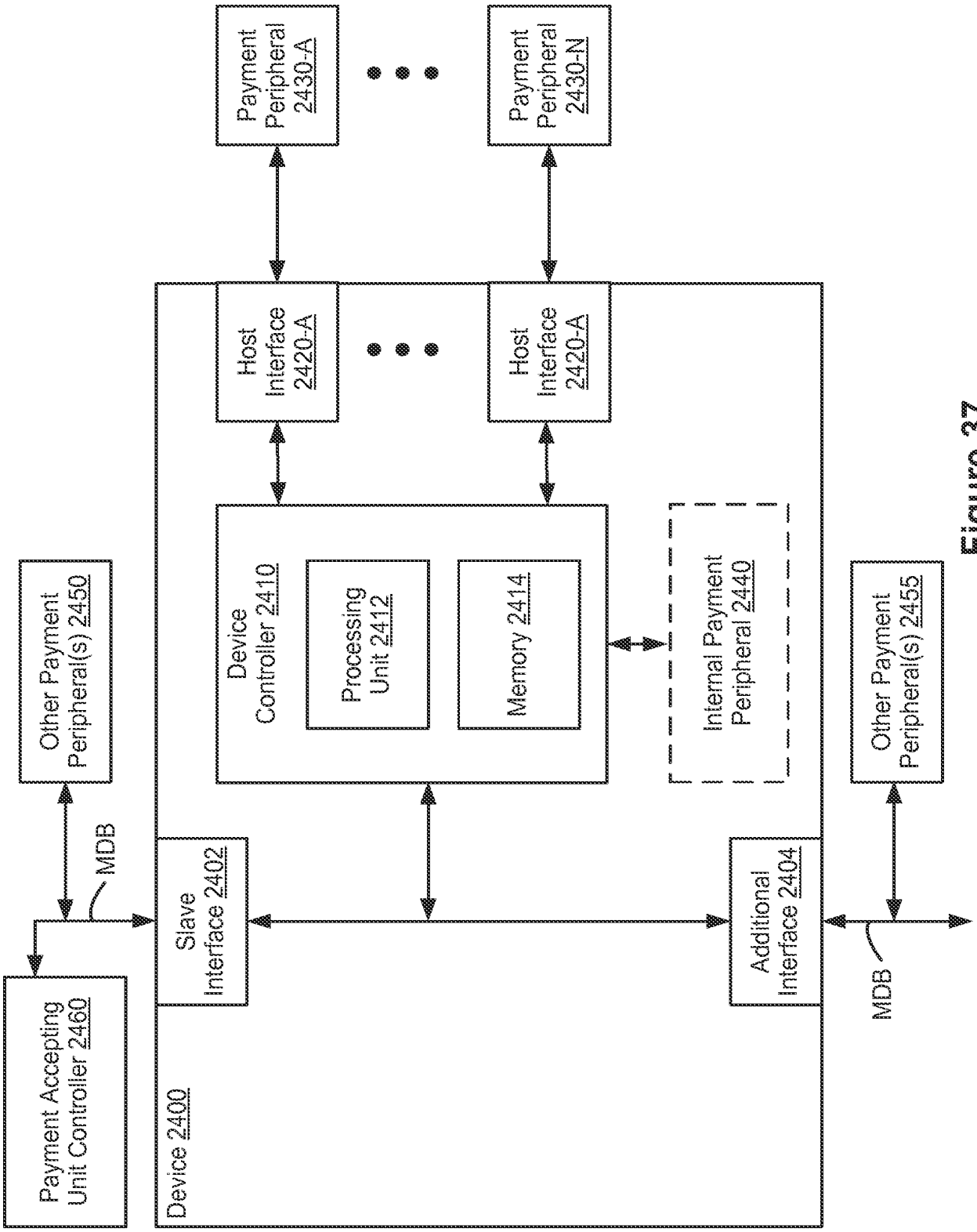

Payment Peripheral 2430-A
Payment Peripheral 2430-N
Host Interface 2420-A
Host Interface 2420-A
Device Controller 2410
Processing Unit 2412
Memory 2414
Internal Payment Peripheral 2440
Other Payment Peripheral(s) 2450
Other Payment Peripheral(s) 2455
MDB
MDB
Slave Interface 2402
Additional Interface 2404
Payment Accepting Unit Controller 2460
Device 2400

Figure 37

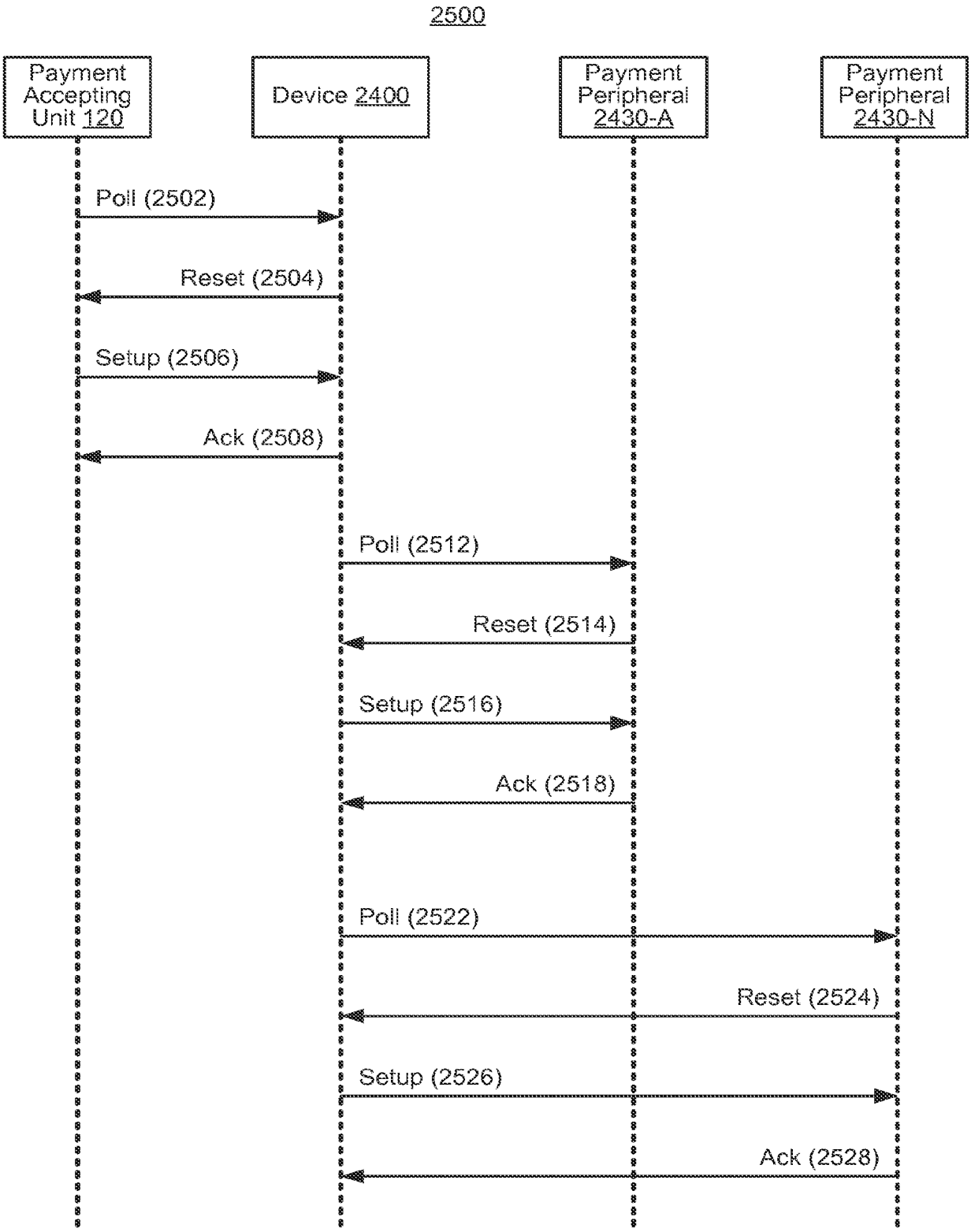

2500
Payment Accepting Unit 120
Device 2400
Payment Peripheral 2430-A
Payment Peripheral 2430-N
Poll (2502)
Reset (2504)
Setup (2506)
Ack (2508)
Poll (2512)
Reset (2514)
Setup (2516)
Ack (2518)
Poll (2522)
Reset (2524)
Setup (2526)
Ack (2528)

Figure 38

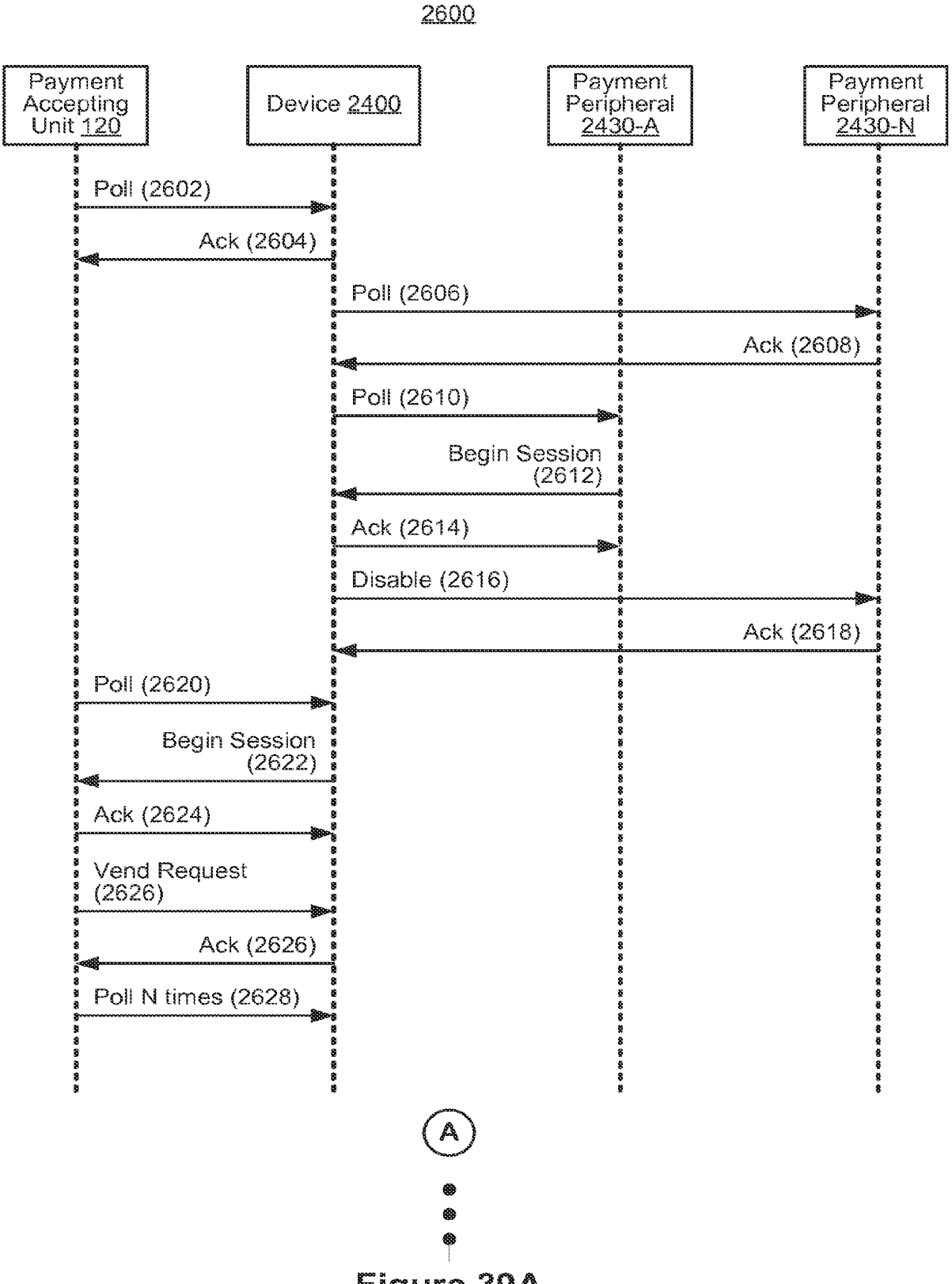

2600
Payment Accepting Unit 120
Device 2400
Payment Peripheral 2430-A
Payment Peripheral 2430-N
Poll (2602)
Ack (2604)
Poll (2606)
Ack (2608)
Poll (2610)
Begin Session (2612)
Ack (2614)
Disable (2616)
Ack (2618)
Poll (2620)
Begin Session (2622)
Ack (2624)
Vend Request (2626)
Ack (2626)
Poll N times (2628)
A

Figure 39A

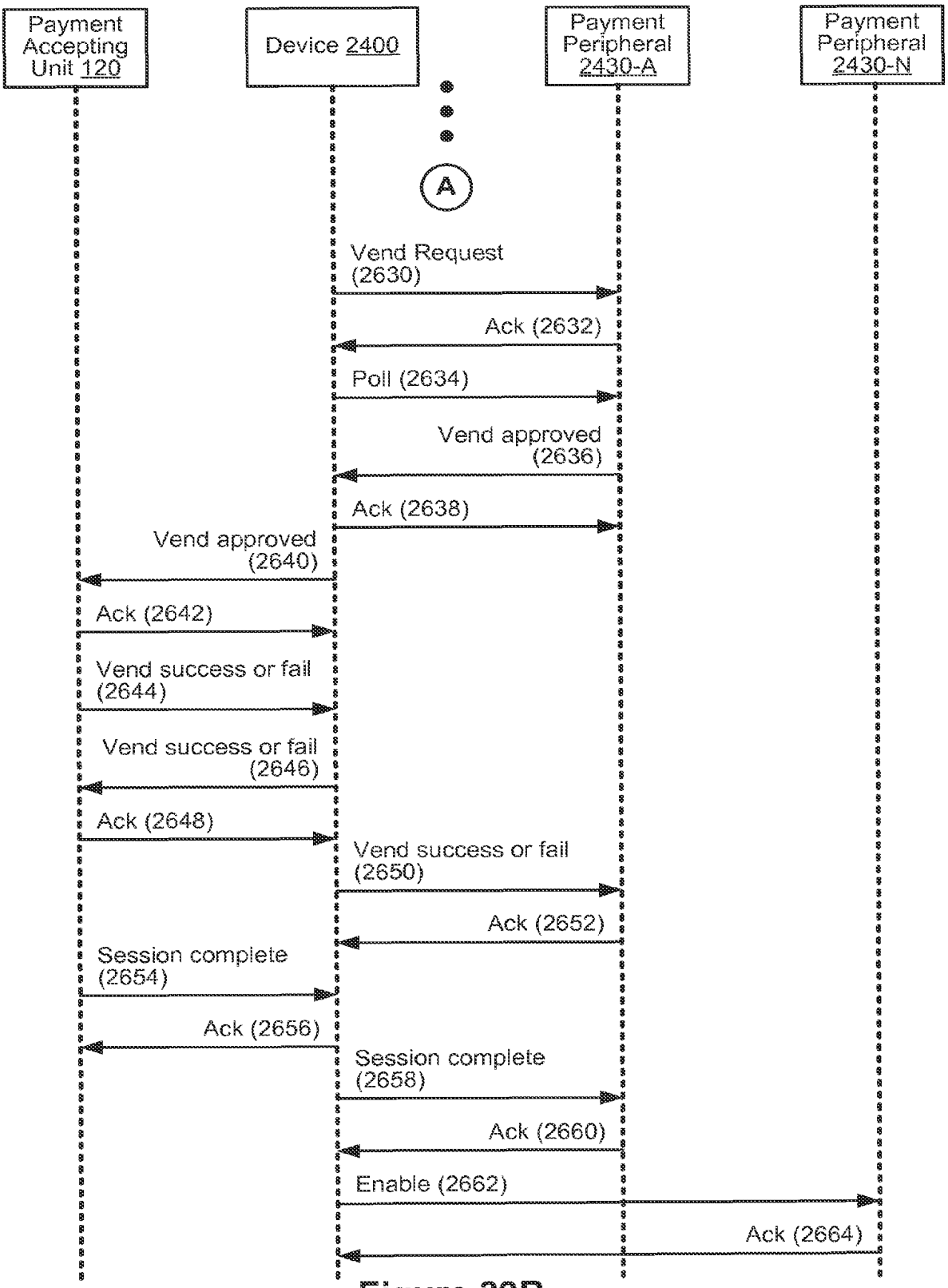

Payment Accepting Unit 120
Device 2400
Payment Peripheral 2430-A
Payment Peripheral 2430-N
A
Vend Request (2630)
Ack (2632)
Poll (2634)
Vend approved (2636)
Ack (2638)
Vend approved (2640)
Ack (2642)
Vend success or fail (2644)
Vend success or fail (2646)
Ack (2648)
Vend success or fail (2650)
Ack (2652)
Session complete (2654)
Ack (2656)
Session complete (2658)
Ack (2660)
Enable (2662)
Ack (2664)

Figure 39B

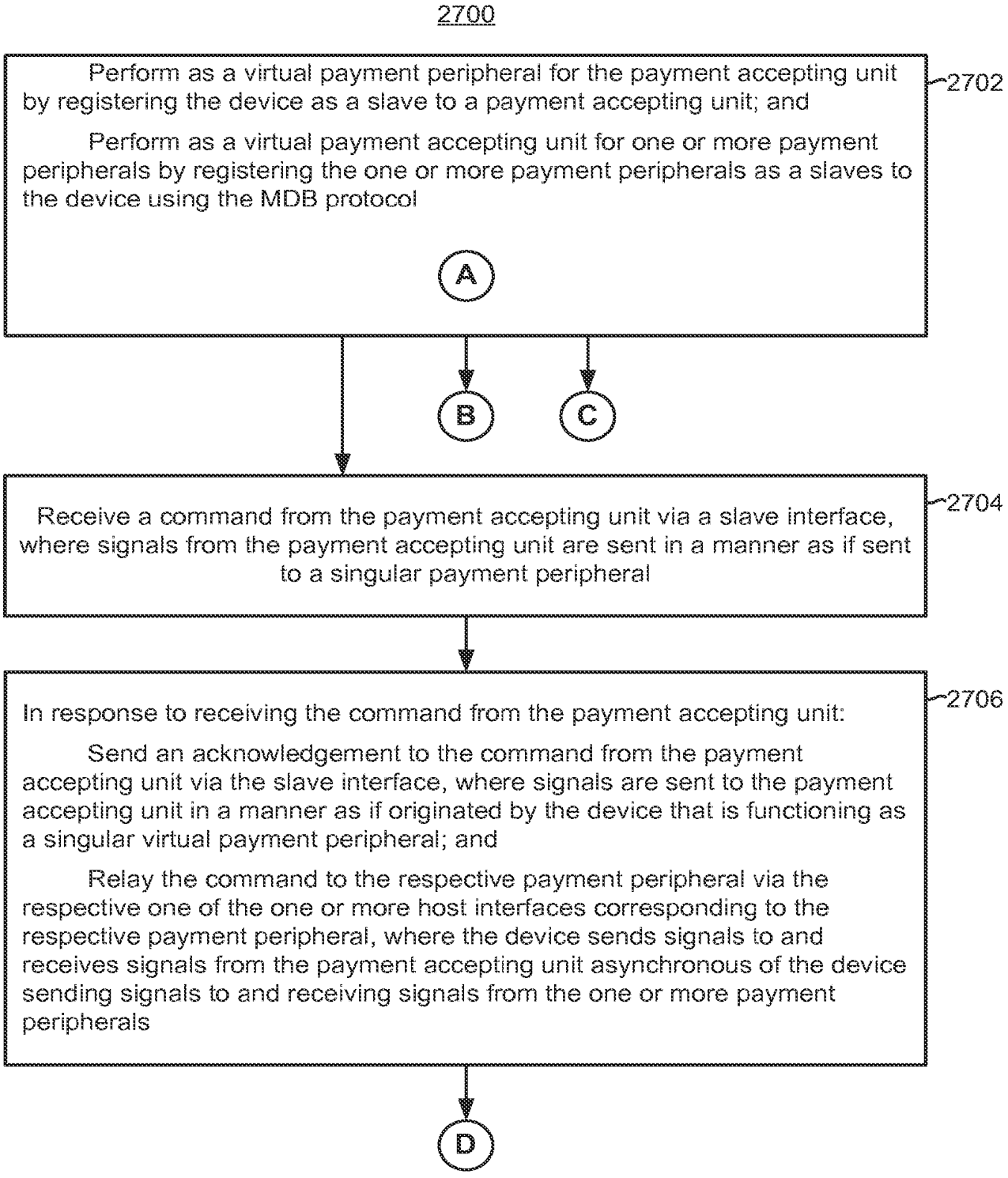

2700
Perform as a virtual payment peripheral for the payment accepting unit by registering the device as a slave to a payment accepting unit; and
Perform as a virtual payment accepting unit for one or more payment peripherals by registering the one or more payment peripherals as a slaves to the device using the MDB protocol
2702
A
B
C
Receive a command from the payment accepting unit via a slave interface, where signals from the payment accepting unit are sent in a manner as if sent to a singular payment peripheral
2704
In response to receiving the command from the payment accepting unit:
Send an acknowledgement to the command from the payment accepting unit via the slave interface, where signals are sent to the payment accepting unit in a manner as if originated by the device that is functioning as a singular virtual payment peripheral; and
Relay the command to the respective payment peripheral via the respective one of the one or more host interfaces corresponding to the respective payment peripheral, where the device sends signals to and receives signals from the payment accepting unit asynchronous of the device sending signals to and receiving signals from the one or more payment peripherals
2706
D

Figure 40A

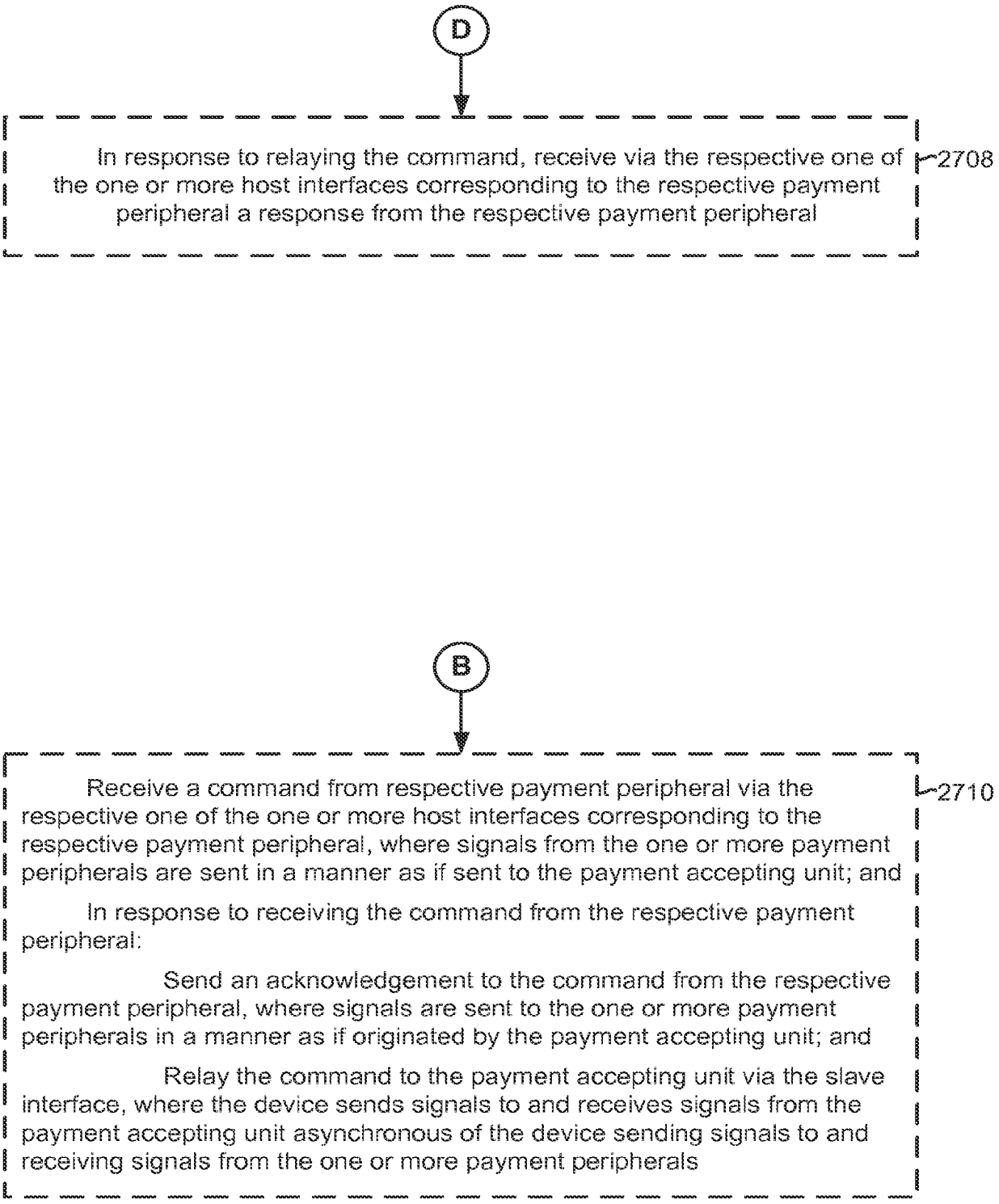

D
In response to relaying the command, receive via the respective one of the one or more host interfaces corresponding to the respective payment peripheral a response from the respective payment peripheral
2708
B
Receive a command from respective payment peripheral via the respective one of the one or more host interfaces corresponding to the respective payment peripheral, where signals from the one or more payment peripherals are sent in a manner as if sent to the payment accepting unit; and
In response to receiving the command from the respective payment peripheral:
Send an acknowledgement to the command from the respective payment peripheral, where signals are sent to the one or more payment peripherals in a manner as if originated by the payment accepting unit; and
Relay the command to the payment accepting unit via the slave interface, where the device sends signals to and receives signals from the payment accepting unit asynchronous of the device sending signals to and receiving signals from the one or more payment peripherals
2710

Figure 40B

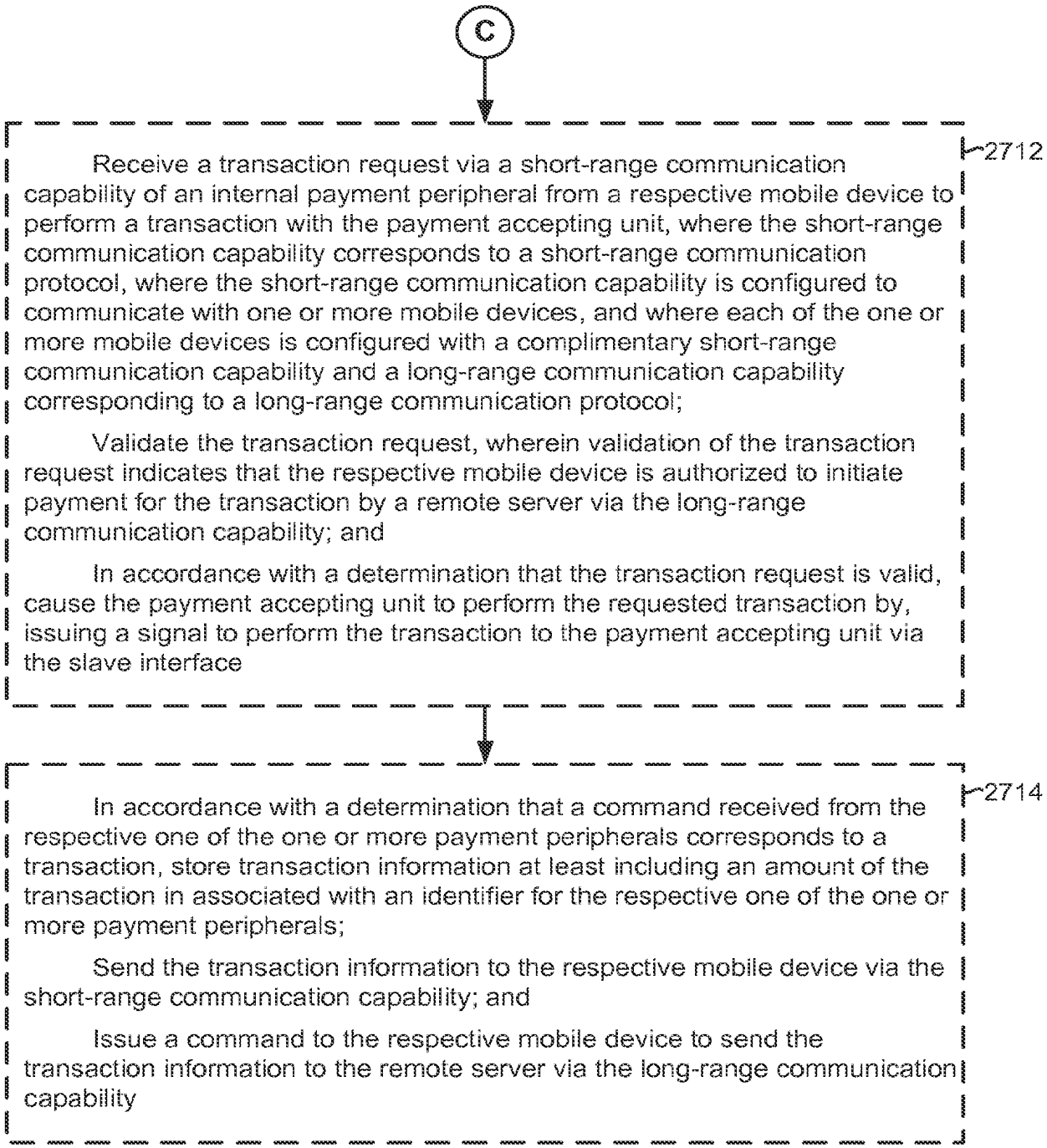

C
2712
Receive a transaction request via a short-range communication capability of an internal payment peripheral from a respective mobile device to perform a transaction with the payment accepting unit, where the short-range communication capability corresponds to a short-range communication protocol, where the short-range communication capability is configured to communicate with one or more mobile devices, and where each of the one or more mobile devices is configured with a complimentary short-range communication capability and a long-range communication capability corresponding to a long-range communication protocol;
Validate the transaction request, wherein validation of the transaction request indicates that the respective mobile device is authorized to initiate payment for the transaction by a remote server via the long-range communication capability; and
In accordance with a determination that the transaction request is valid, cause the payment accepting unit to perform the requested transaction by, issuing a signal to perform the transaction to the payment accepting unit via the slave interface
2714
In accordance with a determination that a command received from the respective one of the one or more payment peripherals corresponds to a transaction, store transaction information at least including an amount of the transaction in associated with an identifier for the respective one of the one or more payment peripherals;
Send the transaction information to the respective mobile device via the short-range communication capability; and
Issue a command to the respective mobile device to send the transaction information to the remote server via the long-range communication capability

Figure 40C

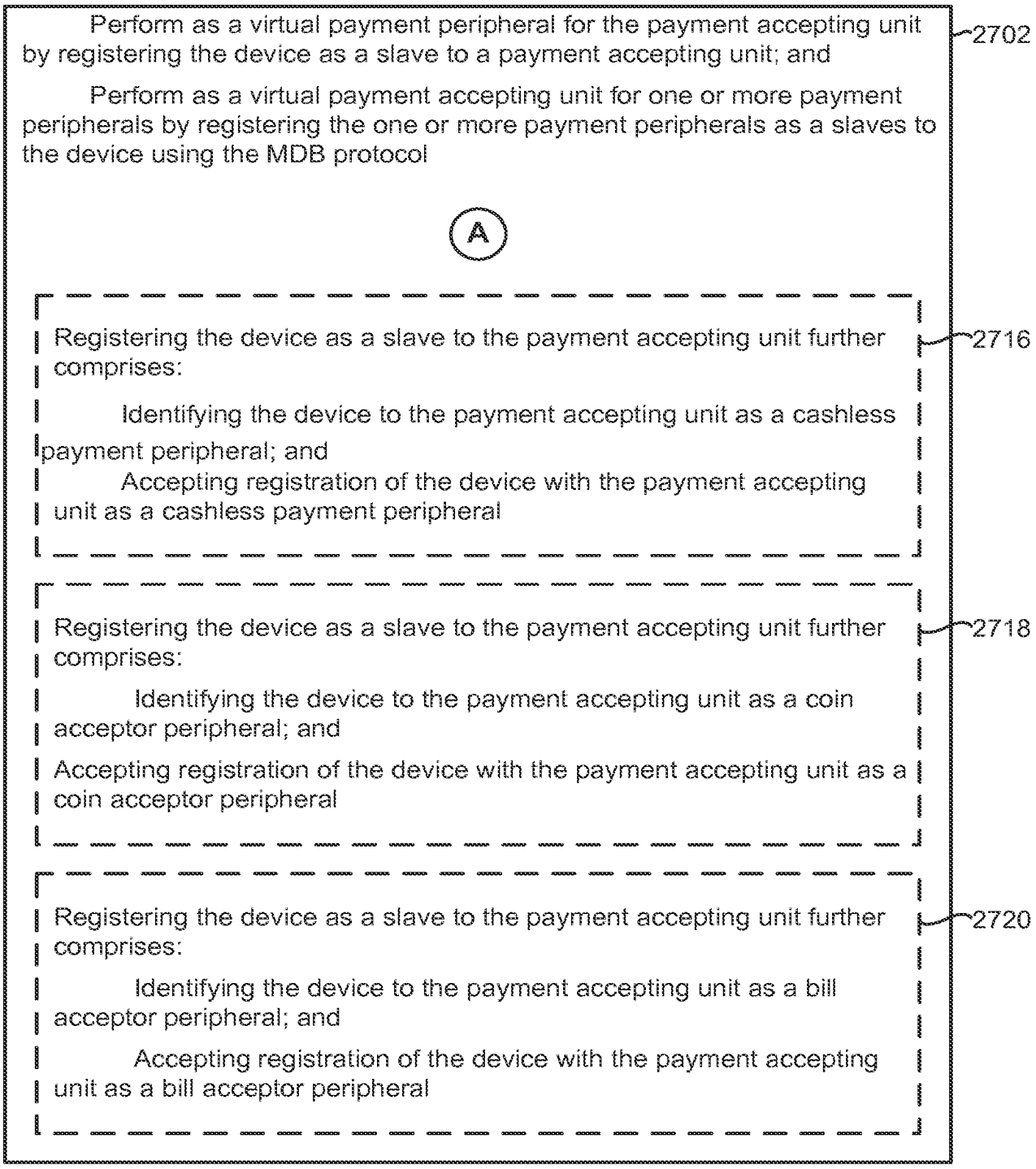

Perform as a virtual payment peripheral for the payment accepting unit by registering the device as a slave to a payment accepting unit; and
Perform as a virtual payment accepting unit for one or more payment peripherals by registering the one or more payment peripherals as a slaves to the device using the MDB protocol
2702
A
Registering the device as a slave to the payment accepting unit further comprises:
Identifying the device to the payment accepting unit as a cashless payment peripheral; and
Accepting registration of the device with the payment accepting unit as a cashless payment peripheral
2716
Registering the device as a slave to the payment accepting unit further comprises:
Identifying the device to the payment accepting unit as a coin acceptor peripheral; and
Accepting registration of the device with the payment accepting unit as a coin acceptor peripheral
2718
Registering the device as a slave to the payment accepting unit further comprises:
Identifying the device to the payment accepting unit as a bill acceptor peripheral; and
Accepting registration of the device with the payment accepting unit as a bill acceptor peripheral
2720

Figure 40D

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 21 16 5692

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	US 7 690 495 B1 (KOLLS H BROCK [US] ET AL) 6 April 2010 (2010-04-06) * column 4, line 35 - column 115, line 10 * * figures * -----	1-13	INV. G06Q20/32 G06Q20/36 G06Q20/40 G06Q30/06 G06Q20/18 G06Q20/38 G07F9/00 G07F9/02
A	US 2013/267176 A1 (HERTEL PHILIPP [US] ET AL) 10 October 2013 (2013-10-10) * paragraph [0015] - paragraph [0063] * * figures * -----	1-13	
A	US 2011/251910 A1 (DIMMICK JAMES [US]) 13 October 2011 (2011-10-13) * page 3, paragraph 32 - page 4, paragraph 48 * * page 4, paragraph 50 - page 5, paragraph 53 * * page 5, paragraph 55 - page 6, paragraph 62 * * page 6, paragraph 68-69 * * page 7, paragraph 73 - page 8, paragraph 86 * * figure 1 * -----	1-13	TECHNICAL FIELDS SEARCHED (IPC) G06Q G07F
A	WO 2013/132995 A1 (SONY CORP [JP]) 12 September 2013 (2013-09-12) * page 7, paragraph 18 - page 36, paragraph 122 * * figure 1 * -----	1-13	
A	US 2005/101295 A1 (RUPP STEPHAN [DE] ET AL) 12 May 2005 (2005-05-12) * page 2, paragraph 19 - page 4, paragraph 37 * * figures * -----	1-13	
	-/--		

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	6 September 2021	Rachkov, Vassil

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 21 16 5692

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	US 8 438 066 B1 (YUEN BILLY [US] ET AL) 7 May 2013 (2013-05-07) * column 3, line 40 - column 12, line 8 * * figures 1, 3A *	1-13	
A	US 2012/316963 A1 (MOSHFEGHI MEHRAN [US]) 13 December 2012 (2012-12-13) * page 2, paragraph 22 - page 15, paragraph 127 *	1-13	
A	US 2012/108173 A1 (HAHM SEONG-IL [KR] ET AL) 3 May 2012 (2012-05-03) * paragraph [0030] - paragraph [0108] * * figures *	1-13	
			TECHNICAL FIELDS SEARCHED (IPC)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	6 September 2021	Rachkov, Vassil

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 21 16 5692

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

06-09-2021

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7690495 B1	06-04-2010	NONE	
US 2013267176 A1	10-10-2013	AU 2013246053 B2	30-10-2014
		CA 2869917 A1	17-10-2013
		EP 2837104 A1	18-02-2015
		KR 20140140642 A	09-12-2014
		US 2013267176 A1	10-10-2013
		US 2014323051 A1	30-10-2014
		WO 2013155122 A1	17-10-2013
US 2011251910 A1	13-10-2011	NONE	
WO 2013132995 A1	12-09-2013	CN 104145284 A	12-11-2014
		EP 2824629 A1	14-01-2015
		JP 5935871 B2	15-06-2016
		JP WO2013132995 A1	30-07-2015
		US 2015073994 A1	12-03-2015
		WO 2013132995 A1	12-09-2013
US 2005101295 A1	12-05-2005	AT 339742 T	15-10-2006
		CN 1614641 A	11-05-2005
		DE 60308385 T2	20-09-2007
		EP 1530177 A1	11-05-2005
		US 2005101295 A1	12-05-2005
US 8438066 B1	07-05-2013	NONE	
US 2012316963 A1	13-12-2012	US 10467617 B1	05-11-2019
		US 2012316963 A1	13-12-2012
		US 2015058125 A1	26-02-2015
US 2012108173 A1	03-05-2012	AU 2011324229 A1	04-04-2013
		CN 103190093 A	03-07-2013
		EP 2636167 A1	11-09-2013
		JP 5937095 B2	22-06-2016
		JP 2014500656 A	09-01-2014
		KR 20120049957 A	18-05-2012
		US 2012108173 A1	03-05-2012
		US 2016014823 A1	14-01-2016
		WO 2012060646 A1	10-05-2012

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

Innovative VDI Standards: Moving an Industry Forward

Michael L. Kasavana, Ph.D., NAMA Professor in Hospitality Business,
Michigan State University, USA

ABSTRACT

Effective self-service automation requires data sharing among non-proprietary system components. Historically, the original self-service provider, the vending industry, has failed to develop and implement open architecture standards that enable seamless data sharing among disparate devices and servers in a network. This lack of interoperability among the main entities in a sophisticated vending system (e.g. vending machine controller, telemetry device, vending management software, and cashless payment equipment) has contributed to a lack of innovation and creativity in the evolution of unattended point-of-sale retailing. Recently crafted Vending Data Interchange (VDI) standards from the National Automatic Merchandising Association (NAMA) provide a non-proprietary means by which to share machine-level data among diverse technology providers. The VDI standards are designed to ensure reliability, continuity, and longevity among installed hardware, software, and netware. In essence, VDI standards contain technical specifications that bundle vending machine-level data for easy distribution throughout a vending operator's technology network and can be implemented by a qualified provider without operator involvement.

Keywords: *data interchange, vending, self-service technologies, unattended point-of-sale*

INTRODUCTION

The NAMA Technology Leadership Committee, under the direction of the NAMA Board of Directors, formed a specialized technology task force charged with developing data interchange standards that could be implemented industry-wide. The task force was directed to develop a set of data transport protocols enabling the sharing of vending machine data among competing back-end technology providers. These standards had to ensure reliability, continuity, and longevity. Reliability relating to each participating technology provider of a vending operator receiving identical data files, continuity in terms of data retrieval and distribution throughout a vending operator's network, and longevity by providing assurance to vending operators that interfaces between installed applications would remain viable going forward. As a result, the task force produced NAMA VDI (**V**ending **D**ata **I**nterchange) standards. These standards contain technical specifications that bundle vending machine-level data for rapid distribution throughout a vending operator's technology network and can be implemented by technology providers without vending operator intervention.

Simply stated, vending operators desire technology capable of reliably passing data sets from one application service provider to another so that multiple application service providers can contribute to a single networked solution. The essence of the NAMA VDI standards is to enable data movement through a messaging technique that ensures data integrity of a transmitted set of data, regardless of whether it was pulled or pushed to a server. In other words, NAMA VDI standards render vending technology capable of linking together diverse software solutions, from different vending technology providers, into unified

applications and likely represent a tipping point in the accelerated adoption of vending technologies as operator concerns related to supplier-dependence are significantly reduced.

Data Sets

NAMA VDI is an innovative set of protocols designed to package vending machine-level data (e.g. DEX and MDB data, alerts data, cashless transaction data, etc.) into a message format that can be shared among diverse supplier systems to enable multiple software applications on the identical data set. For example, consider the situation in which a telemetry provider remotely polls DEX data from a vending machine (e.g. Company "X"). The telemetry provider transfers machine-level generated data file to its server (e.g. "X" Server). The server in turn authenticates the file with a NAMA VDI message wrapper and labels its contents for subsequent communication to any other provider's server in a vending operator's network (e.g. Company "Y" or Company "Z" etc.). Additionally, the vending operator may have machine-installed cashless readers that collecting electronic payment data for transmission to cashless gateway for reconciliation. The polled data set would consist of both DEX data and electronic payment data and packaged into an aggregated data set. Movement of the data set to a host vending management software (VMS) system capable of processing DEX data could occur while simultaneously forwarded data to a cashless gateway system could be applied for processing and settlement. This multiple tasking one a single data set is indicative of the robust nature of VDI messaging.

The functionality of VDI standards is somewhat analogous to an email communication in that the file of machine captured data file forms the content of the message while VDI programming places a wrapper, akin to an email message envelope that enables distribution among any number of file servers (e.g. email recipients), regardless of supplier, provider, or manufacturer. NAMA VDI standards, for example, allow for DEX data to be transmitted by a telemetry device or server in real time. This approach provides a platform for a vending operator's VMS to upload data nightly for use in pre-packaging (also referred to as pre-kitting) and/or dynamic scheduling algorithms that rely on variable replenishment strategies. The goal of NAMA VDI standards is to ensure that a vending operator can confidently implement multiple, diverse vending technology solutions while utilizing operational data in existing application software (regardless of supplier). NAMA VDI specifications are open architecture technology standards designed to be extensible, uniform, stable, and manufacturer neutral.

Vending Technology

More than two decades ago, in an effort to standardize the control of machine-level transaction and event data collection, storage, and transmission technical specification committee members of the National Automatic Merchandizing Association (NAMA)[i] and the European Vending Association (EVA) collaborated to develop a set of protocols necessary for efficient data handling and processing. [i]One of the outcomes of this effort is DEX data. DEX, which is an acronym for Data EXchange standard, is capable of capturing machine-level cash in/out data, product movement data, and financial audit data. DEX data is designed to assist operators with product replenishment strategies, product mix rotations, and cash management safeguards. In order to optimize contribution margins, while controlling operating expenses, DEX data plays an important role in productivity and profitability analysis. Accompanying the advent of DEX, a Data Transfer Standard (DTS) was devised so that the DEX data could be exported from the machine in a decipherable electronic format. Once the data was transmitted, it could be entered into a vending management software system (VMS) and used in combination with product mappings to evaluate route coverage, cash handling procedures, and sales performance. It is for this reason that the DTS protocol is often considered an integral part of the DEX standard; not a separate element.

More recently, the Multi-Drop Bus (MDB) protocol emerged is an internal communication protocol designed to ensure that coin mechanisms, bill validators, and cashless payment devices could be effectively interfaced to a vending machine controller (VMC) without regard to proprietary manufacturing specifications. MDB, often compared to USB standards used in generic computer component interfacing, replaced prior practices built on supplier-specific design connectivity. An MDB cable (also termed a machine harness) provides the physical connectivity for attaching peripheral devices (e.g. card reader, bill validator, etc.) to the VMC of a vending machine. MDB is credited with initiating the movement toward open system architecture in vending technology. Since vending machine-level data capture involves the retrieval of stored audit information (akin to a snapshot) via local or remote transfer, there is a need to apply data in various ways to produce a comprehensive analysis of transactions. In fact, some telemetry providers actively monitor the MDB bus to detect, in real-time, product movement and operational alerts (e.g. bill jam, change shortage, door open, temperature variance, etc.).

As a result of prior developments, vending machine-level data formatting and content derivation conforms to the European Vending Association-Data Transfer Standard (EVA-DTS) and provides access to system status data, transactional data, and machine configuration information. In a typical data connection, a polling device actively surveys the vending machine for stored data then follows DTS standards for transmission to an external device. Once the data transfer is complete, the received vending machine-level data can be wrapped in a NAMA VDI message format for subsequent distribution to installed vending system servers (see Figure 1).

DEX Data Collection	>	**DEX Data Extraction**	>	**DEX Data Transmission**	>	**NAMA VDI Standard**

Figure 1. Machine Data Transmission to NAMA VDI Application

MDB Standards

A more recently developed vending machine technology standard than DEX is the multi-drop bus (MDB) standard. Vending machines have one master communication channel and it is labeled the vending machine controller (VMC). The role of the VMC is to define the functionality of peripheral equipment (coin changer, bill validator, card reader, etc.) that must be interfaced with the electronic circuitry of the vender to work properly. MDB is the short form of multi-drop bus/internal communication protocol (MDB/ICP). MDB/ICP is an open global standard governing the interface between a vending machine controller and payment system peripheral devices and is maintained by NAMA and the European Vending Association (EVA). The MDB standard defines a serial bus interface for electronically controlled vending machines. It also standardizes vending machines that employ electronic controls so that all vending and peripheral equipment communicate identically.

Basically, MDB defines and performs as the serial bus interface for electronically controlled vending machines. It also standardizes processes for vending machine interoperability and communication among various peripheral components (i.e. coordination and validation of peripheral equipment interactivity). The serial bus, or MDB, is typically configured as a master-slave arrangement enabling a single master the capability of communicating with up to thirty-two peripheral devices. The VMC serves as the system master unit. The purpose of MDB is to ensure that the necessary functionality of any device on the bus (i.e. interfaced peripheral equipment) is compatible with the capabilities of the VMC. Hence, the software employed by both the peripherals and VMC must be compatible, but not

necessarily at the same level of capability in order to establish peripheral functionality. Within the MDB standard, the capability of each peripheral device is designated by a classification Level identifier. Levels of peripheral functionality were established in response to the addition of major extension in the capability of add-on devices. Level designations are used to avoid potential conflicts that may arise when a VMC Level and a peripheral Level are incompatible. Should this occur, neither the VMC nor the peripheral will be able to issue or reply to a command not supported by the device.

For example, connecting a Level 2 MDB peripheral device to a Level 1 VMC creates a condition of incompatibility that prohibits the proper functioning of the peripheral device (e.g. payment acceptor). The current level for the MDB standard is referred to as Version 3/Level 3 (first introduced March 2003) that pioneered the incorporation of features associated with cashless transactions. In essence, the VMC must initially determine the Level of a peripheral device in order to determine the command set to communicate with the device. A VMC can only issue commands that are supported by the peripheral. For example, a Level 3 command may only be issued for a Level 3 or higher peripheral and will not work with a command issued for a Level 1 or 2 peripheral. A cashless payment peripheral, for instance, deciphers the Level of a VMC through a setup command designed to establish compatibility and functionality.

A compilation of component Levels indicates varying functionality among installed peripherals. For each classification of peripheral device there are a set of mandatory requirements that equipment developers must adhere to ensure compatibility and Level designation. It is a fundamental principle of the MDB standard that all peripheral devices must be implementable with both backward and forward compatibility relative to a machine VMC. The VMC typically gravitates to the highest common Level among peripheral devices.

MDB Interfaces

MDB/ICP enables the VMC to determine what coins the coin changer, and what bills a bill validator, can accept as payment. Additionally, MDB/ICP establishes the amount of credit available through a payment card reader. Through coordination of components, the VMC is able to manage and direct the coin changer in determining how much change to pay out; bills to recycle, or outstanding credit balance to return to the card. There are additional peripheral devices, beyond coin changers, bill acceptors, and card readers, like paykey and other closed systems, for which the VMC can be configured via an MDB interface. Historically, the manufacturer of components being placed into a vending machine had to manually define technical functionality among machine components. Since each component manufacturer acted independently, a number of proprietary device interfaces emerged. The problem with a proprietary interface is that its uniqueness adds unnecessary costs and complexity to vending machine configuration and operation. The MDB/IPC standard was adopted to establish a communication method that allowed the devices in a vending machine to use a common interface. Despite the fact several devices can tie into the same MDB interface, each will operate independently on the interface. Since each interfaced device is assigned a unique address, the VMC can determine which device is active and communicating.

A majority of vending machines support the MDB/IPC standard and thereby are capable of allowing a vending operator to choose payment and other devices primarily based on reliability, performance, and price. Since the MDB/IPC standard establishes the manner by which each component device communicates with the VMC, the connection to each device tends to be identical. Every device has basically two MDB/IPC connectors to allow it to both connect to the MDB/IPC bus in the machine while providing a linked connection to another device (if needed). This design reduces the number of

MDB connectors needed as well as allowing for additional devices. Hence, adding an additional peripheral device to a vending machine is simplified since the requisite hardware and bus connectors to add the device are already in the machine. The MDB/IPC is an internationally supported interface. Through the cooperative efforts of the National Automatic Merchandising Association (NAMA) in the U.S. and the European Vending Association (EVA) and the European Vending Machine Manufacturers Association (EVMMA) in Europe, the standard was developed with provisions for varying currency acceptance and payment technologies.

MBD and DEX

Technically, the MDB/IPC standard defines a serial master-slave communication bus used by the internal devices in the machine, like the coin acceptor. MDB allows for instantaneous updating of the current status of the machine (i.e. data changes as each product is sold). It is for this reason that the MDB standard is considered a transaction-based mechanism, unlike DEX, which is a cumulative-based reporting system. The MDB protocol allows for the attachment of an audit (DEX) device that, acting as a passive slave, receives information of all events that happened on the machine (e.g. vends, sold outs, coins and bills accepted, etc.). On the other hand, DEX involves the retrieval of stored information (a snapshot) through a serial plug designed for connectivity with a handheld terminal (HHT) or small PC. The connection conforms to the EVA-DTS standard and provides access to status data, testing routines, and machine setup. In a DEX connection, the connected device actively polls the machine for stored information.

Cashless transactions are not dependent on DEX but rely on MDB processes. The fundamental difference between DEX and MDB is that MDB is the only method for a bill acceptor or a coin changer to report credit deposited to authorize a vending transaction. DEX cannot do this. This fact makes it necessary to have MDB installed; DEX, while needed for sales reporting, is not mandatory for the machine to operate. Hence, from a cashless payment perspective, MDB is more useful than DEX since it details the transaction (card number, transaction value, product(s) sold, date, and time) for reconciliation. The results of the transaction will be posted as an MDB record. For operators not employing cashless vending, DEX data is often sufficient to provide necessary information for a vending management system. It is for this reason, some vending operators only use MDB for cashless transactions, and ignore DEX data. For those operators desiring DEX data, a DEX cable can be used to transfer the DEX file along with the cashless MDB data.

DEX is the key to technological advancements in the vending industry worldwide as it enables data capture at the point of purchase. DEX has earned international recognition and support and can be used to facilitate consistent data formatting throughout the vending channel. In the past, machine manufacturers varied in how data exchange transmissions occurred. In response, DEX designers and equipment engineers have established standards governing data recordation, file formatting, and file exportation through common interface linkages. As a consequence, vending machines are manufactured as DEX-enabled and are often labeled "DEX-compliant." From a sales perspective, DEX provides the vending operator the ability to track brand and/or product preferences at the point of purchase. DEX has been found to improve sales performance, reduce operating expenses, and minimize machine malfunctions. In addition, DEX enables space to sales analysis, for machine-level column allocation optimization, in vending management software. This is an important outcome of a DEX-compliant device. The main benefit of line item tracking is accountability and machine plan-o-gram (i.e. rotating menu of product offering) development.

The fact that vending equipment tends to be strategically placed in disparate locations presents a challenge to efficient replenishment, sales analyses, malfunction notification, and comprehensive audit reporting. Fortunately, machine-level transactional data can be captured through an electronic control board installed within each vending machine. Aggregating machine-level data enables remote review of transactions and inventory without having to have a physical presence at the machine. The fact data can be exported to a remote warehouse, central office, or product fulfillment center extends the opportunity for more thorough, immediate, and frequent analysis. A majority of v-commerce applications are the result of DEX implementation.

In the past, machine manufacturers varied in how data exchanges and transmissions occurred. Recently released DEX software (Edition 6 and higher) tightens the specifications of the protocol to prevent possible misinterpretations in accountability or brand identification. Since there has been a proliferation of diverse vending products, and several variations in the packaging of the same product, the DEX standard has been refined to acknowledge and differentiate between product offerings. While not all vending operators demand identical informational output, vending machine circuit boards are built to possess similar data collection capabilities to ensure the delivery of consistent content. For example, three data elements referenced in the DEX standard are: 1) number of bills held in the bill stacker, 2) quantity and denomination of coins stored in the coin box, and 3) number of vends or products sold.

A DEX-compliant machine relies upon DEX architecture to enable vending machine polling. The vending machine exports its unique identification number and stored data to an external system for analysis and processing. An optional element of this data stream is the machine's service history, including the last date the machine was serviced. Once DEX data is exchanged with a vending management system various transaction audits can be performed. Since captured data is not accessible or editable prior to interfacing to an auxiliary system, cash accountability will be accurate and complete. Also, the ability to track product information at the machine-level enhances productivity, as machine fulfillment is improved and manual data entry eliminated. The DEX protocol enables different makes and models of vending machines to communicate in a consistent manner. DEX data sets include sales mix, cash collection, product movement, and malfunction alerts. Additionally, DEX specifications may soon include a standard for reporting error codes for payment validation, dispensing jams, and other operational problems. Proposed specifications are pending approval.

Since vending machines have an average life of ten years, it may take a generation of new machine installations to fully realize the DEX potential. Basically, DEX provides an indisputable, auditable accounting method for cash collections, units sold, and product price recordation that capable of enhancing route efficiency and improving warehouse operations. For example, how much cash should be in a machine at the close of a sales period? A route driver, unable to view the DEX electronic record, will have cash collections compared against the machine-level electronic record. Balancing cash against collections provides management with a unique level of information and control.

DEX Polling

NAMA and the European Vending Association (EVA) have jointly adopted a communication protocol for the electronic retrieval of machine-level information via data polling. As a consequence, vending machines are now manufactured as DEX-enabled. Each vending machine is given a unique identifying number by which the DEX data extracted is labeled. During a polling session, this unique number and the date and time that the service occurred, are transmitted to the polling device. DEX data is polled an audit can be performed. Since captured data is not accessible or editable by the route driver,

cash accountability is assumed accurate and complete. Also, the ability to track product information at the machine level enhances productivity as route time is improved and manual data entry is eliminated. DEX specifies a data format to enable all different types of machines and machine models to communicate electronically in a similar manner. The DEX information available includes: sales, cash collections, product movement and other vending machine activities. Additionally, the DEX specification contains a standard for reporting error codes for payment validation, jams, and other operational problems. Line item tracking is important to both accountability and assistance in future machine menu development. DEX data retrieval can be accomplished via three distinct polling modes: 1-local polling, 2-dial-up polling or 3-wireless polling.

Local Polling – local polling incorporates a hand-held device (or pocket probe) designed to plug connect to a vending machine's DEX-port or to communicate through an IR port. Once the connection is established, the device is used to extract (upload) transactional data from the machine to the handheld device. A typically DEX data upload takes approximately five seconds. Field collected data can be transferred from the handheld device to a central office computer (downloaded) for processing, analysis, and report generation.

Dial-up Polling – dial-up polling involves use of a modem and telephone line. Once a valid connection is established, DEX data can be transported to a remote office or warehouse location for evaluation over an Internet or virtual private network (VPN) connection. This design enables the machine to be remotely monitored with respect to cash, inventory, and machine malfunctions.

Wireless Polling – similar to dial-up polling, wireless polling enables remote access to DEX data via a cellular network. Wireless polling however relies upon cellular network connectivity to establish the proper linkage. The advancement of wireless technology has emerged as an attractive alternative. Wireless applications possess tremendous potential for the vending industry, an industry that desires mobility, flexibility, and reliability in enterprise-wide operations. Vending practitioners dissatisfied with the constraints and complexities of hard wiring are migrating to the convenience of design portability and user mobility that wireless technology solutions provide.

Common network connectivity options include both the Internet and virtual private networks (VPN). Cellular connectivity presents challenges based on the architectural structure surrounding the vending equipment combined with strength of signal requirements. While connectivity to a VPN tends to be more direct and less susceptible to structural infringements, it is likely to be more costly. Historically, vending operators have benefiting from such devices as hand-held terminals, personal digital assistants, smart paging units, global positioning systems, telecommunication links (telemetry), proximity transponders, and related applications.

VDI Standards

The purpose of the NAMA VDI standard is to establish transparent, non-proprietary interfaces that enable transportation of data among the main components of a vending system (e.g. vending machine, telemetry system, cashless payment system, specialty applications, and vending management software). The non-proprietary nature of NAMA VDI renders it an open standard. NAMA VDI relies on messaging standards to satisfy data interchange needs and is not concerned with the entity transmitting or receiving such messages. For example, a messaging standard governing the transmission of machine-level DEX data may originate from the vending machine, an advanced telemetry device, or the file server of another entity. NAMA VDI mandates that the message format conform to the technical specifications of the standard, regardless of the entity creating the message.

VDI Messaging

The NAMA VDI Task Force has identified the following seven elements as important to vending data messaging (interchangeable/exchangeable data files):

1- DEX data messaging – sent or requested captured DEX data file
2- Alert data messaging – may originate from the VMC, DEX, or MDB depending on telemetry provider.
3- Device Status -- device configuration and/or service request
4- Device Configuration – sent device configuration and/or status reporting
5- Security Authorization – defines cooperative agreement partners
6- Machine Message – reconfigures machine to EVA standards
7- Device Messaging - provides confirmation of download instructions

Cooperative Relationships

The NAMA VDI standard incorporates 'cooperative agreements' among competing vending technology suppliers so that interchanged data will be more meaningfully consumed and effectively applied. Cooperative agreements, often referred to as trading partner agreements, involve written documentation that informs both sender (producer) and receiver (consumer) of NAMA VDI messages the specifics of the message(s) being shared. Descriptive elements include such items as: company profile, security authorization, machine identification, location identification, and type of connectivity (server, web-service, email, etc.). For example, if Provider X is to pass a NAMA VDI data message to Provider Y then the cooperating parties must have transaction information to successfully distribute and utilize the desired data messages. User names, passwords, and web-based SSL encryption also can be used to help insure data transfers are secure and accessible by authorized entities.

Early Adopters

Seven major vending technology providers volunteered to implement and fine tune the VDI standards prior to release of Version 1.0. Cantaloupe Systems, CompuVend, Crane Merchandising, InOne Technology, MEI Group, USA Technologies, and Validata worked cooperatively to field test and validate VDI specifications and procedures. This is similar to the replacement of specialized railroad car connectors with non-specialized couplers that enable assembly of cars in any order or sequence (see Figure 2).

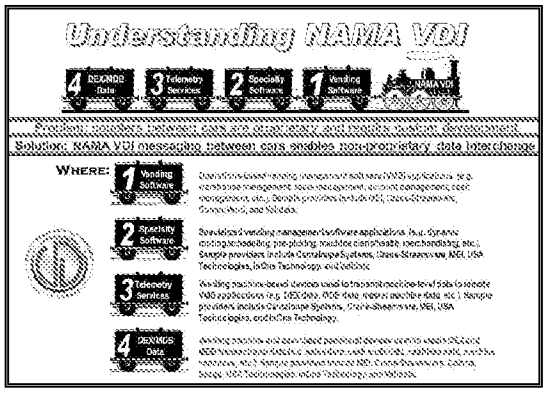


Figure 2. Understanding NAMA VDI Standards

The NAMA VDI standards are being released beginning in first quarter 2010 and are planned to have additional features added and released in sequential stages as developments are completed. Release

1.0 addresses the most critical area among data transactions: DEX messaging. It is anticipated that as the remaining data messages are developed, they will be released and labeled in ascending numerical order (e.g. Release 1.1, 1.2, and so forth).

VDI Benefits

The NAMA VDI standards afford several direct benefits to operators and bottlers, especially those embarking on technology decisions. When purchasing vending technology from a company adhering to NAMA VDI standards, the buyer can be assured that:

1. investment in the compliant technology will be compatible across major suppliers
2. there is no longer a need to rely on the success of a single supplier
3. multiple telemetry devices will work with a variety of VMS providers
4. selling or acquiring VDI compatible components simplifies continued operations

See Figure Three for an illustration of the NAMA VDI standards as applied to data for processing by a vending management software (VMS) program as well as transactional data for processing via a cashless gateway.

SUMMARY

Major vending technology providers participating in NAMA VDI development have created a tipping point for accelerating the implementation of vending technology. Increased interest in addressing operator concerns has resulted in an unprecedented cooperation among vending technology suppliers to enable harmonic data interchange. NAMA VDI ensures that operators can feel confident in technology investment, choice of suppliers, and be assured that hardware and software will work together now and in the future. There has never been a better or safer time to invest in cashless vending, remote machine monitoring, or VMS technology.

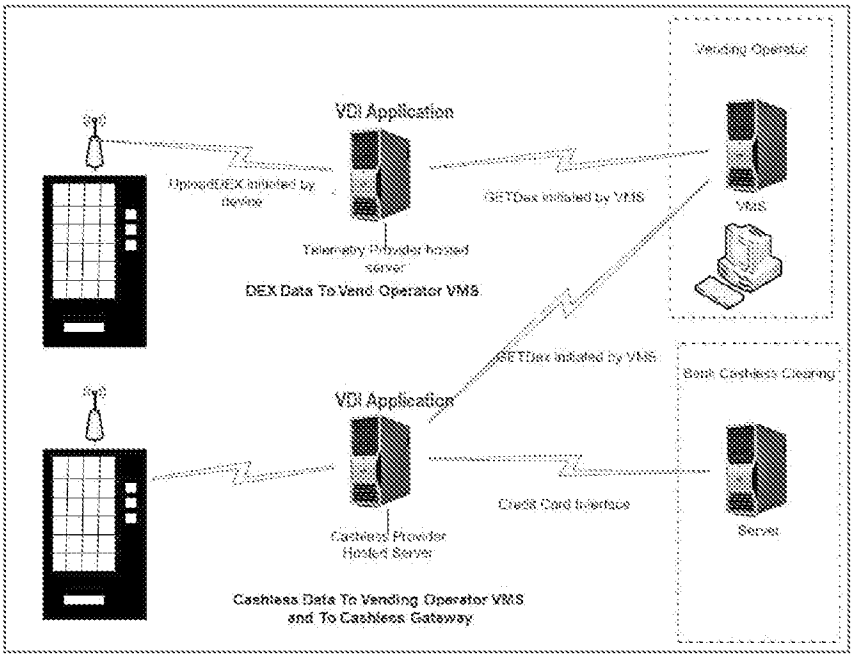


Figure 3. DEX Data transported via telemetry to multiple provider servers reporting into a VMS

REFERENCES

European Vending Association, DEX Standards V 6.1, www.vending-europe.eu/en/standards/eva-dts.html

Kasavana, Michael L. (2006). Understanding MDB, DEX, and DTS, NAMA White Paper (2) 5-17

Kasavana, Michael L., V-Commerce: Understanding Vending Machine Terminology, HFTP Bottomline Magazine, www.hospitalitynet.org/news/4011592.html

MDB/ICP, Version 4.0, April 2009, www.vending.org/technology/MDB_Version_4.pdf

NAMA Vending Data Interchange Standards, www.vending.org/technology/VDI_Standards.pdf

National Automatic Merchandising Association, MDB Standards V 4.0, www.vending.org/technology/MDB_Version_4.pdf

BLUETOOTH DOC	Date / Year-Month-Day 2005-11-25	Approved Adopted	Revision V10r00	Document No HFP1.5_SPEC
Prepared Car Working Group	e-mail address car-feedback@bluetooth.org			N.B.

HANDS-FREE PROFILE 1.5

Abstract

The Hands-Free Profile (HFP) 1.5 specification defines the minimum set of functions such that a Mobile Phone can be used in conjunction with a Hands-Free device (e.g. installed in the car or represented by a wearable device such as a headset), with a *Bluetooth*® Link providing a wireless means for both remote control of the Mobile Phone by the Hands-Free device and voice connections between the Mobile Phone and the Hands-Free device.

Compliance with this specification assures interoperability between a Bluetooth enabled Hands-Free device and any Bluetooth equipped Mobile Phone supporting this profile.

Revision History

Revision Number	Date	Comments
RC10.50	01-01-2912-11-2003	Hands-Free Profile 0.50 published1st draft for SubWG12 Review
RC21.00m VD	22-11-200403-14-03	CR document derived from FIPD07r04Additional comments from BTI/BQRB reviews
D10r08	16-02-2005	Editing for Prototyping Specification standards
D12r00	19-03-2005	Incorporated Prototyping Specifications
D12r01	22-04-2005	Errata and corrections from IOP.
D12r02	28-04-2005	HFP 1.0 Errata- 13, 261, 317, 549, 550, 575, 586, 635, 706, 731, 746
D15r03	13-05-2005	Correct formatting problems and comments from review.
D15r04	27-05-2005	Errata 819, 820, 821, 822
D15r05	07-06-2005	Edits from BARB Review. Errata 823.
D15r06	07-18-2005	Changes from BARB review. Editorial changes for language and readability.
D15r07	08-01-2005	Comments from BTI/BARB review.
D15r08	09/02/2005	Comments from BTI review.
D15r09	09/22/2005	Comments from BTI review
D15r09	09/26/2005	Add 3GPP 27.07 version and remove reference to Call Waiting for AT+CHUP
D15r10-11	10/04/2005	BTI Comments
D15r12	10/05/2005	Editorial changes
D15r13	10/07/2005	Prepare for publication
V10r00	11/25/2005	Adopted by the *Bluetooth* Board of Directors

Contributors

Name	Company
Aaron WEINFIELD	Denso
Basam MASRI	Denso
Don LIECHTY	Extended Systems
Stephen RAXTER	Johnson Controls
Vartika AGARWAL	Motorola
Leonard HINDS	Motorola
Burch SEYMOUR	Motorola
Stephane BOUET	Nissan
Jamie MCHARDY	Nokia
Jurgen SCHNITZLER	Nokia
Guillaume POUJADE	Parrot
Dmitri TOROPOV	Siemens
Erwin WEINANS	Sony Ericsson
Tim REILLY	Stonestreet One
Akira MIYAJIMA	Toyota
Ryan BRUNER	Visteon
Scott WALSH	Plantronics
Patrick CLAUBERG	Nokia
Neil MACMULLEN	CSR
Michael BUNTSCHECK	BMS
Florencio CEBALLOS	Visteon
Bill BERNARD	Visteon

Disclaimer and Copyright Notice

The copyright in this specification is owned by the Promoter Members of *Bluetooth*® Special Interest Group (SIG), Inc. ("*Bluetooth* SIG"). Use of these specifications and any related intellectual property (collectively, the "Specification"), is governed by the Promoters Membership Agreement among the Promoter Members and *Bluetooth* SIG (the "Promoters Agreement"), certain membership agreements between *Bluetooth* SIG and its Adopter and Associate Members (the "Membership Agreements") and the *Bluetooth* Specification Early Adopters Agreements (1.2 Early Adopters Agreements) among Early Adopter members of the unincorporated *Bluetooth* SIG and the Promoter Members (the "Early Adopters Agreement"). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to *Bluetooth* SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of *Bluetooth* SIG or a party to an Early Adopters Agreement (each such person or party, a "Member"), is prohibited. The legal rights and obligations of each Member are governed by their applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement or Early Adopters Agreement and other liability permitted by the applicable agreement or by applicable law to *Bluetooth* SIG or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the *Bluetooth* technology ("*Bluetooth* products") may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of *Bluetooth* products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their *Bluetooth* Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their *Bluetooth* products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. **NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.**

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST *BLUETOOTH* SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserve the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate.

Contents

1 Introduction

1.1 Scope

This document defines the protocols and procedures that shall be used by devices implementing the Hands-Free Profile. The most common examples of such devices are in-car Hands-Free units used together with cellular phones, or wearable wireless headsets.

The profile defines how two devices supporting the Hands-Free Profile shall interact with each other on a point-to-point basis.

An implementation of the Hands-Free Profile typically enables a headset, or an embedded Hands-Free unit to connect, wirelessly, to a cellular phone for the purposes of acting as the cellular phone's audio input and output mechanism and allowing typical telephony functions to be performed without access to the actual phone.

1.2 Profile Dependencies

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by explicitly referencing it. Dependency is illustrated in the figure below.

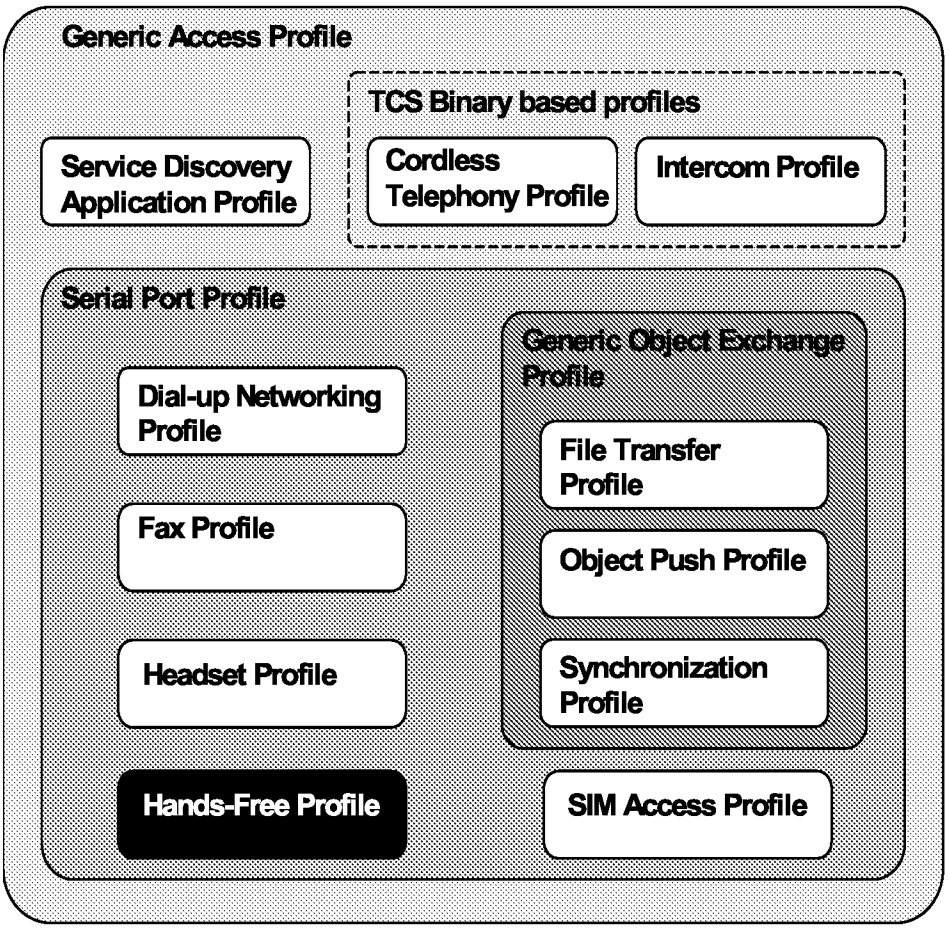


Figure 1.1: Bluetooth Profiles

As indicated in the figure, the Hands-Free Profile is dependent upon both the Serial Port Profile [5] and the Generic Access Profile [4]. Details are provided in Sections 5 (Serial Port Profile) and 6 (Generic Access Profile).

1.3 Symbols and Conventions

1.3.1 Requirement Status Symbols

In this document, the following symbols are used:

- "M" for mandatory to support
- "O" for optional to support
- "X" for excluded (used for capabilities that may be supported by the device, but the Hands-Free Profile shall not use these capabilities)

- "C" for conditional to support
- "N/A" for not applicable (in the given context this capability is not defined)

Some capabilities or features (identified as "X"), mandated according to the relevant Bluetooth specifications, are excluded with respect to this profile because they may degrade the operation of devices in the particular use case. Therefore, features or capabilities labeled "X" shall never be activated while operating in a use case where they are labeled as such.

1.3.2 Naming Conventions

In this document, the following naming conventions are used:

- Where "Core Specification" is said it refers to the Bluetooth Core Specification 1.1 or later adopted by the Bluetooth® SIG.
- Where "LMP link" is said, it means a Link Manager (LM) level link over which only Link Manager Protocol (LMP) commands are conveyed.
- Where "RFCOMM connection" is said, it means the presence of a virtual serial port as specified in [5].
- Where "Service Level Connection" is said, it means a synchronized high-level protocol connection involving a portion of the protocol stack. In this specific case, it refers to the presence of a RFCOMM connection, and assumes that the HF has synchronized itself to the state of the AG using the specified Service Level Connection initialization procedure.
- Where "Service Level Connection initialization" is said, it means the execution of the set of AT commands and responses specified by the profile necessary to synchronize the state of the HF with that of the AG.
- Where "Service Level Connection establishment" is said, it means the combined process of establishing the RFCOMM connection, as well as the necessary device synchronization using Service Level Connection initialization.
- Where "Synchronous Connection" is said, it means a SCO or eSCO logical link intended for supporting a full duplex Audio Connection.
- Where "Audio Connection" is said, it means a Synchronous Connection including the means to provide a complete audio path between two devices assuming roles within this profile.
- Where "incoming call" is said, it means a call connection in the direction "Phone Network=>AG", such that it is initiated by the Network to which the AG is attached.
- Where 'outgoing call' is said, it means a call connection in the direction "AG=>Phone Network", such that it is initiated by the AG towards the Network to which it is attached.

1.3.3 Signaling Diagram Conventions

The signaling diagrams in this specification are informative only. Within the diagrams, the following conventions are used to describe procedures:

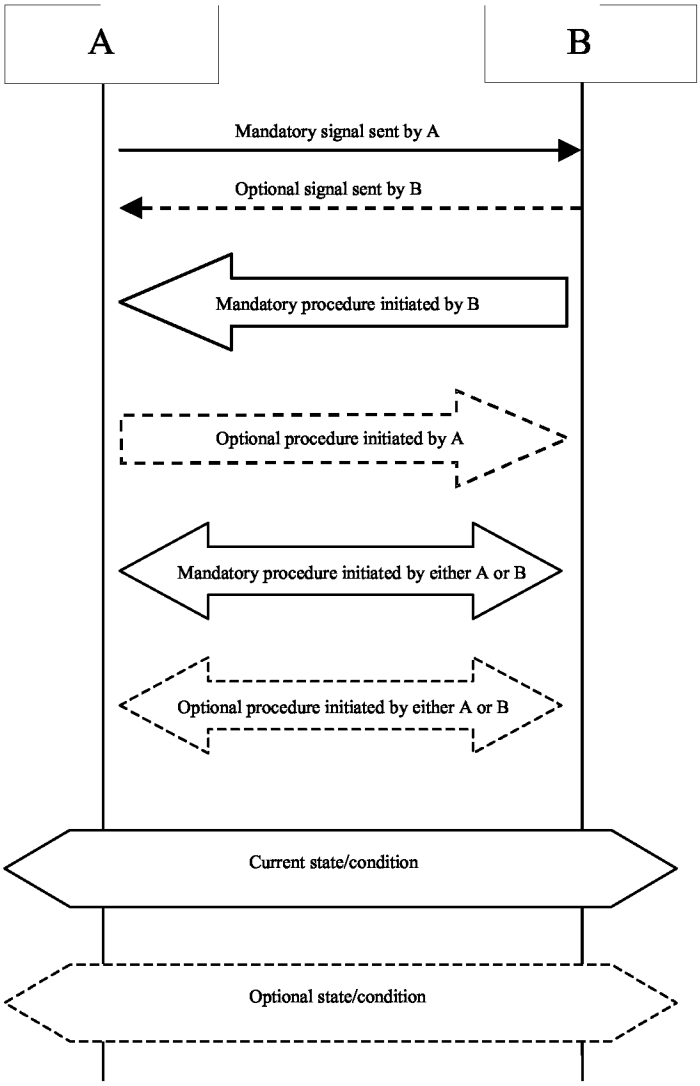


Figure 1.2: Conventions used in signaling diagrams

2 Profile Overview

2.1 Protocol Stack

The figure below shows the protocols and entities used in this profile.

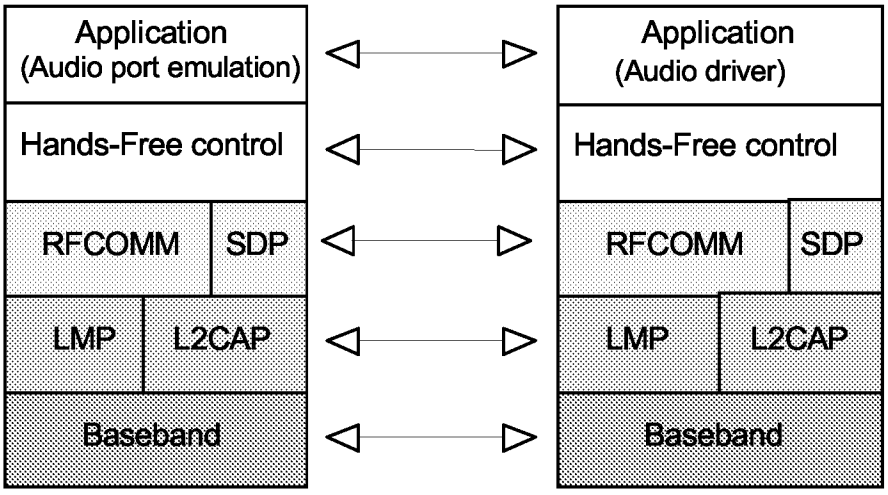


Figure 2.1: Protocol stack

The Baseband, LMP and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM is the Bluetooth serial port emulation entity. SDP is the Bluetooth Service Discovery Protocol. See [1] for more details on these topics.

Compatibility to v1.1 or later Core Specification is required.

Hands-Free control is the entity responsible for Hands-Free unit specific control signaling; this signaling is AT command based.

Although not shown in the model above, it is assumed by this profile that Hands-Free Control has access to some lower layer procedures (for example, Synchronous Connection establishment).

The audio port emulation layer shown in Figure 2.1 is the entity emulating the audio port on the Audio Gateway, and the audio driver is the driver software in the Hands-Free unit.

For the shaded protocols/entities in Figure 2.1, the Serial Port Profile [5] is used as the base standard. For these protocols, all mandatory requirements stated in the Serial Port Profile apply except in those cases where this specification explicitly states deviations.

2.2 Configuration and Roles

Figure 2.2 below shows typical configurations of devices for which the Hands-Free Profile is applicable:

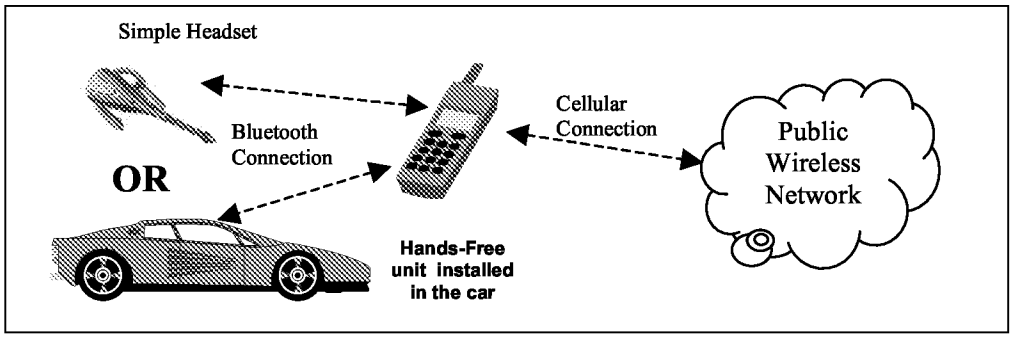


Figure 2.2: Typical Hands-Free Use

The following roles are defined for this profile:

Audio Gateway (AG) – This is the device that is the gateway of the audio, both for input and output. Typical devices acting as Audio Gateways are cellular phones.

Hands-Free unit (HF) – This is the device acting as the Audio Gateway's remote audio input and output mechanism. It also provides some remote control means.

These terms are used in the rest of this document to designate these roles.

2.3 User Requirements and Scenarios

The following rules apply to this profile:

a) The profile states the mandatory and optional features when the "Hands-Free Profile" is active in the Audio Gateway and the Hands-Free unit.

b) The profile mandates the usage of CVSD for transmission of audio (over the Bluetooth link). The resulting audio is monophonic, with a quality that, under normal circumstances, does not have perceived audio degradation.

c) Between the Hands-Free unit and the Audio Gateway, only one Audio Connection per Service Level Connection at a time is supported.

d) Both the Audio Gateway and the Hands-Free unit may initiate Audio Connection establishment and release. Valid speech data shall exist on the Synchronous Connection in both directions after the Audio Connection is established.

e) Whenever an "Audio Connection" exists, a related "Service Level Connection" shall also exist.

f) The presence of a "Service Level Connection" shall not imply that an "Audio Connection" exists. Releasing a "Service Level Connection" shall also release any existing "Audio Connection" related to it.

2.4 Profile Fundamentals

Baseband authentication and encryption is optional for both the Hands-Free unit and the Audio Gateway. If both devices support authentication and encryption, the application on either device may require its use.

A Hands-Free unit may be able to use the services of the Audio Gateway without the creation of a secure connection. It is implementation specific whether the Hands-Free unit provides or supports security enforcement for the user.

Whenever baseband authentication and/or encryption is used, the two devices shall create a secure connection using the GAP authentication procedure as described in Section 5.1 of the Generic Access Profile [4]. This procedure may include entering a Bluetooth PIN code and creation of proper link keys. In cases when the UI of the Hands-Free unit is limited, a fixed Bluetooth PIN code may be used during the GAP authentication procedure.

If a LMP link is not already established between the Hands-Free unit and the Audio Gateway, the LMP link shall be set up before any other procedure is performed.

There are no fixed master of slave roles in the profile.

The Audio Gateway and Hand-Free unit provide serial port emulation. For the serial port emulation, RFCOMM (see [1]) is used. The serial port emulation is used to transport the user data including modem control signals and AT command from the Hands-Free unit to the Audio Gateway. The AT commands are parsed by the Audio Gateway and responses are sent to the Hands-Free unit via the Bluetooth serial port connection.

2.5 Conformance

If conformance to this profile is claimed, all capabilities indicated as mandatory for this profile shall be supported in the specified manner (process mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory, optional and conditional capabilities, for which support is indicated, are subject to verification as part of the Bluetooth Qualification Program.

3 Application layer

This section describes the feature requirements for units complying with Hands-Free Profile.

Table 3.1 below shows the feature requirements for this profile.

	Feature	Support in HF	Support in AG
1.	Connection management	M	M
2.	Phone status information	M(note 1)	M
3.	Audio Connection handling	M	M
4	Accept an incoming voice call	M	M
5.	Reject an incoming voice call	M	O
6.	Terminate a call	M	M
7.	Audio Connection transfer during an ongoing call	M	M
8.	Place a call with a phone number supplied by the HF	O	M
9.	Place a call using memory dialing	O	M
10.	Place a call to the last number dialed	O	M
11.	Call waiting notification	O	M
12.	Three way calling	O(note 2)	O(note 3)
13.	Calling Line Identification (CLI)	O	M
14.	Echo canceling (EC) and noise reduction (NR)	O	O
15.	Voice recognition activation	O	O
16.	Attach a Phone number to a voice tag	O	O
17.	Ability to transmit DTMF codes	O	M
18.	Remote audio volume control	O	O
19.	Respond and Hold	O	O
20.	Subscriber Number Information	O	M
21a.	Enhanced Call Status	O	M
21b.	Enhanced Call Controls	O	O

Note 1: The HF shall support at least the two indicators "service" and "call".

Note 2: If "Three way calling" is supported by the HF, it shall support AT+CHLD values 1 and 2. The HF may additionally support AT+CHLD values 0,3 and 4.

Note 3: If "Three way calling" is supported by the AG, it shall support AT+CHLD values 1and 2. The AG may additionally support AT+CHLD values 0,3 and 4.

Table 3.1: Application layer procedures

Table 3.2 below maps each feature to the procedures used for that feature. All procedures are mandatory if the feature is supported.

	Feature	Procedure	Ref.
1.	Connection management	Service Level Connection establishment	4.2
		Service Level Connection release	4.3
2.	Phone status information	Transfer of Registration Status Transfer of Signal Strength Indication Transfer of Roaming Status Indication Transfer of Battery Level Indication Query of Operator Selection Extended Audio Gateway Error Codes Transfer of Call, Call Setup and Call Held Status	4.4 4.5 4.6 4.7 4.8 4.9 4.10
3.	Audio Connection handling	Audio Connection set up Audio Connection release	4.11 4.12
4.	Accept an incoming voice call	Answer an incoming call	4.13
5.	Reject an incoming voice call	Reject an incoming call	4.14
6.	Terminate a call	Terminate a call process	4.15
7.	Audio Connection transfer during an ongoing call	Audio Connection transfer towards the HF Audio Connection transfer towards the AG	4.16 4.17
8.	Place a call with the phone number supplied by the HF	Place a call with the phone number supplied by the HF	4.18
9.	Place a call using memory dialing	Memory dialing from the HF	4.19
10.	Place a call to the last number dialed	Last number re-dial from the HF	4.20
11.	Call waiting notification	Call waiting notification activation	4.21
12.	Three way calling	Three way call handling	4.22
13.	Calling Line Identification (CLI)	Calling Line Identification (CLI) notification	4.23
14.	Echo canceling (EC) and noise reduction (NR)	HF unit requests turning off the AG's EC and NR	4.24
15.	Voice recognition activation	Voice recognition activation	4.25
16.	Attach a phone number to a voice tag	Attach a voice tag to a phone number	4.26
17.	Ability to transmit DTMF codes	Transmit DTMF code	4.27

	Feature	Procedure	Ref.
18.	Remote audio volume control	Remote audio volume control Volume level synchronization	4.28
19.	Response and Hold	Query response and hold status Put an incoming call on hold from HF Put an incoming call on hold from AG Accept a held incoming call from HF Accept a held incoming call from AG Reject a held incoming call from HF Reject a held incoming call from AG Held incoming call terminated by caller	4.29 4.29 4.29 4.29 4.29 4.29 4.29 4.29
20.	Subscriber Number Information	Subscriber Number Information	4.30
21a.	Enhanced Call Status	Query Call List Indication of Held Call Status	4.31 4.31
21b.	Enhanced Call Control	Release Specified Call Private Consult Mode	4.32 4.32

Table 3.2: Application layer feature to procedure mapping

4 Hands-Free Control Interoperability Requirements

4.1 Introduction

The interoperability requirements for the Hands-Free Control entity are completely contained in this section. Sections 4.2 through 4.28 specify the requirements for the procedures directly related to the application layer features.

The procedures listed in this section are primarily based on the use of a minimum set of AT commands as the control protocol. Section 4.33 specifies these AT commands and their result codes.

Section 4.2 specifies how Service Level Connections are handled in general and specifically states how the layers beneath the Hands-Free Control entity are used to establish and release a Service Level Connection.

4.2 Service Level Connection Establishment

Upon a user action or an internal event, either the HF or the AG may initiate a Service Level Connection establishment procedure.

A Service Level Connection establishment requires the existence of a RFCOMM connection, that is, a RFCOMM data link channel between the HF and the AG.

Both the HF and the AG may initiate the RFCOMM connection establishment. If there is no RFCOMM session between the AG and the HF, the initiating device shall first initialize RFCOMM.

The RFCOMM connection establishment shall be performed as described in Section 7.3 of Generic Access Profile [4] and Section 3 of Serial Port Profile [5].

4.2.1 Service Level Connection Initialization

When an RFCOMM connection has been established the Service Level Connection Initialization procedure shall be executed.

First in the initialization procedure the HF shall send the AT+BRSF=<HF supported features> command to the AG to both notify the AG of the supported features in the HF, as well as to retrieve the supported features in the AG using the +BRSF result code.[1]

After having retrieved the supported features in the AG, the HF shall determine which indicators are supported by the AG, as well as the ordering of the supported indicators. This is because, according to the 3GPP 27.007 specification [2], the AG may support additional indicators not provided for by the Hands-Free Profile, and because the ordering of the indicators is implementation specific. The HF uses the AT+CIND=? Test command to retrieve information about the supported indicators and their ordering.

[1] Audio Gateways supporting the 0.96 version of Hands-Free Profile will return ERROR as a response to AT+BRSF

Once the HF has the necessary supported indicator and ordering information, it shall retrieve the current status of the indicators in the AG using the AT+CIND? Read command.

After having retrieved the status of the indicators in the AG, the HF shall then enable the "Indicators status update" function in the AG by issuing the AT+CMER command, to which the AG shall respond with OK. As a result, the AG shall send the +CIEV unsolicited result code with the corresponding indicator value whenever a change in service, call, or call setup status occurs. When an update is required for both the call and call setup indicators, the AG shall send the +CIEV unsolicited result code for the call indicator before sending the +CIEV unsolicited result code for the call setup indicator. The HF shall use the information provided by the +CIEV code to update its own internal and/or external indications.

Once the "Indicators status update" function has been enabled, the AG shall keep the function enabled until either the AT+CMER command is issued to disable it, or the current Service Level Connection between the AG and the HF is dropped for any reason.

After the HF has enabled the “Indicators status update” function in the AG, and if the “Call waiting and 3-way calling” bit was set in the supported features bitmap by both the HF and the AG, the HF shall issue the AT+CHLD=? test command to retrieve the information about how the call hold and multiparty services are supported in the AG. The HF shall not issue the AT+CHLD=? test command in case either the HF or the AG does not support the "Three way calling" feature.

The HF shall consider the Service Level Connection fully initialized, and thereby established, in either of the following cases:

- After the HF has successfully retrieved information about how call hold and multiparty services are supported in the AG using the AT+CHLD command, if and only if the “Call waiting and 3-way calling” bit was set in the SupportedFeatures attribute of the SDP records for both HF and AG.
- After the HF has successfully enabled the “Indicator status update” using the AT+CMER command, if and only if the “Call waiting and 3-way calling” bit was not set in the SupportedFeatures attribute of the SDP records for either the HF or the AG.

If the HF receives an indication from the AG that a call is currently active, the HF may determine if an unanswered call is currently on hold by querying the Response and Hold state of the AG (see section 4.29.1).

The AG shall consider the Service Level Connection to be fully initialized, and thereby established, in either of the following cases:

- After that the AG has successfully responded with information about how call hold and multiparty services are supported in the AG using +CHLD as well as responded OK, if and only if the “Call waiting and 3-way calling” bit was set in the supported features bitmap for both HF and AG.

- After the AG has successfully responded with OK to the AT+CMER command (to enable the "Indicator status update" function), if and only if the "Call waiting and 3-way calling" bit was not set in the supported features bitmap for either the HF or the AG.

Refer to Section 4.33 for more information on the AT+CIND, AT+CMER, AT+CHLD and AT+BRSF commands and the +CIEV unsolicited result code.

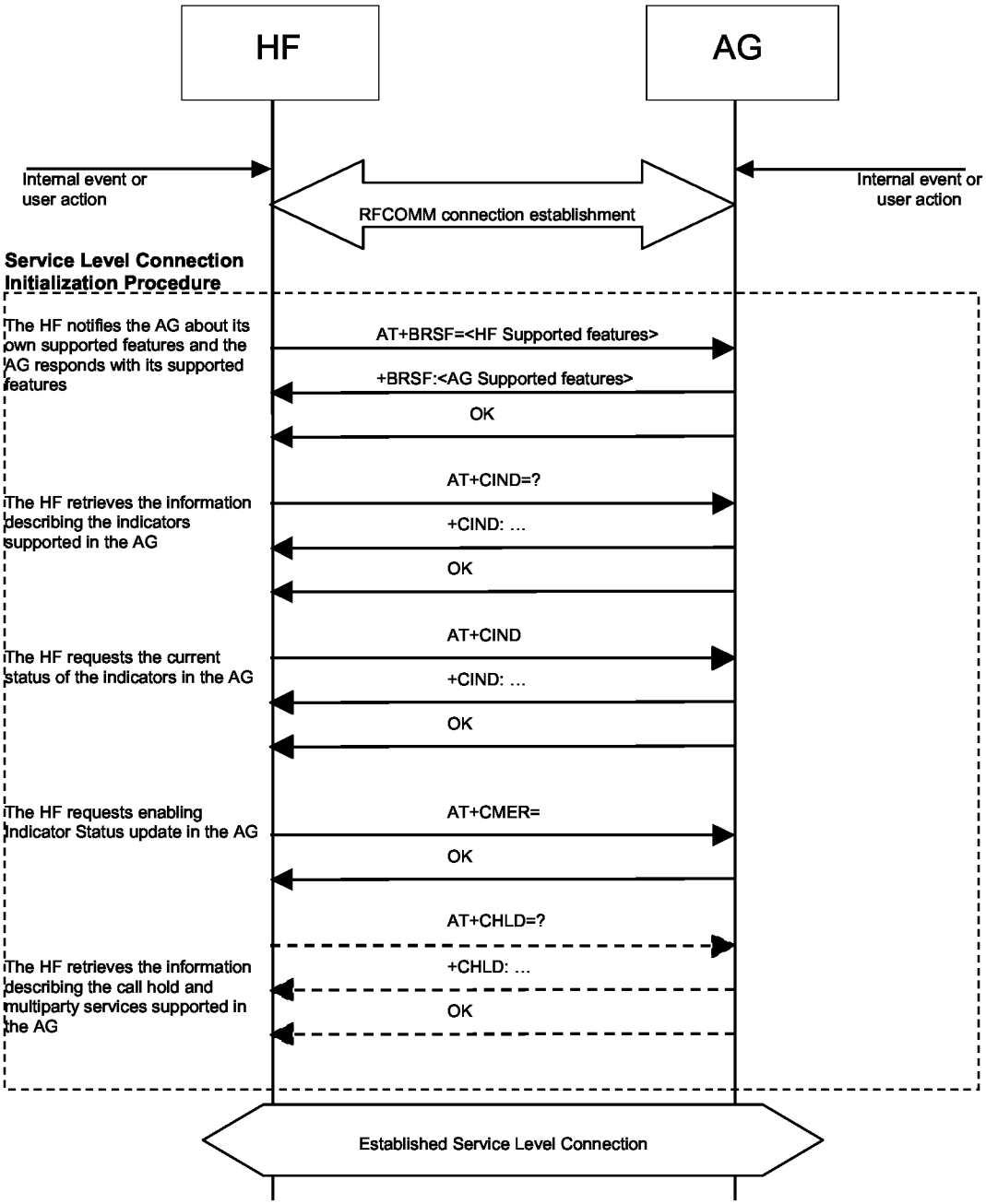


Figure 4.1: Service Level Connection establishment

4.2.2 Link Loss Recovery

This section addresses the link loss recovery from a HF unit. The HF unit may reconnect with the AG whenever there is loss of Bluetooth link.

When a Service Level Connection is disconnected due to explicit termination at one end (using the "Service connection release" as described in Section 4.3), then both devices (AG and HF unit) shall wait for an explicit user action before an attempt is made to re-establish the Service Level Connection.

If the HF unit determines that the Service Level Connection was disconnected due to a link supervision timeout, then the HF unit may execute the "Service Level Connection establishment" procedure as described in Section 4.2 to establish a new Service Level Connection to the AG. Following a link loss due to link supervision timeout, the HF unit shall not assume that the service level connection state from the previous connection is valid (such as Call Status, Service Status).

4.3 Service Level Connection Release

This section describes the procedure for releasing a Service Level Connection.

The disconnection of a Service Level Connection shall result in the immediate removal of the corresponding RFCOMM data link channel between the HF and the AG. Also, an existing audio connection has to be removed as consequence of the removal of the Service Level Connection. The removal of the L2CAP and link layers is optional.

An established Service Level Connection shall be released using a “Service Level Connection removal” procedure.

- Either the HF or AG shall initiate the “Service Level Connection release” procedure due to an explicit user request.
- The “Service Level Connection release” procedure shall be initiated if the Bluetooth functionality is disabled in either the HF or AG.
- The “Service Level Connection release” procedure may be initiated if an “Audio Connection transfer towards the AG”, as stated in section 4.12, is performed during an ongoing call in the AG. In the case that the Service Level Connection is removed, the AG shall attempt to re-establish the Service Level Connection once the call is dropped.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist.

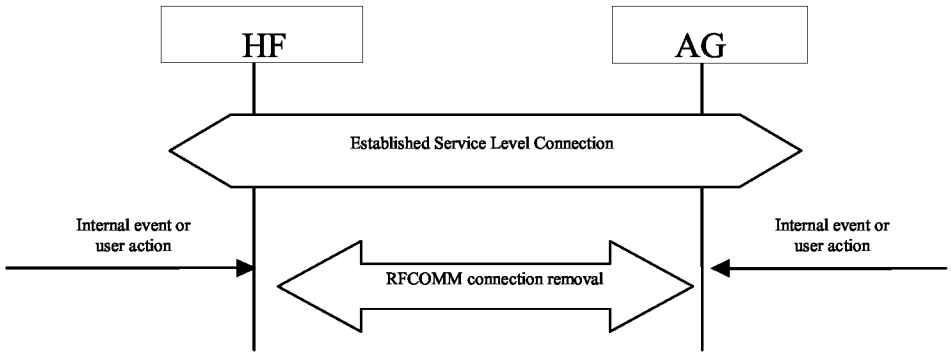


Figure 4.2: Service Level Connection removal

4.4 Transfer of Registration Status

The AT+CMER command, as described in Section 4.2, enables the "Registration status update" function in the AG. When this function is enabled, the AG shall send the +CIEV unsolicited result code with the corresponding service indicator and value whenever the AG's registration status changes. The HF unit shall be capable of interpreting the information provided by the +CIEV result code to determine the service availability status as listed in Section 4.33.2.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the AG shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

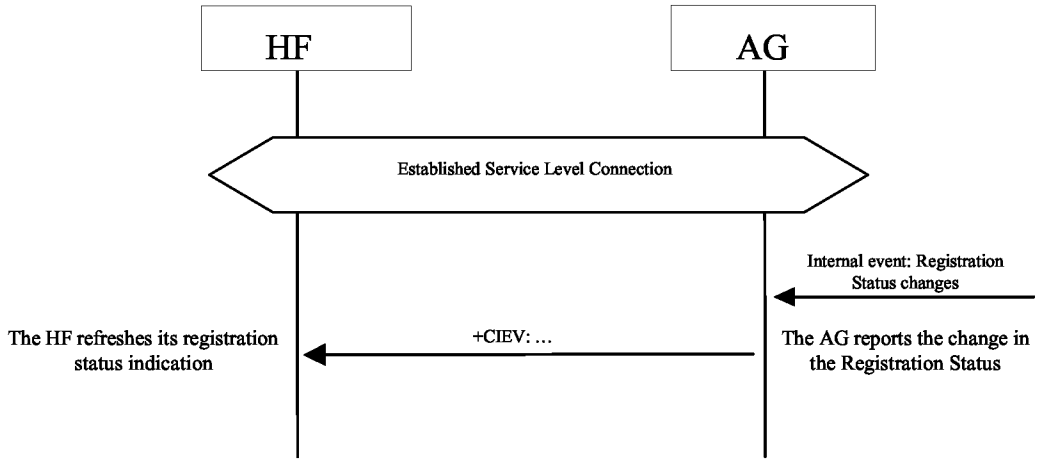


Figure 4.3: Typical Registration Status update

4.5 Transfer of Signal Strength Indication

This procedure enables the AG to send to HF unsolicited result codes with the signal strength values.

The following pre-condition applies for this procedure:

- An ongoing connection between the AG and the HF shall exist. If this connection does not exist, the AG shall establish a connection using "Service Level Connection set up" procedure described in section 4.2.

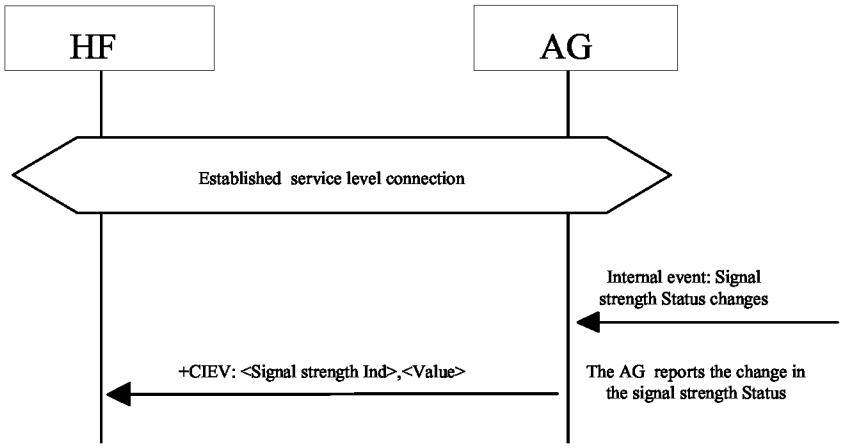


Figure 4.4: Transfer of Signal strength indication

- As a result, the AG shall send the +CIEV unsolicited result code with the corresponding signal strength value whenever its signal strength changes.

4.6 Transfer of Roaming Status Indication

This procedure enables the HF to know the "Roaming Status" of the AG.

The AT+CMER command, as described in Section 4.24.2, enables the "Roaming status Indication" in the AG. As a result, the AG shall send the +CIEV unsolicited result code with the corresponding indicator values whenever its roaming status changes.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall establish the Service Level Connection using the proper procedure as described in Section 4.2.

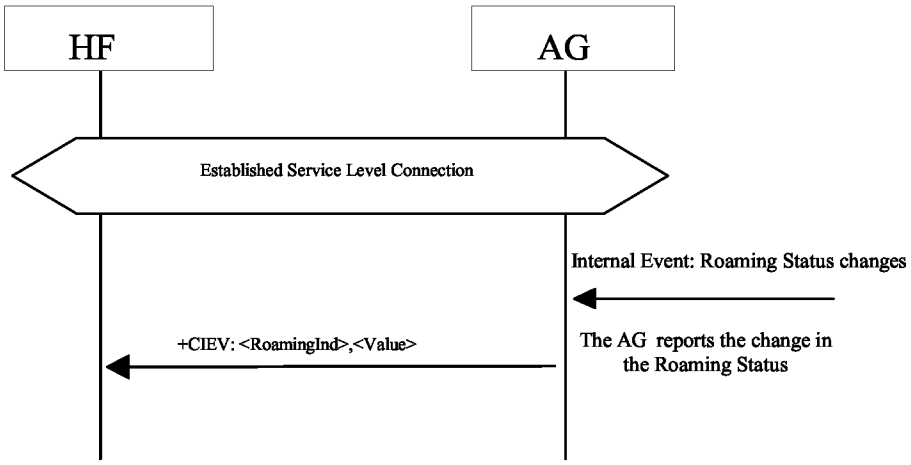


Figure 4.5: Transfer of Roaming Status Indication

4.7 Transfer of Battery Level Indication of AG

This procedure allows the HF to know the battery level of the AG.

The AT+CMER command, as described in Section 4.24.2, enables the "Battery level update" in the AG. As a result, the AG shall send the +CIEV unsolicited result code with the corresponding battery level values whenever its level changes.

The following pre-condition applies for this procedure:

- An ongoing connection between the AG and the HF shall exist. If this connection does not exist, the AG shall establish a connection using "Service Level Connection set up" procedure described in Section 4.2.

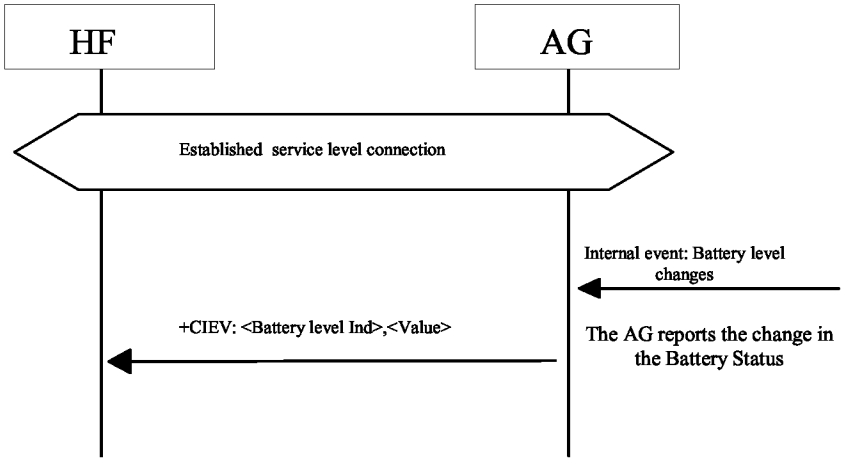


Figure 4.6: Transfer of Battery level indication

4.8 Query Operator Selection

The HF shall execute this procedure to find out the name of the currently selected Network operator by AG.

The following pre-condition applies for this procedure:

- An ongoing connection between the AG and the HF shall exist. If this connection did not exist, the AG shall establish a connection using "Service Level Connection set up" procedure described in Section 4.2.

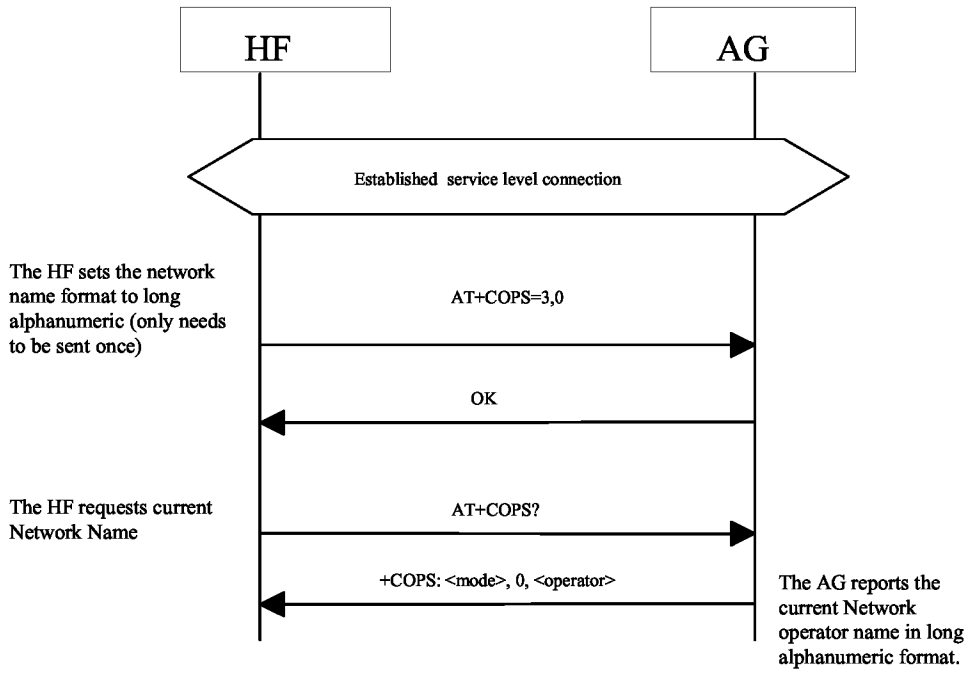


Figure 4.7: Query currently selected Network operator

- HF shall send AT+COPS=3,0 command to set name format to long alphanumeric. Long alphanumeric is defined as a maximum of 16 characters. The value of 3 as the first parameter indicates that this command is only affecting the format parameter (the second parameter). The second parameter, 0, is the value required to set the format to "long alphanumeric."
- Upon an internal event or user-initiated action, HF shall send the AT+COPS? (Read) command to find the currently selected operator.
- AG shall respond with +COPS response indicating the currently selected operator. If no operator is selected, <format> and <operator> are omitted.

4.9 Report Extended Audio Gateway Error Results Code

The HF shall execute this procedure to enable/disable Extended Audio Gateway Error result codes in the AG.

The following pre-condition applies for this procedure:

- An ongoing connection between the AG and the HF shall exist. If this connection did not exist, the AG shall establish a connection using "Service Level Connection set up" procedure described in section 4.2.

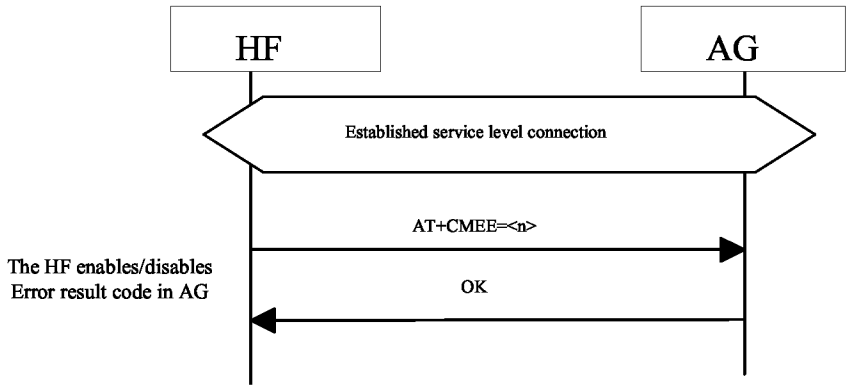


Figure 4.8: Enable/Disable AG Error result code

- The HF shall issue the AT+CMEE command to enable/disable the "Extended Audio Gateway Error Result Code" in the AG. The parameter <n> controls the activation/deactivation of the "Extended Error result code" notification.
- Whenever there is an error relating to the functionality of the AG as a result of AT command, the AG shall send +CME ERROR: <err> response to the HF.

4.10 Transfer of Call, Call Setup and Held Call Status

The AT+CMER command, as described in Section 4.2, enables the "Call Status indicators update" function in the AG. When this function is enabled, the AG shall issue a +CIEV unsolicited result code with the corresponding call indicator and value whenever the AG's current call status changes. Likewise, the AG shall issue a +CIEV unsolicited result code with the corresponding callsetup indicator and value whenever the AG's current call setup status changes. The AG shall also issue a +CIEV unsolicited result code with corresponding callheld indicator and value whenever the AG's current held call status changes.

The HF unit shall be capable of interpreting the information provided by the +CIEV result code to determine the call status as listed in Section 4.33.2.

Furthermore, the HF unit may also be capable of interpreting the optional callsetup state information provided by the +CIEV result code as listed in Section 4.33.2.

The HF unit shall be able to accept unknown indicators provided by the +CIEV result code. The HF unit may ignore unknown indicators provided by the +CIEV result code.

Note: Although the HF unit is required to parse the +CIEV result codes, the HF unit is not required to provide User Interface indicators for the +CIEV result codes.

4.11 Audio Connection Setup

Upon a user action or an internal event, either the HF or the AG shall initiate the establishment of an Audio Connection whenever necessary. Further internal actions may be needed by the HF or the AG to internally route the audio paths.

An Audio Connection set up procedure always means the establishment of a Synchronous Connection and it is always associated with an existing Service Level Connection.

In principle, setting up an Audio Connection by using the procedure described in this section is not necessarily related to any call process.

Once an Audio Connection between the HF and the AG exists, the AG shall utilize the HF as its primary audio port. The AG shall keep the audio paths, call related or not, routed towards HF for all the operations (e.g. voice, alert, key press tones) involving presence of audio.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the initiator of the procedure shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

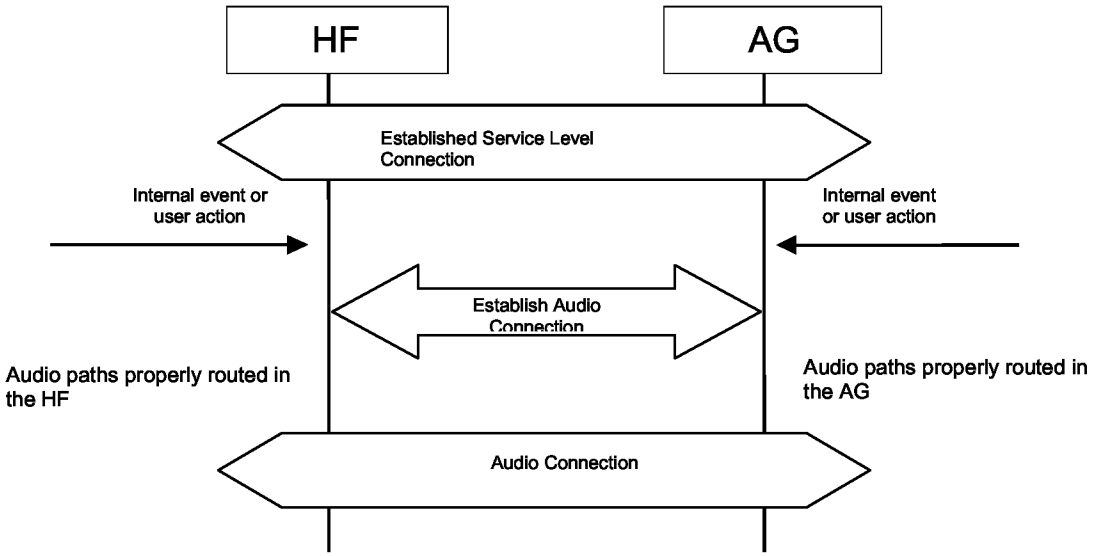


Figure 4.9: Audio Connection set up

Both the initiator and the acceptor shall notify the presence of the new Audio Connection.

4.12 Audio Connection Release

Upon a user action or an internal event, either the HF or the AG shall release an existing Audio Connection whenever necessary.

As pre-condition for this procedure, an ongoing Audio Connection between the AG and the HF shall exist.

An Audio Connection removal always means the disconnection of its corresponding Synchronous Connection.

When the audio connection is released, the audio path shall be routed to the AG.[2]

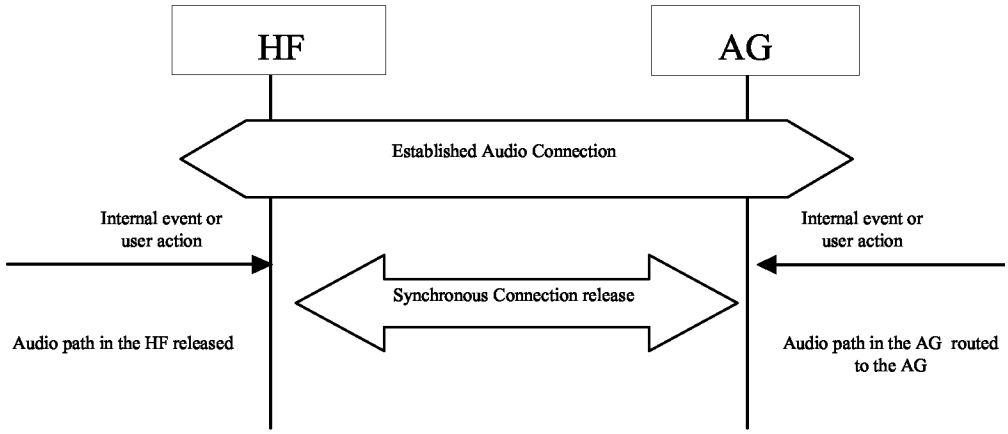


Figure 4.10: Audio Connection release

4.13 Answer an Incoming Call

Upon an incoming call, the AG shall send a sequence of unsolicited RING alerts to the HF. The RING alert shall be repeated for as long as the call acceptance is pending, or until the incoming call is interrupted for any reason.

The HF shall produce a local alerting in reaction to the RING.

If the AG's SDP record (or +BRSF message) indicates "In-band ring tone" is supported, the AG shall send in-band ring tones unless subsequently changed using procedures defined in Section 4.13.4.

The AG may abort the incoming call when necessary. It shall then stop sending the RING alert to the HF.

4.13.1 Answer Incoming Call from the HF – In-Band Ringing

Optionally, the AG may provide an in-band ring tone.

This case is described in Figure 4.11 below and implies, as pre-condition, that an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the AG shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

As the figure below shows, if an in-band ring tone is used, the AG shall send the ring tone to the HF via the established Audio Connection.

[2] In principle, removing an Audio Connection by using the procedure described in this section is not necessarily related to any call process.

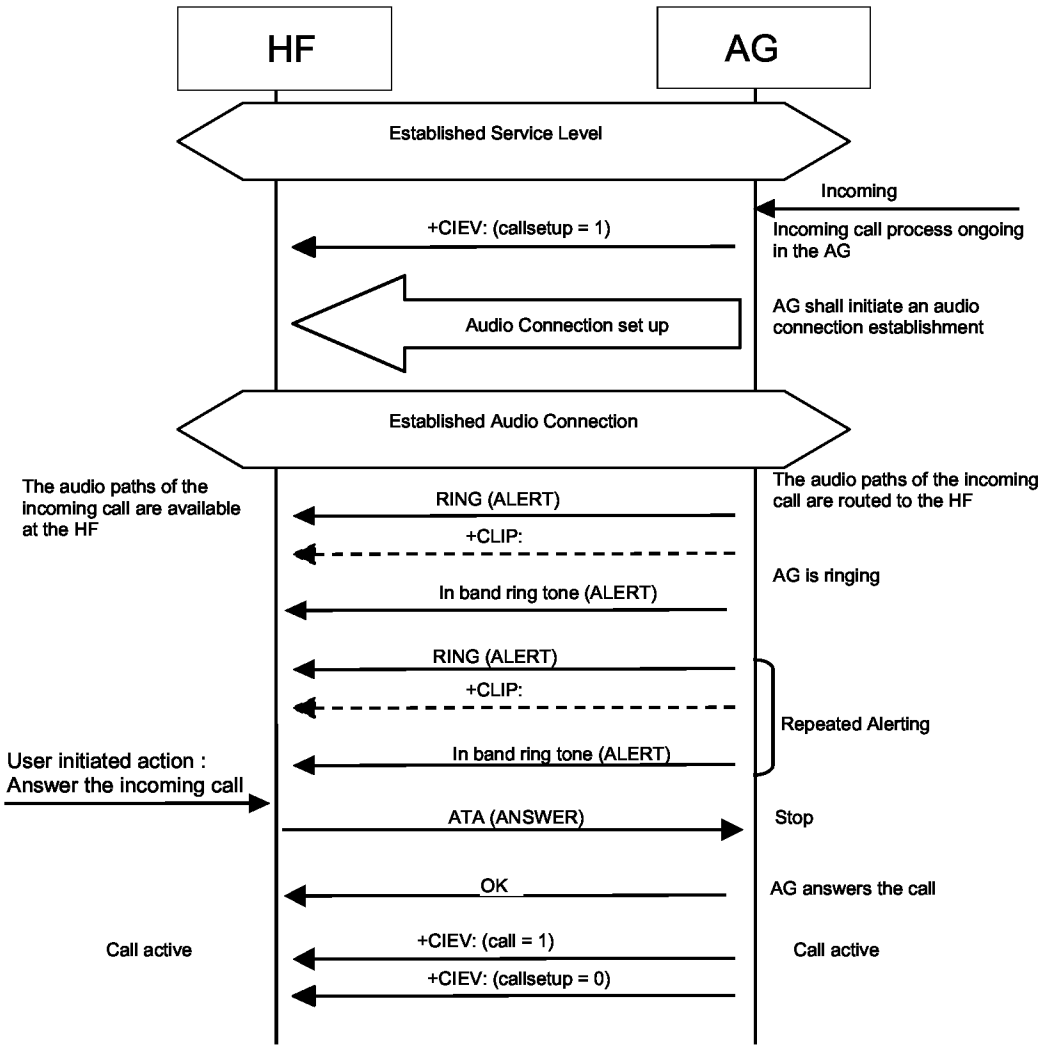


Figure 4.11: Answer an incoming call from the HF – in-band ring tone

The user accepts the incoming voice call by using the proper means provided by the HF. The HF shall then send the ATA command (see Section 4.33) to the AG. The AG shall then begin the procedure for accepting the incoming call.

If the normal incoming call procedure is interrupted for any reason, the AG shall issue the +CIEV result code, with the value indicating (callsetup=0) to notify the HF of this condition (see also Section 4.14.2).

4.13.2 Answer Incoming Call from the HF – No In-Band Ringing

As pre-condition, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the AG shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

As the figure below shows, if no in-band ring tone is used and an Audio Connection does not exist, the AG shall set up the Audio Connection and route the audio paths to the HF upon answering the call.

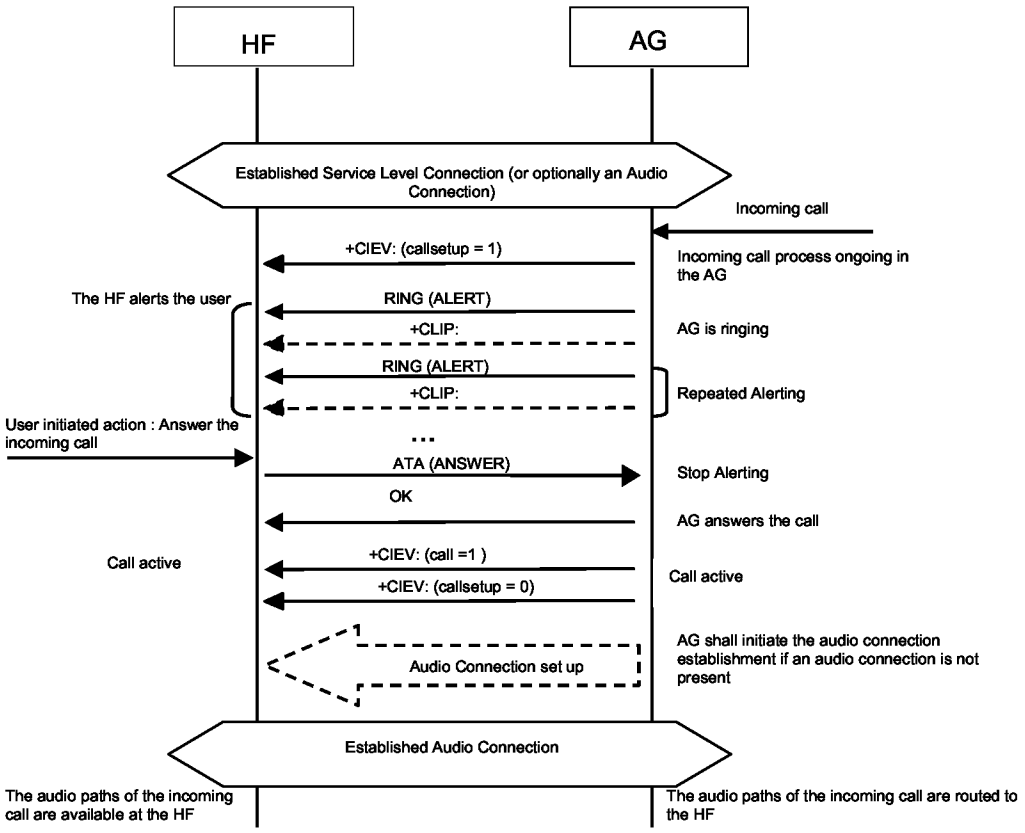


Figure 4.12: Answer an incoming call from the HF – no in-band ring tone

The user accepts the incoming voice call by using the proper means provided by the HF. The HF shall then send the ATA command (see Section 4.33) to the AG, and the AG shall start the procedure for accepting the incoming call and establishing the Audio Connection if an Audio Connection does not exist (refer to Section 4.11).

If the normal incoming call procedure is interrupted for any reason, the AG shall issue the +CIEV result code, with the value indicating (callsetup=0) to notify the HF of this condition (see also Section 4.14.2).

4.13.3 Answer Incoming Call from the AG

The following pre-conditions apply for this procedure:

- As a pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist.
- The AG shall alert the HF using either of the two procedures described in Sections 4.13.1 and 4.13.2.
- The HF shall alert the user.

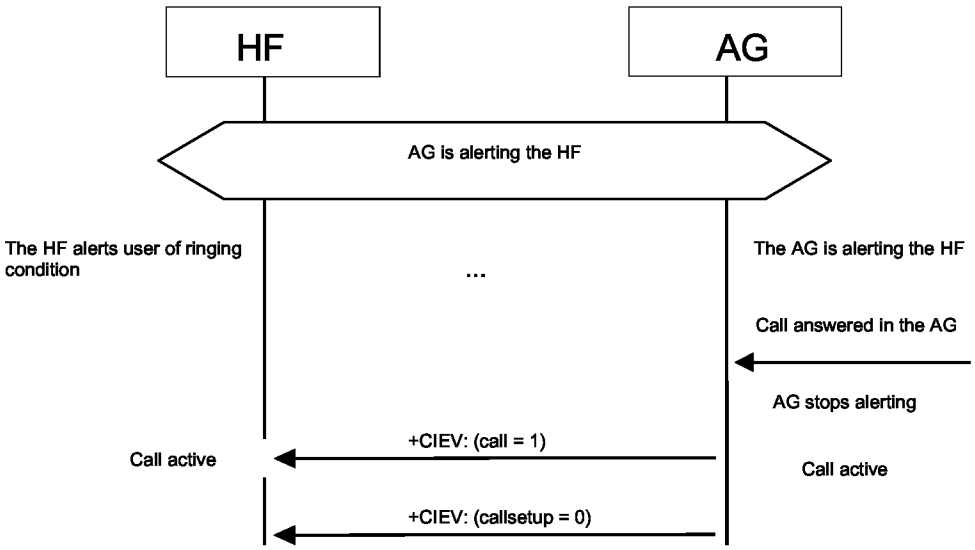


Figure 4.13: Answer an incoming call from the AG

The user accepts the incoming call by using the proper means provided by the AG.

If the normal incoming call procedure is interrupted for any reason, the AG shall issue the +CIEV result code, with the value indicating (callsetup=0) to notify the HF of this condition (see also Section 4.14.2).

4.13.4 Change the In-Band Ring Tone Setting

The SDP record entry “In-band ring tone” of the “Supported features” record (see table 5.4) informs the HF if the AG is capable of sending an in-band ring tone or not. If the AG is capable of sending an in-band ring tone, it shall send the in-band ring tone by default. The AG may subsequently change this setting.

In case the AG wants to change the in-band ring tone setting during an ongoing service level connection, it shall use the unsolicited result code +BSIR (Bluetooth Set In-band Ring tone) to notify the HF about the change. See Figure 4.14 for details.

Refer to Section 4.33 for more information on the +BSIR unsolicited result code.

The in-band ring tone setting may be changed several times during a Service Level Connection.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the AG shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

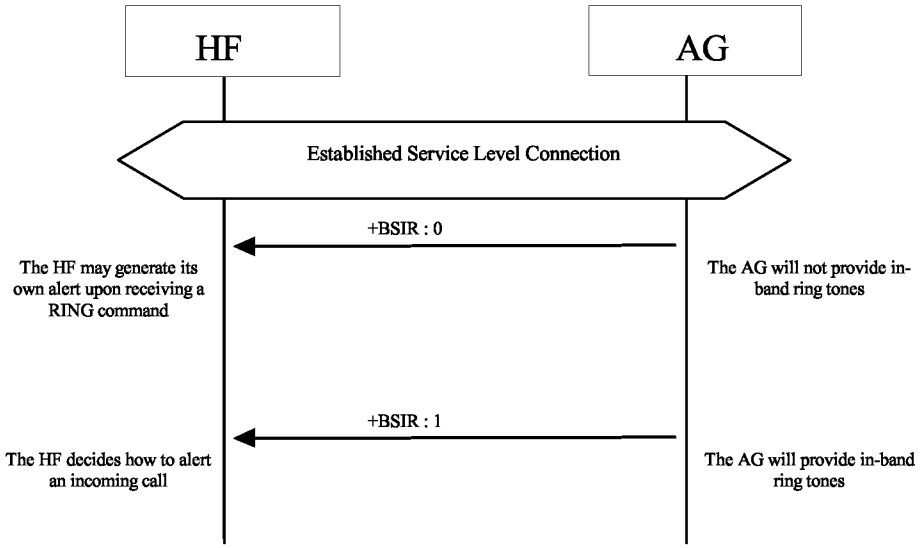


Figure 4.14: Change of the in-band ring tone setting initiated by the AG

In case the HF does not want to use the AG's in-band ring tone, it may mute the Audio Connection after it has received +CIEV:(callsetup=1). The HF shall un-mute the Audio Connection upon receiving the +CIEV:(callsetup=0) indication.

4.14 Reject an Incoming Call

In case of an incoming call, the AG shall alert the HF by either one of the two procedures described in Sections 4.13.1 and 4.13.2.

Instead of answering the call, the user may reject the incoming call process by user action at the HF or the AG. These two procedures are described in the following sections.

4.14.1 Reject an Incoming Call from the HF

As a precondition to this procedure, the AG shall alert the HF using either of the two procedures described in Sections 4.13.1 and 4.13.2.

The user rejects the incoming call by using the User Interface on the Hands-Free unit. The HF shall then send the AT+CHUP command (see Section 4.33) to the AG. This may happen at any time during the procedures described in Sections 4.13.1 and 4.13.2.

The AG shall then cease alerting the HF of the incoming call and send the OK indication followed by the +CIEV result code, with the value indicating (callsetup=0).

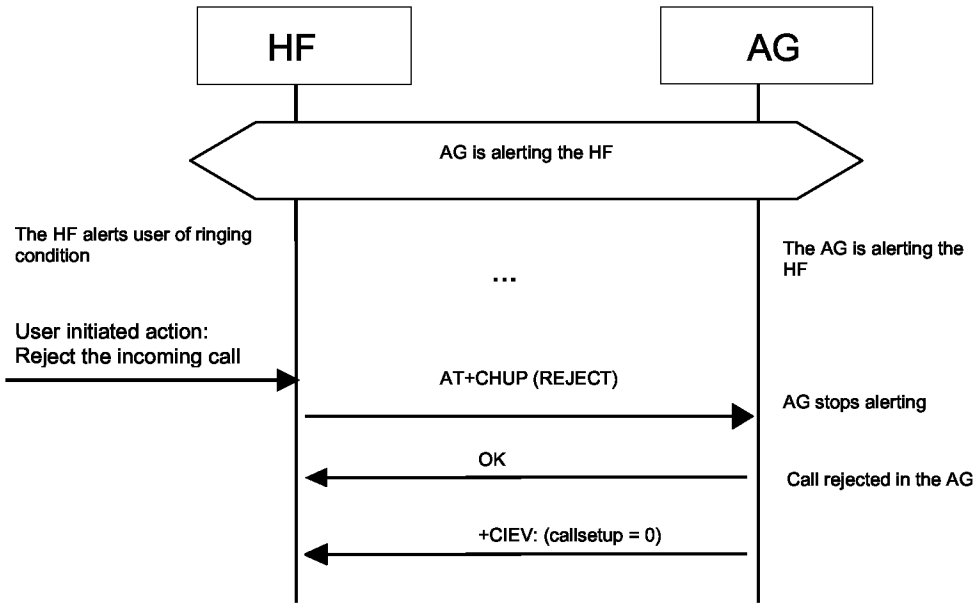


Figure 4.15: Reject an incoming call from the HF

4.14.2 Rejection/Interruption of an Incoming Call in the AG

As a precondition to this procedure, the AG shall alert the HF using either of the two procedures described in Sections 4.13.1 and 4.13.2.

The user rejects the incoming call by using the User Interface on the AG. Alternatively the incoming call process may be interrupted in the AG for any other reason.

As consequence of this, the AG shall send the +CIEV result code, with the value indicating (callsetup=0). The HF shall then stop alerting the user.

This may happen at any time during the procedures described in Sections 4.13.1 and 4.13.2.

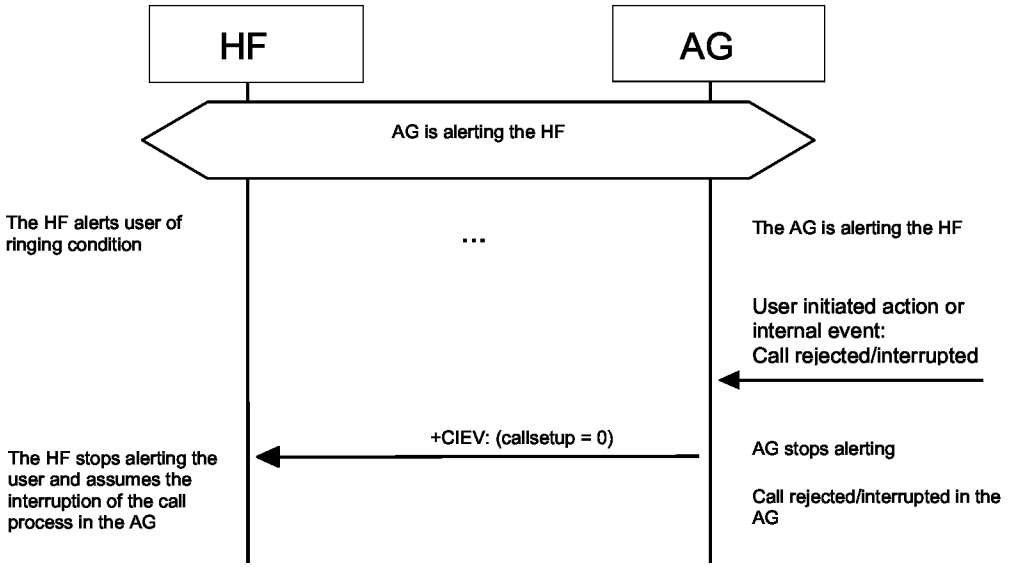


Figure 4.16: Rejection/interruption of an incoming call in the AG

4.15 Terminate a Call Process

An ongoing call process may be terminated by either the HF or the AG by means of a user action or any other event.

4.15.1 Terminate a Call Process from the HF

The following pre-conditions apply for this procedure:

- An ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.
- A call related process is ongoing in the AG.

Although not required for the call termination process, an Audio Connection is typically present between the HF and AG.

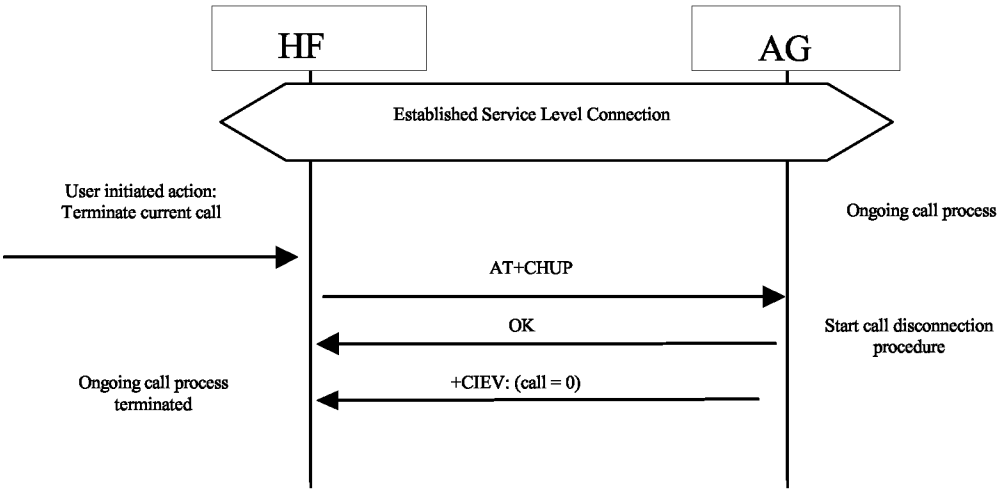


Figure 4.17: Terminate ongoing call - HF initiated

The user may abort the ongoing call process using whatever means provided by the Hands-Free unit. The HF shall send AT+CHUP command (see Section 4.33) to the AG, and the AG shall then start the procedure to terminate or interrupt the current call procedure. The AG shall then send the OK indication followed by the +CIEV result code, with the value indicating (call=0).

Performing a similar procedure, the AT+CHUP command described above may also be used for interrupting a normal outgoing call set-up process.

4.15.2 Terminate a Call Process from the AG

The following pre-conditions apply for this procedure:

- An ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the AG shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.
- A call related process is ongoing in the AG.

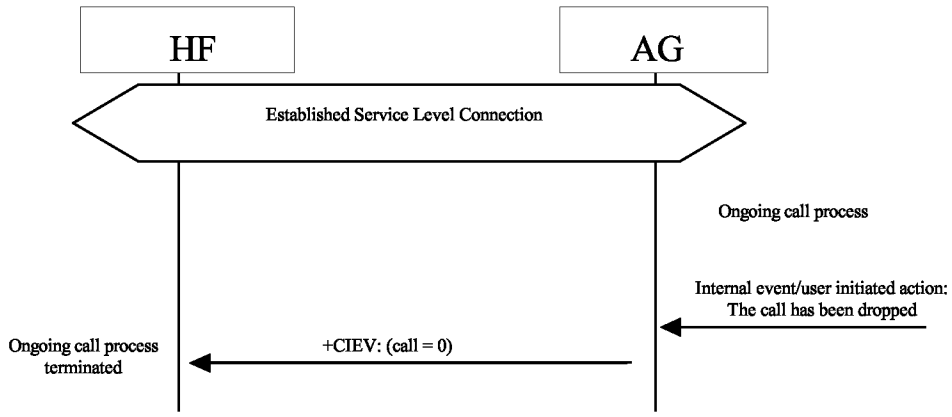


Figure 4.18: Terminate ongoing call - AG initiated

This procedure is fully applicable for cases in which an ongoing call process is interrupted in the AG for any reason.

In this case the AG shall send the +CIEV result code, with the value indicating (call=0).

4.16 Audio Connection Transfer Towards the HF

The audio paths of an ongoing call may be transferred from the AG to the HF. This procedure represents a particular case of an “Audio Connection set up” procedure, as described in Section 4.11.

The call connection transfer from the AG to the HF is initiated by a user action either on the HF or on the AG side. This shall result in either the HF or the AG, respectively, initiating an “Audio Connection set up” procedure with the audio paths of the current call being routed to the HF.

This procedure is only applicable if there is no current Audio Connection established between the HF and the AG. In fact, if the Audio Connection already exists, this procedure is not necessary because the audio path of the AG is assumed to be already routed towards the HF.

The following pre-conditions apply for this procedure:

- An ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the initiator of the “Audio Connection transfer towards the HF” procedure shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.
- An ongoing call exists in the AG, with the audio paths routed to the AG means.

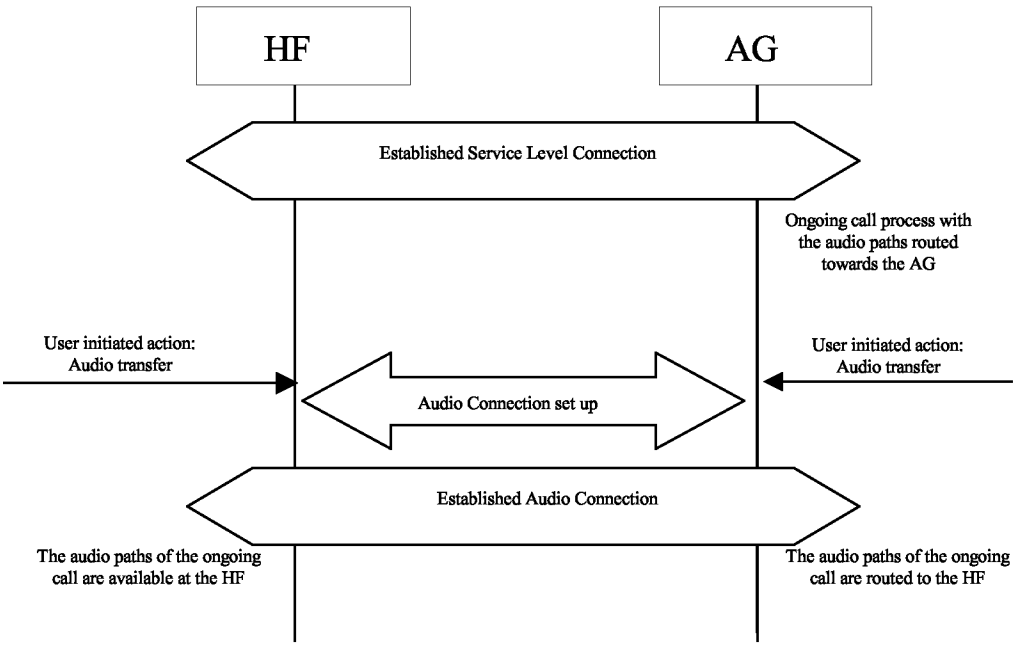


Figure 4.19: Audio Connection transfer to the HF

4.17 Audio Connection Transfer Towards the AG

The audio paths of an ongoing call may be transferred from the HF to the AG. This procedure represents a particular case of an “Audio Connection release” procedure, as described in Section 4.12.

The call connection transfer from the HF to the AG is initiated by a user action in the HF or due to an internal event or user action on the AG side. This results in an “Audio Connection release” procedure being initiated either by the HF or the AG respectively, with the current call kept and its audio paths routed to the AG.

If as a consequence of an HF initiated “Audio Connection transfer towards the AG” procedure, the existing Service Level Connection is autonomously removed by the AG, the AG shall attempt to re-establish the Service Level Connection once the current call ends.

As pre-condition for this procedure, an ongoing call process shall exist in the AG. The audio paths of the ongoing call shall be available in the HF via an Audio Connection established between the AG and the HF.

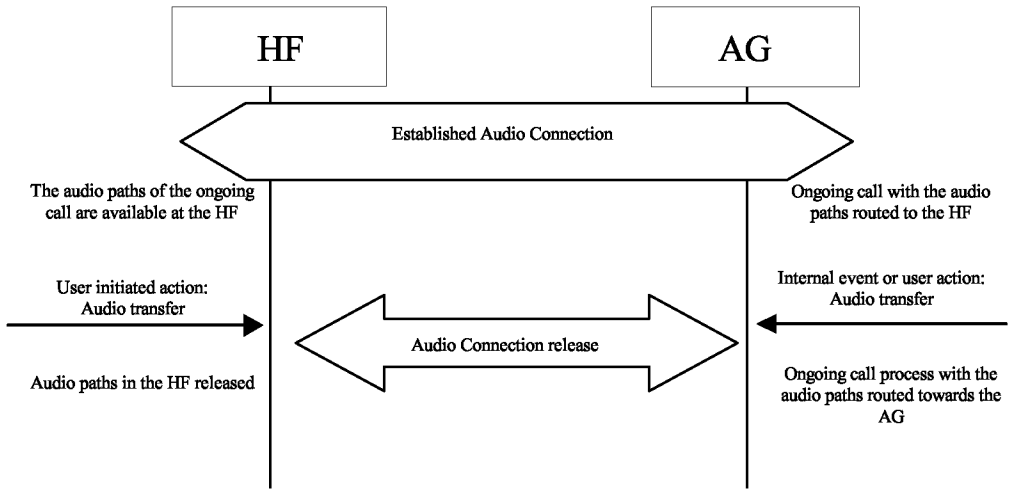


Figure 4.20: Audio Connection transfer to the AG

4.18 Place a Call With the Phone Number Supplied by the HF

The HF may initiate outgoing voice calls by providing the destination phone number to the AG. To start the call set-up, the HF shall initiate the Service Level Connection establishment (if necessary) and send a proper ATDdd…dd; command to the AG. The AG shall then start the call establishment procedure using the phone number received from the HF and issues the +CIEV result code, with the value (callsetup=2) to notify the HF that the call set-up has been successfully initiated.

Refer to Section 4.33 for more information on the ATDdd…dd; command.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

If an Audio Connection is not established the AG shall establish the proper Audio Connection and route the audio paths of the outgoing call to the HF immediately following the commencement of the ongoing call set up procedure.

Once the AG is informed that the alerting of the remote party has begun, the AG shall issue the +CIEV result code, with the value indicating (callsetup=3). If the wireless network does not provide the AG of an indication of alerting the remote party, the AG may not send this indication.

Upon call connection the AG shall send issue the +CIEV result code, with the value indicating (call=1).

If the normal outgoing call establishment procedure is interrupted for any reason, the AG shall issue the +CIEV result code, with the value indicating (callsetup=0), to notify the HF of this condition (see Section 4.15.2).

If the AG supports the “Three-way calling” feature and if a call is already ongoing in the AG, performing this procedure shall result in a new call being placed to a third party with

the current ongoing call put on hold. For details on how to handle multiparty calls refer to Section 4.22.2.

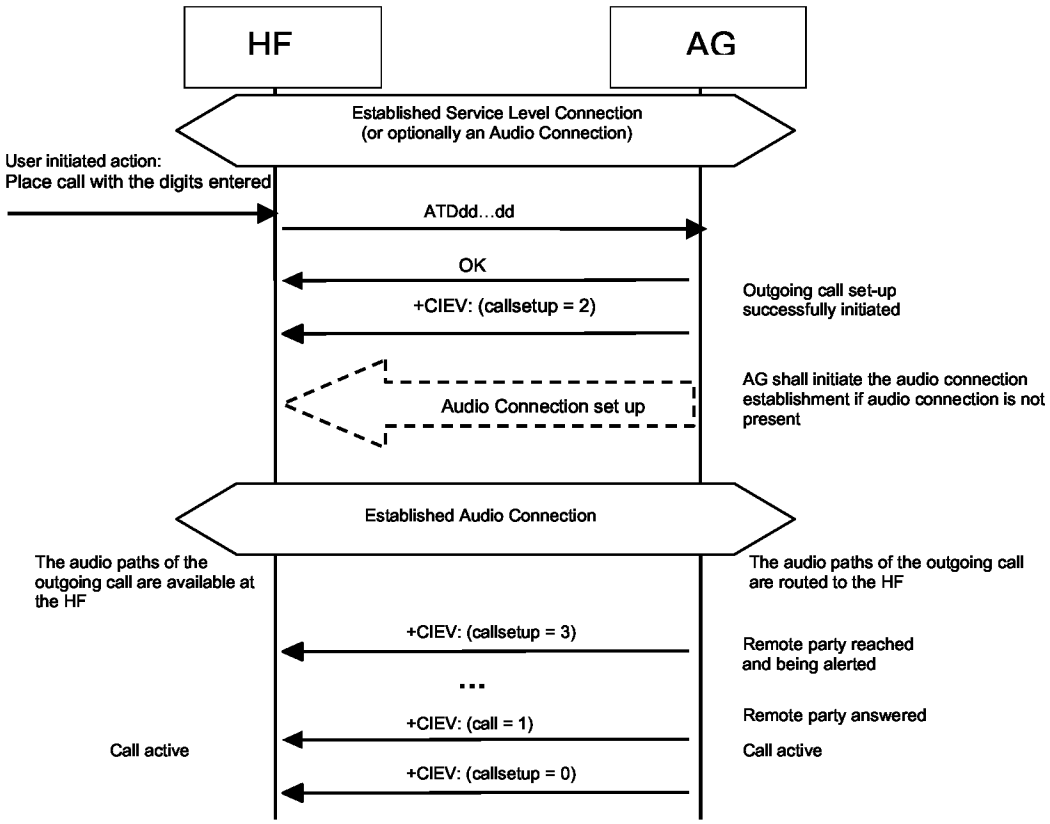


Figure 4.21: Place an outgoing voice call with the digits entered in the HF

4.19 Memory Dialing from the HF

The HF may initiate outgoing voice calls using the memory dialing feature of the AG. To start the call set-up, the HF shall initiate the Service Level Connection establishment (if necessary) and send an ATD>nnn…; command to the AG. The AG shall then start the call establishment procedure using the phone number stored in the AG memory location given by nnn…; and issue the +CIEV result code, with the value (callsetup=2) to notify the HF that the call set-up has been successfully initiated.

Refer to Section 4.33 for more information on the ATD>nnn… command.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

If an Audio Connection is not established, the AG shall establish the proper Audio Connection and route the audio paths of the outgoing call to the HF immediately following the commencement of the ongoing call set up procedure.

Once alerting of the remote party begins, the AG shall issue the +CIEV result code, with the value indicating (callsetup=3).

Upon call connection the AG shall send issue the +CIEV result code, with the value indicating (call=1).

If the normal outgoing call establishment procedure is interrupted for any reason, the AG shall issue the +CIEV result code, with the value indicating (callsetup=0), to notify the HF of this condition (see Section 4.15.2).

If the AG supports the "Three-way calling" feature and if a call is already ongoing in the AG, performing this procedure shall result in a new call being placed to a third party with the current ongoing call put on hold. For details on how to handle multiparty calls refer to Section 4.22.2.

If there is no number stored for the memory location given by the HF, the AG shall respond with ERROR.

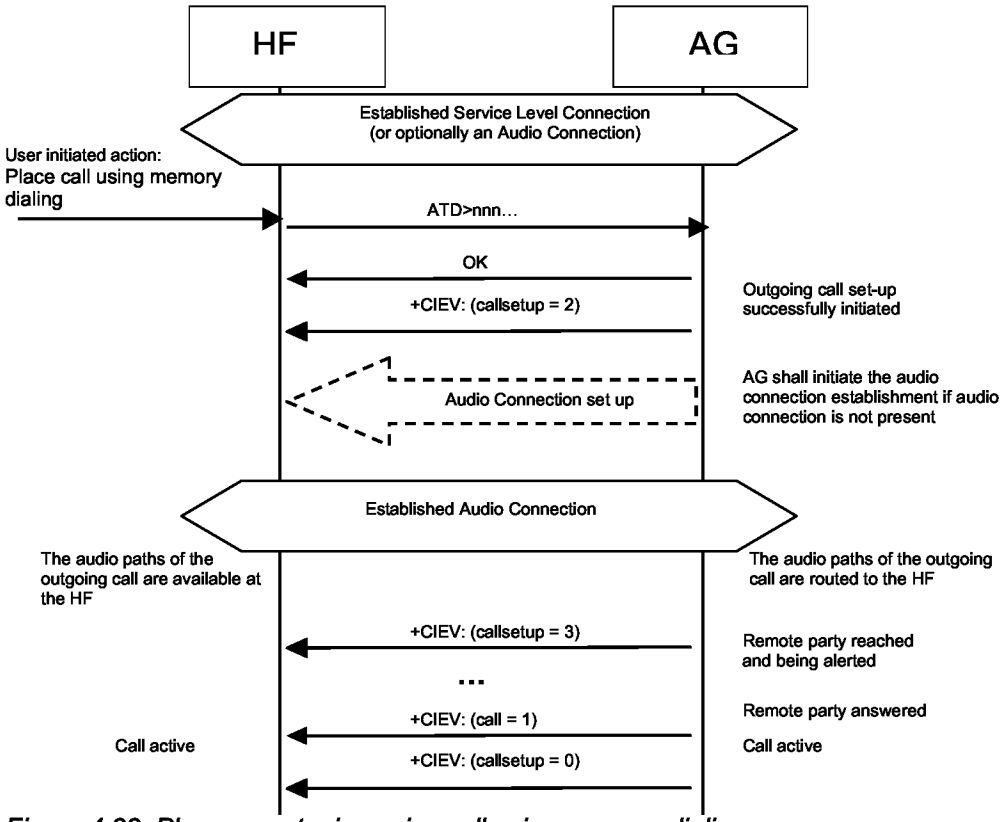


Figure 4.22: Place an outgoing voice call using memory dialing

4.20 Last Number Re-Dial from the HF

The HF may initiate outgoing voice calls by recalling the last number dialed by the AG. To start the call set-up, the HF shall initiate the Service Level Connection establishment (if necessary) and send an AT+BLDN command to the AG. The AG shall then start the call establishment procedure using the last phone number dialed by the AG, and issues the +CIEV result code, with the value (callsetup=2), to notify the HF that the call set-up has been successfully initiated.

Refer to Section 4.33 for more information on the AT+BLDN command.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

If an Audio Connection is not established, the AG shall establish the proper Audio Connection and route the audio paths of the outgoing call to the HF immediately following the commencement of the ongoing call set up procedure.

Once alerting of the remote party begins, the AG shall issue the +CIEV result code, with the value indicating (callsetup=3).

Upon call connection the AG shall send issue the +CIEV result code, with the value indicating (call=1).

If the normal outgoing call establishment procedure is interrupted for any reason, the AG shall issue the +CIEV result code, with the value indicating (callsetup=0), to notify the HF of this condition (see Section 4.15.2).

If the AG supports the “Three-way calling” feature and if a call is already ongoing in the AG, performing this procedure shall result in a new call being placed to a third party with the current ongoing call put on hold. For details on how to handle multiparty calls refer to Section 4.22.2.

If there is no number stored for the memory location given by the HF, the AG shall respond with ERROR.

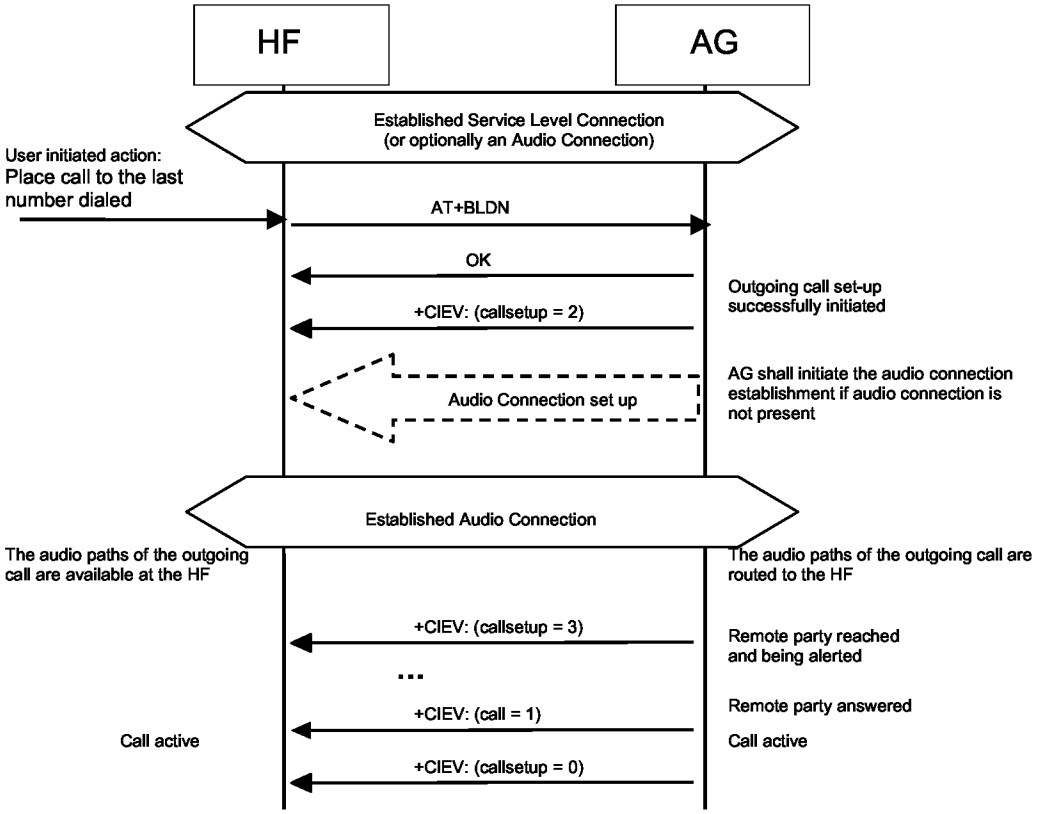


Figure 4.23: Place an outgoing voice call with the last number dialed

4.21 Call Waiting Notification Activation

The HF may issue the AT+CCWA command to enable the "Call Waiting notification" function in the AG. Once the "Call Waiting notification" is enabled, the AG shall send the corresponding +CCWA unsolicited result code to the HF whenever an incoming call is waiting during an ongoing call. It is always assumed that the "call waiting" service is already active in the network.

Once the HF issues the AT+CCWA command, the AG shall respond with OK. It shall then keep the "Call Waiting notification" enabled until either the AT+CCWA command is issued to disable "Call Waiting notification," or the current Service Level Connection between the AG and the HF is dropped for any reason.

Refer to Section 4.33 for more information on the AT+CCWA command.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

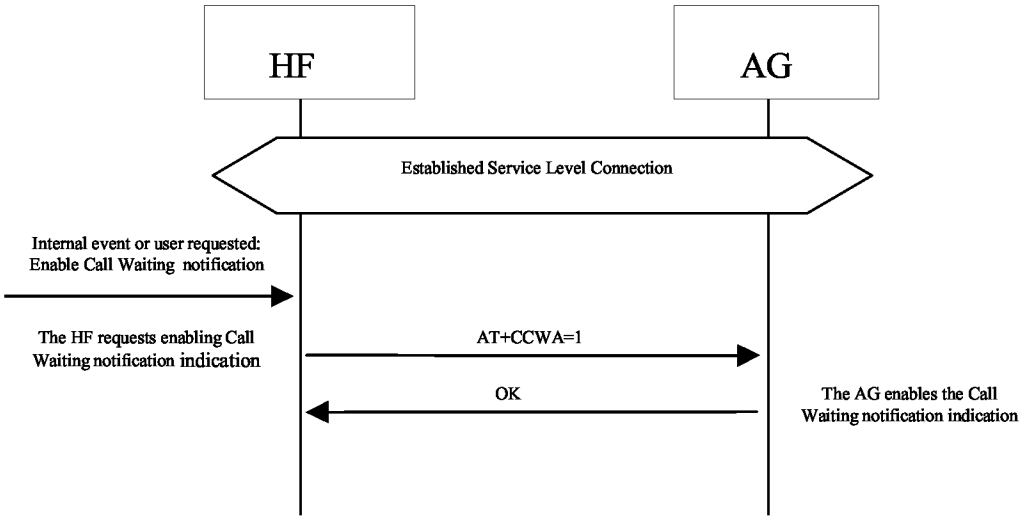


Figure 4.24: Activation of Call waiting notification

4.22 Three Way Call Handling

Proper management of several concurrent calls shall be accomplished by performing the procedures described in [2] but with some limitations stated in this specification. For more details, refer to Section 4.33.

The HF device cannot always assume that the "call hold and/or multiparty" services are available in the network. If the AG determines that a requested action by the HF device cannot be performed due to the inability of the network to support that feature or lack of subscriber subscription, the AG shall return a +CME error.

There are two +CME ERROR codes that are used to indicate network related failure reasons to the HF :

30 - No Network Service. Indicates that an AT+CHLD command cannot be implemented due to network limitations.

31 - Network Timeout. Indicates that an AT+CHLD command cannot be implemented due to network problems.

In general, when the user deals with multiple concurrent calls, the HF shall issue the corresponding AT+CHLD command as a result of user actions. This command allows the control of multiple concurrent calls and provides means for holding calls, releasing calls, switching between two calls, and adding a call to a multiparty conference.

When this feature is supported, the HF and AG are only mandated to implement the “basic Three Way calling” commands AT+CHLD = 1 and 2.

This section covers two cases. In one case the third party call is received in the AG, and notification is sent to the HF via a Call Waiting notification. In the second case, the third party call is placed from the HF.

Refer to Section 4.33 for more information on the AT+CHLD command.

The following pre-conditions apply for these procedures:

- As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the initiator of the procedure shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.
- An ongoing call in the AG shall exist.

4.22.1 Three Way Calling—Call Waiting Notification

In addition to the two previously stated preconditions, the Call Waiting notification to the HF shall already be enabled in the AG (that is, the procedure stated in Section 4.21 has been performed).

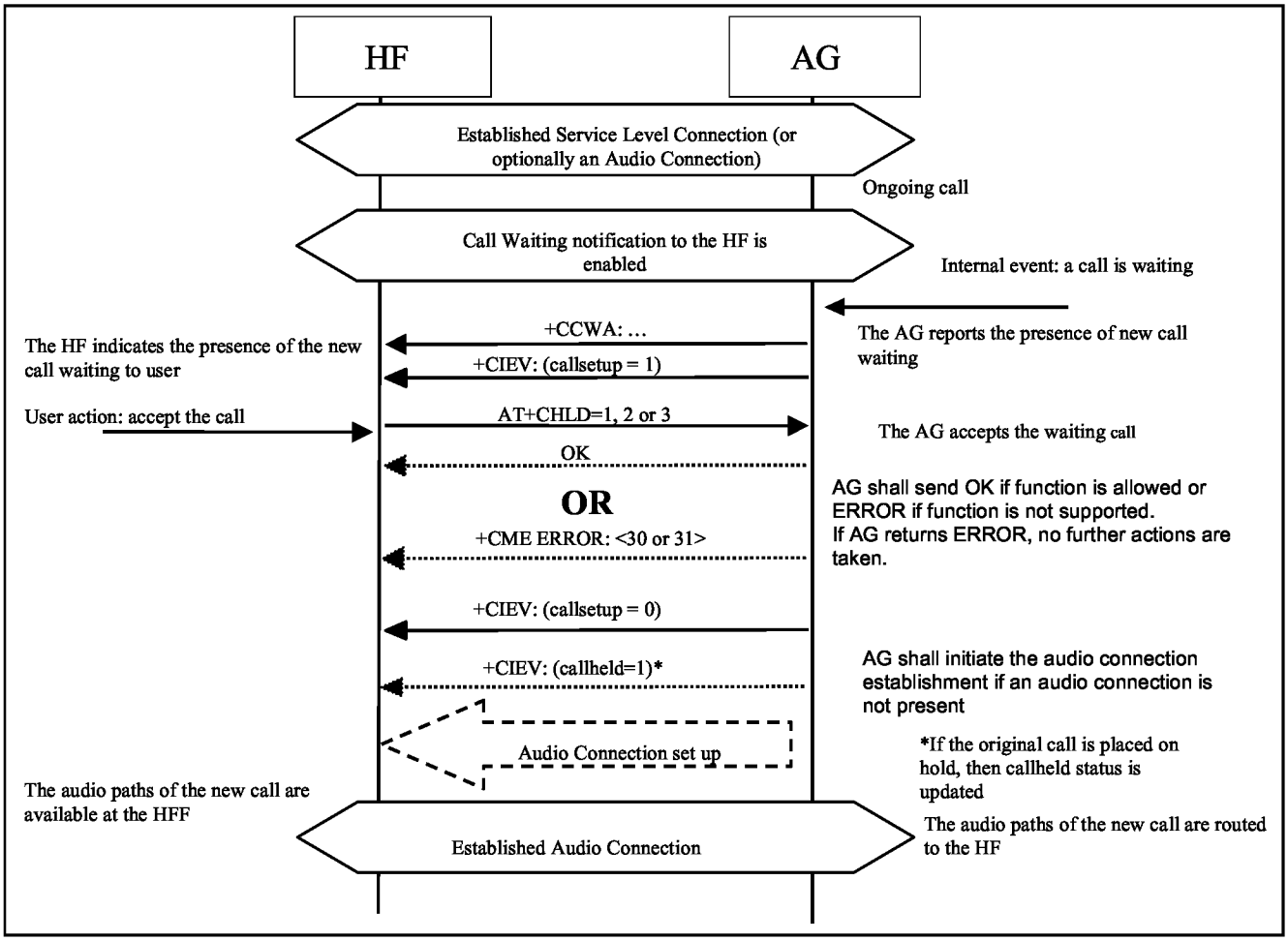


Figure 4.25: Typical Call Waiting indication followed by a three way call set up process

If the AG receives a third party call, it shall send the call waiting notification +CCWA and +CIEV result code, with the value indicating (callsetup=1), to the HF. If the user accepts the call at the HF, it shall send the AT+CHLD with parameter 1, 2 or 3 to the AG. The AG shall then accept the waiting call and respond with OK, and issue the +CIEV result code with the value indicating (callsetup=0). If the HF elects to send AT+CHLD=2 (placing the original call on hold), then the AG shall send the +CIEV result code with the value indicating a held call (callheld=1).

Optionally, the HF may then use the AT+CHLD command, in order to change the status of the held and active calls.

If the normal incoming call procedure is interrupted for any reason, the AG shall issue the +CIEV result code, with the value indicating (callsetup=0), to notify the HF of this condition (see Section 4.14.2).

4.22.2 Three Way Calls – Third Party Call Placed from the HF

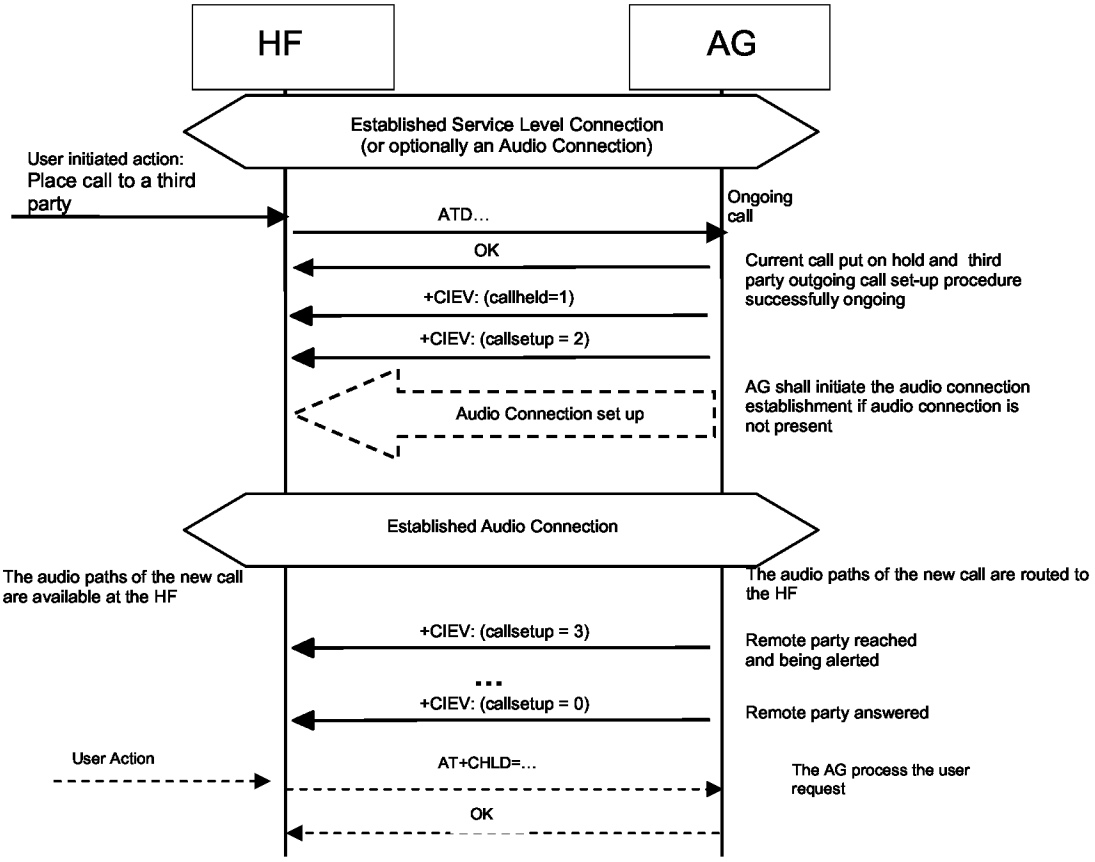


Figure 4.26: Three way call handling when the third party call is placed from the HF

If a third party call is placed from the HF using the ATD command, the AG shall send the OK indication and +CIEV result code, with the value indicating (callsetup=2), to the HF. The AG shall then place the active call in hold status in order to establish the new call.

Once the AG is informed that the alerting of the remote party has begun, the AG shall issue the +CIEV result code, with the value indicating (callsetup=3). If the wireless network does not provide the AG of an indication of alerting the remote party, the AG may not send this indication.

If the remote party answers the call, the AG shall issue the +CIEV result code with the value indicating (callsetup=0).

Optionally, the HF may then use the AT+CHLD command in order to change the status of the held and active calls. If the AT+CHLD command results in the change in a held call status the AG shall provide the status indication using the +CIEV result code with the value indicating the call held status (callheld=<0,1,2>).

If the normal outgoing call procedure is interrupted for any reason, the AG shall issue the +CIEV result code, with the value indicating (callsetup=0), to notify the HF of this condition (see Section 4.15.2).

4.23 Calling Line Identification (CLI) Notification

The HF may issue the AT+CLIP command to enable the "Calling Line Identification notification" function in the AG.

If the calling subscriber number information is available from the network, the AG shall issue the +CLIP unsolicited result code just after every RING indication when the HF is alerted in an incoming call. See Section 4.13 for more details.

Once the HF issues the AT+CLIP command, the AG shall respond with OK. The AG shall then keep the "Calling Line Identification notification" enabled until either the AT+CLIP command is issued by the HF to disable it, or the current Service Level Connection between the AG and the HF is dropped for any reason.

Refer to Section 4.33 for more information on the AT+CLIP command.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

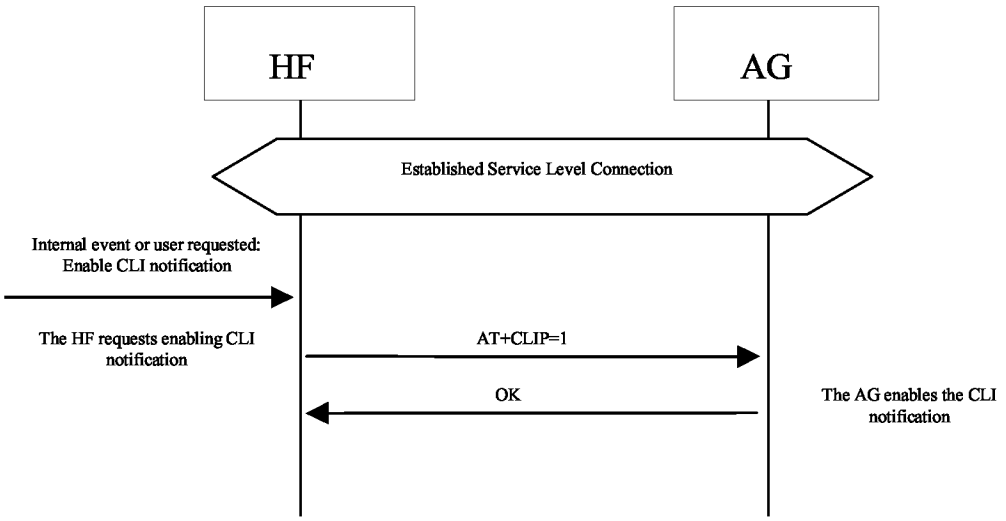


Figure 4.27: Activation of CLI notification

4.24 The HF Requests Turning Off the AG's EC and NR

The HF may disable the echo canceling and noise reduction functions resident in the AG via the AT+NREC command.

If the HF supports embedded EC and/or NR functions it shall support the AT+NREC command as described in the procedures in this section. Moreover, if the HF has these functions enabled, it shall perform this procedure before any Audio Connection between the HF and the AG is established.

By default, if the AG supports its own embedded echo canceling and/or noise reduction functions, it shall have them activated until the AT+NREC command is received. From then on, and until the current Service Level Connection between the AG and HF is

dropped for any reason, the AG shall disable these functions every time an Audio Connection between the HF and the AG is used for audio routing.

If the AG does not support any echo canceling and noise reduction functions, it shall respond with the ERROR indicator on reception of the AT+NREC command.

Refer to Section 4.33 for more information on the AT+NREC command.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

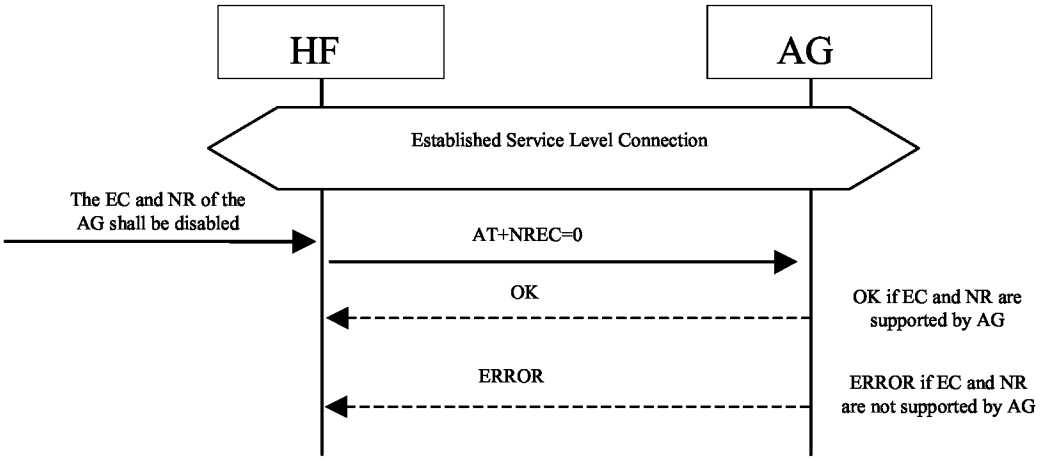


Figure 4.28: NR and EC functions available in the AG

The HF sends the AT+NREC command and AG confirms with either OK or ERROR indication.

4.25 Voice Recognition Activation

The HF, via the AT+BVRA command, or the AG autonomously, may activate/deactivate the voice recognition function resident in the AG. Beyond the audio routing and voice recognition activation capabilities, the rest of the voice recognition functionality is implementation dependent.

Whenever the AG supports a voice recognition function it shall support the AT+BVRA command as described in the procedures in this section.

If the HF issues the AT+BVRA command, the AG shall respond with the OK result code if it supports voice recognition, then initiate an Audio Connection to the HF (if the Audio Connection does not already exist) and begin the voice input sequence.

If the AG does not support voice recognition, the AG shall respond with the ERROR indication.

When the voice recognition function is activated from the AG, it shall inform the HF via the +BVRA: 1 unsolicited result code and the AG shall initiate an Audio Connection to the HF (if the Audio Connection does not already exist) and begin the voice input sequence.

Once activated, depending upon the voice recognition implementation, the AG shall then keep the voice recognition function enabled:

- For the duration of time supported by the implementation ("momentary on" voice recognition implementation). In this case, the AG shall notify the HF by sending a +BVRA: 0 unsolicited result code.
- Or until the AT+BVRA command is issued to disable voice recognition from the HF.
- Or until the current Service Level Connection between the AG and the HF is dropped for any reason.

Refer to Section 4.33 for more information on the AT+BVRA command and the +BVRA result code.

As pre-condition for these procedures, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the initiator of the procedure shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

4.25.1 Voice Recognition Activation – HF Initiated

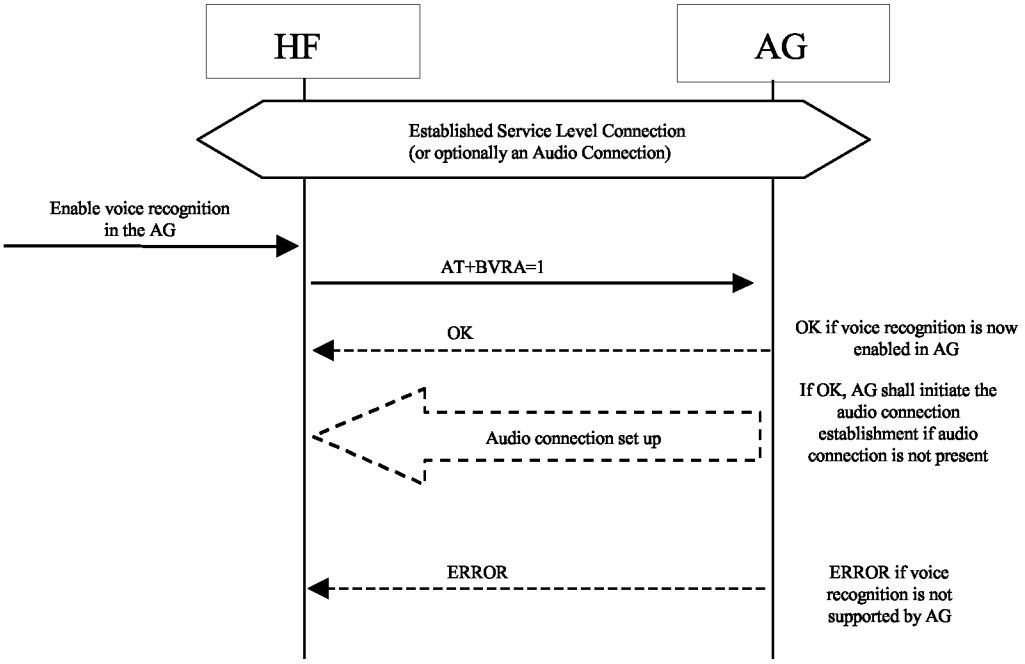


Figure 4.29: Voice recognition activation – HF initiated

4.25.2 Voice Recognition Activation – AG Initiated

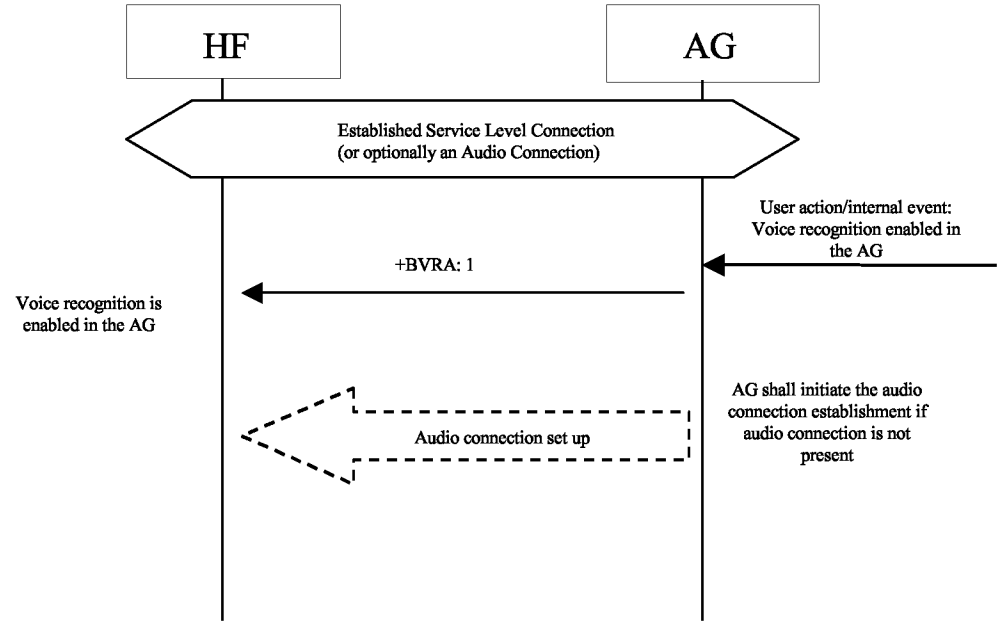


Figure 4.30: Voice recognition activation – AG initiated

4.25.3 Voice Recognition Deactivation

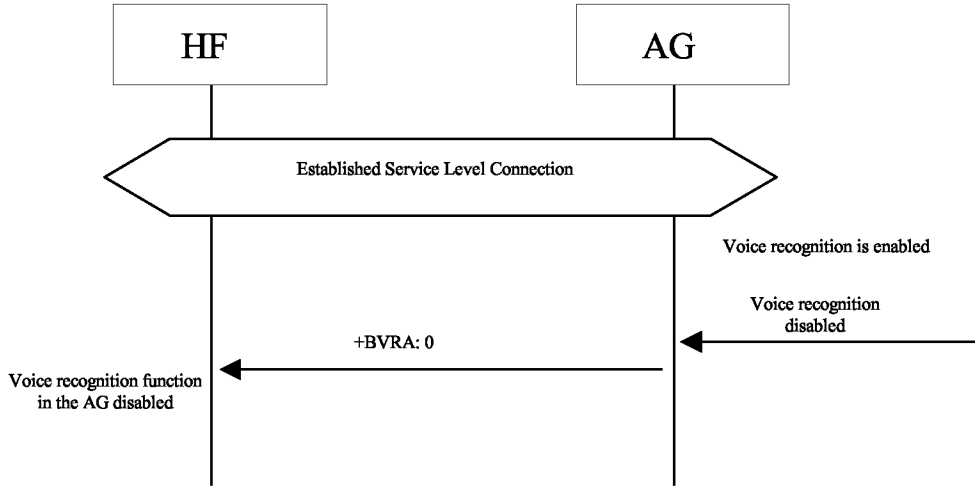


Figure 4.31: Voice recognition deactivation – "momentary on" approach

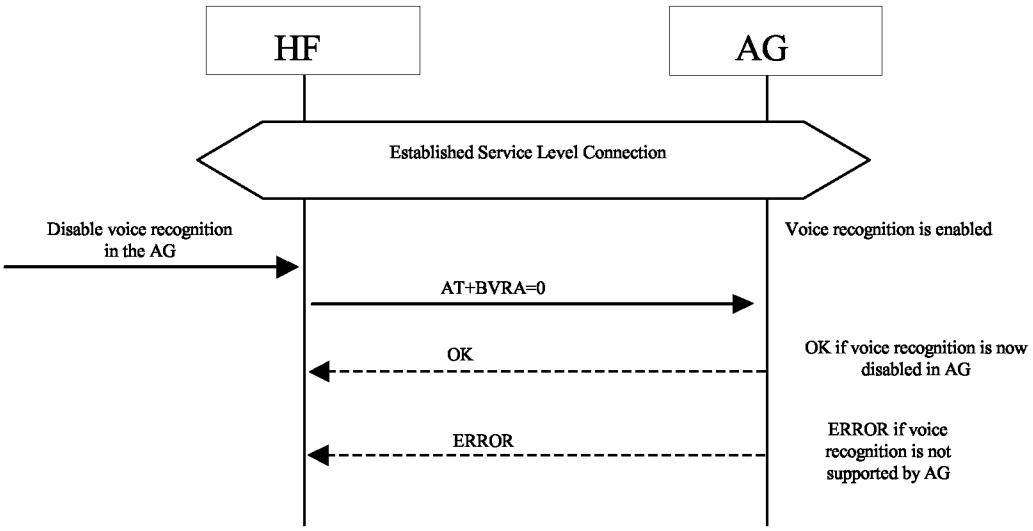


Figure 4.32: Voice recognition deactivation from the HF

4.26 Attach a Phone Number to a Voice Tag

This procedure is applicable to HF units supporting internal voice recognition functionality. It provides a means to read numbers from the AG for the purpose of creating a unique voice tag and storing the number and its linked voice tag in the HF unit's memory. The HF unit may then use its internal Voice Recognition to dial the linked phone numbers when a voice tag is recognized by using the procedure "Place a call with the phone number supplied by the HF" described in Section 4.18.

Upon an internal event or user action, the HF may request a phone number from the AG by issuing the AT+BINP=1 command. Depending on the current status of the AG, it may either accept or reject this request.

If the AG accepts the request, it shall obtain a phone number and send the phone number back to the HF by issuing the +BINP response.

If the AG rejects the request from the HF, it shall issue the ERROR result code to indicate this circumstance to the HF.

When this procedure is executed multiple times (to retrieve multiple AG phone numbers to be linked to voice tags), it is the responsibility of the AG to provide the next phone number to be passed to the HF each time the procedure is executed.

Refer to Section 4.33 for more information on the AT+BINP command and the +BINP response.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

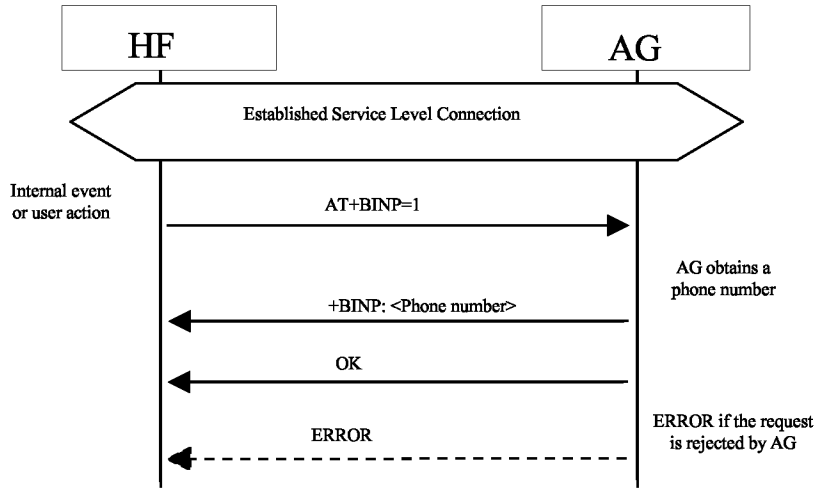


Figure 4.33: Request phone number to the AG

4.27 Transmit DTMF Codes

During an ongoing call, the HF transmits the AT+VTS command to instruct the AG to transmit a specific DTMF code to its network connection.

Refer to Section 4.33 for more information on the AT+VTS command.

The following pre-conditions apply for this procedure:

- An ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.
- An ongoing call in the AG exists.

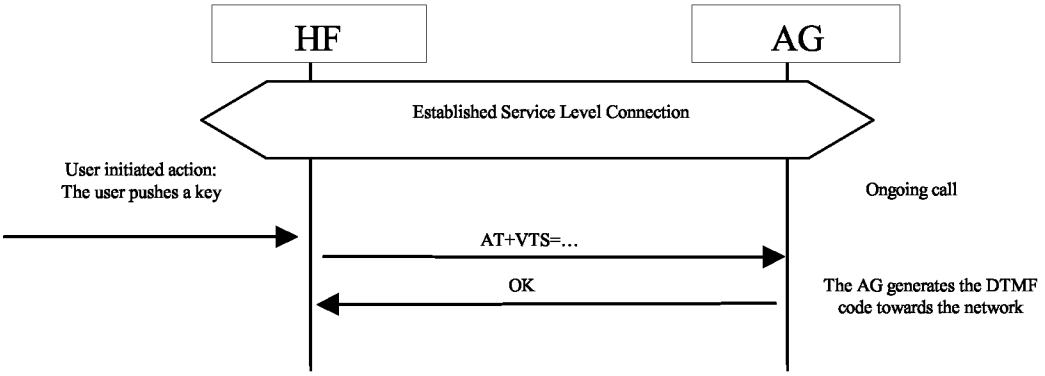


Figure 4.34: Transmit DTMF code

4.28 Remote Audio Volume Control

4.28.1 Audio Volume Control

This procedure enables the user to modify the speaker volume and microphone gain of the HF from the AG.

The AG may control the gain of the microphone and speaker of the HF by sending the unsolicited result codes +VGM and +VGS respectively. There is no limit in the amount and order of result codes.

If the remote audio volume control feature is supported in the HF device, it shall support at least remote control of the speaker volume.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the AG shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

An audio connection is not a necessary pre-condition for this feature.

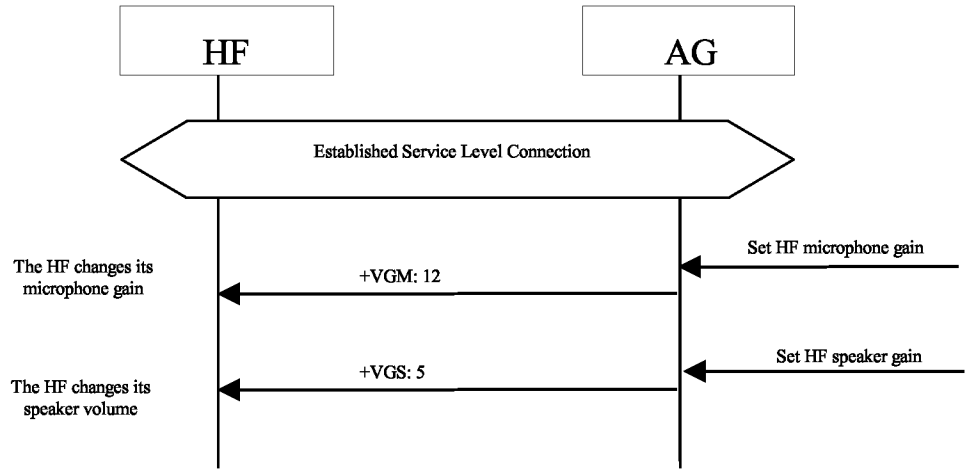


Figure 4.35: Typical example of audio volume control

Both the speaker and microphone gains are represented as parameter to the +VGS and +VGM, on a scale from 0 to 15. The values are absolute values, and relate to a particular (implementation dependent) volume level controlled by the HF.

Refer to Section 4.33 for more information on these commands and unsolicited result codes.

4.28.2 Volume Level Synchronization

This procedure allows the HF to inform the AG of the current gain settings corresponding to the HF's speaker volume and microphone gain.

On Service Level Connection establishment, the HF shall always inform the AG of its current gain settings by using the AT commands AT+VGS and AT+VGM.

If local means are implemented on the HF to control the gain settings, the HF shall also use the AT commands AT+VGS and AT+VGM to permanently update the AG of changes in these gain settings.

In all cases, the gain settings shall be kept stored, at both sides, for the duration of the current Service Level Connection. Moreover, if the Service Level Connection is released

as a consequence of an HF initiated “Audio Connection transfer towards the AG”, as stated in Section 4.17, the HF shall also keep the gain settings and re-store them when the Service Level Connection is re-established again.

The HF is only mandated to support microphone gain synchronization when it supports remote microphone gain control.

As pre-condition for this procedure, an ongoing Service Level Connection between the AG and the HF shall exist. If this connection does not exist, the HF shall autonomously establish the Service Level Connection using the proper procedure as described in Section 4.2.

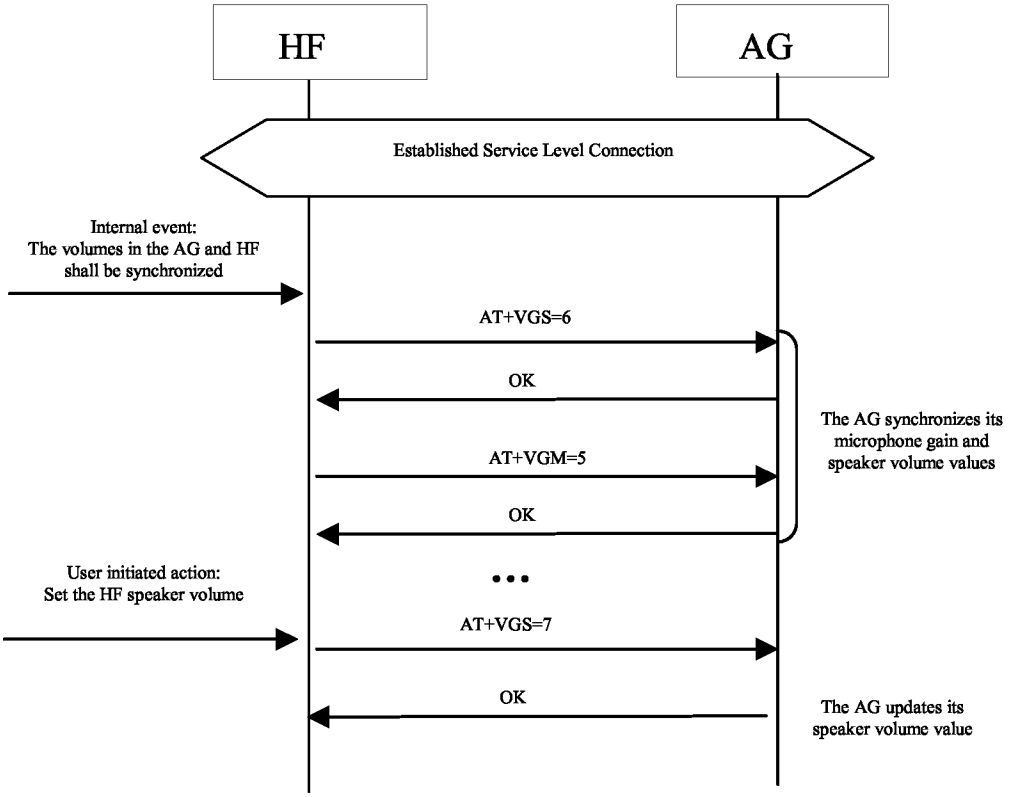


Figure 4.36: Typical example of volume level synchronization

Refer to Section 4.33 for more information on these commands and unsolicited result codes.

4.29 Response and Hold

This procedure allows the user to put an incoming call on hold and then accept or reject the call from the HF or AG. This feature is specific to the limited markets where PDC and CDMA networks support this function.

4.29.1 Query Response and Hold Status

The HF shall execute this procedure to query the status of the "Response and Hold" state of the AG.

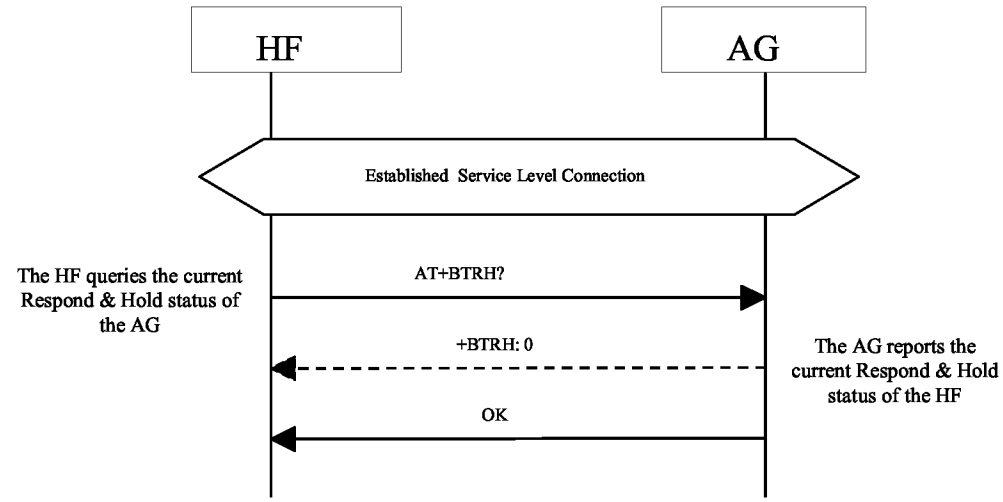


Figure 4.37: Query Response and Hold State of AG

- The HF shall issue AT+BTRH? command to query the current "Response and Hold" state of the AG.
- If the AG is currently in any of the Response and Hold states, then the AG shall send a +BTRH: Response with the parameter set to 0. If the AG is not in the Response and Hold states, then no response shall be sent.
- The AG shall send OK response to signal completion of the AT+BTRH? command.

4.29.2 Put an Incoming Call on Hold from HF

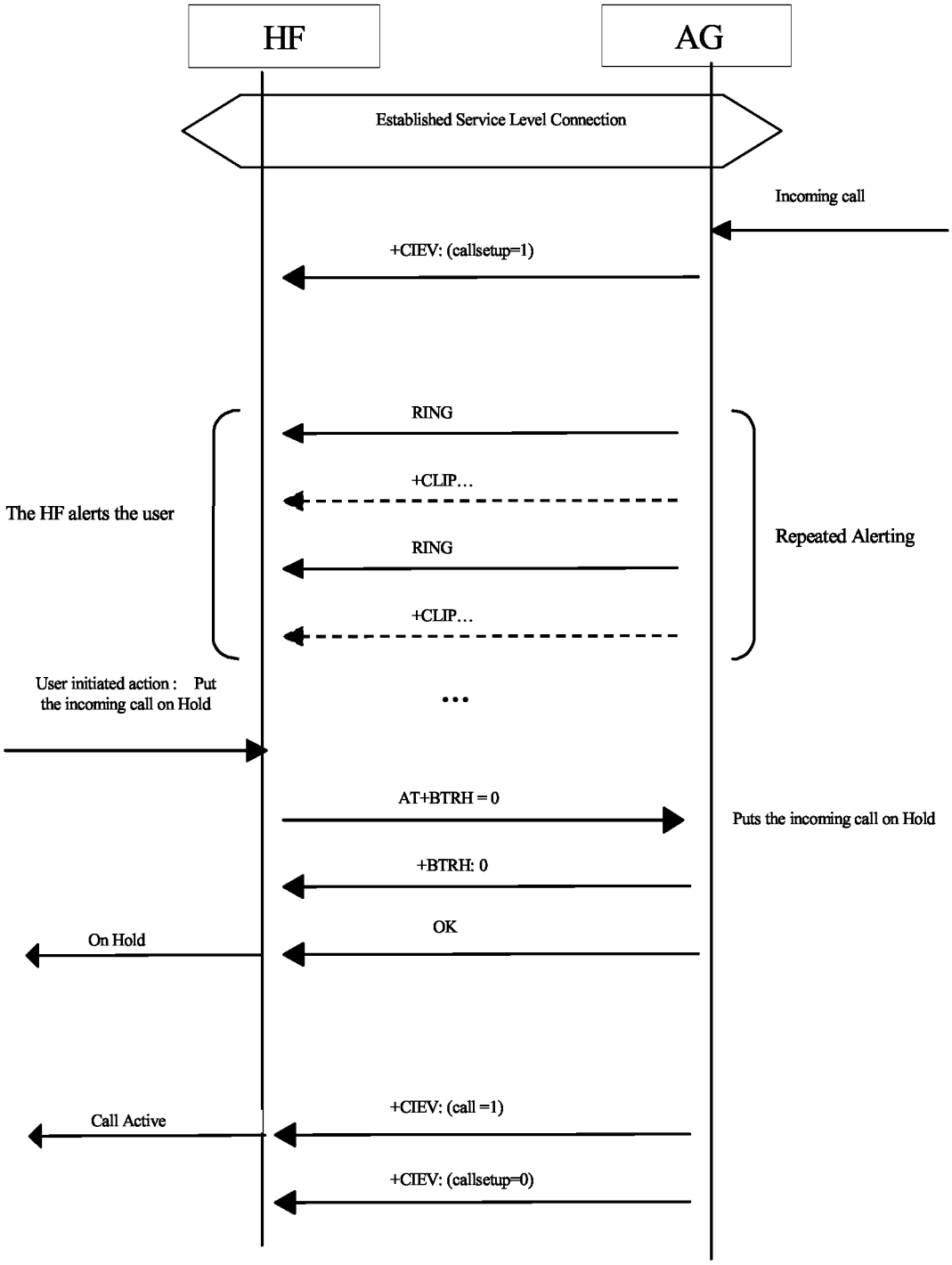


Figure 4.38: Put an incoming call on Hold from HF

- As a pre-condition to this procedure, the AG shall not have an active call or a call on hold.
- The AG shall send a sequence of unsolicited RING alerts to the HF. The RING alert shall be repeated until the HF accepts the incoming call or until the incoming call is interrupted for any reason.

- If the HF has enabled the Calling Line Identification [CLI], the AG shall send a +CLIP Response to HF.
- The user may put the incoming voice call on hold by using the proper means provided by the HF unit. The HF shall then send the AT+BTRH command with the parameter <n> set to 0. The AG shall then begin the procedure for putting the incoming call on hold.
- The AG shall send +BTRH Response with the parameter set to 0 as soon as the incoming call is put on hold.
- The AG shall send the +CIEV Response with the call status set to 1.
- The AG shall send the +CIEV Response with the callsetup status set to 0.

4.29.3 Put an Incoming Call on Hold from AG

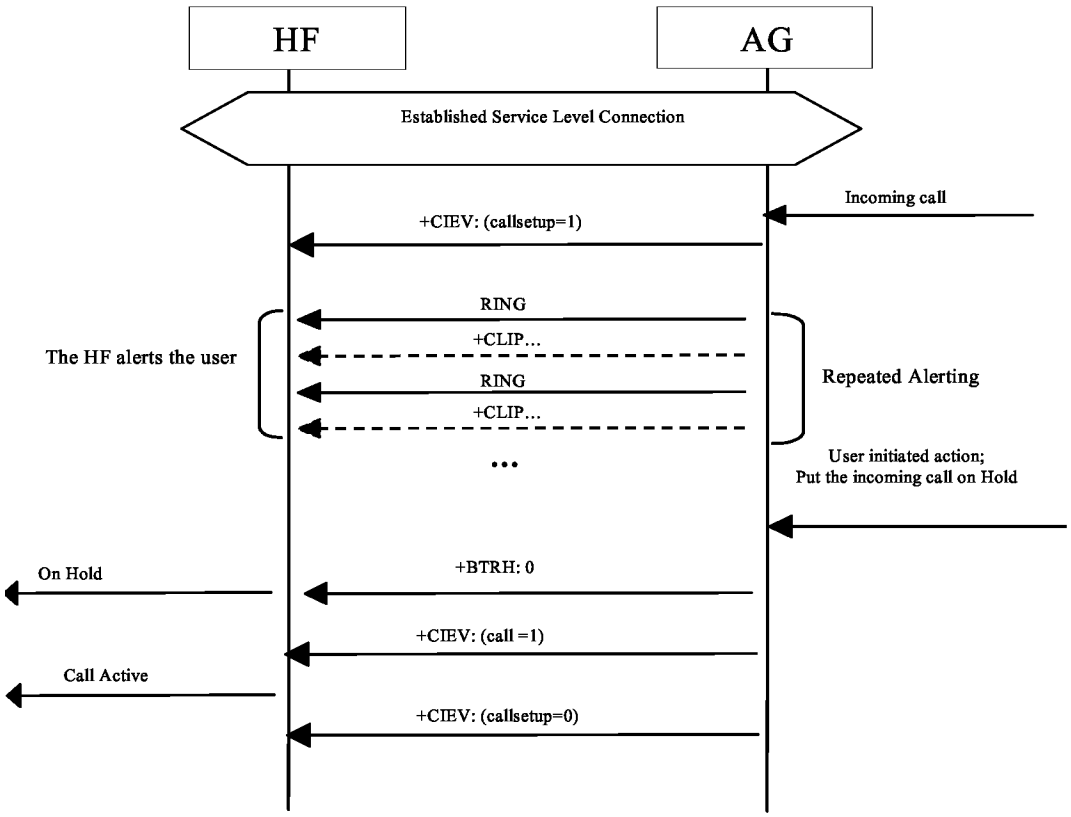


Figure 4.39: Put an incoming call on Hold from AG

As a pre-condition to this procedure, the AG shall not have an active call or a call on hold.

- The AG shall send a sequence of unsolicited RING alerts to the HF. The RING alert shall be repeated until the HF accepts the incoming call or until the incoming call is interrupted for any reason.
- If the HF has enabled the Calling Line Identification [CLI], the AG shall send a +CLIP Response to the HF.
- The user may put the incoming voice call on hold by using the proper means provided by the AG unit. The AG shall then send +BTRH Response with the parameter <n> set to 0 to indicate that the incoming call is on hold.
- Depending on whether in band ringing is enabled or disabled, there may or may not be a synchronous connection established between the HF and AG. The synchronous connection state (enabled or disabled) shall not be changed when an incoming call is placed on hold.
- The AG shall send the +CIEV Response with the call status set to 1.
- The AG shall send the +CIEV Response with the callsetup status set to 0.

4.29.4 Accept a Held Incoming Call from HF

The following additional pre-condition applies to this procedure:

- An incoming call was put on hold.

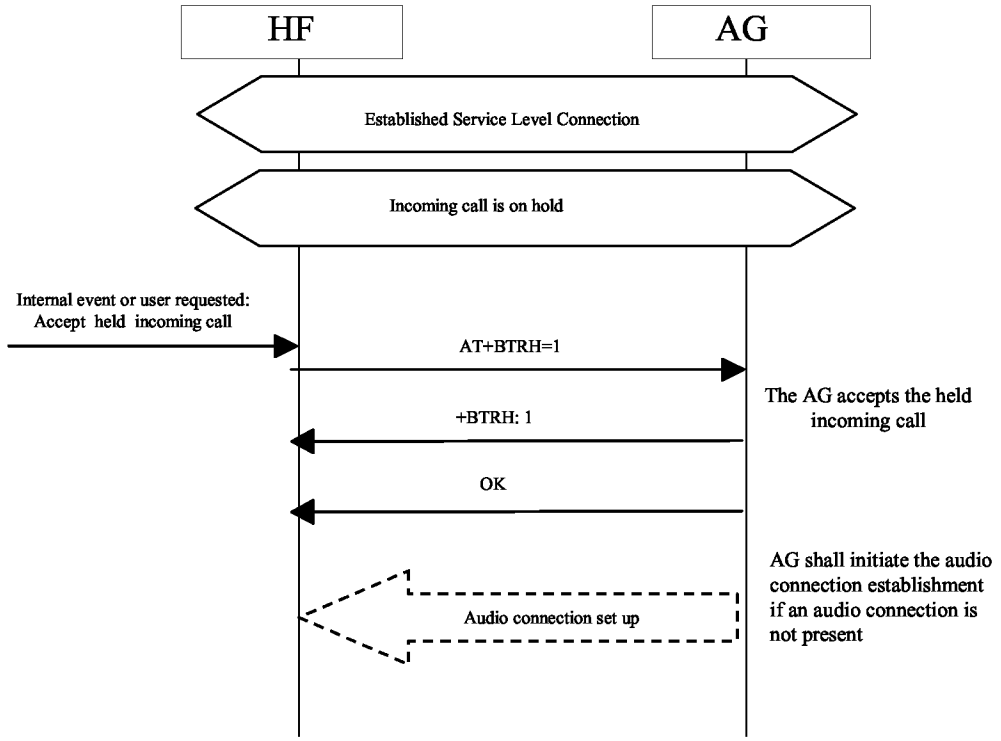


Figure 4.40: Accept a held incoming call from HF

- The user may accept the incoming voice call on hold by using the proper means provided by the HF unit. The HF shall then send the AT+BTRH command with the

parameter <n> set to 1. The AG shall then begin the procedure for accepting the incoming call that was put on hold.

- The AG shall then send +BTRH Response with the parameter <n> set to 1 to notify HF that the held incoming call was accepted.
- The AG shall start the procedure for establishing the audio connection and route the audio paths to the HF only if the audio connection was not established.

4.29.5 Accept a Held Incoming Call from AG

The following additional pre-condition applies to this procedure:

- An incoming call was put on hold.

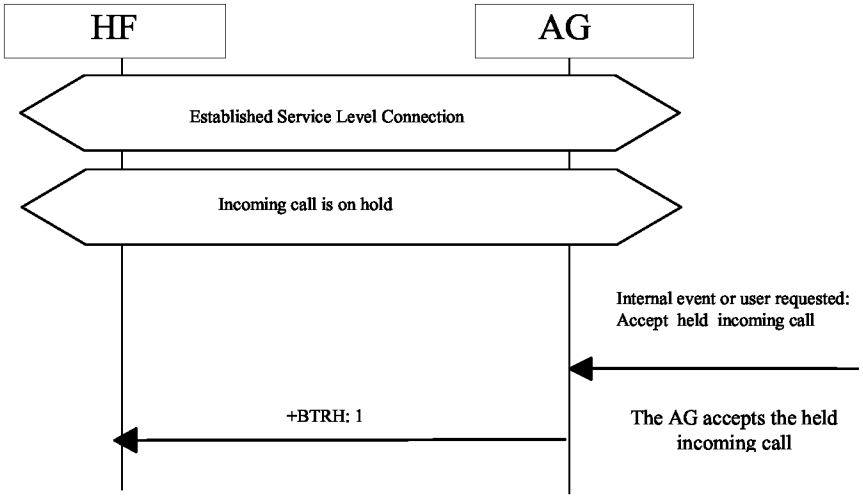


Figure 4.41: Accept a held incoming call from AG

- The user may accept the incoming voice call on hold by using the proper means provided by the AG unit. The AG shall then send +BTRH Response with the parameter <n> set to 1 to notify the HF that the held incoming call was accepted.

4.29.6 Reject a Held Incoming Call from HF

The following additional pre-condition applies to this procedure:

- An incoming call was put on hold.

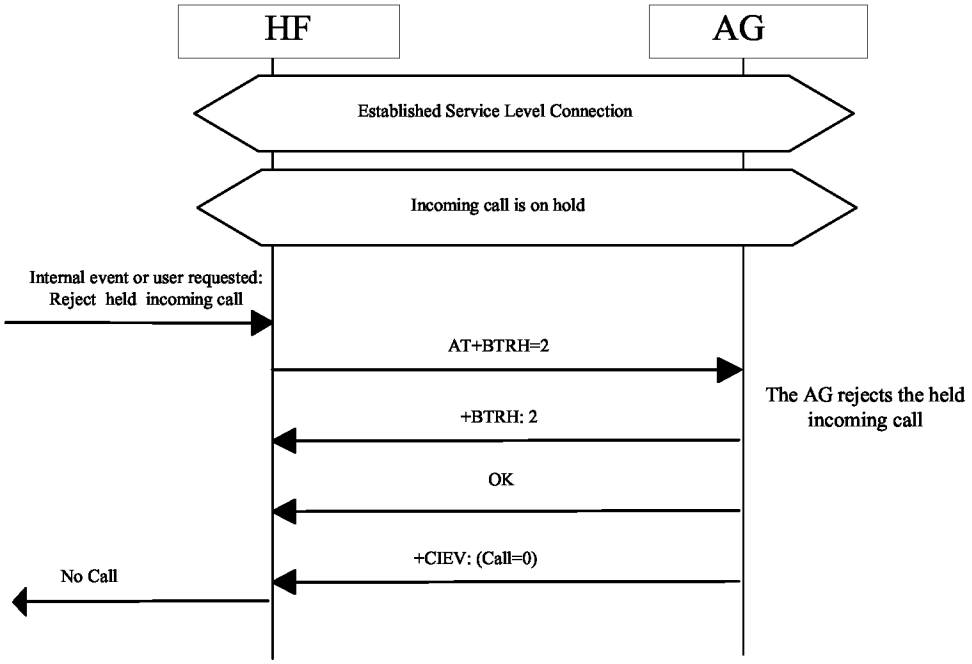


Figure 4.42: Reject a held incoming call from HF

- The user may reject the incoming voice call on hold by using the proper means provided by the HF unit. Either of the following two sequences shall be permissible by the HF and AG:
  - The HF may send the AT+BTRH command with the parameter <n> set to 2. The AG shall then begin the procedure for rejecting the incoming call that was put on hold. The AG shall send +BTRH Response with the parameter <n> set to 2 to notify the HF that the held incoming call was rejected.
  - The HF may send the AT+CHUP command to reject the held incoming call. The AG shall reject the held call and send the OK indication to the HF.
- The AG shall send the +CIEV Response with the call status set to 0.

4.29.7 Reject a Held Incoming Call from AG

The following additional pre-condition applies to this procedure:

- An incoming call was put on hold.

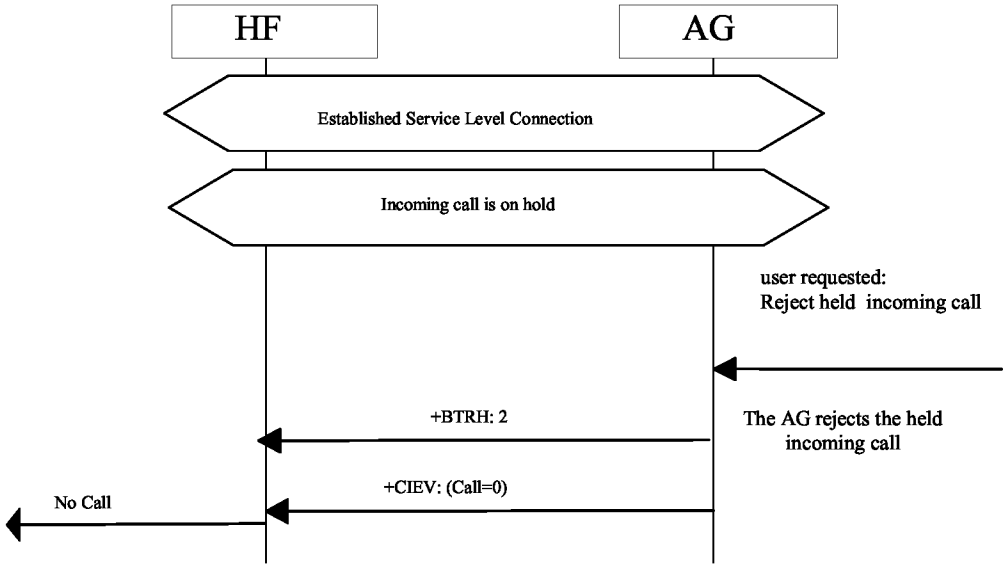


Figure 4.43: Reject a held incoming call from AG

- The user may reject the incoming voice call on hold by using the proper means provided by the AG unit. The AG shall then send +BTRH Response with the parameter <n> set to 2 to notify HF that the held incoming call was rejected.
- The AG shall also send the +CIEV Response with the call status parameter set to 0 to indicate that the AG is currently not in a call.

4.29.8 Held Incoming Call Terminated by Caller

The following additional pre-condition applies to this procedure:

- An incoming call was put on hold.

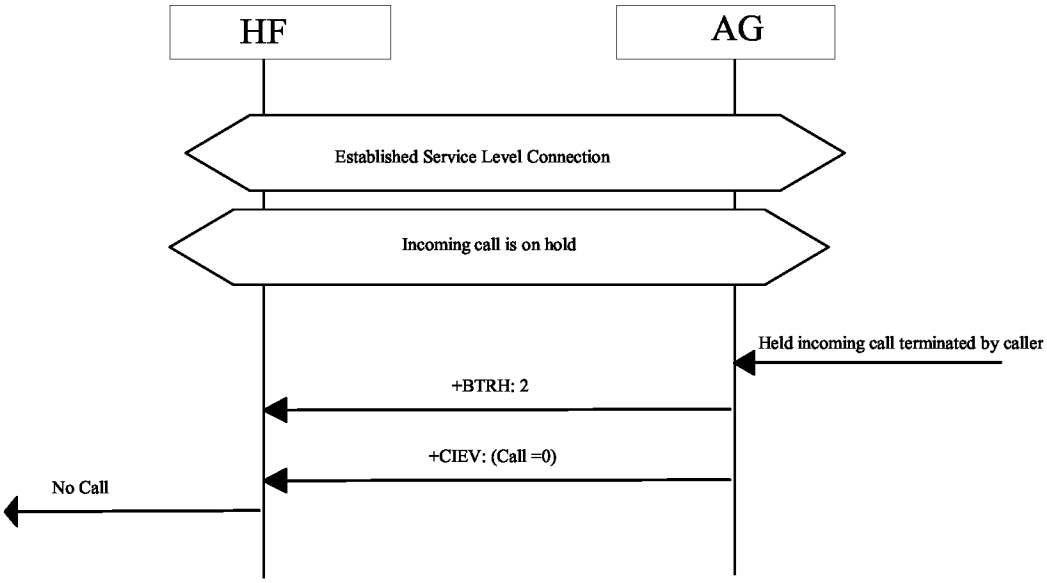


Figure 4.44: Held incoming call terminated by caller

- The caller may terminate the held incoming call. The AG shall then send +BTRH Response with the parameter <n> set to 2 to notify the HF that the held incoming call was terminated.
- The AG shall send the +CIEV Response with the Call status parameter set to 0 to indicate that the AG is currently not in a call.

4.30 Subscriber Number Information

This procedure allows HF to query the AG subscriber number.

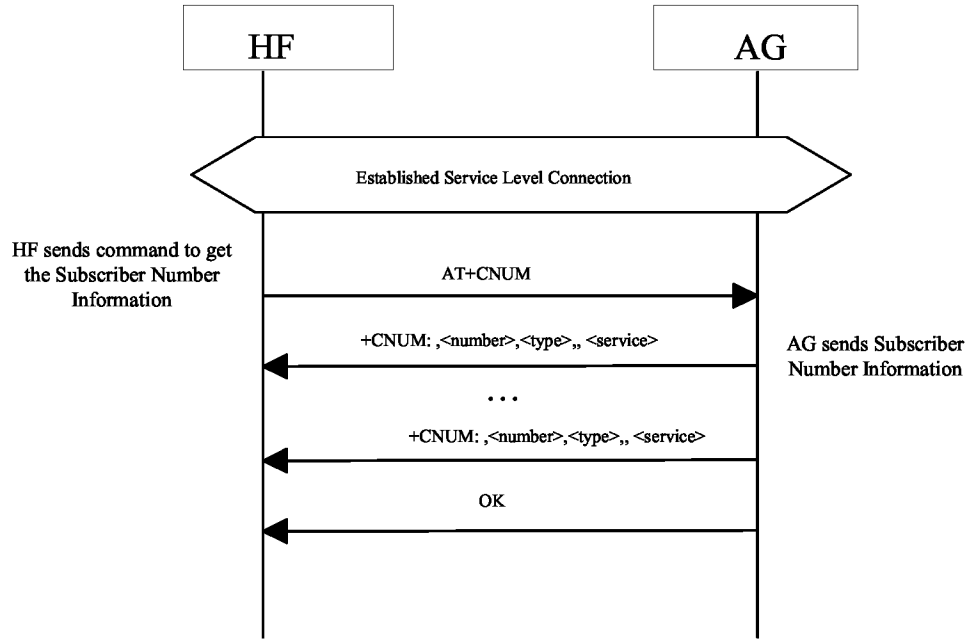


Figure 4.45: Query Subscriber Number Information of AG

This procedure illustrates AG response to the query of an empty subscriber number.

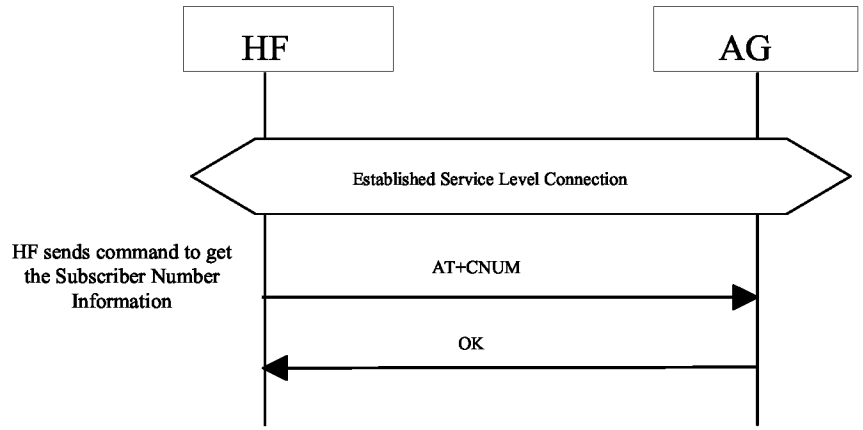


Figure 4.46: Empty Subscriber Number Information from AG

The following pre-condition applies for this procedure:

- An ongoing Service Level Connection between the HF and AG shall exist. If this connection does not exist, the HF shall establish a connection using the "Service Level Connection set up" procedure described in Section 4.2.
- The HF shall send the AT+CNUM command to query the AG subscriber number information.

- If the subscriber number information is available, the AG shall respond with the +CNUM response. If multiple numbers are available, the AG shall send a separate +CNUM response for each available number.
- The AG shall signal the completion of the AT+CNUM action command with an OK response. The OK will follow zero or more occurrences of the +CNUM response. (See figures 4.45 and 4.46).

4.31 Enhanced Call Status Indications

4.31.1 Query List of Current Calls in AG

The HF shall execute this procedure to query the list of current calls in AG.

The following pre-condition applies for this procedure:

- A SLC must exist between the AG and HF devices. If no current SLC exists, the HF shall first initiate a SLC.

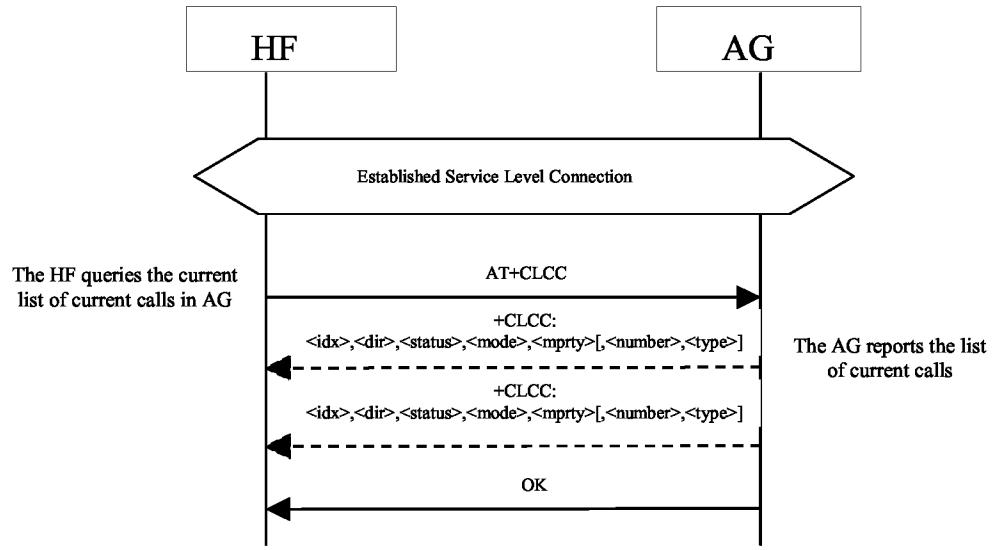


Figure 4.47: Query List of Current Calls

- HF shall find out the list of current calls in AG by sending the AT+CLCC command.
- If the command succeeds and if there is an outgoing (Mobile Originated) or an incoming (Mobile Terminated) call in AG, AG shall send a +CLCC response with appropriate parameters filled in to HF.
- If there are no calls available, no +CLCC response is sent to HF.
- The AG shall always send OK response to HF.

4.31.2 Indication of Status for Held Calls

Upon the change in status of any call on hold in the AG, the AG shall execute this procedure to advise the HF of the held call status. The values for the callheld indicator are:

0= No calls held

1= Call is placed on hold or active/held calls swapped
(The AG has both and active AND a held call)
2= Call on hold, no active call

The following pre-condition applies for this procedure:

- The HF shall have enabled the Call Status Indicators function in the AG.
- A SLC shall exist between the AG and HF devices.

Whenever an active call is placed on hold such that the AG now has both an active and held call or the active/held call positions swapped by a request from the HF or by action on the AG the AG shall issue a +CIEV unsolicited result code with the callheld indicator value of "1".

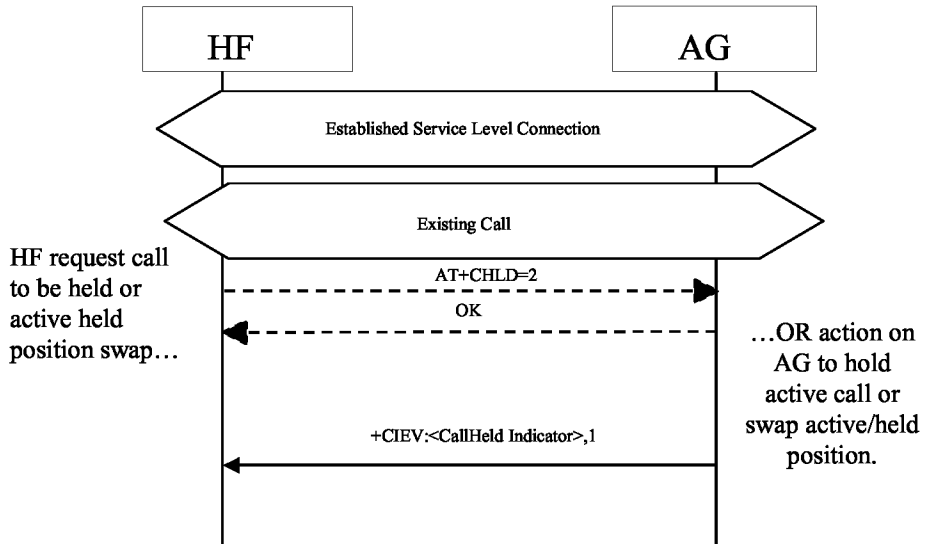


Figure 4.48: Call Held or Active/Held Position Swap

Consequently, upon the release of any call on hold by the HF, the AG or by network event, or actions by the HF or AG to retrieve a held call, the AG shall issue a +CIEV unsolicited result code with the callheld indicator value of "0".

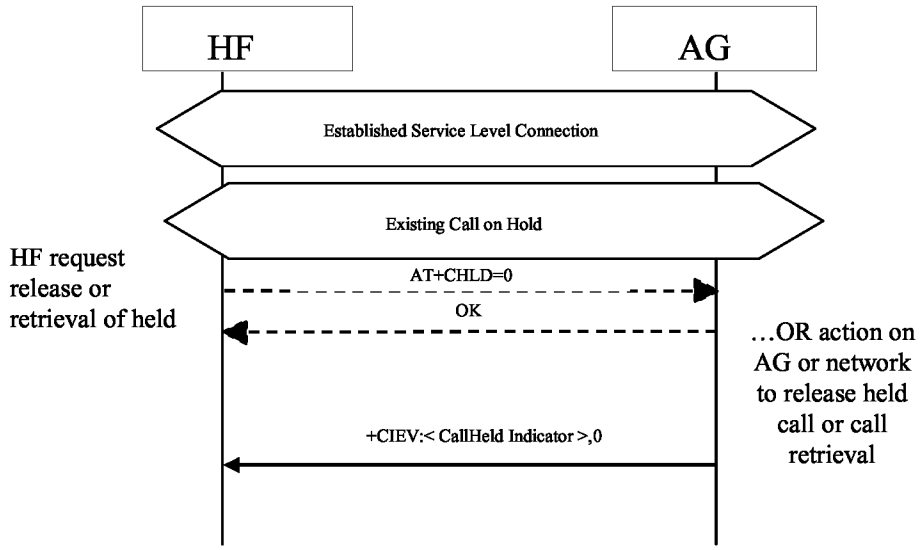


Figure 4.49: Held Call Release

If a call is still on hold when an active call is terminated or a single active call is put on hold, the AG shall issue a +CIEV unsolicited result code with the callheld indicator value of "2".

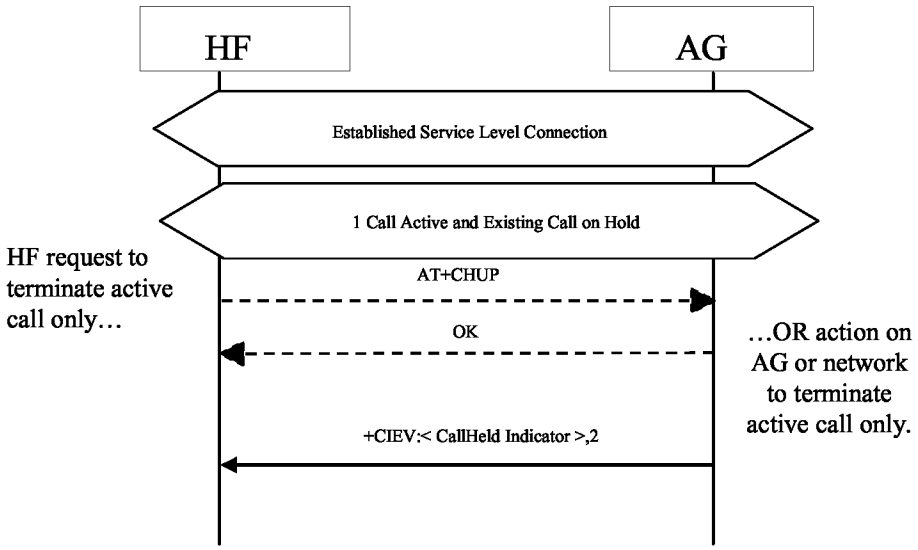


Figure 4.50: Active Call Terminated/Call Remains Held

4.32 Enhanced Call Control Mechanisms

As stated earlier, the Enhanced Call Control mechanism is simply an extension of the current AT+CHLD command. These extensions are defined as additional arguments to the AT+CHLD command. The new arguments for this command include an index of a specific call as indicated in the +CLCC response.

4.32.1 Release Specified Call Index

The HF shall execute this procedure to release a specific call in the AG.

The following pre-condition applies for this procedure:

- A SLC must exist between the AG and HF devices. If no current SLC exists, the HF shall first initiate a SLC.

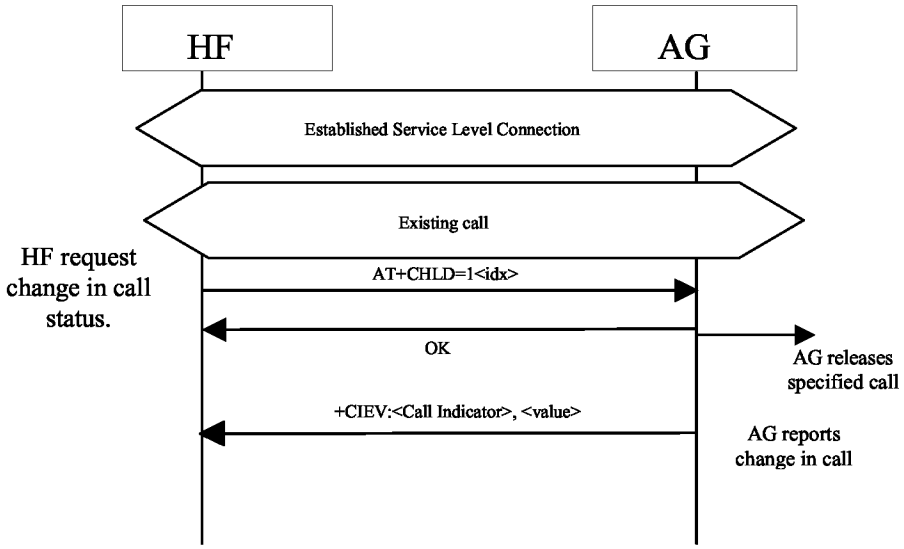


Figure 4.51: Release Specified Active Call

- The HF shall send the AT+CHLD=1<idx> command to release a specific active call.
- The AG shall release the specified call.
- If the released call was an active call and a call is currently held, the AG shall retrieve the held call.
- In the event that there are multiple held calls the AG shall retrieve the call associated with the lowest call index.
- The AG shall report the change in call status.

If the index (<idx>) is not valid, the AG shall report the proper error code.

4.32.2 Private Consultation Mode

The HF shall execute this procedure to place all parties of a multiparty call on hold with the exception of the specified call.

The following pre-condition applies for this procedure:

- A SLC must exist between the AG and HF devices. If no current SLC exists, the HF shall first initiate a SLC.
- Existing multiparty call is active in AG.

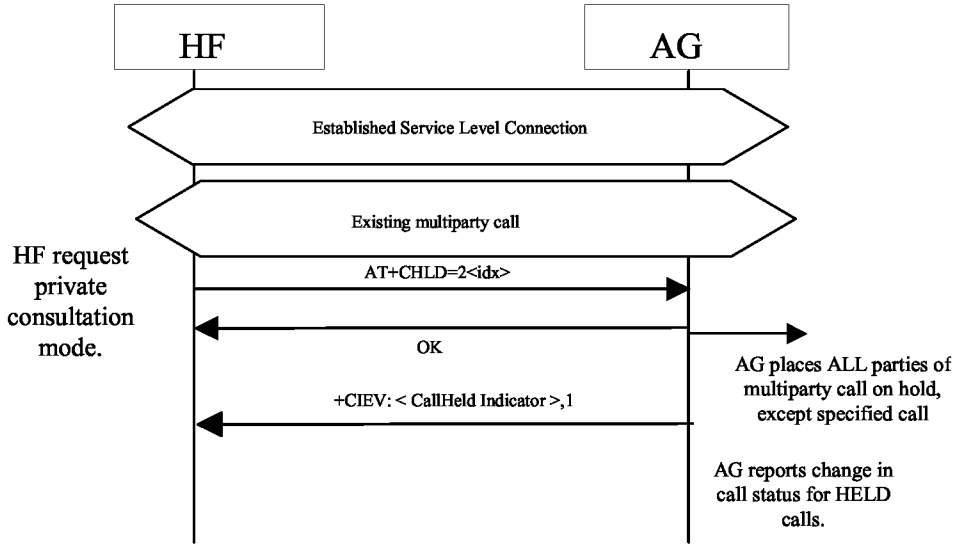


Figure 4.52: Request Private Consultation Mode

- HF shall send the AT+CHLD=2<idx> command to request private consultation mode.
- AG shall place all other parties of call on hold.
- AG shall report the change in status of the held parties.
- If the index (<idx>) is not valid, the AG shall respond with the proper error code.

4.33 AT Command and Results Codes

4.33.1 General

For the exchange of the commands and unsolicited results codes, the format, syntax and procedures of 3GPP 27.007 [2] shall be taken as reference. The following rules specifically apply for the HFP specification:

- Only one command (or unsolicited result code) per command line needs to be expected.
- The AG, by default, shall not echo the command characters.
- The AG shall always transmit result codes using verbose format.

- The characters below shall be used for AT commands and result codes formatting:

  <cr> corresponds to the *carriage return (0/13)* as stated in [6]

  <lf> corresponds to the *line feed (0/10)* as stated in [6]
- The format of an AT command from the HF to the AG shall be:

  <AT command><cr>
- The format of the OK code from the AG to the HF shall be:

  <cr><lf>OK<cr><lf>
- The format of the generic ERROR code from the AG to the HF shall be:

  <cr><lf>ERROR<cr><lf>
- The format of an unsolicited result code from the AG to the HF shall be:

  <cr><lf><result code><cr><lf>

The Hands-Free Profile uses a subset of AT commands and result codes from existing standards; these are listed in Section 4.33.2. Section 4.33.3 lists the new Bluetooth defined AT commands and result codes not re-used from any existing standard.

In general, the AG shall use the OK code, as described in Section 4.33.2, for acknowledgement of the proper execution of a command and respond with the proper error indication to any unknown command received from the HF.

It is mandatory for the AG to properly respond to any error condition and for the HF to properly process the corresponding error indication code received from the AG. The code ERROR, as described in Section 4.33.2, shall be used as error indication for this purpose.

The HF shall always ignore any unknown or unexpected indication code received from the AG. The only exception is the case in which the AG issues a “Mobile Equipment Error” indication using the +CME ERROR: result code (see [2]). In this case, the HF shall interpret this result code in the same way as if it was a generic ERROR code.

As a general rule, when an AT command or result code of this specification is implemented, support for the associated parameters “covered” in this specification, and all their corresponding possible values, shall be considered mandatory unless otherwise explicitly stated in each particular case.

4.33.2 AT Capabilities Re-Used from GSM 07.07 and 3GPP 27.007

The re-used AT commands and unsolicited result codes for implementing the functionality described in this specification are listed below:

As a convention, if a parameter of an AT command or result code is not “covered” in this specification, it shall not be present in the corresponding AT command, and the HF shall ignore the parameter whenever it is received in a result code.

- **ATA**

  Standard call answer AT command. Refer to Annex G in [2].

- **ATDdd…dd;**

  Standard AT command intended for placing a call to a phone number. Only voice calls are covered in this specification. Refer to Section 6.2 in [2].

- **ATD>nnn...;**

Extension of the standard ATD command, intended for memory dialing. Only voice calls are covered in this specification. Refer to Section 6.3 in [2].

- **ERROR**

  Standard error indication code. It shall be issued on detection of any syntax, format or procedure error condition. The "Mobile Equipment Error" report code "+CME ERROR:" is covered below. Refer to Annex B in [2].

- **OK**

  Standard acknowledgement to the execution of a command. Refer to Annex B in [2].

- **NO CARRIER, BUSY, NO ANSWER, DELAYED, BLACKLISTED**

  Extended response indication codes for AT commands. These codes shall be issued from the AG to the HF as responses to AT commands from the HF to the AG or from the AG as unsolicited result codes. These are in addition to the +CME ERROR: responses.

- **RING**

  Standard "incoming call" indication. Refer to Annex B in [2].

- **AT+CCWA**

  Standard "Call Waiting notification" AT command. Within the AT+CCWA=[<n>[,<mode>[,<class>]]]command, only enabling/disabling of the Call Waiting notification unsolicited result code +CCWA , using the <n> parameter, is covered in this specification. Refer to Section 7.12 in [2].

- **+CCWA**

  Standard "Call Waiting notification" unsolicited result code.

  In the +CCWA result code only <number> and <type> parameters are covered in this specification. Other parameters are not considered relevant in this specification and shall be ignored by the HF.

  The <number> parameter shall be a text string and shall always be contained within double-quotes.

  The <type> field specifies the format of the phone number provided, and can be one of the following values:

  - values 128-143: The phone number format may be a national or international format, and may contain prefix and/or escape digits. No changes on the number presentation are required.

- values 144-159: The phone number format is an international number, including the country code prefix. If the plus sign ("+") is not included as part of the number and shall be added by the AG as needed.
- values 160-175: National number. No prefix nor escape digits included.
- Refer to Section 7.12 in [2].

- **AT+CHLD**

  Standard call hold and multiparty handling AT command. In the AT+CHLD=<n> command, this specification only covers values for <n> of 0, 1, 1<idx>, 2, 2<idx>, 3 and 4, where:

  0 = Releases all held calls or sets User Determined User Busy (UDUB) for a waiting call.
  1 = Releases all active calls (if any exist) and accepts the other (held or waiting) call.
  1<idx> = Releases specified active call only (<idx>).
  2 = Places all active calls (if any exist) on hold and accepts the other (held or waiting) call.
  2<idx> = Request private consultation mode with specified call (<idx>). (Place all calls on hold EXCEPT the call indicated by <idx>.)
  3 = Adds a held call to the conversation.
  4 = Connects the two calls and disconnects the subscriber from both calls (Explicit Call Transfer). Support for this value and its associated functionality is optional for the HF.

  The test command AT+CHLD=? may be used for retrieving information about the call hold and multiparty services available in the AG (refer to Section 4.2.1).

  Refer to Section 7.13 in [2] and Section 4.5.5.1 in [8] for details.

- **AT+CHUP**

  Standard hang-up AT command. Execution command causes the AG to terminate the currently active call. This command shall have no impact on the state of any held call. Refer to Section 6.5 in [2].

  AT+CHUP is also used as the command to reject any incoming call prior to answer.

- **AT+CIND**

  Standard indicator update AT command. Only read command AT+CIND? and test command AT+CIND=? are required in this specification.

  The AT+CIND? read command is used to get current status of the AG indicators.

  The AT+CIND=? test command is used to retrieve the mapping between each indicator supported by the AG and its corresponding range and order index. It shall be issued at least once before any other command related to these indicators (AT+CIND? or AT+CMER) is used.

  The following indicators are covered in this specification:

- service: Service availability indication, where:

  <value>=0 implies no service. No Home/Roam network available.

  <value>=1 implies presence of service. Home/Roam network available.

- call: Standard call status indicator, where:

  <value>=0 means no call active.

  <value>=1 means a call is active.

- callsetup: Bluetooth proprietary call set up status indicator[3]. Support for this indicator is optional for the HF. When supported, this indicator shall be used in conjunction with, and as an extension of the standard call indicator. Possible values are as follows:

  <value>=0 means not currently in call set up.

  <value>=1 means an incoming call process ongoing.

  <value>=2 means an outgoing call set up is ongoing.

  <value>=3 means remote party being alerted in an outgoing call.

  Refer to Section 8.9 in [2].

- callheld: Bluetooth proprietary call hold status indicator. Support for this indicator is mandatory for the AG, optional for the HF. Possible values are as follows:

  0= No calls held
  1= Call is placed on hold or active/held calls swapped
  (The AG has both and active AND a held call)
  2= Call on hold, no active call

- signal: Signal Strength indicator, where:

  <value>= ranges from 0 to 5

- roam: Roaming status indicator, where:

  <value>=0 means roaming is not active

  <value>=1 means a roaming is active

- battchg: Battery Charge indicator of AG, where:

  <value>=ranges from 0 to 5

- **+CIND**

  Standard list of current phone indicators. Refer to section 8.9 in [2].

- **AT+CLCC**

  Standard list current calls command. Refer to section 7.18 in [2].

- **+CLCC**

  Standard list current calls result code. Refer to section 7.18 in [2].

[3] This status indicator is not defined in the GSM 07.07 specification

Supported parameters are as follows:

- idx= The numbering (starting with 1) of the call given by the sequence of setting up or receiving the calls (active, held or waiting) as seen by the served subscriber. Calls hold their number until they are released. New calls take the lowest available number.
- dir= 0 (outgoing), 1 (incoming)
- status= 0 = Active
  1 = Held
  2 = Dialing (outgoing calls only)
  3 = Alerting (outgoing calls only)
  4 = Incoming (incoming calls only)
  5 = Waiting (incoming calls only)
- mode= 0 (Voice), 1 (Data), 2 (FAX)
- mpty= 0 (Not Multiparty), 1 (Multiparty)
- number (optional)
- type (optional)

- **AT+COPS**

  The AT+COPS=3,0 shall be sent by the HF to the AG prior to sending the AT+COPS? command. AT+COPS=3,0 sets the format of the network operator string to the long format alphanumeric.

  The AT+COPS? command is used for reading network operator. This profile shall only support the "reading" of the name of the network operator. The response to this command from the AG shall return a +COPS:<mode>,<format>,<operator> where:

  <mode> contains the current mode and provides no information with regard to the name of the operator.

  <format> specifies the format of the <operator> parameter string, and shall always be 0 for this specification.

  <operator> specifies a quoted string in alphanumeric format representing the name of the network operator. This string shall not exceed 16 characters. Refer to Section 7.3 in [2].

- **AT+CMEE**

  Standard AT command used to enable the use of result code +CME ERROR: <err> as an indication of an error relating to the functionality of the AG.

  The set command AT+CMEE=1 is covered in this specification.

- **+CME ERROR**

  This is the Extended Audio Gateway Error Result Code response. Format of the response is: +CME ERROR: <err>. The format of <err> shall be numeric in this specification. The possible values for <err> covered in this specification are described below. These error codes may be provided instead of the standard ERROR response code to provide additional information to the HF. The ERROR

response code is still allowed while using the Extended Audio Gateway Error Result Codes.

+CME ERROR: 0 – AG failure
+CME ERROR: 1 – no connection to phone
+CME ERROR: 3 – operation not allowed
+CME ERROR: 4 – operation not supported
+CME ERROR: 5 – PH-SIM PIN required
+CME ERROR: 10 – SIM not inserted
+CME ERROR: 11 – SIM PIN required
+CME ERROR: 12 – SIM PUK required
+CME ERROR: 13 – SIM failure
+CME ERROR: 14 – SIM busy
+CME ERROR: 16 – incorrect password
+CME ERROR: 17 – SIM PIN2 required
+CME ERROR: 18 – SIM PUK2 required
+CME ERROR: 20 – memory full
+CME ERROR: 21 – invalid index
+CME ERROR: 23 – memory failure
+CME ERROR: 24 – text string too long
+CME ERROR: 25 – invalid characters in text string
+CME ERROR: 26 – dial string too long
+CME ERROR: 27 – invalid characters in dial string
+CME ERROR: 30 – no network service
+CME ERROR: 32 – Network not allowed – Emergency calls only

- **AT+CLIP**

  Standard “Calling Line Identification notification” activation AT command. It enables/disables the Calling Line Identification notification unsolicited result code +CLIP. Refer to Section 7.6 in [2].

- **+CLIP**

  Standard “Calling Line Identification notification” unsolicited result code.

  In the +CLIP: <number>, type> [,<subaddr>,<satype> [,[<alpha>] [,<CLI validity>]]] result code. Only <number> and <type> parameters are covered in this specification. Other parameters are not considered relevant in this specification and shall be ignored by the HF.

  The <number> parameter shall be a text string and shall always be contained within double-quotes.

  The <type> field specifies the format of the phone number provided, and can be one of the following values:

  - values 128-143: The phone number format may be a national or international format, and may contain prefix and/or escape digits. No changes on the number presentation are required.

- values 144-159: The phone number format is an international number, including the country code prefix. If the plus sign ("+") is not included as part of the number and shall be added by the AG as needed.
- values 160-175: National number. No prefix nor escape digits included.

  Refer to Section 7.11 in [2].

- **AT+CMER**

  AT+CMER

  Standard event reporting activation/deactivation AT command.

  In the AT+CMER=[<mode>[,<keyp>[,<disp>[,<ind> [,<bfr>]]]]] command, only the <mode>, and <ind> parameters are relevant for this specification. Only their values <mode>=(0,3) and <ind>=(0,1) are covered in this specification. Refer to Section 8.10 in [2].

  The following examples show how the AT+CMER command may be used for activating or deactivating the “indicator events reporting” result code:

  AT+CMER=3,0,0,1 activates “indicator events reporting”.

  AT+CMER=3,0,0,0 deactivates “indicator events reporting”.

- **+CIEV**

  Standard “indicator events reporting” unsolicited result code.

  In the +CIEV: <ind>,<value> result code, only the indicators stated in the AT+CIND command above are relevant for this specification where:

  - <ind>: Order index of the indicator within the list retrieved from the AG with the AT+CIND=? command. The first element of the list shall have <ind>=1.
  - <value>: current status of the indicator.

  If the HF receives any unknown indicator or value, it shall ignore it.

  Refer to Section 8.10 in [2].

- **AT+VTS**

  Standard DTMF generation AT command. Only the AT+VTS=<DTMF> command format is covered in this specification.

  Refer to Annex C.2.11 in [2].

- **AT+CNUM**

  Syntax: AT+CNUM (Retrieve Subscriber Number Information)
  AT+CNUM=? (Test Subscriber Number Information – Not Implemented)

  Description:

  Command issued by HF for the “Subscriber Number Information” feature in the AG.

  Only the action command AT+CNUM format is used.

- **+CNUM**

  Syntax: +CNUM: [<alpha>],<number>, <type>,[<speed>] ,<service> (Response for AT+CNUM)

  Description:

  Standard Response used for sending the “Subscriber Number Information” from AG to HF.

  The AG shall send the +CNUM: response for the AT+CNUM from the HF.

  Values:

  -<alpha>: This optional field is not supported, and shall be left blank.

  -<number>: Quoted string containing the phone number in the format specified by <type>.

  -<type> field specifies the format of the phone number provided, and can be one of the following values:

  - values 128-143: The phone number format may be a national or international format, and may contain prefix and/or escape digits. No changes on the number presentation are required.

  - values 144-159: The phone number format is an international number, including the country code prefix. If the plus sign ("+") is not included as part of the number and shall be added by the AG as needed.

  - values 160-175: National number. No prefix nor escape digits included.

  -<speed>: This optional field is not supported, and shall be left blank.

  -<service>: Indicates which service this phone number relates to. Shall be either 4 (voice) or 5 (fax).

  Example:

  +CNUM: ,"5551212",129,,4

  Refer to section 7.1 in [2].

4.33.3 Bluetooth Defined AT Capabilities

The GSM 07.07 [2] format and syntax rules shall be taken as the reference for these commands.

The new Bluetooth specific AT capabilities are listed below:

- **AT+BINP** *(Bluetooth INPut)*

  Syntax: AT+BINP=<datarequest>

  Expected response: +BINP: <dataresp$_1$>…<dataresp$_n$>

  Description:

  Command used for requesting some specific data input from the AG[4]. On reception of this command the AG shall perform the proper actions such that the requested information is sent back to the HF using the +BINP response.

  The type of data the HF shall expect in the <dataresp> parameter returned by the AG depends on the information requested in each case.

  Only support for execution command is mandated. Neither the read nor test commands are mandatory.

  Values:

  <datarequest>: 1, where

  1 = Phone number corresponding to the last voice tag recorded in the HF.

  <dataresp$_{1..n}$>: Data parameters returned by the AG. Their contents depends on the value of the <datarequest> parameter as follows:

<datarequest> value	<dataresp> parameters
1	<Phone number>: Phone number string (max. 32 digits). The format (type of address) of the phone number string shall conform with the rules stated in [7], sub-clause 10.5.4.7, for a value (in integer format) of the *type of address octet* of 145, if dialing string includes international access code character "+", and for a value of 129 otherwise.

- **AT+BLDN** *(Bluetooth Last Dialed Number)*

  Syntax: AT+BLDN

  Description:

[4] AT+BINP was created with future extensibility in mind. While the Hands-Free Profile only specifies a <datarequest> value of 1 (i.e. phone number), future profiles may choose to add values for <datarequest> to support the retrieval of additional data from the AG.

Command used for calling the last phone number dialed. On reception of this command, the AG shall set up a voice call to the last phone number dialed.

Only support for execution command is mandated. Neither the read nor test commands are mandatory.

- **AT+BVRA** *(Bluetooth Voice Recognition Activation)*

  Syntax: AT+BVRA=<vrec>

  Description:

  Enables/disables the voice recognition function in the AG.

  Only support for execution command is mandated. Neither the read nor test commands are mandatory.

  Values:

  <vrec>: 0, 1, entered as integer values, where

  0 = Disable Voice recognition in the AG

  1 = Enable Voice recognition in the AG

- **+BVRA** *(Bluetooth Voice Recognition Activation)*

  Syntax: +BVRA: <vrect>

  Description:

  Unsolicited result code used to notify the HF when the voice recognition function in the AG is activated/deactivated autonomously from the AG.

  The unsolicited +BVRA:1 result code shall not be sent by the AG to the HF if the corresponding voice recognition activation has been initiated by the HF. Likewise, the unsolicited +BVRA:0 result code shall not be sent by the AG to the HF if the corresponding voice recognition deactivation has been initiated by the HF, regardless of which side initiated the voice recognition activation.

  Values:

  <vrect>: 0, entered as integer value, where

  0 = Voice recognition is disabled in the AG

  1 = Voice recognition is enabled in the AG

- **AT+BRSF** *(Bluetooth Retrieve Supported Features)*

  Syntax: AT+BRSF=<HF supported features bitmap>

  Description:

  Notifies the AG of the supported features available in the HF, and requests information about the supported features in the AG. The supported features shall be represented as a decimal value.

  Values:

<HF supported features bitmap>: a decimal numeric string, which represents the value of a 32 bit unsigned integer. The 32 bit unsigned integer represents a bitmap of the supported features in the HF as follows:

Bit	Feature
0	EC and/or NR function
1	Call waiting and 3-way calling
2	CLI presentation capability
3	Voice recognition activation
4	Remote volume control
5	Enhanced call status
6	Enhanced call control
7-31	Reserved for future definition

The reserved bits [7-31] shall be initialized to Zero.

- **+BRSF** *(Bluetooth Retrieve Supported Features)*

  Syntax: +BRSF: <AG supported features bitmap>

  Description:

  Result code sent by the AG in response to the AT+BRSF command, used to notify the HF what features are supported in the AG. The supported features shall be represented as a decimal value.

  Values:

  <AG supported features bitmap>: a decimal numeric string, which represents the value of a 32 bit unsigned integer. The 32 bit unsigned integer represents a bitmap of the supported features in the AG as follows:

Bit	Feature
0	Three-way calling
1	EC and/or NR function
2	Voice recognition function
3	In-band ring tone capability
4	Attach a number to a voice tag
5	Ability to reject a call
6	Enhanced call status
7	Enhanced call control
8	Extended Error Result Codes
9-31	Reserved for future definition

  The reserved bits (9-31) shall be initialized to Zero.

- **AT+NREC** *(Noise Reduction and Echo Canceling)*

  Syntax: AT+NREC=<nrec>

  Description:

Command issued to disable any Echo Canceling and Noise Reduction functions embedded in the AG.

Only support for execution command is mandated. Neither the read nor test commands are mandatory.

Values:

<nrec>: 0, entered as integer value, where

0 = Disable EC/NR in the AG

- **AT+VGM** *(Gain of Microphone)*

  Syntax: AT+VGM=<gain>

  Description:

  Command issued by the HF to report its current microphone gain level setting to the AG. <gain> is a decimal numeric constant, relating to a particular (implementation dependent) volume level controlled by the HF. This command does not change the microphone gain of the AG; it simply indicates the current value of the microphone gain in the HF.

  Only support for execution command is mandated. Neither the read nor test commands are mandatory.

  Values:

  <gain>: 0 -15, entered as integer values, where

  0 = Minimum gain

  15 = Maximum gain

- **AT+VGS** *(Gain of Speaker)*

  Syntax: AT+VGS=<gain>

  Description:

  Command issued by the HF to report its current speaker gain level setting to the AG. <gain> is a decimal numeric constant, relating to a particular (implementation dependent) volume level controlled by the HF. This command does not change the speaker gain of the AG; it simply indicates the current value of the speaker volume in the HF.

  Only support for execution command is mandated. Neither the read nor test commands are mandatory.

  Values:

  <gain>: 0 -15, entered as integer values, where

  0 = Minimum gain

  15 = Maximum gain

- **+VGM** *(Gain of Microphone)*

  Syntax: +VGM:<gain>

  Description:

Unsolicited result code issued by the AG to set the microphone gain of the HF. <gain> is a decimal numeric constant, relating to a particular (implementation dependent) volume level controlled by the HF.

Due to the small inconsistency between the GSM standard ([2]) and the current Headset specification ([3]), the HF shall also accept the "**=**" symbol, in place of "**:**", as a valid separator for this unsolicited result code.

Values:

<gain>: 0 -15, integer values, where

0 = Minimum gain

15 = Maximum gain

- **+VGS** *(Gain of Speaker)*

  Syntax: +VGS:<gain>

  Description:

  Unsolicited result code issued by the AG to set the speaker gain of the HF. <gain> is a decimal numeric constant, relating to a particular (implementation dependent) volume level controlled by the HF.

  Due to the small inconsistency between the GSM 07.07 standard ([2]) and the current Headset specification ([3]), the HF shall also accept the "**=**" symbol, in place of "**:**", as valid separator for this unsolicited result code.

  Values:

  <gain>: 0 -15, integer values, where

  0 = Minimum gain

  15 = Maximum gain

- **++BSIR** *(Bluetooth Setting of In-band Ring tone)*

  Syntax: +BSIR: <bsir>

  Description:

  Unsolicited result code issued by the AG to indicate to the HF that the in-band ring tone setting has been locally changed. The HF may react accordingly by changing its own alert method.

  Values:

  <bsir>: 0 = the AG provides no in-band ring tone

  1 = the AG provides an in-band ring tone

- **AT+BTRH (Bluetooth Response and Hold Feature)**

  Syntax: AT+BTRH=<n> (Set command)
  AT+BTRH? (Read Current Status)

  Description:

Command issued by the HF for the "Response and Hold" feature in the AG.
This specification defines the use of the set and read command. The AT+BTRH? command shall be used by the HF to query the current "Response and Hold" state of the AG.

Values:

<n>: 0, 1, 2 entered as integer values, where
0 = Put Incoming call on hold
1 = Accept a held incoming call
2 = Reject a held incoming call

- **+BTRH (Bluetooth Response and Hold Feature)**

Syntax: +BTRH: <n> (Response for AT+BTRH)

Description:

Result code used to notify the HF when-ever the incoming call is either put on hold or accepted or rejected. The AG shall also respond back with this response for the AT+BTRH? command from the HF.

Values:

<n>: 0,1,2 entered as integer value, where
0 = Incoming call is put on hold in the AG
1 = Held incoming call is accepted in the AG
2 = Held incoming call is rejected in the AG

5 Serial Port Profile

This profile requires compliance to the Serial Port Profile [5]. The following text together with the associated sub-clauses defines the requirements with regard to this profile in addition to the requirements as defined in the Serial Port Profile.

For the Hands-Free Profile, both the AG and the HF may initiate connection establishment. Therefore, for the purposes of reading the Serial Port Profile [5], both the AG and the HF may assume the role of Device A or B.

5.1 RFCOMM Interoperability Requirements

For the RFCOMM layer, no additions to the requirements as stated in the Serial Port Profile [5] Section 4 apply.

5.2 L2CAP Interoperability Requirements

For the L2CAP layer, no additions to the requirements as stated in the Serial Port Profile [5] Section 5 apply.

5.3 SDP Interoperability Requirements

The following service records are defined for the Hands-Free Profile. There is one service record applicable to the Hands-Free unit and another for the Audio Gateway.

The attribute “SupportedFeatures” states the features supported in each device. This attribute is not encoded as a data element sequence; it is simply a 16-bit unsigned integer. The set of features supported in each case is bit-wise defined in this attribute on a yes/no basis. The mapping between the features and their corresponding bits within the attribute is listed below in Table 5.2 for the HF and in Table 5.4 for the AG. If a device indicates support for a feature, then it shall support that feature in the manner specified by this Profile, and be subject to verification as part of the Bluetooth Qualification Program.

The codes assigned to the mnemonics used in the Value column, as well as the codes assigned to the attribute identifiers (if not specifically mentioned in the AttrID column), are listed in the Bluetooth Assigned Numbers (see URL [9]).

The values of the “SupportedFeatures” bitmap given in Table 5.2 shall be the same as the values of the Bits 0 to 4 of the AT-command AT+BRSF (see Section 4.33.3).

Item	Definition	Type	Value	Status	Default
ServiceClassIDList				M	
ServiceClass0		UUID	Hands-Free	M	
ServiceClass1		UUID	Generic Audio	M	
ProtocolDescriptorList				M	
Protocol0		UUID	L2CAP	M	

Item	Definition	Type	Value	Status	Default
Protocol1		UUID	RFCOMM	M	
ProtocolSpecificParameter 0	Server Channel	Uint8	N=server channel #	M	
BluetoothProfileDescriptorList				M	
Profile0	Supported Profiles	UUID	Hands-Free	M	Hands-Free
Param0	Profile Version	Uint16	0x0105[5]	M	
ServiceName	Display-able Text name	String	*Service-provider defined*	O	"Hands-Free unit"
SupportedFeatures	Features supported	Uint16	*Device dependent*	M	0x0000

Table 5.1: Service Record for the HF

Bit position (0=LSB)	Feature	Default in HF
0	EC and/or NR function (yes/no, 1 = yes, 0 = no)	0
1	Call waiting and three way calling(yes/no, 1 = yes, 0 = no)	0
2	CLI presentation capability (yes/no, 1 = yes, 0 = no)	0
3	Voice recognition activation (yes/no, 1= yes, 0 = no)	0
4	Remote volume control (yes/no, 1 = yes, 0 = no)	0

Table 5.2: "SupportedFeatures" attribute bit mapping for the HF

The "Network" attribute states, if the AG has the capability to reject incoming calls[6]. This attribute is not encoded as a data element sequence; it is simply an 8-bit unsigned integer. The information given in the "Network" attribute shall be the same as the information given in Bit 5 of the unsolicited result code +BRSF (see Section 4.33.3). An attribute value of 0x00 is translated to a bit value of 0; an attribute value of 0x01 is translated to a bit value of 1.

The values of the "SupportedFeatures" bitmap given in Table 5.4 shall be the same as the values of the Bits 0 to 4 of the unsolicited result code +BRSF (see Section 4.33.3).

[5] Indicating version HFP 1.5.
[6] In previous versions of the Hands-Free Profile, the attribute values were called "GSM like" and "others".

Item	Definition	Type	Value	Status	Default
ServiceClassIDList				M	
ServiceClass0		UUID	AG Hands-Free	M	
ServiceClass1		UUID	Generic Audio	M	
ProtocolDescriptorList				M	
Protocol0		UUID	L2CAP	M	
Protocol1		UUID	RFCOMM	M	
ProtocolSpecificParameter0	Server Channel	Uint8	N=server channel #	M	
BluetoothProfileDescriptorList				M	
Profile0	Supported Profiles	UUID	Hands-Free	M	Hands-Free
Param0	Profile Version	Uint16	0x0105[7]	M	
ServiceName	Display-able Text name	String	*Service-provider defined*	O	"Voice gateway"
Network		Uint8	0x01 – Ability to reject a call 0x00 – No ability to reject a call	M	
SupportedFeatures	Features supported	Uint16	*Device dependent*	M	0x0009

Table 5.3: Service Record for the AG

Bit position (0=LSB)	Feature	Default in AG
0	Three-way calling (yes/no, 1 = yes, 0 = no)	1
1	EC and/or NR function (yes/no, 1 = yes, 0 = no)	0
2	Voice recognition function (yes/no, 1 = yes, 0 = no)	0
3	In-band ring tone capability (yes/no, 1 = yes, 0 = no)	1
4	Attach a phone number to a voice tag (yes/no, 1 = yes, 0 = no)	0

Table 5.4: "SupportedFeatures" attribute bit mapping for the AG

[7] Indicating version HFP 1.5

5.3.1 Interaction with Hands-Free Profile Rev 0.96 Implementations

HF implementations, which are according to the Hands-Free Profile specification Rev. 0.96, will not send the AT+BRSF command. Likewise, AG implementations, which are according to the Hands-Free Profile specification Rev. 0.96, will not be able to respond to AT+BRSF with the +BRSF unsolicited result code. Instead they will respond with ERROR.

In order to retrieve the "SupportedFeatures" information from an HF, which does not send AT+BRSF, Service Discovery should be used by the AG implementation. Whenever the "SupportedFeatures" attribute is not present in the HF service record, or if the AG does not perform the Service Discovery procedure, default values as stated in Table 5.2 shall be assumed.

In order to retrieve the "SupportedFeatures" and "Network" information from an AG, which does not send +BRSF, Service Discovery should be used by the HF implementation. Whenever the "SupportedFeatures" attribute is not present in the AG service record, or if the HF does not perform the Service Discovery procedure, default values as stated in Table 5.4 shall be assumed.

5.4 Link Manager (LM) Interoperability Requirements

The profile adopts the requirements for the Link Manager as stated in the "Serial Port Profile" [5].

Additionally this profile mandates that both the AG and HF devices shall support synchronous logical transports, subject to the requirements in Section 5.6.

5.5 Link Control (LC) Interoperability Requirements

Table 5.5 shows the changes from Link Controller requirements in the Serial Port Profile [5].

Table 5.5: Link Controller requirements

	Capability	Support in AG	Support in HF
1.	Inquiry		O
2.	Inquiry scan	O	
7	Voice CODEC		
C	CVSD	M	M

5.5.1 Class of Device

A device implementing the HF role of HFP shall set the "Audio" bit in the Service Class field. Optionally, if the HF intends to be discovered as a "Hands-Free", it may use the following values in the Class of Device field:

1. Indicate "Audio" as Major Device class.
2. Indicate “Hands-Free” as the Minor Device class.

An inquiring AG may use this information to filter the inquiry responses.

5.6 Synchronous Connection Interoperability Requirements

Synchronous connections may be realized by a SCO or by an eSCO logical transport. Only the support for SCO logical transports is mandated.

The remainder of this section relates to devices supporting eSCO logical transports. Here, "initiating" and "responding" refers to the initiating and responding (i.e. accept or reject) role in setting up the synchronous connection.

Table 5.6 defines eSCO configuration parameter sets S1, S2 and S3. HCI level parameters are given as a reference. On systems not incorporating HCI, values for LMP level eSCO parameters T_{eSCO}, W_{eSCO} and packet length shall be associated that correspond to these HCI parameters and fall into the mandatory parameter ranges for these packet types as given in the LMP specification, and the Voice Setting parameter translates into the air mode parameter of LMP.

eSCO parameter set	**S1 "Safe Settings"**	**S2**	**S3**
Packet type	EV3	2-EV3	2-EV3
Transmit/Receive Bandwidth	8000	8000	8000
Voice_Setting (air coding)	CVSD	CVSD	CVSD
Max_Latency	0x0007 (7 ms)	0x0007 (7 ms)	0x000A (10ms)
Retransmission_Effort	0x01	0x01	0x01

Table 5.6: eSCO synchronous connections (HCI Reference parameters)

The following requirements apply to the support and use of eSCO logical transports and are based on parameter sets S1, S2 and S3:

- The device starting the request for a Synchronous Connection is known as the Initiator, the device receiving the request from the Initiator is known as the Responder. The Responder is able to accept or reject a request for eSCO transport. The Responder shall always accept a request for SCO transport.
- If support for eSCO logical transports is indicated at the Controller level, the Initiator may request the setup of an eSCO logical transport instead of SCO.
- The Initiators request for an eSCO transport may involve any configuration parameters matching the bidirectional throughput requirements of the voice codec (see section 5.5). If an HCI is supported on this device, the request for setting up a synchronous connection may include single or multiple packet types masked within the same request.
- The Responder may choose to accept or reject the request from the Initiator. It may reject the request for an eSCO transport, or may accept it with parameters that do not match the requested parameters. In this case the Initiator may retry the Synchronous Connection setup with different configuration parameters.

- If one or subsequent requests for an eSCO logical transport fails, the Initiator shall not abandon the setup of an eSCO transport without having requested eSCO using the “safe settings” S1.
- Only for HCI-based devices: if the Responder does not reject the request for an eSCO transport, the response shall include the parameters corresponding to the “safe settings” S1 when accepting a request. The Responder shall not request eSCO parameters that would inhibit the ability of the Initiator to negotiate the S1 settings.
- If the Initiator fails to establish an eSCO transport with the S1 settings, the Initiator shall request the setup of a SCO transport.
- Only for HCI-based devices: the Responder shall include the parameters for a SCO transport when accepting a request for a Synchronous Connection.

The following requirements apply if a device supports both eSCO logical transports and Enhanced Data Rate (as of Bluetooth core specification v2.0 + EDR or later).

- The Controller shall support the packet type 2-EV3, hence mandatory eSCO parameters ranges as given in the LMP specification and contained in settings S2 and S3.
- On an HCI-Responder, at least the settings S2 shall be included in the list of acceptable parameters.

6 Generic Access Profile

This section defines the support requirements for the capabilities as defined in the "Generic Access Profile" of the Core Specification.

6.1 Modes

The table shows the support status for GAP Modes in this profile.

Procedure	Support in HF
General discoverable mode	M
Procedure	**Support in AG**
Pairable mode	M

Table 6.1: Modes

6.2 Security Aspects

There are no changes to the security requirements as stated in the Generic Access Profile.

6.3 Idle Mode Procedures

Table 6.2 shows the support status for Idle mode procedures within this profile.

Procedure	Support in AG
Initiation of general inquiry	M
Initiation of general bonding	O
Initiation of dedicated bonding	O

Table 6.2: Idle mode procedures

7 References

[1] "Specification of the Bluetooth System; Core, v1.1 or later"

[2] 3GPP 27.007 v6.8.0 now supersedes and replaces ETS 300 916, "Digital cellular telecommunications system (Phase 2+); AT command set for GSM Mobile Equipment (ME) (GSM 07.07 version 7.5.0)"

http://www.3gpp.org/ftp/Specs/html-info/27007.htm

[3] "Specification of the Bluetooth System; Profiles, v1.1 or later, Headset Profile"

[4] "Specification of the Bluetooth System; Core, v1.1 or later, Generic Access Profile"

[5] "Specification of the Bluetooth System; Profiles, v1.1 or later, Serial Port Profile"

[6] "ITU-T50, Terminal Equipment and Protocols for telematic services: International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 IA5). Information technology – 7-Bit coded character set for information interchange"

[7] "Digital cellular telecommunication system (Phase 2+); Mobile radio interface layer 3 specification", (GSM 04.08 version 6.11.0)

[8] "GSM 02.30 (version 7.1.0): Digital cellular telecommunications system (Phase 2+); Man-Machine Interface (MMI) of the Mobile Station (MS)"

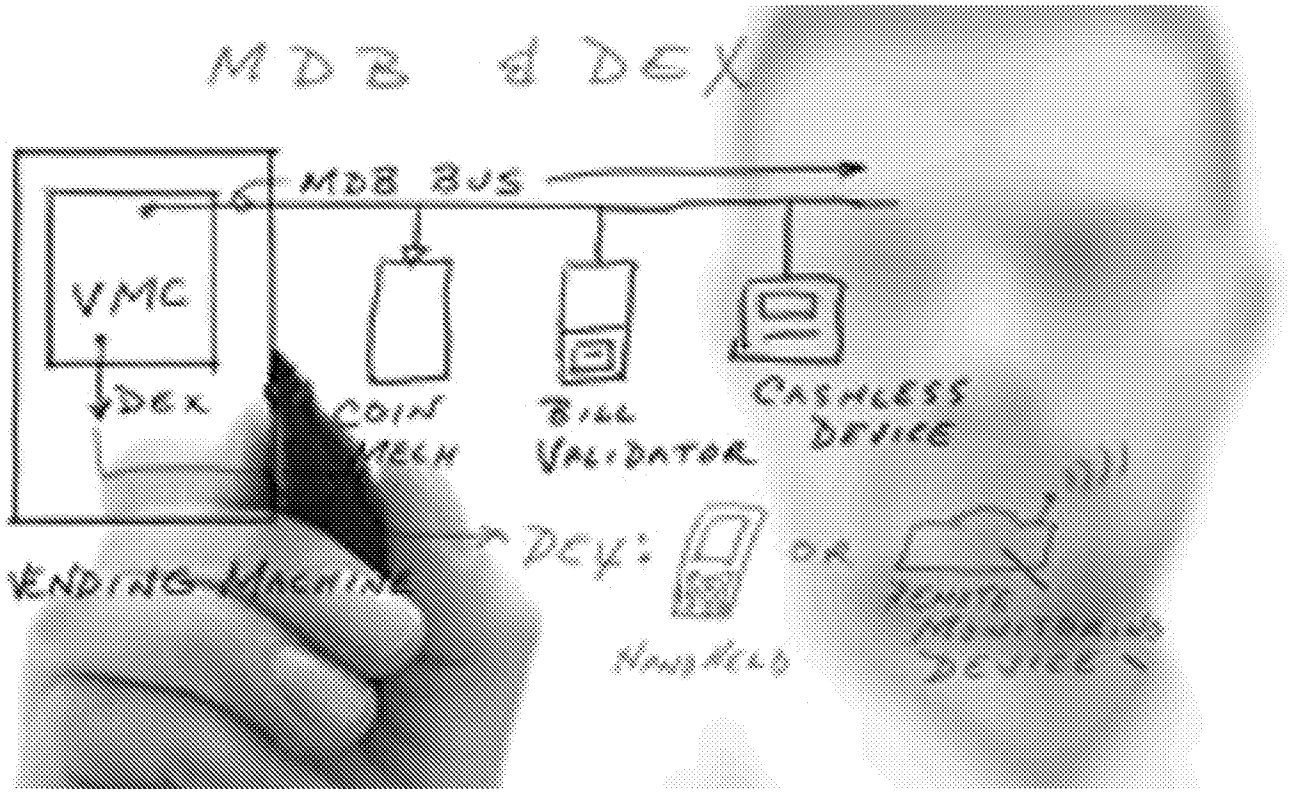
[9] Bluetooth Assigned Number URL is
https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers

8 List of Acronyms and Abbreviations

Abbreviation or Acronym	Meaning
AG	Audio Gateway
AT	Attention
CLI	Calling Line Identification
CODEC	COder DECoder
CVSD	Continuous Variable Slope Delta modulation
DTMF	Dual Tone Multi-Frequency
EC	Echo Cancellation
EDR	Enhanced Data Rate
eSCO	Extended Synchronous Connection Oriented
GAP	Generic Access Profile
GSM	Global System for Mobile communication
HF	Hands-Free unit
L2CAP	Logical Link Control and Adaptation Protocol
LMP	Link Manager Protocol
NR	Noise Reduction
OSI	Open System Interconnection
PIN	Personal Identification Number
RFCOMM	Serial port transport protocol over L2CAP
SCO	Synchronous Connection Oriented
SDP	Service Discovery Protocol
UI	User Interface
UUID	Universally Unique Identifier

9 List of Figures

10 List of Tables



DEX and MDB: A Primer For Vendors

Technology Basics 101: Both technologies are important but serve different functions.

Feb 7th, 2008

Two of the most oft-mentioned and misunderstood technologies in our industry are MDB (multi-drop bus) and DEX (digital exchange). It amazes me how frequently I hear people confuse MDB and DEX, as if they are related. Allow me to end that rumor right here. The only correlation between DEX and MDB is that they are two separate and distinct technologies that happen to reside in modern day vending machines.

DEX BRINGS IMPROVED AUDIT

DEX was brought to the industry in the late 1980s to provide better audit capabilities. The bottlers brought DEX, a uniform commercial code set up across many industries to

vending when they implemented DEX for communications between a route handheld and a grocery store's computer system. Since many bottler route drivers performed direct store delivery (DSD) as well as service of can/bottle machines, it made sense for their handheld to communicate with the vending machines they serviced as well as the stores. As often happened due to their size, resources and commitment to implementing technology, the bottlers took the leadership position, and the National Automatic Merchandising Association Technology Committee (made up mostly of engineers and industry suppliers) followed suit, adopting DEX as our industry standard.

VENDING USAGE INFORMATION

So what is DEX? DEX is our standard for an ASCII code-based electronic audit file, a way to communicate information such as sales, cash in bill validators, coins in coin boxes, sales of units by selection, pricing, door openings, and much more. It is created either locally by the VMC (Vending Machine Controller often called the "brain" of an electronic machine) or created by a retrofit DEX device in older electromechanical (dip switch) machines.

DEX is the result of the VMC storing information on an interval basis (the interval of time since the last DEX reading) and cumulative basis (since the VMC was first installed or the machine went into service). The VMC accumulates the data and transmits it in DEX format (see sidebar) over the DEX port when requested.

DEX data is quite useful and extensive. It eliminates the need for route people to write what they loaded into a machine on a route card. It also makes it unnecessary to manually input this information into a handheld. But the feature of DEX that gets most companies excited and starting to "DEX" their machines is the accuracy of cash accountability. There is no more second guessing what was to be collected out of the machine.

DEX IMPROVES ROUTE ACCOUNTING

DEX data is downloaded to a handheld device or transmitted via a remote monitoring device over to software that can parse the information into useful reports. DEX is downloaded using a 0.25-inch stereo plug (exactly like the one with your old stereo headphones from the 70s). When downloaded to a handheld, DEX is parsed and compared to planogram information unique to that machine that was stored in the handheld. This informs the route driver how many units of each product he/she has to load back into the machine to bring it back up to par.

Remote monitoring devices (wireless, LAN or telephone) can forward DEX, usually via the Internet, to a central computer where the software performs the same tasks as the handheld, but from the headquarters. This gives vendors the opportunity to pre-assemble

items for locations before drivers leave and efficiently pack route trucks with only the necessary products.

Approximately 60 to 70 percent of the machines currently deployed have "native" DEX, meaning the machines come with a VMC that produces DEX. Sometimes a newer version of firmware for the VMC and a DEX download cable are required to be added to enable DEX.

Older electronic and electromechanical machines not equipped with DEX can be retrofitted with either a new VMC that provides DEX (and many of the features found in new machines) or with a retrofit DEX audit device.

DEX File Interpretation Chart - View this chart in PDF format.

MULTI DROP BUS RELATES TO PAYMENT

MDB (multi drop bus) relates to the different payment systems interfacing together. When vending machines were electromechanical (using dip switches), bill validators and cashless systems had to run through the coin mechanisms. There were a slew of different connectors to interface to all the different types of coin mechs on the market, and it was very confusing since there was no industry standard. Even early electronic machines, which had VMC, didn't have standard connections. They used a serial interface (such as MicroMech), but additional devices, like bill validators or cashless systems, still had to be connected to and emulate the coin mechanisms.

If it wasn't for the NAMA and European Vending Association (EVA) getting together in the 1990s and working in a cooperative spirit to write the MDB specification, we would probably still be struggling through proprietary interfaces and the nightmare of connectors. MDB is an international standard co-authored by NAMA and EVA, and is present in almost every vending machine worldwide except for the Far East, which has its own standards.

MDB = ELECTRICAL BUS FOR INTERFACING

MDB was the first attempt by the industry to come up with a standard interface for all transactional electronic devices (i.e., coin mechanism, bill validator or cashless system) to be able to interface through an electrical bus to the VMC. This electrical bus provides one standard male and female connector, both of which are found on all MDB vending transactional electronic devices. An MDB device should have a y-MDB connection, providing for a piggyback connection from one MDB device to another.

I typically like to compare MDB to the USB port on a personal computer (PC). USB is an international electrical bus standard which supplies an electrical connection and protocol for connecting peripheral devices (such as a mouse) to a PC. Likewise, the MDB is the vending industry's international standard for providing an electrical connection with protocol for peripheral devices (in this case, an example would be a coin mech) to the VMC.

The one thing MDB does that USB doesn't do is that MDB provides sufficient power to operate the transactional device. (USB can power very low draw devices, but it wasn't designed to power most PC peripheral devices.)

When an MDB device is connected to an MDB machine, the device identifies itself to the machine as to the type of device it is (coin mech, bill validator or cashless system) and the currency for which the MDB device is programmed to receive. The VMC recognizes and enables the MDB device for operation, after which the MDB device and VMC communicate constantly.

The dialogue establishes that a machine is active for taking in currency or cashless, transmitting each activity that occurs with the MDB device, such as each occurrence of a coin being accepted into a coin mechanism; a bill being accepted into a bill validator; or a credit card, tap-and-go device or keyfob being accepted by a cashless system). The machine establishes a monetary credit and shows the credit on the display.

Since the VMC is the brains of the machine, it determines if enough credit is present in the machine to enable a vend. When a vend occurs, the VMC communicates back to the transactional device MDB to complete the transaction. For a coin mech, it means pay back change; for a bill validator, stack the bill from escrow; for a cashless device, it means transmit the vend price and transactional information over to the processor or local card server (college); and for a stored value cashless system, it means writing new stored value back to the magnetic card or smart token or keyfob.

ERROR MESSAGE COMMUNICATION

One of the very nice features of MDB is that MDB devices communicate status to the VMC. This means if there is a problem with a device, the device communicates a message to the VMC indicating the error. Examples of this are bill jams, bill stacker capacity status, coin mech problems, etc. This feature is particularly useful when used with remote data collection systems, where error messages can be forwarded to field service personnel via text messages or email.

TRACKING VENDING ACTIVITY THROUGH MDB

When MDB was originally conceived, MDB communications was limited to transaction device identification and operational communications between the device and the VMC. Information such as vend selection was not available, mainly because it is internal to the VMC and does not need to be transmitted on the MDB.

Eventually, cashless device suppliers lobbied NAMA/EVA to change the specifications for the MDB to accommodate transmission of selection information on the MDB, so that information is now available.

Some cashless and remote data communication providers choose to bypass DEX and derive audit information from MDB communications. While it is possible to derive sales and selection choices, the information produced by MDB is not as detailed as DEX, because it was never intended to be.

DEX and MDB are clearly distinct technologies. DEX allows product auditing, cash accountability and possible pre-kitting, while MDB is the means in which various transactional devices operate and communicate with the brains of the vending machine. DEX is used with a handheld unit or remote monitoring, of which the MDB is an internal component. Both DEX and MDB were meant to make it easier to deploy useful technology in vending equipment.

PRESS RELEASE
September 10, 2013

Apple Announces iPhone 5s—The Most Forward-Thinking Smartphone in the World

iPhone 5s Features 64-bit A7 chip, All-New 8 Megapixel iSight Camera with True Tone Flash & Introduces Touch ID Fingerprint Sensor

CUPERTINO, California—September 10, 2013—Apple® today announced iPhone® 5s, the most forward-thinking iPhone yet, featuring an all-new A7 chip, making iPhone 5s the world's first smartphone with 64-bit desktop-class architecture for blazing fast performance in the palm of your hand. iPhone 5s redefines the best smartphone experience in the world with amazing new features all packed into a remarkable thin and light design, including an all-new 8 megapixel iSight® camera with True Tone flash and introducing Touch ID™, an innovative way to simply and securely unlock your phone with just the touch of a finger. iPhone 5s comes with iOS 7, the most significant iOS update since the original iPhone, engineered for 64-bit technology and featuring hundreds of great new features, including Control Center, Notification Center, improved Multitasking, AirDrop®, enhanced Photos, Safari®, Siri® and iTunes Radio℠.

"iPhone 5s is the most forward-thinking smartphone in the world, delivering desktop class architecture in the palm of your hand," said Philip Schiller, Apple's senior vice president of Worldwide Marketing. "iPhone 5s sets a new standard for smartphones, packed into its beautiful and refined design are breakthrough features that really matter to people, like Touch ID, a simple and secure way to unlock your phone with just a touch of your finger."

The all-new A7 chip in iPhone 5s brings 64-bit desktop-class architecture to a smartphone for the first time. With up to twice the

CPU and graphics performance, almost everything you do on iPhone 5s is faster and better than ever, from launching apps and editing photos to playing graphic-intensive games—all while delivering great battery life. Apple also engineered iOS 7 and all the built-in apps to maximize the performance of the A7 chip. iPhone 5s is the best mobile gaming device with access to hundreds of thousands of games from the App Store℠, the A7 chip's 64-bit architecture and support for OpenGL ES version 3.0. iPhone 5s delivers incredibly rich and complex visual effects, previously only possible on Macs, PCs and gaming consoles.

Every iPhone 5s includes the new M7 motion coprocessor that gathers data from the accelerometer, gyroscope and compass to offload work from the A7 for improved power efficiency. Developers can also access new CoreMotion APIs that take advantage of M7, so they can create even better fitness and activity apps that go well beyond what other mobile devices offer. The M7 motion coprocessor continuously measures your motion data, even when the device is asleep, and saves battery life for pedometer or other fitness apps that use the accelerometer all day.

iPhone 5s introduces Touch ID, an innovative way to simply and securely unlock your iPhone with just the touch of a finger. Built into the home button, Touch ID uses a laser cut sapphire crystal, together with the capacitive touch sensor, to take a high-resolution image of your fingerprint and intelligently analyze it to provide accurate readings from any angle. Setting up Touch ID to recognize your fingerprint is easy, and every time you use it, it gets better. The Touch ID sensor recognizes the touch of a finger so the sensor is only activated when needed, preserving battery life. All fingerprint information is encrypted and stored securely in the Secure Enclave inside the A7 chip on the iPhone 5s; it's never stored on Apple servers or backed up to iCloud®. Touch ID can also be used as a secure way to approve purchases from the iTunes Store®, App Store or iBooks Store℠.

iPhone 5s makes it even easier to take great photos with the world's most popular camera. The all-new 8 megapixel iSight camera features a larger f/2.2 aperture and a new, larger sensor with 1.5µ pixels for better sensitivity and low-light performance, resulting in better pictures. These improvements, along with the Apple-designed image signal processor in the A7 chip and the new Camera app in iOS 7, provide up to two-times faster auto-focus, faster photo capture, automatic image and video stabilization, and better dynamic range. iPhone 5s introduces the new True Tone flash—the world's first for any camera—that variably adjusts color and intensity for over 1,000 combinations, so photos taken with a flash appear more natural. iPhone 5s also includes a new Burst Mode, Slo-Mo video with 120 fps, a new FaceTime® HD camera for better low-light performance and audio-only FaceTime calls with iOS 7.

iPhone 5s features a remarkable thin and light, precision-crafted design that customers around the world love, including an anodized aluminum body with diamond cut chamfered edges, a stunning 4-inch Retina® display and glass inlays. iPhone 5s is available in three gorgeous metallic finishes including gold, silver and space gray. To complement iPhone 5s, Apple designed premium leather cases in six rich colors—beige, black, blue, brown, yellow and (RED)—with soft, color-matched microfiber lining.

iPhone 5s makes it even easier to connect to high-speed networks with support for up to 13 LTE[1] wireless bands, more than any other smartphone in the world. With download speeds up to 100 Mbps[2], you can browse, download and stream content even faster. iPhone 5s includes dual-band 802.11 a/b/g/n Wi-Fi support for up to 150 Mbps[2] and Bluetooth 4.0. iPhone 5s delivers an amazing 10 hours of talk time on 3G networks, up to 10 hours of web browsing on Wi-Fi and LTE networks and up to 8 hours on 3G networks, and up to 10 hours of video playback and up to 40 hours of audio playback.[3]

iPhone 5s comes with iOS 7, the most significant iOS update since the original iPhone, engineered to support the A7 chip's 64-bit architecture, the new iSight camera and Touch ID fingerprint sensor. iOS 7 features a stunning new user interface, completely redesigned with an elegant color palette, distinct, functional layers and subtle motion that make it feel more alive. iOS 7 has hundreds of great new features, including Control Center, Notification Center, improved Multitasking, AirDrop, enhanced Photos, Safari, Siri and introduces iTunes Radio, a free Internet radio service based on the music you listen to on iTunes®.[4]

iPhone 5s customers have access to the revolutionary App Store, which offers more than 900,000 apps to iPhone, iPad® and iPod touch® users in 155 countries around the world. More than 50 billion apps have been downloaded from the App Store to date, offering customers an incredible range of apps in 23 categories, including newspapers and magazines in Newsstand, games and entertainment, business, news, sports, health and fitness and travel.

Designed specifically for iOS, iPhoto®, iMovie®, Pages®, Numbers® and Keynote® are among the most popular apps in the App Store and are now available as free downloads with the purchase of iPhone 5s. iPhoto and iMovie enable you to do more than you ever thought possible with your photos and movies, and with Pages, Numbers and Keynote you can create, edit and share stunning documents, spreadsheets and presentations on your iPhone, iPad or iPod touch.

Pricing & Availability

iPhone 5s comes in gold, silver or space gray, and will be available in the US for a suggested retail price of $199 (US) for the 16GB model and $299 (US) for the 32GB model and $399 (US) for the 64GB

model.[5] iPhone 5s will be available from the Apple Online Store (www.apple.com), Apple's retail stores, and through AT&T, Sprint, T-Mobile, Verizon Wireless and select Apple Authorized Resellers. iPhone 5s cases will be available in beige, black, blue, brown, yellow and (RED) for a suggested retail price of $39 (US) through the Apple Online Store (www.apple.com), Apple's retail stores and select Authorized Apple Resellers. iPhone 5s will be available in the US, Australia, Canada, China, France, Germany, Hong Kong, Japan, Puerto Rico, Singapore and the UK on Friday, September 20. A new iPhone 4S 8GB model will also be available for free.[5] iOS 7 will be available as a free software update starting on Wednesday, September 18 for iPhone 4 and later, iPad 2 and later, iPad mini and iPod touch (fifth generation). Some features may not be available on all products.

[1] LTE is available through select carriers. Network speeds are dependent on carrier networks, check with your carrier for details.
[2] Based on theoretical speeds, actual speeds may vary.
[3] Battery life depends on device settings, usage and other factors. Actual results vary.
[4] iTunes Radio will be available with the launch of iOS 7 in the US.
[5] For qualified customers.

Apple designs Macs, the best personal computers in the world, along with OS X, iLife, iWork and professional software. Apple leads the digital music revolution with its iPods and iTunes online store. Apple has reinvented the mobile phone with its revolutionary iPhone and App Store, and is defining the future of mobile media and computing devices with iPad.

Press Contacts:
Teresa Brewer
Apple
tbrewer@apple.com
(408) 974-6851

Natalie Kerris
Apple
nat@apple.com
(408) 974-6877

Newsroom

The latest news and updates, direct from Apple.

Read more ›

NATIONAL AUTOMATIC MERCHANDISING ASSOCIATION

Serving the Vending / Foodservice management industry

Multi-Drop Bus / Internal Communication Protocol

MDB / ICP

Supported by the Technical Members of:

NAMA	**N**ational **A**utomatic **M**erchandising **A**ssociation
EVA	**E**uropean **V**ending **A**ssociation
EVMMA	**E**uropean **V**ending **M**achine **M**anufacturers **A**ssociation

Version 3.0

March 26, 2003

National Automatic Merchandising Association
20 N. Wacker Drive, Suite 3500
Chicago, Illinois 60606-3120 USA
312-346-0370 FAX 312-704-4140
E-Mail: Tech@vending.org

(this page intentionally left blank)

Multi-Drop Bus / Internal Communication Protocol

Table of Contents

Appendix 1
Currency Codes

Appendix 2
Battery Operated Card Reader

Revisions

Version 3.0

Version 3.0 of this specification is the third release of the international **Multi-Drop Bus / Internal Communication Protocol** (**MDB / ICP**). This specification is the continued effort put forth by technical members of NAMA and the EVA. The basis for this specification is the Version 2.0 international **Multi-Drop Bus / Internal Communication Protocol** (**MDB / ICP**) released on October 4, 2002.

Of special note are the four major changes that were made to the specification:

- Added a second Cashless Device peripheral address in Section 7
- Replaced the Audit Unit with the Communications Gateway in Section 8
- Added the Coin Hopper or Tube – Dispenser in Section 10 (new)
- Assigned 2 addresses to be used for experimental peripherals

The following lists the primary revisions to the Version 3.0 of the **MDB / ICP.**

Section 1 – General Information

Section 1.3

- Changed the Level and Options chart for the Communications Gateway and the Coin Hopper or Tube – Dispenser

Section 2 – Communication Format

Section 2.2

- Added headers for the Response Codes
- Clarified non response processing for Master-to-Peripheral and Peripheral-to-Master communication.

Section 2.3

- Updated the Peripheral Address table for the Communications Gateway, Coin Hopper, Cashless Payment 1, and Experimental addresses
- Defined the use of the experimental addresses

Section 2.5

- Added new RESET examples F & G.

Section 5 – Coin Acceptor / Changer

Section 5.2

- Renamed the STATUS command to SETUP command
- Added a new Possible Credited Coin Removal status code (0Dh)

Section 6 – Bill Validator

Section 6.2

- Renamed the STATUS command to SETUP command
- Added a new Possible Credited Bill Removal status code (0Ch)

Section 7 – Cashless Device(s) **(New Cashless Device #2)**

Changed name from Cashless Payment to Cashless Device

Section 7.1

- Added information regarding the dual addresses for two Cashless Device peripherals (10h and 60h)

Section 7.3

- Updated Command & Response table for dual addresses

Section 7.4

- Updated Command/Response Formats for dual addresses

Section 8 – Communications Gateway **(New Peripheral)**

Sections 8.1 through 8.6

- Replaced former Audit Unit sections with new Communications Gateway Sections

Section 9 –Universal Satellite Device (USDC)

Section 9.3

- Updated POLL table with proper number of bytes (FTL portion)
- Changed "numeric row and column" to "Item Number"

Section 10 – Coin Hopper or Tube – Dispenser **(New Peripheral)**

Sections 10.1 through 10.5

- Added complete new sections

Version 2.0

Version 2.0 of this specification is the second release of the international **Multi-Drop Bus / Internal Communication Protocol** (**MDB / ICP**). This specification is the culmination of effort put forth by technical members of NAMA, the EVMMA, and the EVA. The basis for this specification is the Version 1.0 international **Multi-Drop Bus / Internal Communication Protocol** (**MDB / ICP**) released on October 14, 1998.

The following lists the primary revisions to the Version 2.0 of the **MDB / ICP**

Introduction

Foreword

- Clarified that the Standard is a communication interface

Section 1 - General Information

Section 1.1

- Added 3rd paragraph noting interface specification vs. system specification

Section 1.3

- Added entire Levels and Options section

Section 2 - Communication Format

Section 2.1

- Changed Mode Bit Master-to-Peripheral text

Section 2.2

- Removed "command" from Master-to-Peripheral 4th paragraph
- Changed RET description

Section 2.3

- Defined address 0000xxxB (00H) for VMC
- Provided address information to show hexadecimal format

Section 2.4

- Changed format to 2.4.X sub-sections and added 2.4.4 on Levels

Section 2.5

- Changed RET description

Section 2.6

- Added complete File Transport Layer Section

Section 3 - Bus Timing

Section 3.1

- Added 2nd sentence to t_{setup}

Section 4 - Hardware Specification

Section 4.3

- Modified complete section and added AMP as alternate source to Molex

Section 4.4

- Added pin numbers to schematic

Section 5 - Coin Acceptor / Changer

Section 5.1

- Provided additional address information

Section 5.3

- Added recommended RESET command sequence
- Modified STATUS response to indicate Country / Currency Codes
- Modified County / Currency Code to include ISO 4217 (Appendix A1)
- Added Note 2 to DISPENSE (ODH) command
- Added FTL POLLed responses
- Added FTL "b3" option bit
- Added FTL expansion commands
- Cosmetic changes to all EXPANSION commands
- Split ALTERNATIVE PAYOUT (0FH-02H) and PAYOUT STATUS (0FH-03H) command into two separate commands (cosmetic change only)
- Added text to ALTERNATIVE PAYOUT (0FH-02H) Y1 description
- Added Note 3 to ALTERNATIVE PAYOUT STATUS (0FH-03H)

Section 5.5

- Added "**See Note 2 …**" text
- Added "If both peripherals supported" to Note

Section 6 - Bill Validator

Section 6.1

- Provided additional address information

Section 6.3

- Added recommended RESET command sequence
- Modified STATUS response to indicate Country / Currency Codes
- Modified County / Currency Code to include ISO 4217 (Appendix A1)
- Added Level 2 information
- Added Level 2 option bytes w/ new EXPANSION COMMANDs:
  37H 01H Level 2 Option Bit Enable
  37H 02H Level 2 Identification
- Added FTL POLLed responses
- Added FTL "b0" option bit
- Added FTL expansion commands
- Modified last sentence in SECURITY command to link to Z9-Z10 STATUS response
- Cosmetic changes to all EXPANSION commands

Section 6.5

- Added "If both peripherals supported" to Note

Section 7 - Cashless Payment

Section 7.2 & 7.2.7

- Added Level 03 Negative Vend Request

Section 7.2.2

- Changed 1st sentence to link Setup to 7.4.1 information

Section 7.2.4

- Added Negative Vend and Revalue

Section 7.2.7

- Added Level 03 Negative Vend Request

Section 7.3

- Added bold text regarding defining currency at the beginning of a session
- Broke uninterruptable table into VMC Command and Reader Response
- Added Level 03 NEGATIVE VEND REQUEST to VMC Command table
- Added Level 03 DATA ENTRY REQUEST to Reader Response table
- Highlighted command out of sequence hard resets from VMC
- Moved Vend Failure Sequence to 7.4.8

Section 7.3 – Table 1

- Changed name to COMMANDS & RESPONSES
- Changed Comment column to VMC / Reader Level Support
- Linked all commands and responses to Levels
- Added DATA ENTRY REQUEST POLLed responses
- Added FTL POLLed responses
- Added FTL commands
- Added NEGATIVE VEND REQUEST responses
- Defined 14H-1AH and 20H-FEH as “For Future Use”

Section 7.4.1

- Cosmetically modified RESET command sequence
- Added 32 bit SETUP MAX/MIN PRICE
- Changed text following **Reader response**

Section 7.4.2

- Clarified Level 01 information (reader has no revalue capability)
- Added Level 03 information
- Modified SETUP response to indicate Country / Currency Codes
- Modified County / Currency Code to include ISO 4217 (Appendix A1)
- Added bold Note in Z3-Z4 County / Currency Code
- Added definition for Miscellaneous Options “b4 – b7”

Section 7.4.3

- Added Level 03 SETUP if Expanded Currency Mode

Section 7.4.4

- Added Level 03 BEGIN SESSION response if Expanded Currency Mode
- Added Level 03 VEND APPROVED response if Expanded Currency Mode
- Added Level 03 PERIPHERAL ID response if Expanded Currency Mode
- Clarified COMMAND OUT OF SEQUENCE definition

- Added Level 03 REVALUE LIMIT AMOUNT response if Expanded Currency Mode
- Added Level 03 DATA ENTRY REQUEST response if Data Entry Mode
- Added Level 03 DATA ENTRY CANCEL response if Data Entry Mode
- Added Level 03 FTL REQ TO RCV response if FTL Mode
- Added Level 03 FTL RETRY / DENY response if FTL Mode
- Added Level 03 FTL SEND BLOCK response if FTL Mode
- Added Level 03 FTL OK TO SEND response if FTL Mode
- Added Level 03 FTL REQ TO SEND response if FTL Mode

Section 7.4.5

- Added Level 03 VEND command if Expanded Currency Mode
- Added Level 03 VEND APPROVED response if Expanded Currency Mode

Section 7.4.8

- Added Vend Failure (from 7.3)

Section 7.4.10

- Added Level 03 VEND command if Expanded Currency Mode

Section 7.4.11 (new)

- Added complete Level 03 NEGATIVE VEND Request section

Section 7.4.15 (new)

- Added complete Level 03 DATA ENTRY Request section

Section 7.4.16

- Added Level 03 REVALUE Request command if Expanded Currency Mode

Section 7.4.17

- Added Level 03 REVALUE Limit Request command if Expanded Currency Mode

Section 7.4.18

- Added Level 03 EXPANSION REQUEST ID response if Expanded Currency Mode

Section 7.4.22

- Added Level 03 EXPANSION ENABLE OPTIONS command

Section 7.4.23

- Added Level 03 FTL REQ TO RCV command & responses if FTL Mode

Section 7.4.24

- Added Level 03 FTL RETRY / DENY command if FTL Mode

Section 7.4.25

- Added Level 03 FTL SEND BLOCK command & response if FTL Mode

Section 7.4.26

- Added Level 03 FTL OK TO SEND command if FTL Mode

Section 7.4.27

- Added Level 03 FTL REQ TO SEND command & responses if FTL Mode

Section 7.7

- Added Example Vend Session #10 (Single Negative Vend)

Section 8 - Audit Device

Section 8.1

- Provided additional address information

Section 8.3

- Added FTL POLLed responses
- Added FTL "b3" option bit
- Added FTL expansion commands

Section 9 - Universal Satellite Device

Section 9.1

- Provided additional address information

Section 9.3

- Added FTL POLLed responses
- Added FTL "b2" option bit
- Added FTL expansion commands

Document Revision History

- Deleted

Appendix 1 - Currency Codes

- Added entire section (based on **ISO 4217**)

Appendix 2 - Battery Operated Card Reader

- Added entire section

Version 1.0

Version 1.0 of this specification is the first release of the international **Multi-Drop Bus / Internal Communication Protocol** (**MDB / ICP**). This specification is the culmination of effort put forth by technical members of NAMA, the EVMMA, and the EVA. The basis for this specification is the **International Multi-Drop Bus Interface Standard** published by NAMA and the **Internal Communication Protocol** published by the EVMMA. The NAMA document was originally introduced on October 19, 1993 and later revised on August 19, 1994, June 20, 1997, and October 15, 1997. The EVMMA document was adopted in 1994 and later revised in 1995.

The following lists the primary revisions to the original two documents which were "combined" to create Version 1.0 of the **MDB / ICP**. In actuality, the NAMA **MDB** was the basis of the **MDB / ICP** with the exception of Section 7 which came from the EVMMA **ICP**. Besides typographical corrections and actual feature changes (below), the entire document was edited to provide a more uniform appearance.

The following lists the primary revisions to the Version 1.0 of the **MDB / ICP.**

Hardware Specification - Section 4.3

- Added drawings of the MDB male and female connectors.

Coin Acceptor / Changer - Section 5.3

- Added Expansion commands:
  - 0F-05 Send Current Diagnostic Status
  - 0F-06 Send Controlled Manual Fill Report
  - 0F-07 Send Controlled Manual Payout Report

Coin Acceptor / Changer - Section 5.5

- Added coin acceptance and coin payout power requirements for coin changers using motorized payout mechanisms.
- Added note about simultaneously supplying bill validator transport power.

Bill Validator - Section 6.5

- Added note about simultaneously supplying coin mechanism coin acceptance power.

Cashless Payment - Section 7.2.6

- Added Level 02 Revalue capability.

Cashless Payment - Section 7.3

- Added Level 02 REVALUE REQUEST.
- Removed NAK (NCK) response from uninterruptable state and unexecutable command descriptions.
- Eliminated the BUSY response to vend failure sequences.
- Modified Table 1 per above.

Cashless Payment - Section 7.4.1

- Further defined the initializing sequence following a RESET command.

Cashless Payment - Section 7.4.2

- Further defined the Z7 Application Maximum Response Time.
- Added Z8 – b3 for supporting the VEND/CASH SALE subcommand.

Cashless Payment - Section 7.4.4

- Begin Session (03h) - Added Level 02 Reader Z4-Z10 data.
- Malfunction/Error (0Ah) - Added error code 1100 (refund error).
- Command Out of Sequence (0Bh) - Added Z2 data.
- Eliminated Busy (0Ch) response.
- Added Level 02 Reader Revalue Approved (0Dh) response.
- Added Level 02 Reader Revalue Denied (0Eh) response.
- Added Level 02 Reader Revalue Limit Amount (0Fh) response.
- Added Level 02 Reader User File Data (10h) response.

- Added Level 02 Reader Time/Date Request (11h) response.

Cashless Payment - Section 7.4.10

- Added Level 01 Reader CASH SALE (13h/05h) VMC command.

Cashless Payment - Section 7.4.14

- Added Level 02 Reader Revalue - Request (15h/00h) VMC command.

Cashless Payment - Section 7.4.15

- Added Level 02 Reader Revalue – Limit Request (15h/01h) VMC command.

Cashless Payment - Section 7.4.17

- Obsoleted EXPANSION – Read User File (17h/01h) VMC command.

Cashless Payment - Section 7.4.18

- Obsoleted EXPANSION – Write User File (17h/02h) VMC command.

Cashless Payment - Section 7.4.19

- Added Level 02 Reader Write Time/Date File (17h/03h) VMC command.

Cashless Payment - Section 7.5

- Further defined the non-response time with the "Application Maximum Response Time" Z7.

Cashless Payment - Section 7.6 (original ICP Spec)

- Moved this section (ICP Payment Media Return Button) to Section 7.3.2.

Cashless Payment - Section 7.6 (MDB/ICP Spec)

- Previously was the ICP 7.7 with no modifications.

(*this page intentionally left blank)*

Introduction

Foreword

This voluntary Standard contains basic requirements for a vending machine communication interface within the limitations given below and in the General Information section of this Standard. These requirements are based on sound engineering principles, research, field experience, and an appreciation of the problems of manufacture, installation, and use derived from consultation with and information obtained from manufacturers, users, and others having specialized experience. These requirements are subject to revision as further experience and investigation may show it necessary or desired.

NAMA, in performing its functions in accordance with its objectives, does not assume or undertake to discharge any responsibility of the manufacturer or any other party. The opinions and findings of NAMA represent its professional judgment given with due consideration to the necessary limitations of practical operation and state of the art at the time the NAMA Standard is processed. NAMA shall not be responsible to anyone for use or reliance upon Standard by anyone. NAMA shall not incur any obligation or liability for damages, including consequential damages, arising out of or in connection with the use, interpretation of, reliance upon this Standard.

Standard Review

A complete review of this standard shall be conducted at least every five years to keep requirements consistent with technology. These reviews shall be conducted by representatives from industry and user groups on the NAMA Vending Technology Standards Committee at that time.

(this page intentionally left blank)

Section 1

General Information

1.1 Introduction

This document defines a serial bus interface for electronically controlled vending machines. The interface is a 9600 baud Master-Slave arrangement where all peripherals are Slaves to a Master controller.

The intent of this document is to standardize vending machines that employ electronic control (traditionally known as vending mechanism controller - VMC) so that all vending and peripheral equipment communicates identically.

It should be noted that this document is a vending machine interface / protocol specification and **not** a vending machine system specification. Each machine manufacturer should provide a specification on the overall operation of the machine.

1.2 Operational and Application Notes

The serial bus, or Multi-Drop Bus (MDB) is configured for Master-Slave operation. There is one Master with capability of communicating with up to thirty-two peripherals. The Master is defined as the Vending Machine Controller (VMC).

Each peripheral is assigned a unique address and command set. The master will "poll" the Bus for peripheral activity. That is, each peripheral is asked for activity, and responds with either an acknowledge, negative acknowledgment, or specific data dependent on its current activity. If a peripheral does not respond within a predefined time, (t-non-response as defined in the peripheral sections) it is assumed that it is not present on the Bus.

Bus interference, or "crashes" are prevented because each peripheral only responds upon being polled. Since there is only one master, and all communication is initiated by the Master, Bus "crashes" are easily precluded.

All peripherals will recognize a disable command, or commands, sent by the Master. This allows for disabling of individual peripherals for various reasons, for example, power management techniques.

Error checking and correction is accomplished by using checksums (CHK) and a retransmit command.

1.3 Levels and Options

Since the introduction of the earliest Multi-Drop Bus specification, functional levels and operational options have been established for most of the peripherals on the MDB/ICP interface. These have provided the capability for new features to be implemented as new requirements and features were needed for the international vending industry.

1.3.1 Levels

Levels of peripheral functionality were established when a major change occurred in the peripheral that added extended commands and responses. Due to potential conflicts between a VMC level and a peripheral level, neither the VMC nor the peripheral should issue a command or reply with a response that is not supported by the other device.

The VMC must initially determine (via the appropriate STATUS or SETUP command) the level of a peripheral before determining which commands it can issue to that device. **A VMC must only send commands that are supported by the peripheral.** For example, a Level 3 command may only be issued to a Level 3 or higher peripheral and must not be issued to a Level 1 or 2 peripheral.

The Cashless Payment and the Universal Satellite Device can also learn the respective level of the VMC for that device. This information is sent via the SETUP command. **It is the responsibility of the peripheral to only send responses that are supported by the VMC.** For example, a Level 3 response may only be sent to a Level 3 or higher VMC and must not be sent to a Level 1 or 2 VMC. Effectively, the VMC and peripheral should support the highest common level.

For total compatibility, VMCs and peripherals should support all lower levels. **For new designs after July 2000, it is strongly recommended that VMCs and peripherals must support all lower levels.** Commercial or regional issues may cause machine or peripheral manufacturers to implement only specific levels; however, this is a decision (and risk) made by the machine or peripheral manufacturer.

1.3.2 Options

Options were established in the peripherals to provide various additional operational features that may be required for specific vending applications. As the name implies, these features are “above and beyond” the standard core of required functionality.

At power on and after a Bus Reset or a RESET command, all options are disabled. During the initialization command sequences, the VMC determines the optional features supported by the peripherals. The VMC will then enable the features it is going to use. Until the feature is enabled, it is the responsibility of the peripheral to ignore feature specific commands and not respond with feature specific responses.

1.3.3 Currently Established Levels and Options

The following table provides a brief description of each of the currently established levels and options of the various MDB/ICP peripherals. Please refer to the specific sections for each device for more detailed information.

Peripherals	Levels	Options	Description
Coin Changer	1	n/a	Never released
	2	none	Supports standard commands
	3	below	Supports Expansion ID command and optionally supports commands for features below
		b0	Alternative Payout Method
		b1	Extended Diagnostics
		b2	Controlled Manual Fill and Payout
		b3	File Transport Layer (FTL)
Bill Validator	1	none	Supports standard commands and Expansion ID command without options
	2	below	Supports expansion ID command with options and optionally supports commands for features below
		b0	File Transport Layer (FTL)
Cashless Device #1 & #2	1	below	Supports standard commands and Expansion ID command. Readers do not have revaluation capability
		b0*	Reader is capable of restoring funds to card
		b1*	Reader is multivend capable
		b2*	Reader has a display available
		b3*	Reader supports VEND-CASH SALE command
			*bits in the SETUP-Config command
	2	above	Supports Revalue, Time/Date, Read User File (obsolete), and Write User File (obsolete) commands

Peripherals	Levels	Options	Description
Cashless Device #1 & #2 (continued)	3	above & below	Supports expansion ID command with options and optionally supports commands for features below (bits in the Level 3 Expansion ID command)
		b0**	File Transport Layer (FTL)
		b1**	16 or 32 Bit Monetary Format
		b2**	Multi Currency / Multi Lingual
		b3**	Negative Vend
		b4**	Data Entry
			**bits in the Level 3 Expansion ID command
Communications Gateway	1	none	Obsolete (former Audit Unit)
	2	none	Obsolete (former Audit Unit)
	3	below	Supports Expansion ID command and optionally supports commands for features below
		b0	File Transport Layer (FTL)
		b1	Verbose Mode
Universal Satellite Device (USD)	1	below	Supports all basic commands and optionally supports commands for features below
		b0	USD is capable of storing and controlling pricing
		b1	USD is capable of selecting items to vend
		b2	File Transport Layer (FTL)
Coin Hopper or Tube - Dispenser	1	below	Supports Expansion ID command and optionally supports commands for features below
		b0	File Transport Layer (FTL)

Section 2

Communication Format

2.1 Byte Format

Baud Rate: 9600 NRZ

Serial Bit Format:
1 Start Bit
8 Data Bits
1 Mode Bit
1 Stop Bit

11 Bits Total

LSB									MSB
Start	0	1	2	3	4	5	6	7	Mode	Stop

Mode Bit: Master-to-Peripheral

The mode bit differentiates between ADDRESS bytes and DATA bytes. ADDRESS bytes must be read by all peripherals, DATA bytes are only read by the peripheral that has been addressed.

The mode bit is set (logic one) to indicate an ADDRESS byte, and not set (logic zero) to indicate a DATA byte.

Mode Bit: Peripheral-to-Master

The mode bit must be set on the last byte sent when data is sent from a Slave to the Master.

2.2 Block Format

Master-to-Peripheral

A Communication Block for Master-to-Slave transmissions is defined as an Address byte, optional data bytes, and a CHK byte. A block is limited to a maximum of thirty-six (36) bytes.

The upper five bits (MSB) of the Address Byte will be used for addressing. That is, bits 7,6,5,4,3 of the previous byte description will be used for addressing.

The lower three bits (i.e. 2,1,0) of the Address Byte will contain peripheral specific commands. This will allow up to eight instructions to be embedded in the first byte of a block.

The VMC Master will respond to data from a peripheral with an Acknowledgment (ACK), Negative Acknowledgment (NAK), or Retransmit (RET). These are defined later in the document. The 5 mS time-out (t-response) described in the Bus Timing section of this document is the equivalent of a NAK.

If the addressed Slave does not respond within the 5 mS time-out (silence), the Master may repeat the same command, or send a different command, until it receives an answer or until the end of the Non-Response time, as defined in the peripheral sections. See Example in 2.5D. The RESET command should not be used as a recovery method to a 5 mS time-out (t-response) until after exceeding the Non-response time. The VMC may send commands to any other peripheral during this time.

Peripheral-to-Master

A Communication Block for Slave-to-Master transmissions consists of either a data block and a CHK byte, a acknowledgment (ACK), or a negative acknowledgment (NAK).

The 5 mS time-out (t-response) described in the Bus Timing section of this document is the equivalent of a NAK command. In addition, it is recommended that the peripheral use this time-out as the NAK when a reception error of the ADDRESS byte occurs. This will prevent several peripherals from trying to simultaneously respond with a NAK.

A data block consists of one or more data bytes followed by a CHK byte. The CHK byte is defined later in this document.

The data block and CHK byte are limited to a maximum size of 36 bytes.

A CHK byte is not required when a peripheral responds with NAK or ACK byte. ACK and NAK are defined later in this document.

The peripheral must set the mode bit on the last byte sent to signify end of transmission. This will be either the CHK byte of a block, a NAK byte, or an ACK byte. The mode bit must not be set except for the conditions above.

A peripheral response of ACK or NAK signifies the end of the exchange.

When a peripheral responds with a data block, the VMC must respond with an ACK, NAK or RET. If the Master cannot respond within the 5 mS time-out (t-response) the peripheral must repeat the data block, or append it, at the next possible occasion (i.e. to a later POLL). The same behavior is to apply when the Master responds with NAK.

CHK Byte

A CHK byte must be sent at the end of each block of data. The CHK byte is a checksum calculated by adding the ADDRESS byte and all DATA bytes. The CHK byte is not included in the summation. The carry bit for CHK additions is ignored since the CHK byte is limited to eight bits.

The following example shows a CHK byte calculation for a possible response to a STATUS command sent to a USA changer slave. See section 5 for details of byte meanings.

02H	Changer feature level
00H	Country code for USA
01H	Country code for USA
05H	Coin scaling factor
02H	Decimal place
00H	Coin type routing
07H	Coin type routing
01H	Coin type 0 has value of 1 scaling factor
02H	Coin type 1 has value of 2 scaling factor
05H	Coin type 2 has value of 5 scaling factor
14H	Coin type 3 has value of 20 scaling factor
FFH	Coin type 4 is a token
12CH	Therefore the CHK byte would be equal to 2CH

A checksum will be performed on all full blocks of communication. A checksum will not be performed on ACK, NAK, or RET bytes.

Response Codes

The following codes are reserved for the ACK, NAK and RET bytes:

ACK	00H	(acknowledgment/checksum correct)
RET	AAH	(Retransmit the previously sent data. **Only the VMC can transmit this byte**)
NAK	FFH	(Negative acknowledge)
		The VMC and peripheral must also recognize the 5 mS time-out (t-response) as a NAK.

NOTE: To improve system reliability it is recommended that when receiving ACK, NAK, or RET the receiving device counts the number of bits set in the byte. This method will require at least two bit errors in the byte before the byte can be mis-interpreted.

Bus Reset

The VMC may reset all peripherals by pulling the transmit line "active" for a minimum of 100 mS. This informs all peripherals to abort any activity and return to its power-on reset state. Details of this state for each peripheral are provided in later sections of this document. It is recommended that the VMC re-initialize each peripheral after this type of reset.

2.3 Peripheral Addresses

The addresses below are defined. Note again that the bits shown are the upper five bits (7,6,5,4,3) of the Address Byte and will be used for all addressing including the File Transport Layer described in Section 2.6. The lower three bits (2,1,0) are used for the command.

Address		Definition
00000xxxB	(00H)	Reserved for VMC
00001xxxB	(08H)	Changer
00010xxxB	(10H)	Cashless Device #1
00011xxxB	(18H)	Communications Gateway
00100xxxB	(20H)	Display
00101xxxB	(28H)	Energy Management System
00110xxxB	(30H)	Bill Validator
00111xxxB	(38H)	Reserved for Future Standard Peripheral
01000xxxB	(40H)	Universal Satellite Device #1
01001xxxB	(48H)	Universal Satellite Device #2
01010xxxB	(50H)	Universal Satellite Device #3
01011xxxB	(58H)	Coin Hopper or Tube - Dispenser
01100xxxB	(60H)	Cashless Device #2
01101xxxB	(68H)	Reserved for Future Standard Peripherals
.	.	.
.	.	.
.	.	.
11011xxxB	(D8H)	Reserved for Future Standard Peripherals
11100xxxB	(E0H)	Experimental Peripheral #1
11101xxxB	(E8H)	Experimental Peripheral #2
11110xxxB	(F0H)	Vending Machine Specific Peripheral #1
11111xxxB	(F8H)	Vending Machine Specific Peripheral #2

Experimental Peripheral Addresses

Experimental Peripheral addresses 11100xxxB (E0H) and 11101xxxB (E8H) are reserved for use by manufacturers when designing and field testing potential new MDB/ICP devices. These addresses are **temporary** and once the new device is approved by NAMA and the EVA, the device will be assigned a different permanent peripheral address. Use of the Experimental Peripheral addresses shall be limited to "in house" testing and "closed site" field trials. Manufacturers must understand that any devices in the field with Experimental Peripheral addresses must be recalled or updated to the permanent address if the device is approved by NAMA and the EVA. If not approved by NAMA and the EVA, the devices must be recalled or have their addresses changed to the Vending Machine Specific peripheral addresses described below.

Vending Machine Specific Peripheral Addresses

Vending Machine Specific peripheral addresses (addresses 11110xxxB (F0H) and 11111xxxB (F8H)) are reserved for Non-Standard or proprietary applications. These devices are allowed a unique set of commands.

All other peripherals are defined as Standard devices. These peripherals must follow the specifications to ensure compatibility between manufacturers.

2.4 Software Operational Rules

2.4.1 Power Budget

The VMC must regulate the power budget. That is, peripherals must be enabled and disabled dependent on power availability. The power bus is defined later in this document.

2.4.2 Bytes

During multi-byte messages the most significant byte is sent first.

> Any bytes within a command or response that are not specifically defined should be left in a 0 state. For Level 03 or lower coin mechanisms, Level 01 bill validators, and Level 01 card readers, this is not a requirement but a suggestion.

2.4.3 Polling

The following are recommendations for the methods of VMC to peripheral software operation.

> Each peripheral should be polled every 25-200 milliseconds. This can be done by the POLL command or any other appropriate command.
>
> If a peripheral has not responded to a poll for its maximum Non-Response time, the VMC should continue to poll the peripheral at least every ten seconds with a RESET command. (See Example G in Section 2.5).

2.4.4 Levels

Due to potential conflicts between a VMC level and a peripheral level, neither the VMC nor the peripheral should issue a command or reply with a response that is not supported by the other device.

The VMC must initially determine (via the appropriate STATUS or SETUP command) the level of a peripheral before determining which commands it can issue to that device. **A VMC must only send commands that are supported by the peripheral.** For example, a Level 3 command may only be issued to a Level 3 or higher peripheral and must not be issued to a Level 1 or 2 peripheral.

The Cashless Payment and the Universal Satellite Device can also learn the respective level of the VMC for that device. This information is sent via the SETUP command. **It is the responsibility of the peripheral to only send responses that are supported by the VMC.** For example, a Level 3 response may only be sent to a Level 3 or higher

VMC and must not be sent to a Level 1 or 2 VMC. Effectively, the VMC and peripheral should support the highest common level.

For total compatibility, VMCs and peripherals should support all lower levels. **For new designs after July 2000, it is strongly recommended that VMCs and peripherals must support all lower levels.** Commercial or regional issues may cause machine or peripheral manufacturers to implement only specific levels; however, this is a decision (and risk) made by the machine or peripheral manufacturer.

2.5 Typical Session Examples

A. The diagram below represents a typical transmission when a peripheral is idle.

VMC:

ADD* CHK

Peripheral:

ACK*

B. The diagram below represents a typical transmission when a peripheral has data to return.

VMC:

ADD* CHK ACK

Peripheral:

DAT DAT CHK*

C. The diagram below represents a typical transmission when the VMC has data to send.

VMC:

ADD* DAT DAT CHK

Peripheral:

ACK*

*Indicates mode bit set

D. The diagram below represents a typical transmission when the VMC determines a CHK is not correct. The VMC will respond one of two ways:

Send a NAK to the peripheral to indicate that the information was not received correctly then perform other tasks. Note: When the Master answers with NAK (or silence which is treated equally) the slave has to repeat the response, in order to ensure the execution of the response (i.e. coin reception etc.).

OR

The VMC may send a retransmit (RET) command alerting the peripheral to retransmit the previously sent data.

VMC:

ADD* DAT CHK RET ACK

Peripheral:

DAT DAT CHK* DAT DAT CHK*

*Indicates mode bit set.

E. This diagram represents a situation where the peripheral does not respond within the 5 mS time-out (t-response).

VMC:

ADD* CHK ADD* CHK

Peripheral:

[silence…] ACK*

F. This diagram represents a situation where the peripheral does not respond to a command and after its maximum Non-Response time, is reset by the controller.

Controller		Peripheral	Comment
Command X	→		
	←	Response	Normal response
Command Y	→		
	←	[silence…]	No response
Command Y	→		
	←	[silence…]	No response
Command Y	→		
	←	[silence…]	No response
			Peripheral does not response within its allocated Non-Response Time.
RESET	→		Software Reset
	←	[silence…]	Peripheral in initialization routine
RESET	→		
	←	ACK	Peripheral operational again
POLL	→		
	←	JUST RESET	Peripheral indicates finished RESET processing
ACK	→		
			Peripheral initialization sequence is performed as recommended in each peripheral section.

G. This diagram represents a situation where the peripheral is disconnected or goes offline. The controller should send a RESET command every 10 seconds to determine if, and when, the peripheral becomes active again.

Controller		Peripheral	Comment
Command X	→		
	←	Response	Normal response
Command Y	→		
	←	[silence…]	No response
Command Y	→		
	←	[silence…]	No response
Command Y	→		
	←	[silence…]	No response
			Peripheral does not response within its allocated Non-Response Time.
RESET	→		Software Reset
	←	[silence…]	Peripheral offline
RESET	→		Software Reset
	←	[silence…]	Peripheral offline
			Wait 10 seconds
RESET	→		
	←	[silence…]	Peripheral offline
			Wait 10 seconds
RESET	→		
		[silence…]	Peripheral offline
			Wait 10 seconds
RESET	→		
		[silence…]	Peripheral offline

2.6 File Transport Layer

The **F**ile **T**ransport **L**ayer (**FTL**) provides a method to send and/or receive high level information between peripherals or between a peripheral and the VMC. It is **not** intended to be used for standard vending credit and control functions. An example would be loading new validation parameters into a coin changer or bill validator.

Since the MDB/ICP interface is "driven" by the VMC, it has to be a network manager for all FTL data transfers. It acts as a temporary mailbox and data switch for FTL blocks; however, the information that is sent via FTL does not have to be interpreted by the VMC. The VMC simply uses the destination and source address information provided in the MDB/ICP command and response structure to forward the data to the proper recipient.

2.6.1 FTL Process Overview

If a peripheral needs to transfer data to another peripheral (or the VMC):
- The VMC must poll the peripheral,
- The peripheral must answer with a "REQUEST TO SEND",
- The VMC must get approval to forward data (if necessary),
- The VMC requests the first data block,
- The VMC ACKs the first block and forwards to destination,
- The process repeats until all blocks are sent.

If the VMC needs to transfer data to a peripheral:
- The VMC must send a "REQUEST TO SEND",
- The peripheral approves or denies the transfer request,
- If approved, the VMC sends the first data block,
- The peripheral ACKs the first data block,
- The process repeats until all blocks are sent.

If a peripheral (A) needs to request a transfer of data from another peripheral (B):
- The VMC must poll the peripheral A,
- Peripheral A must send a "REQUEST TO RECEIVE",
- The VMC forwards the request to peripheral B,
- Peripheral B decides to honor the request or not,
- If approved, peripheral B sends the first data block,
- The VMC forwards the data block to peripheral A,
- The process repeats until all blocks are sent.

2.6.2 FTL Detailed VMC Operation

The VMC must act as a network manager, it is responsible for checking peripheral status and managing network resources as described below, it must:

- Be aware of which peripherals are active and support the FTL. If a file transfer is requested involving a peripheral that does not support it, the VMC should deny the transfer using RETRY/DENY defined later.
- Poll peripherals to become aware that a data transfer is requested.
- Read data blocks from selected peripherals.
- If VMC receives a NAK, it should attempt to finish current command/response up to 5 times. After that, it should abort file transfer as defined by the protocol.
- Send data blocks to destination device, if not the VMC itself.
- Repeat these steps for all data blocks, as needed.

2.6.3 FTL General Operation

- The FTL "session" would transfer a "file" using several "blocks". The "Dest" and "Src" are switched by the VMC directing each block to its destination.
- All responses can be sent immediately after receipt of command or the command can be ACK'ed and the response sent in a delayed fashion (meeting all appropriate time-outs). However, FTL responses must NOT be combined with responses to any other commands, at any time.
- File transfers less than 256 blocks are terminated by sending an empty data file (SEND BLOCK with no data). File transfers of exactly 256 blocks are terminated by block #FE followed by block #FF.
- It is recommended that files larger than one block:
  1) Include a CRC in their data. The transport layer is not responsible for checking for correct CRCs.
  2) Include a time out mechanism to prevent system dead locks. The transport layer is **not** responsible for checking for dead locked file transfers.
- To prevent a system dead lock, the VMC must poll other peripherals during all data transfers and service them accordingly.
- Since the VMC is not knowledgeable about the contents of the file transfer it should not disable any peripherals due to a transfer request. This will be the responsibility of the peripherals themselves. They may internally disable and report so to the VMC if possible, or they may just stop responding to the VMC until ready. The latter may cause the VMC to try to reset the peripheral.

2.6.4 FTL Command and Response Sets For All Components

The table below defines the VMC commands and peripheral responses that occur during an FTL data transfer. Note that the peripheral responses can either be immediate to the VMC's command or delayed and provided to a subsequent POLL. Definitions are provided on the following page.

Command / Response	VMC Cmd[1]	Resp	Source Data (bytes)	Destination Response
REQ TO SEND	α7/FE	1F	Dest (1) Src (1) File ID (1) Length (1) Control (1)	OK TO SEND or RETRY/DENY
OK TO SEND	α7/FD	1E	Dest (1) Src (1)	SEND BLOCK (repeated until whole file is transferred)
SEND BLOCK	α7/FC	1D	Dest (1) Block # (1) Data (1 to 31)	ACK
RETRY/DENY	α7/FB	1C	Dest (1) Src (1) Retry delay (1)	ACK
REQ TO RCV	α7/FA	1B	Dest (1) Src (1) File ID (1) Max Length (1) Control (1)	SEND BLOCK (repeated until whole file is transferred) or RETRY/DENY

Note 1: The α7 represents the address of the destination device (defined in Section 2.3) logically OR'd with a hexadecimal 0x07.

Dest **1 byte**

The destination address of the peripheral where the data block (**not the whole file**) is being sent to. All addresses refer to the standard MDB defined peripheral addresses as defined in Section 2.3. Note that 00000xxx (00H) will be used for the VMC. Examples are a changer (08H), audit system (18H), bill validator (30H), and universal satellite device #2 (48H).

Src **1 byte**

The source address of the peripheral where the data block (**not the whole file**) is being sent from. All addresses refer to the standard MDB defined peripheral addresses as defined in Section 2.3. Note that 00000xxx (00H) will be used for the VMC. Examples are the same as in the **Dest** above.

File ID **1 byte**

The type of information desired. NAMA will maintain a list of standard file ID's and a definition of what each file type means. Note that if a device responds with a "Retry delay" of FFH it should be interpreted that this device does not support the requested function.

Currently defined file IDs include:

00H: Manufacture ID information. This file must start with the manufactures three character manufactures code, anything after that would be up to the manufacture to define.

01H: DTS defined file. This file must follow the format defined in the EVA-DTS standard. This would include the DXS record as well as all data up to and including the DXE record.

0FH to 0FFH: This range of files may be used for Manufacturer Specific information. The content and format of these files are left up to the manufacturer to define.

Additional ID proposals must be evaluated by the NAMA MDB/ICP technical standard committee.

(Max) Length **1 byte**

The total number of blocks that will (can) be included in the entire file. This byte should be used as a counter to determine the amount of data blocks to be transferred.

Control **1 byte**

This byte contains information that can be used by the VMC and peripherals to determine how the data transfer is conducted. Included controls are:

b0: Reset after transfer. The receiving peripheral should reset itself after the file transfer is complete.
b1: End of File. The last block of the current FTL session contains the end of this file. If clear (=0), then another FTL session will follow with additional blocks. If set (=1), then this is the last (or only) FTL session to be sent.
b2 - b7: Not used, must be set to 0

Block # **1 byte**

The sequential number of this block, within the total file, that is being requested/sent. All data blocks must be identified by a block number, counting up from 0 (first block) to 255.

Data Block **1 to 31 byte(s)**

The actual data portion of the block. All data must fit into a 31 byte, or less, string. The standard MDB CHK byte will signify the end of block. (Peripherals will have to use inter-byte time out when receiving blocks from the VMC.) Knowledge of the contents of this data is only required by the source and destination devices.

Retry Delay **1 byte**

A time delay that the sender should wait before trying to re-send the entire data file again. If a device is not capable of receiving a file in its current state, this byte should represent the number of seconds before it will be ready to receive the data. If the device simply refuses to accept the file it must answer with a “Retry Never” signified by a 00H retry delay. If the device is not present, block synchronization is lost, or other failure mode arises a “Retry Never” should be used to abort/deny the current file transfer.

File Transport Layer Examples

Below are examples of data transfers between the VMC and a peripheral or between two different peripherals via the VMC.

SUCCESSFUL TRANSFER – VMC TO PERIPHERAL A			
Peripheral A	**VMC**	**Peripheral B**	**Comments**
	← REQ TO SEND (α7/FE)		Request to send "n" blocks
OK TO SEND (1E)	→		
	← ACK		
	← SEND BLOCK (α7/FC)		Repeated "n" times
ACK	→		

DENIED TRANSFER – VMC TO PERIPHERAL A			
Peripheral A	**VMC**	**Peripheral B**	**Comments**
	← REQ TO SEND (α7/FE)		
RETRY/00 (1C)	→		Denied
	← ACK		

SUCCESSFUL REQUEST – VMC TO PERIPHERAL A			
Peripheral A	**VMC**	**Peripheral B**	**Comments**
	← POLL (varies)		
REQ TO RCV (1B)	→		Request receive "n" blocks
	← ACK		
	← SEND BLOCK (α7/FC)		Repeated "n" times
ACK	→		

DENIED REQUEST – VMC TO PERIPHERAL A			
Peripheral A	**VMC**	**Peripheral B**	**Comments**
	← POLL (varies)		
REQ TO RCV (1B)	→		Request receive "n" blocks
	← ACK		
	← RETRY/00 (α7/FB)		Denied
ACK	→		

VMC ABORTED TRANSFER – VMC TO PERIPHERAL A

Peripheral A		VMC	Peripheral B	Comments
	←	REQ TO SEND (α7/FE)		Request to send "n" blocks
OK TO SEND (1E)	→			
	←	ACK		
	←	SEND BLOCK (α7/FC)		Repeated "n" times
ACK	→			
	←	RETRY/00 (α7/FB)		Aborted!
ACK	→			

PERIPHERAL ABORT TRANSFER – VMC TO PERIPHERAL A

Peripheral A		VMC	Peripheral B	Comments
	←	REQ TO SEND (α7/FE)		Request to send "n" blocks
OK TO SEND (1E)	→			
	←	ACK		
	←	SEND BLOCK (α7/FC)		
RETRY/00 (1C)	→			Aborted!
	←	ACK		

SUCCESSFUL TRANSFER – PERIPHERAL A TO VMC

Peripheral A		VMC	Peripheral B	Comments
	←	POLL (varies)		
REQ TO SEND (1F)	→			Request to send "n" blocks
	←	ACK		
	←	OK TO SEND (α7/FD)		
SEND BLOCK (1D)	→			Repeated "n" times
	←	ACK		

DENIED TRANSFER – PERIPHERAL A TO VMC

Peripheral A		VMC		Peripheral B	Comments
	←	POLL (varies)			Request to send "n" blocks
REQ TO SEND (1F)	→				
	←	ACK			
	←	RETRY/00 (α7/FB)			Denied
ACK	→				

SUCCESSFUL TRANSFER – PERIPHERAL A TO PERIPHERAL B

Peripheral A		VMC		Peripheral B	Comments
	←	POLL (varies)			
REQ TO SEND (1F)	→				Request to send "n" blocks
	←	ACK			
		REQ TO SEND (1F) (α7/FE)	→		
			←	OK TO SEND (1E)	
		ACK	→		
	←	OK TO SEND (α7/FD)			
SEND BLOCK (1D)	→				Repeated "n" times
	←	ACK			
		SEND BLOCK (α7/FC)	→		
			←	ACK	

DENIED TRANSFER – PERIPHERAL A TO PERIPHERAL B

Peripheral A		VMC		Peripheral B	Comments
	←	POLL (varies)			
REQ TO SEND (1F)	→				Request to send "n" blocks
	←	ACK			
		REQ TO SEND (1F) (α7/FE)	→		
			←	RETRY/00 (1C)	Denied
		ACK	→		
	←	RETRY/00 (α7/FB)			
ACK	→				

SUCCESSFUL REQUEST - PERIPHERAL A TO PERIPHERAL B

Peripheral A	VMC	Peripheral B	Comments
	← POLL (varies)		
REQ TO RCV (1B)	→		Request receive "n" blocks
	← ACK		
	REQ TO RCV (α7/FA) →		
		← SEND BLOCK (1D)	Repeated "n" times
	ACK →		
	← SEND BLOCK (α7/FC)		
ACK	→		

DENIED REQUEST – PERIPHERAL A TO PERIPHERAL B

Peripheral A	VMC	Peripheral B	Comments
	← POLL (varies)		
REQ TO RCV (1B)	→		Request receive "n" blocks
	← ACK		
	REQ TO RCV (α7/FA) →		
		← RETRY/00 (1C)	Denied
	ACK →		
	← RETRY/00 (α7/FB)		
ACK	→		

PERIPHERAL A TRANSFER TO PERIPHERAL B – ABORTED BY A

Peripheral A	VMC	Peripheral B	Comments
	← POLL (varies)		
REQ TO SEND (1F)	→		Request to send "n" blocks
	← ACK		
	REQ TO SEND (α7/FE) →		
		← OK TO SEND (1E)	
	ACK →		
	← OK TO SEND (α7/FD)		
SEND BLOCK (1D)	→		
	← ACK		
	SEND BLOCK (α7/FC) →		
		← ACK	
	.		Repeated "n" times
	.		
	.		
	← POLL (varies)		
RETRY/00 (1C)	→		Aborted!
	← ACK		
	RETRY/00 (α7/FB) →		
		← ACK	

PERIPHERAL A TRANSFER TO PERIPHERAL B – ABORTED BY B

Peripheral A	VMC	Peripheral B	Comments
	← POLL (varies)		
REQ TO SEND (1F)	→		Request to send "n" blocks
	← ACK		
	REQ TO SEND (α7/FE) →		
		← OK TO SEND (1E)	
	ACK →		
	← OK TO SEND (α7/FD)		
SEND BLOCK (1D)	→		
	← ACK		
	SEND BLOCK (α7/FC) →		
		← ACK	
	.		Repeated "n" times
	.		
	.		
	← POLL (varies)REQ BLOCK (α7/FD)		
SEND BLOCK (1D)	→		
	← ACK		
	SEND BLOCK (α7/FC) →		
		← RETRY/00 (1C)	Aborted!
	ACK →		
	← RETRY/00 (α7/FB)		
ACK	→		

Section 3

Bus Timing

3.1 Timing Definitions

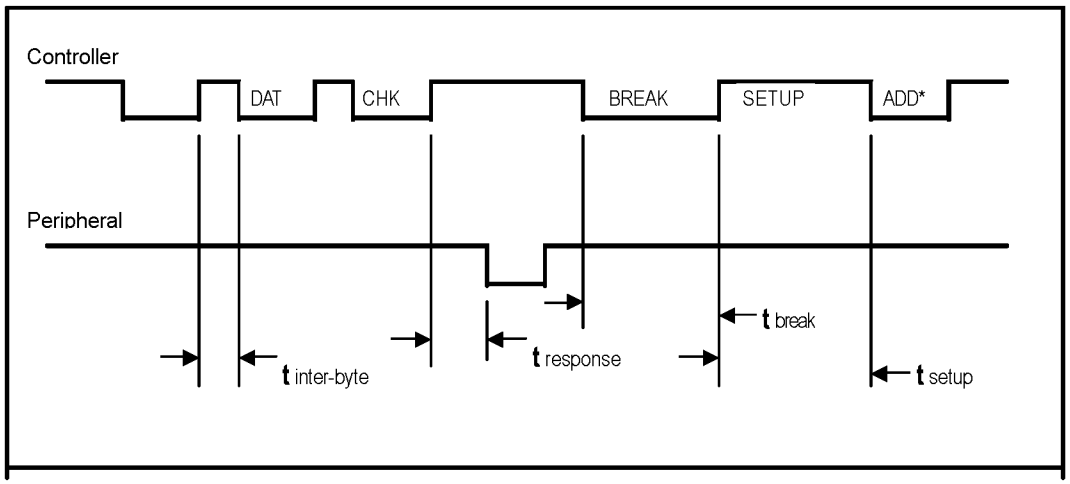
Baud rate	=	The rate of bit transfer per second.
t inter-byte (max.)	=	The maximum time allowed between bytes in a block transmission.
t response (max.)	=	The maximum time any device, master or peripheral, will take to respond to a valid communication.
t break (VMC)	=	The minimum time of the Bus Reset signal sent by the VMC to reset all peripherals.
t setup	=	The minimum set-up time before the VMC attempts to communicate after a reset signal. Peripheral devices may choose to not respond for up to the non-response time defined in each peripheral section.

3.2 Timing Specifications

Baud Rate	=	9600 +1%/-2% NRZ
t inter-byte (max.)	=	1.0 mS
t response (max.)	=	5.0 mS
t break (min.)	=	100 mS
t setup (min.)	=	200 mS

NOTE: All peripherals have the option of not responding to the VMC. Non-response timing is defined in the peripheral specification.

3.3 Timing Diagram



NOTE: * indicates that the mode bit is set

Section 4

Hardware Specification

4.1 Bus Power Supply Definition

The information below defines the minimum VMC voltage output. The actual current ratings per peripheral will be defined in their respective sections.

Power supply filtering is optional, therefore if a peripheral requires more power, or tighter regulation, they may elect to supply their own power, or filtering, from available sources elsewhere in the machine.

VMC Voltage Output:

Minimum	=	20 VDC rms.(rectified and optionally filtered)
Nominal	=	34 VDC unreg.(rectified and filtered) 24 VDC rms.(rectified only)
Maximum	=	42.5* VDC(ripple voltage upper limit) * High line input may allow 45 VDC peak (max.).

4.2 Bus Transmitter / Receiver Specification

The following section describes the 5V, optically isolated, current loop system between the Master and the Slave.

VMC Master:

Transmit:

Minimum source current (active):	100 mA @ 4V
Maximum leakage current (inactive):	100 uA

NOTES: 1) The transmit line must be able to withstand a short while in the active mode.

2) 15 mA should be added for each peripheral over six.

Receive:

Minimum input current (active):	15 mA @ 1V
Maximum input current (inactive):	1 mA

Peripheral Slave:

Receive:

Maximum input current (active):	15 mA @ 4V
Maximum input current (inactive):	100 uA

Transmit:

Minimum sink current (active):	15 mA @ 1V
Maximum leakage current (inactive):	30 uA

4.3 Connector Specification

Connector assemblies supplied by the NAMA approved suppliers, noted in Section 4.3.6, are intermateable and meet or exceed the minimum requirements identified in Sections 4.3.1, 4.3.2, 4.3.3, 4.3.4, and 4.3.5 when tested in the mated condition. NAMA must approve any supplier changes to the fit, form, or function. Discrete components, i.e. contacts, are not required to be inter-changeable between supplier products.

4.3.1. **Material**

4.3.1.1. Terminal: Phosphor Bronze
4.3.1.2. Plating: Tin or Tin/Lead
4.3.1.3. Housing: UL 94V-2 nylon

4.3.2. **Ratings**

Section	Item	Requirement
4.3.2.1.	Rated Voltage (Max)	600 Volts AC
4.3.2.2.	Maximum Rated Current (Six Circuit)	7 Amps
4.3.2.3.	Ambient Temperature Range (including terminal T-rise)	-40°C to +105°C

4.3.3. **Electrical Performance**

Section	Item	Test Condition	Requirement
4.3.3.1.	Contact Resistance	Mate Connectors, measure by dry circuit, 20 mV max., 10 mA. Wire resistance shall be removed from the measured value.	10 mΩ Max.
4.3.3.2.	Insulation Resistance	Mate Connectors, apply 500V DC between adjacent terminal or ground.	1000 MΩ Min.
4.3.3.3.	Dielectric Strength	Mate Connectors, apply 1500V AC for 1 minute between adjacent terminal or ground.	No Breakdown.

4.3.4. **Mechanical Performance**

Section	Item	Test Condition		Requirement
4.3.4.1.	Insertion and Withdrawal Force	Insert and withdraw connectors at a speed rate of 25 +/- 3mm / minute.		Noted Below
	6 Pos Insertion Max.		6 Pos Withdrawal Min.	
	Initial	30th cycle	Initial	30th cycle
	41.2 N	38.2 N	2.9 N	2.4 N
4.3.4.2.	Crimping Pull Out Force	Mount the crimped terminal, apply axial force on the wire at a rate of 25 +/- 3mm minute.		
		16 AWG		88 N Min.
		18 AWG		88 N Min.
		20 AWG		59 N Min.
		22 AWG		39 N Min.
		24 AWG		29 N Min.
		26 AWG		20 N Min.
		28 AWG		10 N Min.
4.3.4.3.	Terminal Insertion Force	Insert the crimped terminal into the housing.		15 N Max.
4.3.4.4.	Terminal/Housing Retention Force	Apply axial pull out force at the speed rate of 25 +/- 3mm / minute.		22 N Min.
4.3.4.5.	Locking / Unlocking Force	Measure force to lock & unlock connector housings (without contacts) at a rate of 25 +/- 3mm / minute.		Lock: 30 N Max. Unlock: 50 N Min.

4.3.5. **Environmental Performance**

Section	Item	Test Condition	Requirement	
4.3.5.1.	Repeated Insertion / Withdrawal	When mated up to 30 cycles repeatedly by rate of 10 cycles per minute.	Contact Resistance	20 mΩ Max.
4.3.5.2.	Temperature Rise	Carrying rated current load.	30°C Rise Max.	
4.3.5.3.	Vibration	Amplitude: 1.5mm P-P Sweep Time: 10-55-10 Hz in 1 minute. Duration: 2 hours in each X,Y,Z axis.	Appearance	No Damage
			Contact Resistance	20 mΩ Max.
			Discontinuity	1 μ sec. Max.
4.3.5.4.	Shock	50 G; 3 strokes in each X,Y,Z axis.	Appearance	No Damage
			Contact Resistance	20 mΩ Max.
			Discontinuity	1 μ sec Max.
4.3.5.5.	Heat Resistance	105 +/- 2°C, 96 hours	Appearance	No Damage
			Contact Resistance	20 mΩ Max.
4.3.5.6.	Cold Resistance	-40 +/- 3°C, 96 hours	Appearance	No Damage
			Contact Resistance	20 mΩ Max.
4.3.5.7.	Humidity	Temperature: 60 +/- 2°C Relative Humidity: 90% - 95% Duration: 96 hours	Appearance	No Damage
			Contact Resistance	20 mΩ Max.
			Dielectric Strength	No Breakdown
			Insulation Resistance	1000 MΩ Min.
4.3.5.8.	Temperature Cycling	5 Cycles: a) - 55°C ; 30 Minutes b) 105°C ; 30 Minutes	Appearance	No Damage
			Contact Resistance	20 mΩ Max.
4.3.5.9.	Salt Spray	48 +/- 4 hours exposure to salt spray from 5 +/- 1% solution at 35 +/- 2°C.	Appearance	No Damage
			Contact Res.	20 mΩ Max.
4.3.5.10.	SO_2 Gas	24 hour exposure to 50 +/- 5 ppm SO_2 gas at 40 +/- 2°C.	Appearance	No Damage
			Contact Res.	Max.

4.3.6 **Approved Suppliers and Part Numbers**

4.3.6.1. Suppliers

Molex : Mini-Fit, Jr.™ Product
AMP: AMP-DUAC™ Product

4.3.6.2. Peripherals

Connector: Six (6) Circuit Receptacle Housing

Molex 39-01-2060
AMP P/N 106527-6

Terminals: Female Contacts (sockets), Tin

Molex 39-00-0065
AMP P/N 106528-2 or 106529-2

Strain Relief: The strain relief shall not exceed a Maximum Form Factor of 0.85 inch wide x 0.75 inch high x 1.90 inch long, excluding integrated hinges and wire ties.

Molex 15-04-0296
AMP P/N 1375618-1

4.3.6.3. Bus Harness

Connector: Six (6) Circuit Plug Housing

Molex 39-01-2061
AMP P/N 794550-6 or 794542-6

Terminals: Male Contacts (pins), Tin

Molex 39-00-0067
AMP P/N 794578-1 or 794576-1

4.3.6.4. VMC Connector (Direct PCB Mount)

Vertical Header: Male Contacts (pins), Tin

Molex 39-28-1063
AMP P/N 794664-6

Right Angle Header: Male Contacts (pins), Tin

Molex 39-30-1060
AMP P/N 794448-1

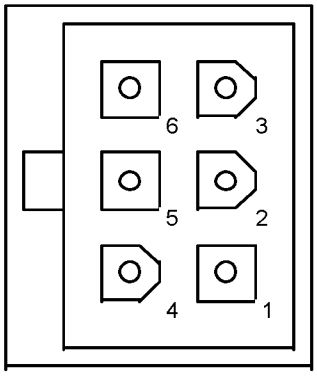
4.3.6.5. Approved Parts – Alternate Form Factors

Select applications may require connector configurations with alternate form factors. Alternate form factor connectors may be used provided they are:

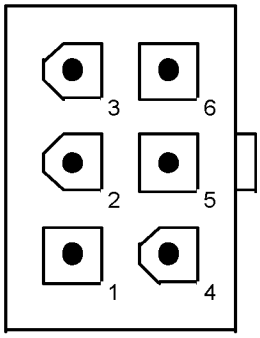
- provided by the Approved Suppliers listed
- part of the Approved Supplier Product Family portfolio
- intermateable with the approved connector part numbers listed
- meet the performance objectives set forth in this specification

Connector Pin-out:

Line 1 - 34 VDC
Line 2 - DC Power Return
Line 3 - N/C
Line 4 - Master Receive
Line 5 - Master Transmit
Line 6 - Communications Common

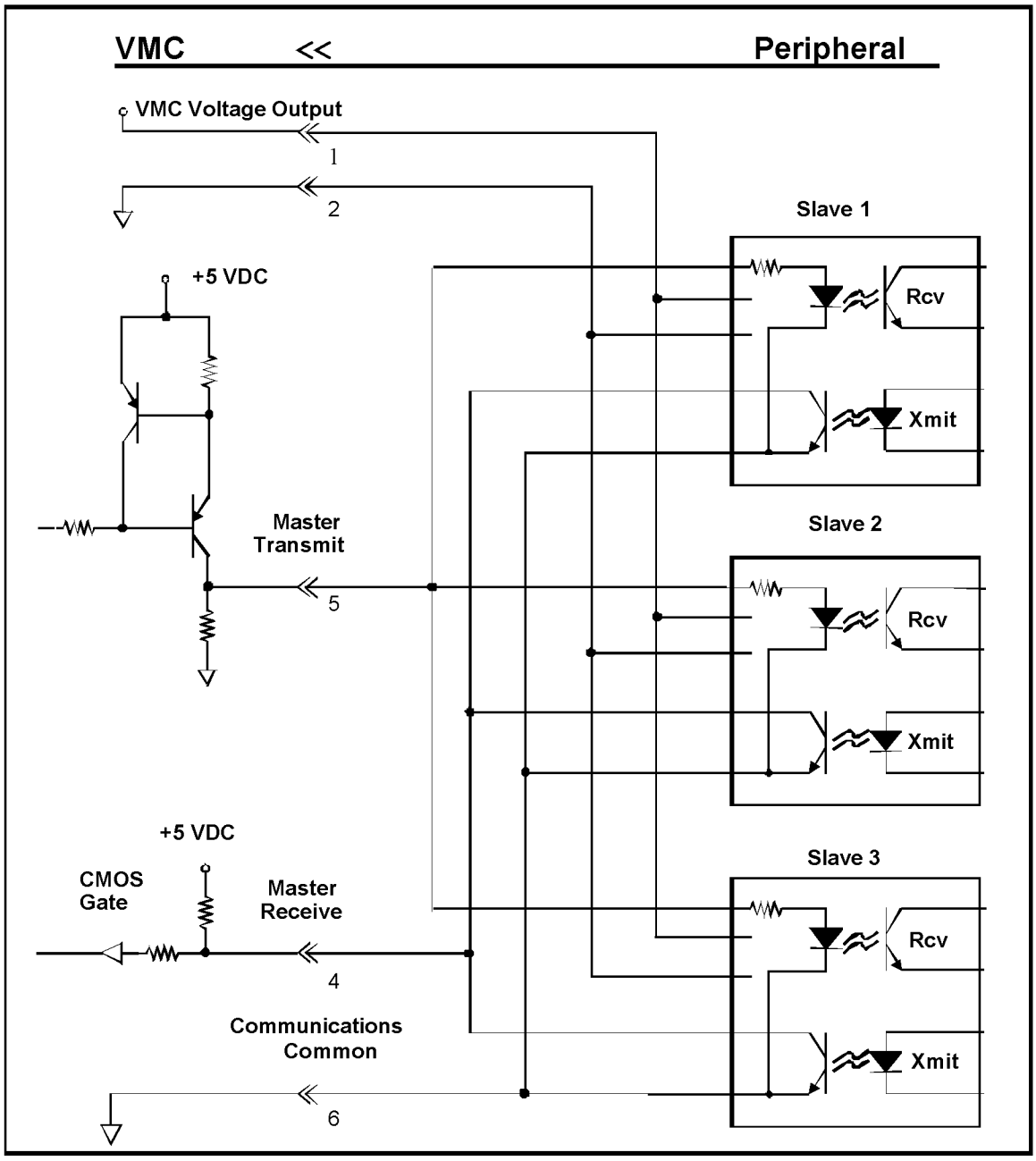


Peripheral Connector
Face View
Receptacle
(Sockets)



VMC / Bus Connector
Face View
Header
(Pins)

4.4 Example Schematic



Section 5

Coin Acceptor/Changer
VMC/Peripheral Communication Specifications

5.1 Introduction

This section defines the communication bytes sent and received by a coin accepting device ("Changer"). As defined in Section 2.3, the changer's address is 00001xxxB (08H).

Unless stated otherwise, all information is assumed to be in a binary format.

There are currently two levels of support defined for the coin mechanism interface, Level 2 and Level 3. The level of coin mechanism operation is sent to the VMC in the response to the STATUS command (defined later in this section). The following paragraphs will define how a VMC should differentiate between each level.

Level 2 Changers

For level 2 changers, VMC operation consists of monitoring inputs from the coin mechanism, accumulating credit, issuing a coin acceptance disable command when appropriate, and issuing appropriate payout commands based on the VMC resident payout algorithms and escrow rules.

Level 3 Changers

For level 3 changers, VMC operation is the same as defined above for level 2, with the addition of the EXPANSION command and its implications (defined later in this section). The VMC has the option of sending the EXPANSION command to the coin mechanism to determine the coin mechanism's manufacturer code, serial number, model/tuning revision, software version, and optional features. Based on the optional feature information the VMC will determine the appropriate operating mode (in other words, modes that both the coin mechanism and the VMC can support), enable any appropriate coin mechanism features by sending an appropriate feature enable command back to the coin mechanism, and enter the proper operating mode. This technique allows all VMCs and peripherals to accommodate existing feature capabilities and provides a means for upgrading Level 3 equipment.

5.2 VMC Commands

Command	Hex Code	Description
RESET	08H	Command for changer to self-reset
SETUP *	09H	Request for changer setup information.
TUBE STATUS	0AH	Request for changer tube status.
POLL	0BH	Request for changer activity status.
COIN TYPE	0CH	Signifies coin types accepted and allowable coin dispensing. This command is followed by setup data. See command format section.
DISPENSE	0DH	Command to dispense a coin type. Followed by coin type to dispense. See command format section.
EXPANSION COMMAND	0FH	Command to allow addition of features and future enhancements. Changers at feature level 2 do not support this command.

NOTE: An EXPANSION command is always followed by a "sub-command." This command allows for feature additions.

* In Version 1.0 & 2.0, **SETUP** was called **STATUS.**

5.3 VMC Command Format

VMC Command	Code	VMC Data
RESET	08H	No data bytes

This command is the vehicle that the VMC should use to tell the changer that it should return to its default operating mode. With the exception of the ACK response, it should abort all communication and disable all acceptance until otherwise instructed by the VMC.

The following initialization sequence is recommended for all new VMCs designed after July, 2000. It should be used after "power up", after issuing the RESET command, or after issuing the Bus Reset (pulling the transmit line "active" for a minimum of 100 mS).

POLL – 0Bh
To obtain "JUST RESET" response

SETUP – 09h
To obtain changer level and configuration information

EXPANSION IDENTIFICATION – 0F 00h (Level 03+ only)
To obtain additional changer information and options

EXPANSION FEATURE ENABLE – 0F 01h (Level 03+ only)
To enable desired options

EXPANSION SEND DIAG STATUS – 0F 05h (Level 03+ & option b1 only)
To request the changer to report its current state of operation

TUBE STATUS – 0Ah (Note 1)
To obtain tube status / change information

COIN TYPE – 0Ch
To enable desired coin acceptance and disable manual coin payout if desired

Note 1 – A minimum 500 msec delay is required between a reset (regardless of type) and the first **TUBE STATUS** command for certain models of the existing MDB coin changer field base.

VMC Command	Code	Changer Response Data
SETUP	09H	23 bytes: Z1 - Z23

Z1 = Changer Feature Level - 1 byte

Indicates the feature level of the changer. This will distinguish the changers feature level to the VMC. Current defined levels:

Level 2: Supports "core" command set. These are: RESET, STATUS, TUBE STATUS, POLL, COIN TYPE, and DISPENSE. (Z1 = 02h)

Level 3: Supports level two and the EXPANSION command addition changer model number, manufacturer code, turning revision, etc. See the details of EXPANSION command later in this document. (Z1=03h)

Z2 - Z3 = Country / Currency Code - 2 bytes

The packed BCD country / currency code of the changer can be sent in two different forms depending on the value of the left most BCD digit.

If the left most digit is a 0, the International Telephone Code is used to indicate the country that the changer is set-up for. For example, the USA code is 00 01H (Z2 = 00 and Z3 = 01).

If the left most digit is a 1, the latest version of the ISO 4217 numeric currency code is used (see Appendix A1). For example, the code for the US dollar is 18 40H (Z2 = 18 and Z3 = 40) and for the Euro is 1978 (Z2 = 19 and Z3 = 78).

All new designs after July, 2000 must use the ISO 4217 numeric currency codes as listed in Appendix A1.

Z4 = Coin Scaling Factor - 1 byte

All accepted coin values must be evenly divisible by this number. For example, this could be set to 05H for the USA nickel.

Z5 = Decimal Places - 1 byte

Indicates the number of decimal places on a credit display. For example, this could be set to 02H in the USA.

Z6 - Z7 = Coin Type Routing - 2 bytes

Indicates what coin types can be routed to the Changer's tubes.

```
b15  b14  b13  b12  b11  b10  b9  b8 | b7  b6  b5  b4  b3  b2  b1  b0
Z6                                                              Z7
```

Bit is set to indicate a coin type can be routed to the tube. Valid coin types are 0 to 15.

Z8 - Z23 = Coin Type Credit - 16 bytes

Indicates the value of coin types 0 to 15. Values must be sent in ascending order. This number is the coin's monetary value divided by the coin scaling factor. Unused coin types are sent as 00H. Unsent coin types are assumed to be zero. It is not necessary to send all coin types. Coin type credits sent as FFH are assumed to be vend tokens. That is, their value is assumed to worth one vend.

The bytes position in the 16 byte string indicates the coin type(s). For example, the first byte sent would indicate the value of coin type 0, the second byte sent would indicate the value of coin type 1, and so on. For example, the USA coin types may be; Coin type 0 = nickel, Coin type 1 = dime, Coin type 2 = quarter, Coin type 3 = dollar.

VMC Command	**Code**	**Changer Response Data**
TUBE STATUS	0AH	18 bytes: Z1 - Z18

Z1 - Z2 = Tube Full Status - 2 bytes

Indicates status of coin tube for coin types 0 to 15.

```
b15  b14  b13  b12  b11  b10  b9  b8 | b7  b6  b5  b4  b3  b2  b1  b0
Z1                                                              Z2
```

A bit is set to indicate a full tube. For example, bit 7 = set would indicate the tube for coin type 7 is full.

Z3 - Z18 = Tube Status - 16 bytes

Indicates the greatest number of coins that the changer "knows" definitely are present in the coin tubes. A bytes position in the 16 byte string indicates the number of coins in a tube for a particular coin type. For example, the first byte sent indicates the number of coins in a tube for coin type 0. Unsent bytes are

assumed to be zero. For tube counts greater than 255, counts should remain at 255.

NOTE: If a changer can detect a tube jam, defective tube sensor, or other malfunction, it will indicate the tube is "bad" by sending a tube full status and a count of zero for the malfunctioning coin type.

VMC Command	Code	Changer Response Data
POLL	0BH	16 bytes: Z1 - Z16

Z1 - Z16 = Changer Activity - 16 bytes

Indicates the changer activity. If there is nothing to report, the changer should send only an ACK. Otherwise, the only valid responses are:

Coins Dispensed Manually:

Z1	Z2
(1yyyxxxx)	(zzzzzzzz)

yyy = The number of coins dispensed.
xxxx = The coin type dispensed (0 to 15)
zzzzzzzz = The number of coins in the tube.

Coins Deposited:

Z1	Z2
(01yyxxxx)	(zzzzzzzz)

yy = Coin routing. 00: CASH BOX
01: TUBES
10: NOT USED
11: REJECT

xxxx = Coin type deposited (0 to 15).

zzzzzzzz = The number of coins in the tube for the coin type accepted.

Status:

(00000001) = Escrow request[1] - An escrow lever activation has been detected.

(00000010) = Changer Payout Busy[2] - The changer is busy activating payout devices.

(00000011) = No Credit[1] - A coin was validated but did not get to the place in the system when credit is given.

(00000100) = Defective Tube Sensor[1] - The changer has detected one of the tube sensors behaving abnormally.

(00000101) = Double Arrival[1] - Two coins were detected too close together to validate either one.

(00000110) = Acceptor Unplugged[2] - The changer has detected that the acceptor has been removed.

(00000111) = Tube Jam[1] - A tube payout attempt has resulted in jammed condition.

(00001000) = ROM checksum error[1] - The changers internal checksum does not match the calculated checksum.

(00001001) = Coin Routing Error[1] - A coin has been validated, but did not follow the intended routing.

(00001010) = Changer Busy[2] - The changer is busy and can not answer a detailed command right now.

(00001011) = Changer was Reset[1] - The changer has detected an Reset condition and has returned to its power-on idle condition.

(00001100) = Coin Jam[1] - A coin(s) has jammed in the acceptance path.

(00001101) = Possible Credited Coin Removal[1] – There has been an attempt to remove a credited coin.

Note:
- changers must have a means to disable this code due to potential older VMC issues.
- virtually all VMCs designed prior to this code's introduction (10/16/02) will not support it.
- It is a vending machine system issue as to what is done when this code is received.

Slug:

(001xxxxx) = xxxxx is the number of slugs since the last activity.

NOTES: The Changer may send several of one type activity, up to 16 bytes total. This will permit zeroing counters such as slug, inventory, and status.

1 Sent once each occurrence.
2 Sent once each POLL

File Transport Layer POLLed responses:

Note that all FTL responses are defined in Section 2.6. For the coin changer, the source address will always be the changer (08H) as defined in Section 2.3.

Z1

1B	REQ TO RCV	The coin changer is requesting to receive data from a device or VMC. Z2 = Destination address of response Z3 = Source address of response (08H) Z4 = File ID Z5 = Maximum length Z6 = Control
1C	RETRY/DENY	The coin changer is requesting a device or VMC to retry or deny the last FTL command. Z2 = Destination address of response Z3 = Source address of response (08H) Z4 = Retry delay
1D	SEND BLOCK	The coin changer is sending a block of data (maximum of 31 bytes) to a device or VMC. Z2 = Destination address of data Z3 = Block # Z4-Z34 = Data (maximum of 31 bytes)
1E	OK TO SEND	The coin changer is indicating that it is OK for a device or VMC to send it data. Z2 = Destination address of response Z3 = Source address of response (08H)
1F	REQ TO SEND	The coin changer is requesting to send data to a device or VMC. Z2 = Destination address of response Z3 = Source address of response (08H) Z4 = File ID Z5 = Maximum length Z6 = Control

VMC Command	Code	VMC Data

COIN TYPE 0CH 4 bytes: Y1 - Y4

Y1 - Y2 = Coin Enable - 2 bytes

```
b15  b14  b13  b12  b11  b10  b9  b8 | b7  b6  b5  b4  b3  b2  b1  b0
Y1                                                                Y2
```

A bit is set to indicate a coin type is accepted. For example, bit 6 is set to indicate coin type 6, bit 15 is set to indicate coin type 15, and so on. To disable the changer, disable all coin types by sending a data block containing 0000H. All coins are automatically disabled upon reset.

Y3 - Y4 = Manual Dispense Enable - 2 bytes

```
b15  b14  b13  b12  b11  b10  b9  b8 | b7  b6  b5  b4  b3  b2  b1  b0
Y3                                                                Y4
```

A bit is set to indicate dispense enable. For example, bit 2 is set to enable dispensing of coin type 2. This command enables/disables manual dispensing using optional inventory switches. All manual dispensing switches are automatically enabled upon reset.

VMC Command	Code	VMC Data
DISPENSE	0DH	1 byte: Y1

b7	b6	b5	b4	b3	b2	b1	b0
				Y1			

Bits b3, b2, b1, b0 indicate coin type to be dispensed. Valid codes are 0H to FH to indicate coin types 0 to 15.

Bits b7, b6, b5, b4 indicate the number of coins to be dispensed.

NOTE 1: If two coin types have the same value, the highest coin type should be paid out first.

NOTE 2: There is no defined limit on how long the actual dispense takes since the command allows for up to 15 coins to be paid out. The payout cycle begins when the changer ACKs the VMC's DISPENSE (0DH) command. This cycle typically lasts a minimum of 100 mS and ends when the changer stops dispensing the desired number of coins. VMCs should monitor the Changer Payout Busy response to the POLL command to determine when the entire payout cycle is completed.

However, it must be noted that other than ACKing the DISPENSE (0DH) command, the changer does not have to respond during the payout cycle provided the payout cycle is less than the changer's non-response time and the changer starts responding again prior to the end of the non-response time. Thus, it is acceptable for the changer to never report Changer Payout Busy, but simply start ACKing the POLL commands upon completion of a payout cycle provided the non-response time has not been exceeded.

LEVEL THREE CAPABILITIES - EXPANSION COMMAND

The following describes the currently defined expansion commands.

Sub-command 00H is used for a changer that has the capability of reporting model number, serial number, and so on.

VMC Command	Code	Sub-Command	Changer Response Data
EXPANSION COMMAND	0FH	00H IDENTIFICATION	33 bytes: Z1 - Z33

Z1 - Z3 = Manufacturer Code - 3 bytes
Identification code for the equipment supplier. Sent as ASCII characters. Currently defined codes are listed in the **EVA** document entitled "***European Vending Association Data Transfer Standard***" (**EVA-DTS**), the Audit Data Lists section, sub-section 2, "Manufacturer Codes".

Z4 - Z15 = Serial Number - 12 bytes

Factory assigned serial number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable.

Z16 - Z27 = Model #/Tuning Revision - 12 bytes

Manufacturer assigned model number and tuning number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable. Each manufacturer should include information concerning the changer tuning revision.

Z28 - Z29 = Software Version - 2 bytes

Current software version. Must be sent in packed BCD.

Z30 - Z33 = Optional Features - 4 bytes

Each of the 32 bits indicate an optional features availability. If the bit is set the feature is available. Bits should be sent in descending order, i.e. bit 31 is sent first and bit 0 is sent last. Currently defined options are:

b0 - Alternative Payout method. This method allows changer designs that determine change payout. That is, the payout algorithm may reside in the changer instead of the VMC.

b1 - Extended Diagnostic command supported. This command allows the VMC to request diagnostic status of the coin changer.

b2 - Controlled Manual Fill and Payout commands supported. These commands allows the VMC to request the number of coin inserted or dispensed while the changer was in a controlled manual fill or payback mode.

b3 - File Transport Layer (FTL) supported as defined in Section 2.6.

b4 - b31 Available for future use

VMC Command	Code	Sub-Command	VMC Data
EXPANSION COMMAND	0FH	01H FEATURE ENABLE	4 bytes: Y1 - Y4

This command is used to enable each of the optional features defined in Z30-Z33 above. To enable a feature a bit is set to one. **All optional features are disabled after reset.**

VMC Command	Code	Sub-command	VMC Data	Changer Response
EXPANSION COMMAND (Alternative Payout)	0FH	02H PAYOUT	Y1	None

Y1 = Value of coins to be paid out - 1 byte

This value is expressed as the number of coin scaling factors that would sum to the value. For example, in a USA system using a scaling factor of 05, if the change to be paid out is 75 cents, then Y1 will equal fifteen. That is, the sum of fifteen nickels equal 75 cents. The coin changer will determine which actual denominations of coins will be paid out. In the 75 cent example, the coins may be 3 quarters; or, 7 dimes & 1 nickel; or, 2 quarters & 2 dimes & 1 nickel, etc.

VMC Command	Code	Sub-command	VMC Data	Changer Response
EXPANSION COMMAND (Alternative Payout)	0FH	03H PAYOUT STATUS	None	16 bytes: Z1-Z16

Z1 - Z16 = Number of each coin type paid out - 16 bytes

This is the changer's response to the last VMC Alternative PAYOUT command (0FH-02H). Bytes are sent in ascending order of coin types. A bytes position in the string indicates the coin type. That is, byte one is the number of coins for coin type 1, byte two is the number of coins for coin type two, and so on. Unsent bytes are assumed to be zero.

The changer clears payout data after an ACK response from the VMC.

The VMC should compare the value of the coins paid out to the (0FH-02H) Alternative PAYOUT command's Y1 .

NOTES: 1) If the changer's payout is busy it will respond to the Alternative PAYOUT STATUS command with an ACK only.

2) If no coins have been paid out, at least one zero valued data byte must be sent.

3) There is no defined limit on how long the actual payout takes. See Note 2 under the DISPENSE (0DH) command.

VMC Command	Code	Sub-command	Changer Response Data
EXPANSION COMMAND (Alternative Payout)	0FH	04H PAYOUT VALUE POLL	1 byte: Z1

Z1 = Changer Payout Activity - 1 byte

An interval value (scaled) which indicates the amount of paid out change since the previous PAYOUT VALUE POLL (or between the initial Alternative PAYOUT command (0FH-02H) and the first PAYOUT VALUE POLL).

An 00H response indicates no coins were paid out since the previous PAYOUT VALUE POLL (or the initial Alternative PAYOUT command (0FH-02H)).

An ACK only indicates that the change payout is finished. This should be followed by the PAYOUT STATUS command (0FH-03H) to obtain the complete payout data.

NOTE: The initial intent of this command is to determine the amount of change paid out so that the credit display can be decremented as coins are dispensed.

VMC Command	Code	Sub-Command	Changer Response Data
EXPANSION	**0FH**	**05H**	**16 bytes: Z1-Z16**
COMMAND	**SEND DIAGNOSTIC STATUS**		

Send Current Diagnostic Status - This command requests the changer to report its current state of operation. The VMC should periodically transmit the command approximately every 1 to 10 seconds.

.

Z1-Z2 **= Current changer diagnostic information**

The changer reports its current state of operation in a 2 byte code. Z1 is the main code and Z2 is the sub-code. The code is reported as long as the condition exists and stops being reported as soon as the condition does not exist. Multiple 2 byte codes may be sent in response to a single command which could result in a maximum of eight 2 byte codes (16 bytes total).

The following tables identify the currently defined extended diagnostic codes:

Z1 / Z2	Status	Cause(s) of Status / Error
01 / 00	Powering up	Changer powering up / initialization
02 / 00	Powering down	Changer powering down
03 / 00	OK	Changer fully operational and ready to accept coins
04 / 00	Keypad shifted	MODE key pressed and held so that LED flashes indicating keypad in shifted state. Reverts to normal mode if no key pressed for 15 seconds
05 / 10	Manual Fill / Payout active	Manual Fill or Manual Payout mode of operation in progress (under control of the changer). This response must be reported at least once to allow the VMC to request a manual fill or manual payout report.
05 / 20	New Inventory Information Available	Changer not in Manual inventory mode, but new inventory information available.
06 / 00	Inhibited by VMC	All coin acceptance inhibited at request of VMC, possibly due to product dispenser jams, completely sold out, etc.
10 / Z2	General changer error	Z2 defined as: **00** Non specific error. **01** Check sum error #1. A check sum error over a particular data range of configuration field detected. **02** Check sum error #2. A check sum error over a secondary data range or configuration field detected. **03** Low line voltage detected. The changer has disabled acceptance or payout due to a low voltage condition.

Z1 / Z2	Status	Cause(s) of Status / Error
11 / Z2	Discriminator module error	Z2 defined as: **00** Non specific discriminator error. **10** Flight deck open. **11** Escrow Return stuck open. **30** Coin jam in sensor. **41** Discrimination below specified standard. **50** Validation sensor A out of range. The acceptor detects a problem with sensor A. **51** Validation sensor B out of range. The acceptor detects a problem with sensor B. **52** Validation sensor C out of range. The acceptor detects a problem with sensor C. **53** Operating temperature exceeded. The acceptor detects the ambient temperature has exceeded the changer's operating range, thus possibly affecting the acceptance rate. **54** Sizing optics failure. The acceptor detects an error in the sizing optics.
12 / Z2	Accept gate module error	Z2 defined as: **00** Non specific accept gate error. **30** Coins entered gate, but did not exit. **31** Accept gate alarm active. **40** Accept gate open, but no coin detected. **50** Post gate sensor covered before gate opened.
13 / Z2	Separator module error	Z2 defined as: **00** Non specific separator error **10** Sort sensor error. The acceptor detects an error in the sorting sensor.
14 / Z2	Dispenser module error	Z2 defined as: **00** Non specific dispenser error.
15 / Z2	Coin Cassette / tube module error	Z2 defined as: **00** Non specific cassette error. **02** Cassette removed. **03** Cash box sensor error. The changer detects an error in a cash box sensor. **04** Sunlight on tube sensors. The changer detects too much ambient light on one or more of the tube sensors.

Diagnostic Status EVA-DTS Correlation

The Extended Diagnostic information reported may be used by the vending machine controller as desired (i.e., service mode displays); however, EVA-DTS **data elements could also be used for reporting to a host system. Examples are:**

- o **Via a translation of the Z1/Z2 code to one of the** Fault Lists **as described in Section 10 of the** EVA-DTS.

- o **Via the EA201 Event Identification element with the format** EAxxyy **where xx = Z1 and yy = Z2.**

- o **Via a customer / manufacture specific coding scheme using the** MA5xx **fields.**

VMC Command	Code	Sub-Command	Changer Response Data
EXPANSION	**0FH**	**06H**	**16 bytes Z1-Z16**
COMMAND	**SEND CONTROLLED MANUAL FILL REPORT**		

Send Controlled Manual Fill Report - **This command requests the changer to report the number of coins inserted during a changer controlled manual fill (controlled bulk fill) mode. While in this mode, the changer must not report coins inserted in response to the** POLL **command.**

Z1-Z16 **= number of controlled manual mode filled coins (by coin type)**

A single byte is reported for each coin type, 0 to 15. For example, Z1 = number of coins of coin type 0 added in a controlled manual fill mode. Any amount above 255 will be reported as 255, i.e. it will reach a maximum limit.

Only coin types *supported* are required to be reported. Counts for unsent coins types will be assumed to be unchanged.

Notes: **After power on, changer reset, closing of the machine door, or a change in controlled manual fill status in the changer (changer indicated it was in controlled manual fill mode via CM0510 then changed to any other state) the machine should request the controlled manual coin fill data from the changer using the above command.**

See EVA-DTS correlation at end of SEND CONTROLLED MANUAL PAYOUT REPORT **(0F-07H) command.**

VMC Command	Code	Sub-Command	Changer Response Data
EXPANSION	**0FH**	**07H**	**16 bytes Z1-Z16**
COMMAND	**SEND CONTROLLED MANUAL PAYOUT REPORT**		

Send Controlled Manual Payout Report - **This command requests the changer to report the number of coins dispensed during a changer controlled manual payout (controlled bulk dispense) mode. Note that this does not include the coins dispensed via the individual dispense switches.**

If the new Controlled Manual Fill / Payout command is implemented in the coin mech and **enabled by the VMC (0Fh, 01h, bit 2 of Y1 to Y4), while in a controlled manual payout (dispense) mode, the changer** must not **report the coins paid out in response to the** POLL **command. Conversely, if the changer does not support the new command or the VMC does not enable it, the changer** should **report the coins paid out in response to the** POLL **command.**

Z1-Z16 **= number of controlled manual mode dispensed coins (by coin type)**

> **A single byte is reported for each coin type 0 to 15. For example, Z1 = number of coins of coin type 0 dispensed in a controlled manual payout mode. Any amount above 255 will be reported as 255, i.e. it will reach a maximum limit.**
>
> **Only coin types supported are required to be reported. Counts for unsent coin types will be assumed to be unchanged.**

Note: **After power on, changer reset, closing of the machine door, or a change in controlled manual payout status in the changer (changer indicated it was in controlled manual payout mode via CM0510 then changed to any other state) the machine should request the controlled manual coin payout data from the changer using the above command.**

Controlled Manual Fill / Payout EVA-DTS Correlation

The controlled manual fill and payout coin information may be used by the vending machine controller as desired (i.e., service mode displays); however, EVA-DTS **data elements could be used for reporting to a host system. Examples are:**

	CA3XX	CA4XX	CA1704	CA1705
Controlled Manual Fill	0F06	n/a	0F06	n/a
VMC Tube Fill	VMC	n/a	VMC	n/a
Controlled Manual Payout	n/a	0F07*	n/a	0F07*
VMC Coin Payout	n/a	VMC	n/a	VMC
Manual Dispense Switches	n/a	0B	n/a	0B

***If extended** 0F06 **&** 0F07 **commands are implemented.**
If extended 0F06 **&** 0F07 **commands are not implemented in the coin mech or not enabled by the VMC, the coin mech will respond to the** POLL **command with the controlled manual payout coins.**

With the above, the CA3XX **&** CA4XX **fields can continue to be the primary fields for cash audit and the** CA1704 **&** CA1705 **fields can be used for indicating controlled manually filled / dispensed coins.**

Coin Tube Audit Fields

As a reference, below are the agreed CA17XX **data elements that provide detailed coin tube count information and controlled-manual coin tube insertion / dispense information. These were approved by the** EVA - DTS **Technical Sub Committee on January 27, 1997.**

Block Identifier Reference	Data Contents	Charac-teristic	Length Min	Length Max	Element
CA17	Coin Type Number (per MDB coin type)	N	01	03	CA1701
	Value of Coin	N	01	08	CA1702
	Number of Coins in Tube	N	01	08	CA1703
	Number of Coins Inserted during Controlled-Manual Fill	N	01	08	CA1704
	Number of Coins Dispensed during Controlled-Manual Payout	N	01	08	CA1705

Definitions:

CA1701 **The coin type number as referred to in the MDB Interface Specification. If not an MDB system, the number represents the coin's position in the coin set starting with the lowest value coin accepted. Note if two or more vintage of the same coin is accepted, the oldest one is first.**

For example, the Canadian coin types may be:

0 Old Nickel	**3 Quarter**
1 New Nickel	**4 $1 Dollar**
2 Dime	**5 $2 Dollar**

CA1702 **The cash value of the coin (units base).**

For example, the Canadian coin types would be:

Nickel	**5**	**$1 Dollar**	**100**
Dime	**10**	**$2 Dollar**	**200**
Quarter	**25**		

CA1703 **The number of coins in the coin tube (or tubes if multiple tubes per coin) that are reported by the coin mech during normal vending operations. Note that this is the "best known tube count" and may be inaccurate if coins were manually added or removed by hand.**

CA1704 **The number of coins inserted while the changer was in a Controlled manual fill mode. Controlled manual fill indicates that the coins are being inserted under the control of the coin mech or VMC. Coins are not being loaded by hand through the tops of the tubes.**

CA1705 **The number of coins dispensed while the changer was in a controlled manual payout mode. Controlled manual payout indicates that the coins are being dispensed under the control of the coin mech or VMC. Coins are not being removed by hand by "dumping" the tubes.**

VMC Command	Code	Sub-command	VMC Data	Changer Response
EXPANSION COMMAND	0FH	FAH **FTL** REQ TO RCV	Y1-Y5	Z1 - Zn (immediate or POLLed)

The VMC is requesting to receiv e data from the changer whose destination address will always be (08H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (08H)
Y2 = Source address of command
Y3 = File ID
Y4 = Maximum length
Y5 = Control

Z1 = 1DH which indicates SEND BLOCK
Z2 = Destination address of data
Z3 = Block #
Z4 - Z34 = Data (maximum of 31 bytes)

or

Z1 = 1CH which indicates RETRY / DENY
Z2 = Destination address of response
Z3 = Source address of response (08H)
Z4 = Retry delay

VMC Command	Code	Sub-command	VMC Data	Changer Response
EXPANSION COMMAND	0FH	FBH **FTL** RETRY / DENY	Y1-Y3	None

The VMC is retrying, denying, or aborting a data transfer to/from the changer whose destinat ion address will always be (08H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (08H)
Y2 = Source address of command
Y3 = Retry delay

VMC Command	Code	Sub-command	VMC Data	Changer Response
EXPANSION COMMAND	0FH	FCH **FTL** SEND BLOCK	Y1-Y33	None

The VMC is sending data to the changer whose destination address will always be (08H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command & data (08H)
Y2 = Block #
Y3 - Y33 = Data (maximum of 31 bytes)

VMC Command	Code	Sub-command	VMC Data	Changer Response
EXPANSION COMMAND	0FH	FDH **FTL** OK TO SEND	Y1-Y2	Z1-Z34 (immediate or POLLed)

The VMC is indicating that it is OK for the changer to transfer data. The destination address will always be the changer (08H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (08H)
Y2 = Source address of command

Z1 = 1DH which indicates SEND BLOCK
Z2 = Destination address of data
Z3 = Source address of data
Z4 - Z34 = Data (maximum of 31 bytes)

VMC Command	Code	Sub-command	VMC Data	Changer Response
EXPANSION COMMAND	0FH	FEH **FTL** REQ TO SEND	Y1-Y5	Z1 (immediate or POLLed)

The VMC is requesting to send data to the changer whose destination address will always be (08H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (08H)
Y2 = Source address of command
Y3 = File ID
Y4 = Maximum length
Y5 = Control

Z1 = 1EH which indicates OK TO SEND
Z2 = Destination address of response
Z3 = Source address of response (08H)

or

Z1 = 1CH which indicates RETRY / DENY
Z2 = Destination address of response
Z3 = Source address of response (08H)
Z4 = Retry delay

VMC Command	Code	Sub-command	VMC Data	Changer Response
EXPANSION COMMAND	0FH	FFH DIAGNOSTICS	Y1-Yn	Z1-Zn

Y1 - Yn = Device manufacturer specific instruction for implementing various manufacturing or test modes. Y1 - Yn implies that any number of bytes can be used for the VMC data to the peripheral.

Z1 - Zn = Device manufacturer specific responses after receiving manufacturing or test instructions. Z1 - Zn implies that any number of bytes can be used for the changer response data from the peripheral.

5.4 Changer Non-Response Time

The maximum non-response time for the changer is two seconds.

5.5 Changer Power Requirements

The current draw for any changer must fall within the following limits. All measurements are at the minimum VMC Voltage Output.

Idle mode = 200 mA. (max.) continuous

Coin acceptance = 1.8 A. (max.) for up to 2 seconds

(For coin changers using solenoid based payout mechanisms - typical of 3 tube changers sold in the US market. Vending machines sold into the US market are required to supply this power.)

1.0 A. (max.) for up to 2 seconds

(For coin changers using motorized payout mechanisms - typical of 4 tube changers.)

Coin payout = 3.6 A. (max.) for 100 mS. with 400 mS. idle current between pulses during the coin payout cycle.

(For coin changers using solenoid based payout mechanisms - typical of 3 tube changers sold in the US market. Vending machines sold into the US market are required to supply this power.)

1.8 A. (max.) during the coin payout cycle.

(For coin changers using motorized payout mechanisms - typical of 4 tube changers.

See Note 2 under the DISPENSE (0DH) command for further information on the coin payout cycles.)

Note: If both peripherals are supported, vending machines should be able to provide sufficient power to simultaneously supply the above power requirements for both the coin changer **Coin Acceptance** and bill validator **Bill Transport** as specified in Section 6.5.

Section 6

Bill Validator
VMC/Peripheral Communication Specifications

6.1 Introduction

This section defines the communication bytes sent and received between a Bill Validator and the VMC. As defined in Section 2.3, the bill validator's address is 00110xxxB (30H).

Unless stated otherwise, all information is assumed to be in a binary format.

There are currently two levels of support defined for the bill validator interface, Level 1 and Level 2. The level of bill validator operation is sent to the VMC in the response to the STATUS command (defined later in this section). The following paragraphs will define how a VMC should differentiate between each level.

Level 1 Bill Validators

Level 1 bill validators support all standard functions, but do not support any optional features.

Level 2 Bill Validators

Level 2 bill validators support all standard functions plus various optional features as defined in Section 6.3 under the Expansion command 37-02H. Based on the optional feature information the VMC will determine the appropriate operating mode (in other words, modes that both the bill validator and the VMC can support), enable any appropriate features by sending an appropriate feature enable command back to the bill validator, and enter the proper operating mode. This technique allows all VMCs and peripherals to accommodate existing feature capabilities and provides a means for upgrading Level 2 equipment.

6.2 VMC Commands

Command	Hex Code	Description
RESET	30H	Command for bill validator to self-reset.
SETUP *	31H	Request for bill validator setup information.
SECURITY	32H	Sets Validator Security Mode
POLL	33H	Request for Bill Validator activity Status.
BILL TYPE	34H	Indicates Bill Type enable or disable. Command is followed by set-up data. See command format.
ESCROW	35H	Sent by VMC to indicate action for a bill in escrow.
STACKER	36H	Indicates stacker full and the number of bills.
EXPANSION COMMAND	37H	Command to allow addition of features and future enhancements. Level 1 and above bill validators must support this command.

NOTE: The expansion command is always followed by a sub-command.

* In Version 1.0 & 2.0, **SETUP** was called **STATUS.**

6.3 VMC Command Format

VMC Command	Code	VMC Data
RESET	30H	No data bytes

This command is the vehicle that the VMC should use to tell the validator that it should return to its default operating mode. It should reject any bills in the validation process, return any bills in the escrow position, and disable all other activity until otherwise instructed by the VMC.

The following initialization sequence is recommended for all new VMCs designed after July, 2000. It should be used after "power up", after issuing the RESET command, or after issuing the Bus Reset (pulling the transmit line "active" for a minimum of 100 mS).

POLL – 33h
To obtain "JUST RESET" response

SETUP – 31h
To obtain bill validator level and configuration information

EXPANSION IDENTIFICATION – 37 00h (Level 01+)
To obtain additional bill validator information

EXPANSION IDENTIFICATION w/ OPTION BITS – 37 02h (Level 02+ only)
To obtain additional bill validator information and options

EXPANSION FEATURE ENABLE – 37 01h (Level 02+ only)
To enable desired options

STACKER – 36h
To obtain stacker status and number of bills

BILL TYPE – 34h
To enable desired bill acceptance and desired bill escrow capability

VMC Command	Code	Validator Response Data
SETUP	31H	27 bytes: Z1 - Z27

Z1 = Bill Validator Feature Level - 1 byte
Indicates current feature level of the bill validator. Currently defined levels are:

Level **1** - does not support option bits (Z1 = 01h)
Level **2** - supports option bits (Z1 = 02h)

Z2 - Z3 = Country / Currency Code - 2 bytes
The packed BCD country / currency code of the bill validator can be sent in two different forms depending on the value of the left most BCD digit.

If the left most digit is a 0, the International Telephone Code is used to indicate the country that the validator is set-up for. For example, the USA code is 00 01H (Z2 = 00 and Z3 = 01).

If the left most digit is a 1, the latest version of the ISO 4217 numeric currency code is used (see Appendix A1). For example, the code for the US dollar is 18 40H (Z2 = 18 and Z3 = 40) and for the Euro is 1978 (Z2 = 19 and Z3 = 78).

All new designs after July, 2000 must use the ISO 4217 numeric currency codes as listed in Appendix A1.

Z4 - Z5 = Bill Scaling Factor - 2 bytes
All accepted bill values must be evenly divisible by this number. For example, this could be set to 0064H for the USA.

Z6 = Decimal Places - 1 byte
Indicates the number of decimal places on a credit display. For example, this could be set to 02H for the USA.

Z7 - Z8 = Stacker Capacity - 2 bytes
Indicates the number of bills that the stacker will hold. For example, 400 bill capacity = 0190H.

Z9 - Z10 = Bill Security Levels - 2 bytes
Indicates the security level for bill types 0 to 15. Since not all validators support multiple security levels, validators that do not have this feature must report a "high" security level.

Z11 = Escrow/No Escrow - 1 byte
Indicates the escrow capability of the bill validator. If Z11 = 00H, the bill validator does not have escrow capability. If Z11 = FFH, the bill
validator has escrow capability.

Z12 - Z27 = Bill Type Credit - 16 bytes
Indicates the value of the bill types 0 to 15. Values must be sent in ascending order. This number is the bill's monetary value divided by the bill scaling factor. Unused bill types are sent as 00H. Unsent bill types are assumed to be zero. FFH bills are assumed to be vend tokens.

VMC Command	Code	VMC Data
SECURITY	32H	2 Bytes: Y1 - Y2

Y1 - Y2 = Bill Type(s) - 2 bytes

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Y1								Y2							

A bit is set to indicate the type of bill(s) which are set to a "high" security level. Note that validators that do not support dual security levels should report a "high" security level in the response bytes Z9-Z10 to the STATUS (31H) command.

VMC Command	Code	Validator Response Data
POLL	33H	16 bytes: Z1 - Z16

Z1 - Z16 = Bill Validator Activity - 16 bytes

Indicates the validator activity, for example, the type and number of bills accepted, and stacker position. If there is nothing to report, the validator should send only an ACK. Otherwise, the only valid responses are:

Bills Accepted:

Indicates the type and number of bills accepted and stacker status.

Z1	
(1yyyxxxx)	**NOTE:** These responses should be used to add or subtract credit.

yyy = Bill Routing;
- 000: BILL STACKED
- 001: ESCROW POSITION
- 010: BILL RETURNED
- 011: NOT USED
- 100: DISABLED BILL REJECTED

xxxx = Bill Type (0 to 15)

Status:

(00000001) = Defective Motor[3] - One of the motors has failed to perform its expected assignment.

(00000010) = Sensor Problem[3] - One of the sensors has failed to provide its response.

(00000011) = Validator Busy[2] - The validator is busy and can not answer a detailed command right now.

(00000100) = ROM Checksum Error[3] - The validators internal checksum does not match the calculated checksum.

(00000101) = Validator Jammed[3] - A bill(s) has jammed in the acceptance path.

(00000110) = Validator was reset[1] - The validator has been reset since the last POLL.

(00000111) = Bill Removed[1] - A bill in the escrow position has been removed by an unknown means. A BILL RETURNED message should also be sent.

(00001000) = Cash Box out of position[3] - The validator has detected the cash box to be open or removed.

(00001001) = Unit Disabled[2] - The validator has been disabled, by the VMC or because of internal conditions.

(00001010) = Invalid Escrow request[1] - An ESCROW command was requested for a bill not in the escrow position.

(00001011) = Bill Rejected[1] - A bill was detected, but rejected because it could not be identified.

(00001100) = Possible Credited Bill Removal[1] – There has been an attempt to remove a credited (stacked) bill.

Note:
- validators must have a means to disable this code due to potential older VMC issues.
- virtually all VMCs designed prior to this code's introduction (10/16/02) will not support it.
- It is a vending machine system issue as to what is done when this code is received.

(010xxxxx) = Number of attempts to input a bill while validator is disabled.[1]

NOTE: The validator may send several of one type activity up to 16 bytes total.

1 Sent once each occurrence.
2 Sent once each POLL
3 Sent once each occurrence. The unit is then disabled until the condition is removed. Validator will respond with unit disabled until repaired or replaced.

File Transport Layer POLLed responses:

Note that all FTL responses are defined in Section 2.6. For the bill validator, the source address will always be the validator (30H) as defined in Section 2.3.

Z1

1B	REQ TO RCV	The bill validator is requesting to receive data from a device or VMC. Z2 = Destination address of response Z3 = Source address of response (30H) Z4 = File ID Z5 = Maximum length Z6 = Control
1C	RETRY/DENY	The bill validator is requesting a device or VMC to retry or deny the last FTL command. Z2 = Destination address of response Z3 = Source address of response (30H) Z4 = Retry delay
1D	SEND BLOCK	The bill validator is sending a block of data (maximum of 31 bytes) to a device or VMC. Z2 = Destination address of data Z3 = Block # Z4-Z34 = Data (maximum of 31 bytes)
1E	OK TO SEND	The bill validator is indicating that it is OK for the device or VMC to send it data. Z2 = Destination address of response Z3 = Source address of response (30H)
1F	REQ TO SEND	The bill validator is requesting to send data to a device or VMC. Z2 = Destination address of response Z3 = Source address of response (30H) Z4 = File ID Z5 = Maximum length Z6 = Control

VMC Command	Code	VMC Data
BILL TYPE	34H	4 bytes: Y1 - Y4

Y1 - Y2 = Bill Enable - 2 bytes

Indicates what type of bills are accepted.

```
b15  b14  b13  b12  b11  b10  b9  b8 | b7  b6  b5  b4  b3  b2  b1  b0
Y1                                                              Y2
```

Bill types are 0 to 15. A bit is set to indicate acceptance of bill type.

NOTE: Sending 0000H disables the bill validator.

Y3 - Y4 = Bill Escrow Enable:

```
b15  b14  b13  b12  b11  b10  b9  b8 | b7  b6  b5  b4  b3  b2  b1  b0
Y3                                                              Y4
```

Bill types are 0 to 15. A bit is set to indicate enable of escrow for a bill type.

NOTE: On power-up or reset all bill acceptance and escrow are disabled.

VMC Command	Code	VMC Data
ESCROW	35H	1 byte: Y1

Y1 = Escrow status - 1 byte

If Y1 = 0;	Return bill in the escrow position.
If Y1 = xxxxxxx1;	Stack the bill ("x" indicates don't care)

NOTE: After an ESCROW command the bill validator should respond to a POLL command with the BILL STACKED, BILL RETURNED, or INVALID ESCROW message within 30 seconds. If a bill becomes jammed in a position where the customer may be able to retrieve it, the bill validator should send a BILL RETURNED message.

VMC Command	Code	Validator Response Data
STACKER	36H	2 bytes: Z1 - Z2

Indicates stacker full condition and the number of bills in the stacker.

Z1 Z2

(Fxxxxxxx) (xxxxxxxx)
F = 1 if stacker is full, 0 if not.

xxxxxxxxxxxxxxx = The number of bills in the stacker.

LEVEL ONE and TWO+ CAPABILITIES - EXPANSION COMMAND

In order to allow existing VMCs to operate with original Level 1 or new Level 2 bill validators, a separate identification sub-command has been introduced to handle the additional 4 bytes of Option Bit information.

The original sub-command 00H is used for obtaining Z1 to Z29 identification information from bill validators. This information includes the model number, serial number, software version, etc, but **not the option bits**. Note that if a Level 2+ bill validator is sent the 00H sub-command, it must **not** report the Z30 to Z33 option bytes.

Sub-command 01H is used for Level 2+ bill validators to enable option bits reported in the expansion command 02H sub-command below.

The new sub-command 02H is used for obtaining Z1 to Z33 identification information from Level 2+ bill validators. This information includes the model number, serial number, software version, etc, and the **option bits (Z30-Z33).**

VMC Command	Code	Sub-Command	Validator Response Data
EXPANSION	37H	00H	29 bytes: Z1 - Z29
COMMAND	LEVEL 1 IDENTIFICATION WITHOUT OPTION BITS		

Z1 - Z3 = Manufacturer Code - 3 bytes
Identification code for the equipment supplier. Sent as ASCII characters. Currently defined codes are listed in the **EVA** document entitled " ***European Vending Association Data Transfer Standard***' (**EVA-DTS**), the Audit Data Lists section, sub-section 2, "Manufacturer Codes".

Z4 - Z15 = Serial Number - 12 bytes
Factory assigned serial number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable.

Z16 - Z27 = Model #/Tuning Revision - 12 bytes
Manufacturer assigned model number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable.

Z28 - Z29 = Software Version - 2 bytes
Current software version. Must be sent in packed BCD.

VMC Command	Code	Sub-Command	VMC Data
EXPANSION COMMAND	37H	01H LEVEL 2+ FEATURE ENABLE	4 bytes: Y1 - Y4

This command is used to enable each of the Level 2+ optional features defined in the Level 2+ Identification response bytes Z30-Z33 below. To enable a feature a bit is set to one. **All optional features are disabled after reset.**

VMC Command	Code	Sub-Command	Validator Response Data
EXPANSION COMMAND	37H	02H LEVEL 2+ IDENTIFICATION WITH OPTION BITS	33 bytes: Z1 – Z33

Z1 - Z3 = Manufacturer Code - 3 bytes
Identification code for the equipment supplier. Sent as ASCII characters. Currently defined codes are listed in the **EVA** document entitled " ***European Vending Association Data Transfer Standard***' (**EVA-DTS**), the Audit Data Lists section, sub-section 2, "Manufacturer Codes".

Z4 - Z15 = Serial Number - 12 bytes
Factory assigned serial number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable.

Z16 - Z27 = Model #/Tuning Revision - 12 bytes
Manufacturer assigned model number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable.

Z28 - Z29 = Software Version - 2 bytes
Current software version. Must be sent in packed BCD.

Z30 - Z33 = Optional Features - 4 bytes
Each of the 32 bits indicate an optional features availability. If the bit is set the feature is available. Bits should be sent in descending

order, i.e. bit 31 is sent first and bit 0 is sent last. Currently defined options are:

b0 - File Transport Layer (FTL) supported as defined in Section 2.6.
b1 - b31 Available for future use

VMC Command	Code	Sub-command	VMC Data	Validator Response
EXPANSION COMMAND	37H	FAH **FTL** REQ TO RCV	Y1-Y5	Z1 (immediate or POLLed)

The VMC is requesting to receive data from the bill validator whose destination address will always be (30H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (30H)
Y2 = Source address of command
Y3 = File ID
Y4 = Maximum length
Y5 = Control

Z1 = 1DH which indicates SEND BLOCK
Z2 = Destination address of data
Z3 = Block #
Z4 - Z34 = Data (maximum of 31 bytes)

or

Z1 = 1CH which indicates RETRY / DENY
Z2 = Destination address of response
Z3 = Source address of response (30H)
Z4 = Retry delay

VMC Command	Code	Sub-command	VMC Data	Validator Response
EXPANSION COMMAND	37H	FBH **FTL** RETRY / DENY	Y1-Y3	None

The VMC is retrying, denying, or aborting a data transfer to/from the bill validator whose destination address will always be (30H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (30H)
Y2 = Source address of command
Y3 = Retry delay

VMC Command	Code	Sub-command	VMC Data	Validator Response
EXPANSION COMMAND	37H	FCH **FTL** SEND BLOCK	Y1-Y33	None

The VMC is sending data to the bill validator whose destination address will always be (30H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command & data (30H)
Y2 = Block #
Y3 - Y33 = Data (maximum of 31 bytes)

VMC Command	Code	Sub-command	VMC Data	Validator Response
EXPANSION COMMAND	37H	FDH **FTL** OK TO SEND	Y1-Y2	Z1-Z34 (immediate or POLLed)

The VMC is indicating that it is OK for the bill validator to transfer data. The destination address will always be the validator (30H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (30H)
Y2 = Source address of command

Z1 = 1DH which indicates SEND BLOCK
Z2 = Destination address of data
Z3 = Source address of data
Z4 - Z34 = Data (maximum of 31 bytes)

VMC Command	Code	Sub-command	VMC Data	Validator Response
EXPANSION COMMAND	37H	FEH **FTL** REQ TO SEND	Y1-Y5	Z1 (immediate or POLLed)

The VMC is requesting to send data to the bill validator whose destination address will always be (30H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (30H)
Y2 = Source address of command
Y3 = File ID
Y4 = Maximum length
Y5 = Control

Z1 = 1EH which indicates OK TO SEND
Z2 = Destination address of response
Z3 = Source address of response (30H)

or

Z1 = 1CH which indicates RETRY / DENY
Z2 = Destination address of response
Z3 = Source address of response (30H)
Z4 = Retry delay

VMC Command	Code	Sub-Command	VMC Data	Val Response
EXPANSION COMMAND	37H	FFH DIAGNOSTICS	Y1-Yn	Z1 - Zn

Y1 - Yn = Device manufacturer specific instruction for implementing various manufacturing or test modes. Y1 - Yn implies that any number of bytes can be used for the VMC data to the peripheral.

Z1 - Zn = Device manufacturer specific responses after receiving manufacturing or test instructions. Z1 - Zn implies that any number of bytes can be used for the bill validator response data from the peripheral.

6.4 Bill Validator Non-Response Time

The maximum non-response time for the bill validator is five seconds.

6.5 Bill Validator Power Requirements

The current draw for any bill validator must fall within the following limits. All measurements are at the minimum VMC Voltage Output.

Idle mode = 200 mA. (avg.) continuous

Bill transport = 2.5 A. (max.) up to 10 seconds

Note: If both peripherals are supported, vending machines should be able to provide sufficient power to simultaneously supply the above power requirements for both the bill validator **Bill Transport** and coin mechanism **Coin Acceptance** as specified in Section 5.5.

(this page intentionally left blank)

Section 7

Cashless Device(s)
VMC/Peripheral Communication Specifications

7.1 Introduction

This section defines the communications bytes sent and received between the cashless device(s) and the Vending Machine Controller (VMC). As defined in Section 2.3, there are two cashless device addresses; Cashless #1, 00010xxxB (10H) and Cashless #2, 1100xxxB (60H). The second address has been assigned to allow for two unique forms of cashless devices to be resident in the vending machine simultaneously. An example would be a card based system as Cashless Device #1 (10H) and a mobile phone based system as Cashless Device #2 (60H). **Everything defined in this section will be common to the two cashless devices – only the addresses will be different.**

Unless otherwise stated, all monetary values used by the cashless devices and the VMC will be sixteen bit (Level 01 & 02) or thirty-two bit (Level 03 if 32 bit option enabled), unsigned binary numbers. The numbers will be sent most significant byte first and scaled using the parameters provided by the cashless device's READER CONFIGURATION DATA response.

7.2 State Definitions

MDB cashless devices may be viewed as state machines. These states are as follows:

1) Inactive
2) Disabled
3) Enabled
4) Session Idle
5) Vend
6) Revalue (Level 02/03 cashless devices)
7) Negative Vend (Level 03 cashless devices)

7.2.1 Inactive

This is the state of the cashless device at power up or after a reset. While in the Inactive state, cashless devices will NOT be accepted for vending purposes. The cashless device cannot leave this state until all Setup information is received from the VMC.

7.2.2 Disabled

The cashless device automatically enters this state from the Inactive state when it has received the Setup information specified in 7.4.1. It will also enter the Disabled state from the Enabled state when it receives the READER DISABLE command. While in the Disabled state, payment medias will NOT be accepted for vending purposes. The cashless device will remain in this state until either a READER ENABLE command is received (when it will enter the Enabled state) or a RESET is received (when it will enter the Inactive state). For power management purposes, current consumption will not exceed idle mode specification during disabled state.

7.2.3 Enabled

In this state, cashless devices may be used for MDB transactions. The cashless device will remain in this state until a valid payment media is read (when it will enter the Session Idle state),

a READER DISABLE command is received (when it will return to the Disabled state) or a RESET is received (when it will enter the Inactive state).

7.2.4 Session Idle

In the Enabled state, when a valid payment media is processed, the cashless device will issue a BEGIN SESSION response to a VMC POLL and enter the Session Idle state. This indicates that the cashless device is available for vending activities. The only structured exit from the Session Idle state is through the SESSION COMPLETE message from the VMC. The SESSION COMPLETE command will cause the cashless device to respond with an END SESSION message and enable/disable itself appropriately. Vend / Negative Vend / Revalue commands will cause the cashless device to leave the Session Idle state and enter the Vend / Negative Vend / Revalue state when products are selected and purchased.

7.2.5 Vend

This state is entered from the Session Idle state upon reception of a VEND REQUEST message from the VMC. The entire Vend state is an uninterruptable command/response sequence. The cashless device will return to the Session Idle state upon completion of this sequence.

7.2.6 Revalue (Level 02 / 03 Cashless Devices)

This state is entered from the Session Idle state upon reception of a REVALUE REQUEST message from the VMC. The entire Revalue state is an uninterruptable command/response sequence. The cashless device will return to the Session Idle state upon completion of this sequence.

7.2.7 Negative Vend Request (Level 03 Cashless Devices)

This state is entered from the Session Idle state upon reception of a NEGATIVE VEND REQUEST message from the VMC. The entire Negative Vend Request state is an uninterruptable command/response sequence. The cashless device will return to the Session Idle state upon completion of this sequence.

7.3 Command Protocol

After the VMC has issued a command, no new commands may be issued until all data generated in response to that command has been received from the cashless device. The complete response may be an ACK only (e.g. the READER ENABLE command). Alternatively, it may consist of an informational response (e.g. READER CONFIGURATION DATA).

The cashless device may provide an informational response in two ways. It may respond immediately with the requested data, or the cashless device may ACK the VMC command. If ACKed, the VMC must issue POLLs until the cashless device responds with the requested data, or until the Application Maximum Response Time (defined in READER CONFIGURATION response) has elapsed.

The cashless device will define the currency type at the beginning of each session. The currency type will be used for all following transactions in that session. If the VMC does not support this currency type, it will end the session.

Below are the uninterruptable VMC commands which require an informational cashless device response and their associated result:

VMC Command	Cashless Device Response	Result
SETUP/CONFIGURATION DATA =>	READER CONFIGURATION DATA	
EXPANSION/REQUEST ID =>	PERIPHERAL ID	
READER CANCEL =>	CANCELLED	
VEND REQUEST... VEND CANCEL =>	VEND DENIED*	
VEND REQUEST =>	VEND DENIED*	
VEND REQUEST =>	VEND APPROVED =>	VEND SUCCESS*
VEND REQUEST =>	VEND APPROVED =>	VEND FAILURE*
NEGATIVE VEND REQUEST =>	NEGATIVE VEND DENIED*	
NEGATIVE VEND REQUEST =>	NEGATIVE VEND APPROVED =>	NEGATIVE VEND SUCCESS*
NEGATIVE VEND REQUEST =>	NEGATIVE VEND APPROVED =>	NEGATIVE VEND FAILURE*
REVALUE REQUEST=>	REVALUE APPROVED/DENIED*	
SESSION COMPLETE =>	END SESSION	

*These VEND / NEGATIVE VEND / REVALUE REQUEST response sequences constitute the Vend / Negative Vend / Revalue Request states.

Below are the uninterruptable POLLed cashless device which require an informational response

from the VMC:

VMC Command & Data	Cashless Device Response	Result
POLL =>	DATA ENTRY REQUEST + DISPLAY REQUEST (optional)	
POLL =>	DATA ENTRY CANCEL	Cancelled
DATA ENTRY RESPONSE w/ FFs =>		Cancelled

Any command may be issued by the VMC at anytime providing the above command protocol is observed. There are four exceptions to this rule:

1) VEND REQUEST, REVALUE REQUEST, and NEGATIVE VEND REQUEST response sequences may only be initiated in the Session Idle state.

2) The VMC may issue a VEND CANCEL command after issuing a VEND REQUEST, but before receiving a VEND APPROVED/DENIED response. In this case the cashless device will issue a VEND DENIED response to satisfy the original VEND REQUEST response requirement.
3) The cashless device may issue DISPLAY REQUESTs in response to POLLs at any time, if the VMC's display is available for use.
4) The RESET command is allowed at any time, it is not subject to any restrictions.

If a VMC command is received by the cashless device while it is in one of the preceding uninterruptable states, the following will occur:

The cashless device will ACK the offending command (no data response will be forthcoming). The cashless device will respond to the next poll with the "COMMAND OUT OF SEQUENCE" response (0BH).

It should be pointed out to cashless device developers that a command out of sequence will always cause the VMC to issue a RESET command to the cashless device.

7.3.1 Multi-Message Response Format

The multi-message response format permits the cashless device to send multiple messages in response to a single command or POLL. Because all messages are of a fixed length, there is no confusion determining where one message ends and the next message begins. (The total message length is subject to the 36 byte limit imposed by Section 2 of this standard.)

For example, if a cashless device fails to correctly write a payment media after a VEND REQUEST, it may need to report:

1) VEND DENIED
2) MALFUNCTION/ERROR subcode 07h
3) SESSION CANCEL REQUEST

The multi-message response (hex) would look like this:

06	0A 07	04	1B *
1	2	3	4

The first byte above (marked 1) is the VEND DENIED message. The next two bytes (marked 2) are the MALFUNCTION/ERROR message. The third and final message is the CANCEL SESSION REQUEST (marked 3). An eight bit checksum with the mode bit set (marked 4) finishes the message.

It is important to note that the controller must service the messages in the order in which they are received. This is necessary to ensure that command protocol is maintained.

7.3.2 Coin Mechanism Escrow Return Actions

If present, the cashless device return button is controlled by the cashless device and it is the responsibility of the cashless device to terminate a vend sequence if the return button is pressed during a vend sequence.

The reaction of the VMC to the coin mechanism escrow return will vary depending upon the state of the system at the time it is pressed. If escrow return is allowed then a coin mechanism escrow return should be interpreted as VEND CANCEL or END OF SESSION.

1) In the Enabled state, the VMC should send a READER CANCEL command to the cashless device. This allows the user to abort a pre-approved on-line authorisation request.

2) In the Session Idle state, the VMC should send a SESSION COMPLETE command to the cashless device. This will return the cashless device to the Enabled state. The escrow return may cause the system to enter the revalue state prior to the VMC sending the "SESSION COMPLETE" command.

3) In the Vend state, before the cashless device has sent a VEND APPROVED or a VEND DENIED, the VMC should send a VEND CANCEL command to the cashless device. This will cancel the vend and cause the cashless device to refund the payment media if necessary.

4) In all other cases, no message is sent from the VMC to the cashless device.

TABLE 1: COMMANDS & RESPONSES

Command	Code	Sub-command / Data	Response	VMC / Cashless Level Support
Reset	10H 60H	(none)	No Data *	(Level 01+)
Setup	11H 61H	00H - Config Data	01H - Reader Config Data	(Level 01+)
		01H - Max/Min Prices	No Data *	(Level 01+)
Poll	12H 62H	(none)	00H - Just Reset	(Level 01+)
			01H - Reader Config Data	(Level 01+)
			02H - Display Request	(Level 01+)
			03H - Begin Session	(Level 01+)
			04H - Session Cancel Request	(Level 01+)
			05H - Vend Approved	(Level 01+)
			06H - Vend Denied	(Level 01+)
			07H - End Session	(Level 01+)
			08H - Cancelled	(Level 01+)
			09H - Peripheral ID	(Level 01+)
			0AH - Malfunction / Error	(Level 01+)
			0BH - Cmd Out Of Sequence	(Level 01+)
			0DH - Revalue Approved	(Level 02+) (option)
			0EH - Revalue Denied	(Level 02+) (option)
			0FH - Revalue Limit Amount	(Level 02+) (option)
			10H - User File Data	(Level 02) **
			11H - Time/Date Request	(Level 02+) (option)
			12H - Data Entry Request	(Level 03+) (option)
				(Level 03+) (option)
			13H - Data Entry Cancel	(For Future Use)
			14H -1AH	(Level 03+) (option)
			1BH - FTL REQ TO RCV	(Level 03+) (option)
			1CH - FTL RETRY / DENY	(Level 03+) (option)
			1DH - FTL SEND BLOCK	(Level 03+) (option)
			1EH - FTL OK TO SEND	(Level 03+) (option)
			1FH - FTL REQ TO SEND	(For Future Use)
				(Level 01+)

			20H - FEH FFH - Diagnostic Response	
Vend	13H 63H	00H - Vend Request	05H - Vend Approved 06H - Vend Denied	(Level 01+) (Level 01+)
		01H - Vend Cancel	06H - Vend Denied	(Level 01+)
		02H - Vend Success	No Data *	(Level 01+)
		03H - Vend Failure	No Data *	(Level 01+)
		04H - Session Complete	07H - End Session	(Level 01+)
		05H - Cash Sale	No Data *	(Level 01+)
		06H - Negative Vend Request	05H – Vend Approved 06H – Vend Denied	(Level 03+) (option) (Level 03+) (option)
Reader	14H 64H	00H - Reader Disable	No Data *	(Level 01+)
		01H - Reader Enable	No Data *	(Level 01+)
		02H - Reader Cancel	08H - Cancelled	(Level 01+)
		03H - Data Entry Response	No Data *	(Level 03+) (option)
Revalue (option)	15H 65H	00H - Revalue Request	0DH - Revalue Approved 0EH - Revalue Denied	(Level 02+) (option) (Level 02+) (option)
		01H - Revalue Limit Request	0FH - Revalue Limit Amount 0EH - Revalue Denied	(Level 02+) (option) (Level 02+) (option)
Expansion	17H 67H	00H - Request ID	09H - Peripheral ID	(Level 01+)
		01H - Read User File	10H - User File Data	(Level 02) **
		02H - Write User File	No Data *	(Level 02) **
		03H - Write (option) Time/Date	No Data *	(Level 02+) (option)
		04H - Optional Feature Enabled	No Data	(Level 03+)
		FAH - FTL (option) REQ TO RCV	1DH - SEND BLOCK 1CH - RETRY / DENY	(Level 03+) (option) (Level 03+) (option)
		FBH - FTL (option) RETRY / DENY	No Data	(Level 03+) (option)
		FCH - FTL (option) SEND BLOCK	No Data	(Level 03+) (option)
		FDH - FTL (option) OK TO SEND	1DH - SEND BLOCK	(Level 03+) (option)
		FEH - FTL (option) REQ TO SEND	1EH - OK TO SEND 1CH - RETRY/DENY	(Level 03+) (option) (Level 03+) (option)
		FFH - Diagnostics	FFH - Diagnostic Response	(Level 01+)

* No Data response = peripheral just responds with ACK or NAK

**** Obsolete Command – Do not use for new designs. Use EXPANSION - Diagnostics.**

The term (option) indicates that the command/response is a feature enabled by option bits.

NOTE: Cashless device responses which are part of request / response sequences are listed more than once in the above table since the cashless device may respond either immediately to the request (within 5 milliseconds) or to a later POLL.

7.4 VMC/ Cashless Device Command/Response Formats

In the following section, the term "Reader" will indicate either Cashless Device #1 or #2.

7.4.1 Reset and Initialising

RESET **(10H / 60H)**

Reader response:

No Data response

If this command is received by a cashless device it should terminate any ongoing transaction (with an appropriate credit adjustment, if appropriate), eject the payment media (if applicable), and go to the Inactive state.

All Level 02 and above VMCs must follow the RESET command with the following cashless device initializing sequence: (Any new Level 01 VMCs are recommended to follow the sequence.)

Note that the example shows commands for Cashless Device #1 (10H) only. They would be the same for Cashless Device #2 (address 60H).

POLL – 12h

To obtain "JUST RESET" response

SETUP CONFIGURATION DATA – 11 00h

To send the VMC's configuration data and obtain the cashless device's data

SETUP MAX/MIN PRICE – 11 01h

To send the maximum and minimum prices in the VMC. These prices must be sent as Level 01/02 16 bit credit.

EXPANSION REQUEST ID – 17 00h

To obtain additional cashless device information and options (options in Level 03+ only)

EXPANSION ENABLE OPTIONS – 17 04h (Level 03+ only)

To enable desired options

SETUP MAX/MIN PRICE – 11 01h (Level 03+ and option bits 1 & 2 only)

If 32 bit currency option and/or multi currency – multi lingual is enabled (i.e. bits 1 & 2 of expansion enable options), perform **SETUP MAX/MIN PRICE** again to get 32 bit credit and/or user currency – user language (this conditions will be known as EXPANDED CURRENCY MODE in the rest of the document).

READER ENABLE – 14 01h

To enable cashless device (if desired)

7.4.2 SETUP - Config Data

SETUP (11H / 61H)	Config Data (00H) Y1	VMC Feature Level Y2	Columns on Display Y3	Rows On Display Y4	Display Info Y5

Y1 : Configuration data.
VMC is sending its configuration data to reader.

Y2 : VMC Feature Level.
Indicates the feature level of the VMC. The available feature levels are:

01 - The VMC is not capable or will not perform the advanced features as specified in Table 1: COMMANDS & RESPONSES following Section 7.3.2. The reader will not provide advanced information to the VMC, but can do the advanced features internally (transparently to the VMC). The reader has no revaluation capability.

02 - The VMC is capable and willing to perform the advanced features as specified in Table 1: COMMANDS & RESPONSES following Section 7.3.2. The reader will provide advanced information to the VMC (if possible) and will not do the advanced features internally.

03 - The VMC is able to support level 02, but also supports some or all of the optional features listed in the EXPANSION ID command (i.e., file transfer, 32 bit credit, multi-currency / language features, negative vend, and / or data entry).

Y3 : Columns on Display. The number of columns on the display. Set to 00H if the display is not available to the reader.

Y4 : Rows on Display.
The number of rows on the display

Y5 : Display Information - xxxxxyyy
xxxxx = Unused
yyy = Display type
000 : Numbers, upper case letters, blank and decimal point.
001 : Full ASCII
010-111: Unassigned

Reader Response:

Reader Config Data (01H) Z1	Reader Feature Level Z2	Country Code High Z3	Country Code Low Z4	Scale Factor Z5	Decimal Places Z6	Application Maximum Response Time Z7	Miscellaneous Options Z8

Z1 : READER - Configuration data.
Indicates the payment media reader is responding to a SETUP – Configuration data request from the VMC.

Z2 : Reader Feature Level.
Indicates the feature level of the reader. Currently feature levels are:

01 - The reader is not capable or will not perform the advanced features as specified in Table 1: COMMANDS & RESPONSES following Section 7.3.2. The reader will not provide advanced information to the VMC, but can do the advanced features internally (transparently to the VMC). The reader has no revaluation capability.

02 - The reader is capable and willing to perform the advanced features as specified in Table 1: COMMANDS & RESPONSES following Section 7.3.2. The reader will provide advanced information to the VMC (if possible) and will not do the advanced features internally.

03 - The reader is able to support level 02, but also supports some or all of the optional features listed in the EXPANSION ID command (i.e., file transfer, 32 bit credit, multi-currency / language features, negative vend, and / or data entry).

Z3-Z4 : Country / Currency Code - packed BCD.
The packed BCD country / currency code of the reader can be sent in two different forms depending on the value of the left most BCD digit.

If the left most digit is a 0, the International Telephone Code is used to indicate the country that the reader is set-up for. For example, the USA code is 00 01H (Z3 = 00 and Z4 = 01).

If the left most digit is a 1, the latest version of the ISO 4217 numeric currency code is used (see Appendix A1). For example, the code for the US dollar is 18 40H (Z2 = 18 and Z3 = 40) and for the Euro is 1978 (Z3 = 19 and Z4 = 78). Use FFFFh if the country code in unknown.

For level 3 cashless devices, it is mandatory to use the ISO 4217 numeric currency code (see Appendix A1).

Z5 : Scale Factor.
The multiplier used to scale all monetary values transferred between the VMC and the reader.

Z6 : Decimal Places.
The number of decimal places used to communicate monetary values between the VMC and the payment media reader.

All pricing information sent between the VMC and the payment media reader is scaled using the scale factor and decimal places. This corresponds to:

$$ActualPrice = P \cdot X \cdot 10^{-Y}$$

where P is the scaled value send in the price bytes, and X is the scale factor, and Y is the number of decimal places. For example if there are 2 decimal places and the scale factor is 5, then a scaled price of 7 will mean an actual of 0.35.

Z7 : Application Maximum Response Time - seconds.
The maximum length of time a reader will require to provide a response to any command from the VMC. The value reported here supercedes the payment reader's default NON-RESPONSE time defined in section 7.5 if the value reported here is greater.

Z8 : Miscellaneous Options – xxxxyyyy

xxxx: Unused (must be set to 0)

yyyy: Option bits

b0=0: The payment media reader is NOT capable of restoring funds to the user's payment media or account. Do not request refunds.

b0=1: The payment media reader is capable of restoring funds to the user's payment media or account. Refunds may be requested.

b1=0: The payment media reader is NOT multivend capable.

Terminate session after each vend.

b1=1: The payment media reader is multivend capable. Multiple items may be purchased within a single session.

b2=0: The payment media reader does NOT have a display.

b2=1: The payment media reader does have its own display.

b3=0: The payment media reader does NOT support the VEND/CASH SALE subcommand.

b3=1: The payment media reader does support the VEND/CASH SALE subcommand.

b4-b7=0 **Any future options must be covered by the EXPANSION COMMAND option bits.**

7.4.3 SETUP – Max / Min Prices

SETUP (11H / 61H)	Max / Min Prices (01H) Y1	Maximum Price Y2-Y3	Minimum Price Y4-Y5

Level 01 / 02 / 03 Readers

Y1 : Max / Min prices
Indicates the VMC is sending the price range to the reader.

Y2 - Y3 : Maximum Price – scaled
This information should be sent as soon as the VMC prices have been established and any time there is a change in the maximum price, If the VMC does not know the maximum price, FFFFh should be sent.

Y4 -Y5 : Minimum Price – scaled
This information should be sent as soon as the VMC prices have been established and any time there is a change in the minimum price. If the VMC does not know the minimum price, 0000h should be sent.

SETUP (11H / 61H)	Max / MinPrices (01H) Y1	Maximum Price Y2-Y5	Minimum Price Y6-Y9	Currency Code Y10-Y11

Level 03 (EXPANDED CURRENCY MODE) Readers

Y1 : Max / Min prices
Indicates the VMC is sending the price range to the reader.

Y2 – Y5 : Maximum Price – scaled
This information should be sent as soon as the VMC prices have been established and any time there is a change in the maximum price, If the VMC does not know the maximum price, FFFFFFFFh should be sent.

Y6 –Y9 : Minimum Price – scaled
This information should be sent as soon as the VMC prices have been established and any time there is a change in the minimum price. If the VMC does not know the minimum price, 00000000h should be sent.

Y10-Y11 Currency Code
The currency code used during this command per ISO 4217 (see Appendix A1). The value is configured as packed BCD with the leading digit a 1 (one). For example, the code for the US dollar would be 1840 (Z10 = 18 and Z11 = 40). and for the Euro is 1978 (Z10 = 19 and Z11 = 78).

Reader response:

No Data response

7.4.4 POLL

POLL
(12H / 62H)

The POLL command is used by the VMC to obtain information from the payment media reader. This information may include user actions (CANCEL SESSION REQUEST), hardware malfunctions (MALFUNCTION /ERROR), software malfunctions (COMMAND OUT OF SEQUENCE) or information explicitly requested by the controller (READER CONFIGURATION DATA). An ACK response indicates that no error states exist, and either no information request is pending or pending information is not yet ready for transmission.

In addition to an ACK, the VMC may receive the following POLL responses from the payment media reader.

Reader responses:

Just Reset (00H)
Z1

Z1 : JUST RESET
Indicates the payment media reader has been reset.
Note: the difference between ACK and JUST RESET responses is:
00H 00H* =JUST RESET
00H* =ACK
*mode bit=1

Reader Config Info (01H)	Reader Feature Level	Country Code High	Country Code Low	Scale Factor	Decimal Places	Application Maximum Response Time	Miscellaneous Options
Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8

See paragraph 7.4.2 for a detailed explanation of this response.

Display Request (02H)	Display Time	Display Data
Z1	Z2	Z3-Z34

Z1 : DISPLAY REQUEST
The payment media reader is requesting a message to be displayed on the VMC's display.

Z2 : Display Time - 0.1 second units
The requested display time. Either the VMC or the payment media reader may overwrite the message before the time has expired.

Z3-Z34 : Display Data – ASCII
The message to be displayed. Formatting (leading and/or trailing blanks) is the responsibility of the payment media reader.

The number of bytes must equal the product of Y3 and Y4 up to a maximum of 32 bytes in the setup/configuration command.

Begin Session (03H)	Funds Available
Z1	Z2-Z3

Level 01 Readers

Z1 : BEGIN SESSION (**level 01 readers**)
Allow a patron to make a selection, but do not dispense product until funds are approved.

Z2-Z3 : Funds Available - scaled

a. Lesser of the user's payment media or account balance or FFFEh units.

b. Not yet determined - FFFFh.

Begin Session (03H)	Funds Available	Payment media ID	Payment Type	Payment Data
Z1	Z2-Z3	Z4-Z7	Z8	Z9-Z10

Level 02 / 03 Readers

Z1 : BEGIN SESSION (**level 02/03 readers**)
Allow a patron to make a selection, but do not dispense product until funds are approved.

Z2-Z3 : Funds Available – scaled

a. Lesser of the user's payment media or account balance or FFFEh units.

b. Not yet determined - FFFFh.

Z4-Z7 : Payment media ID.
00000000h-FFFFFFFEh=Payment media identification number. FFFFFFFFh = unknown payment media ID.

Z8 : Type of payment:
00xxxxxxb = normal vend card (refer EVA-DTS Standard, Appendix A.1.1 Definitions)
x1xxxxxxb = test media
1xxxxxxxb = free vend card
xx000000b -0 VMC default prices.
xx000001b -1 User Group (Z9 = EVA-DTS Element DA701)
Price list number (Z10 = EVA-DTS Element LA101)*
xx000010b -2 User Group (Z9 = EVA-DTS Element DA701)
Discount group index (Z10 = EVA-DTS Element MA403)
xx000011b -3 Discount percentage factor (Z9=00, Z10 = 0 to 100**,

report as positive value in EVA-DTS Element MA404)

xx000100b -4 Surcharge percentage factor (Z9=00, Z10 = 0 to 100**, report as negative value in EVA-DTS Element MA404)

* User Group is a segmentation of all authorized users. It allows selective cost allocation. A User Group usually has no direct relation to a price list.
Price Lists are tables of prices. Each Price List contains an individual price for each product.
Discount Group indicates the Price List on which the Percentage Factor will be applied.
If the User Group, the Price List or Discount Group is unknown

by the VMC, the normal prices are used (Z8 is defaulted to 00h).

Minimum value for Z9 and Z10 is 0.

** Percentages are expressed in binary (00 to 64h)

Note:
These functions may NOT be supported by all VMCs.

Z9-Z10 : Payment data as defined above.

Begin Session (03H)	Funds Available	Payment media ID	Payment Type	Payment Data	User Language	User Currency Code	Card Options
Z1	Z2-Z5	Z6-Z9	Z10	Z11-Z12	Z13-Z14	Z15-Z16	Z17

Level 03 (EXPANDED CURRENCY MODE) Readers

Z1 : BEGIN SESSION (**level 03 readers / EXPANDED CURRENCY MODE**)
Allow a patron to make a selection, but do not dispense product until funds are approved.

Z2-Z5 : Funds Available – scaled

a. Lesser of the user's payment media or account balance or FFFFFFFEh units.

b. Not yet determined - FFFFFFFFh.

Z6-Z9 : Payment media ID.
00000000h-FFFFFFFEh=Payment media identification number. FFFFFFFFh = unknown payment media ID.

Z10 : Type of payment:
00xxxxxxb = normal vend card (refer EVA-DTS Standard, Appendix A.1.1 Definitions)
x1xxxxxxb = test media
1xxxxxxxb = free vend card
xx000000b -0 VMC default prices.
xx000001b -1 User Group (Z11 = EVA-DTS Element DA701)
Price list number (Z12 = EVA-DTS Element LA101)*
xx000010b -2 User Group (Z11 = EVA-DTS Element DA701)
Discount group index (Z12 = EVA-DTS Element MA403)
xx000011b -3 Discount percentage factor (Z11=00, Z12 = 0 to 100**, report as positive value in EVA-DTS Element MA404)
xx000100b -4 Surcharge percentage factor (Z11=00, Z12 = 0 to 100**, report as negative value in EVA-DTS Element MA404)

* User Group is a segmentation of all authorized users. It allows selective cost allocation. A User Group usually has no direct relation to a price list.
Price Lists are tables of prices. Each Price List contains an individual price for each product.
Discount Group indicates the Price List on which the Percentage Factor will be applied.
If the User Group, the Price List or Discount Group is unknown

by the VMC, the normal prices are used (Z10 is defaulted to

00h).

Minimum value for Z11 and Z12 is 0.

** Percentages are expressed in binary (00 to 64h)

Note:
These functions may NOT be supported by all VMCs.

Z11-Z12: Payment data as defined above.

Z13-Z14 User language to use during this session (2 ASCII characters per ISO 639:latest version). The user language is read from the patrons card and, if supported, should be used instead of the VMC default language (taken according to the setup command International Telephone code) up to the next "session complete". If the VMC is not able to support this language, the default setting should be used.

Z15-Z16 User currency code to use during this session per ISO 4217 (see Appendix A1). The value is configured as packed BCD with the leading digit a 1 (one). For example, the code for the US dollar would be 1840 (Z15 = 18 and Z16 = 40).

and for the Euro is 1978 (Z6 = 19 and Z7 = 78).

Z17 Card options (overrides any previous default settings for reader)

- b0=0: The VMC displays the credit if it is programmed to do so
- b0=1: The VMC **must not display** the credit (privacy purpose – user option)
- b1=0: The actual inserted patrons card has no refund capability
- b1=1: The actual inserted patrons card has refund capability (Note: a reader with refund capability may be used with both type of cards)
- b2=0 The actual inserted patrons card has no revalue capability
- b2=1 The actual inserted patrons card has revalue & negative vend capability
- b3-b7: Reserved for future extensions (unused bits must be set to 0)

Refund means the ability to put money back on the inserted patrons card up to the value of the last transaction. Revalue means the ability to put money back on the inserted patrons card up to any value.

The card reader will define the currency type at the beginning of each card session. **The currency type will be used for all following transactions in that session. If the VMC does not support this currency type, it will end the session.**

Session Cancel Request (04H)
Z1

Z1 : SESSION CANCEL REQUEST
The payment media reader is requesting the VMC to cancel the session. The VMC should initiate an eventual SESSION COMPLETE. This response is sent to the VMC whenever the payment media is removed or a request for removal from the reader is made by the user (e.g. if a return button on the reader is pressed).

Vend Approved (05H)	Vend Amount
Z1	Z2-Z3

Level 01 / 02 / 03 Readers

Refer to paragraph 7.4.5 for detailed explanation.

Vend Approved (05H)	Vend Amount

Z1	Z2-Z5

Level 03 (EXPANDED CURRENCY MODE) Readers

Refer to paragraph 7.4.5 for detailed explanation.

Vend Denied (06H) Z1

Refer to paragraph 7.4.5 for detailed explanation.

End Session (07H) Z1

Refer to paragraph 7.4.9 for detailed explanation.

Cancelled (08H) Z1

Refer to paragraph 7.4.14 for detailed explanation.

Peripheral ID (09H) Z1	Manufacturer Code Z2-Z4	Serial Number Z5-Z16	Model Number Z17-Z28	Software Version Z29-Z30

Level 01 / 02 / 03 Readers (If VMC indicates Level 01 or 02)

Z1 : PERIPHERAL ID
Reader is sending peripheral ID information.

Z2 - Z4 : Manufacturer Code - ASCII
Identification code for the equipment supplier. Currently defined codes are listed in the **EVA** document entitled "***European Vending Association Data Transfer Standard***" (**EVA-DTS**), the Audit Data Lists section, sub-section 2, "Manufacturer Codes".

Z5-Z16 : Serial Number – ASCII
Factory assigned serial number.

Z17-Z28 : Model Number - ASCII

Manufacturer assigned model number.

Z29-Z30 : Software Version - packed BCD
Current software version.

Peripheral ID (09H)	Manufacturer Code	Serial Number	Model Number	Software Version	Optional Feature bits
Z1	Z2-Z4	Z5-Z16	Z17-Z28	Z29-Z30	Z31 - Z34

Level 03 Readers (If VMC indicates Level 03)

Z1 : PERIPHERAL ID
Reader is sending peripheral ID information.

Z2 - Z4 : Manufacturer Code - ASCII
Identification code for the equipment supplier. Currently defined codes are listed in the **EVA** document entitled "***European Vending Association Data Transfer Standard***" (**EVA-DTS**), the Audit Data Lists section, sub-section 2, "Manufacturer Codes".

Z5-Z16 : Serial Number – ASCII
Factory assigned serial number.

Z17-Z28 : Model Number - ASCII
Manufacturer assigned model number.

Z29-Z30 : Software Version - packed BCD
Current software version.

Z31- Z34 Optional Feature Bits. Each of the 32 bits indicate an optional feature availability. Bits should be sent in descending order, i.e. bit 31 is sent first and bit 0 is sent last. Options **must be enabled by the VMC** using the Expansion Optional Feature Bit Enable (17H-04H) command and **all features are disabled after a reset**. Currently defined options are:

b0 - File Transport Layer supported
b1 - 0 = 16 bit monetary format, 1 = 32 bit monetary format
b2 - multi currency / multi lingual
b3 - negative vend
b4 - data entry
b5 to b31 not used (should be set to 0)

Note: If 32 bit monetary format (b1) and or multi currency / multi lingual (b2) options are enabled, this condition will be known as **EXPANDED CURRENCY MODE** in the rest of the document.

Malfunction / Error	Error Code
(0AH) Z1	Z2

Z1 : MALFUNCTION/ERROR
The payment media reader is reporting a malfunction or error.

Z2 : Error Code – xxxxyyyy

xxxx	error types
0000:	Payment media Error1
0001:	Invalid Payment media1
0010:	Tamper Error1
0011:	Manufacturer Defined Error1
0100:	Communications Error2
0101:	Reader Requires Service2
0110:	Unassigned2
0111:	Manufacturer Defined Error2
1000:	Reader Failure3
1001:	Communications Error3
1010:	Payment media Jammed3
1011:	Manufacturer Defined Error
1100:	Refund error – internal reader credit lost
1101-1111:	Unassigned

1 Transient error - Reported once
2 Non-transient error - Reported every POLL until cleared. Reader still functional.
3 Non-transient error - Reported every POLL until cleared. Reader not presently functional.

yyyy = Manufacturer defined subcode

Transient Error Handling
The error will be reported to the VMC until it has been ACKnowledged. The error state will be cleared in the reader, and normal operations will continue.

Non-transient Error Handling
The error will be reported to the VMC at each POLL as long as it exists. If the reader is still functional, multi-message responses will allow normal responses in addition to the error report.

Note: Refund error is sent from the media reader when it is not able to refund money to the payment media following a failed or cancelled vend. The reader internally cancels the credit and the credit is lost.

Command Out of Sequence (0BH)
Z1

Level 01 Readers

Z1 : COMMAND OUT OF SEQUENCE (Level 01 readers)
The payment media reader has received a command that is not executable in its current state, or that violates one of the uninterruptable sequences. The offending command should be ACKed but not acted upon the reader. The VMC will send the RESET command to the reader upon reception of this response. Note that the reader will continue with any credit update process prior to resetting.

Command Out of Sequence (0BH)	Status
Z1	Z2

Level 02 / 03 Readers

Z1 : COMMAND OUT OF SEQUENCE. (Level 02/03 readers)
The payment media reader has received a command that is not executable in its current state, or that violates one of the uninterruptable sequences. The offending command should be ACKed but not acted upon the reader. The VMC will send the RESET command to the reader upon reception of this response. Note that the reader will continue with any credit update process prior to resetting.

Z2 : Status
The state of the payment media reader.
01: Inactive state
02: Disabled state
03: Enabled state
04: Session idle state
05: Vend state
06: Revalue state
07: Negative Vend state

Revalue

Approved (0DH) Z1

Level 02 / 03 Readers

Refer to paragraph 7.4.16 for detailed explanation.

Revalue Denied (0EH) Z1

Level 02 / 03 Readers

Refer to paragraph 7.4.16 for detailed explanation.

Revalue Limit Amount (0FH) Z1	Revalue Limit Amount Z2-Z3

Level 02 / 03 Readers

Refer to paragraph 7.4.17 for detailed explanation.

Revalue Limit Amount (0FH) Z1	Revalue Limit Amount Z2-Z5

Level 03 (EXPANDED CURRENCY MODE) Readers

Refer to paragraph 7.4.17 for detailed explanation.

User File Data (10H) Z1	Number of User File Z2	Length Of User File Z3	User Data Z4-Zn

Level 02 Readers

Obsolete Response – Do not use for new designs!! (Use EXPANSION – Diagnostics)

Refer to paragraph 7.4.19 for detailed explanation.

Time/Date Request (11H) Z1

Level 02 / 03 Readers

Z1 : TIME DATE REQUEST
In certain circumstances it will be necessary to synchronize the real time clock of the card reader with real time clock of the VMC. The card reader will respond with TIME/DATE REQUEST to a POLL command of the VMC. The VMC will follow with the EXPANSION-WRITE TIME/DATE FILE to the card reader. Refer to paragraph 7.4.19.

Data Entry Request Response (12H) Z1	Data Entry Length and Repeat Bit Z2

Level 03 Readers (if Data Entry option enabled)

Z1 : DATA ENTRY REQUEST
The reader is making a DATA ENTRY REQUEST.

Z2 : DATA ENTRY LENGTH and REPEAT BIT
rnnnnnnnn
r – Repeat Bit (0 = initial request / 1 = repeated requests
nnnnnnn – number of requested characters / keys

Depending on the type of data being entered, it is a higher level system decision on whether or not the data is displayed on either the vending machine or card reader. If the data is not displayed (a recommendation for certain types of sensitive data) the vending machine or card reader display can still be optionally used to indicate a prompt and/or representation of the data entered for user feedback (i.e., asterisks *****).

If the card reader uses the vending machine's display for Data Entry information, it **must concatenate** the DATA ENTRY REQUEST Response (12H) with the DISPLAY REQUEST response (02H). Upon receipt of the response pair, the vending machine controller will give its display to the card reader for the duration of the Data Entry session plus the amount of time specified in the Z2 Display Time following the end of the session (regardless of a normal or cancelled session). In essence, the vending machine controller will not write anything to its display during the Data Entry session plus the Z2 time. The reader will be able to update the Data Entry information on the vending machine's display by sending additional DISPLAY REQUEST responses during the Data Entry session.

Please see additional DATA ENTRY procedures in Section 7.4.15.

Data Entry Cancel (13H) Z1

Level 03 Readers (if Data Entry option enabled)

Z1 : DATA ENTRY CANCEL
The user has pushed the reader's RETURN button before completing the DATA

ENTRY. The VMC should terminate all DATA ENTRY activity in progress.

FTL
REQ TO RCV
(1BH)
Z1

Level 03 Readers (if File Transport Layer option enabled)

Z1 : **FTL** REQ TO RCV
The reader is requesting to receive data from a device or VMC.

Z2 : **FTL** Destination Address
The destination address of the response as defined in Section 2.6.

Z3 : **FTL** Source Address (Reader = 10H / 60H)
The source address of the response as defined in Section 2.6.

Z4 : **FTL** File ID
The type of information desired as defined in Section 2.6.

Z5 : **FTL** Maximum Length
The total number of blocks in the file as defined in Section 2.6.

Z6 : **FTL** Control
Data transfer control information as defined in Section 2.6.

FTL
RETRY/DENY
(1CH)
Z1

Level 03 Readers (if File Transport Layer option enabled)

Z1 : **FTL** RETRY / DENY
The reader is requesting a device or VMC to retry or deny the last FTL command.

Z2 : **FTL** Destination Address
The destination address of the response as defined in Section 2.6.

Z3 : **FTL** Source Address (Reader = 10H / 60H)
The source address of the response as defined in Section 2.6.

Z4 : **FTL** Retry Delay
The retry delay as defined in Section 2.6.

FTL
SEND
BLOCK
(1DH)
Z1

Level 03 Readers (if File Transport Layer option enabled)

Z1 : **FTL** SEND BLOCK
The reader is sending a block of data (maximum of 31 bytes) to a device or VMC.

Z2 : **FTL** Destination Address
The destination address of the response as defined in Section 2.6.

Z3 : **FTL** Block #
The sequential number of the block as defined in Section 2.6.

Z4- Z34 : **FTL** Data (maximum of 31 bytes)
The actual data portion of the block as defined in Section 2.6.

FTL
OK TO SEND
(1EH)
Z1

Level 03 Readers (if File Transport Layer option enabled)

Z1 : **FTL** OK TO SEND
The reader is indicating that it is OK for the device or VMC to send it data.

Z2 : **FTL** Destination Address
The destination address of the response as defined in Section 2.6.

Z3 : **FTL** Source Address (Reader = 10H / 60H)
The source address of the response as defined in Section 2.6.

FTL REQ TO SEND (1FH) Z1

Level 03 Readers (if File Transport Layer option enabled)

Z1 : **FTL** REQ TO SEND
The reader is requesting to send data to a device or VMC.

Z2 : **FTL** Destination Address
The destination address of the response as defined in Section 2.6.

Z3 : **FTL** Source Address (Reader = 10H / 60H)
The source address of the response as defined in Section 2.6.

Z4 : **FTL** File ID
The type of information desired as defined in Section 2.6.

Z5 : **FTL** Maximum Length
The total number of blocks in the file as defined in Section 2.6.

Z6 : **FTL** Control
Data transfer control information as defined in Section 2.6.

Diagnostics Response (FFH) Z1	User Defined Data Z2-Zn

Refer to paragraph 7.4.28 for detailed explanation.

7.4.5 VEND - Request

	Vend Request (00H) Y1	Item Price Y2-Y3	Item Number Y4-Y5
Vend (13H / 63H)			

Level 01 / 02 / 03 Readers

Y1 : VEND REQUEST
The patron has made a selection. The VMC is requesting vend approval from the payment media reader before dispensing the product.

Y2-Y3 : Item Price - scaled
The price of the selected product.

Y4-Y5 : Item Number
The item number of the selected product. This number is defined by the manufacturer, and set to FFFFh for undefined or not implemented.

Reader response:

Vend Approved (05H) Z1	Vend Amount Z2-Z3

Z1 : VEND APPROVED
Allow the selected product to be dispensed.

Z2-Z3 : Vend Amount - scaled
This is the amount deducted from the user's payment media or account. This may not match the amount specified in the VEND REQUEST command; it may be surcharged or discounted.
FFFFh - an electronic token was used.

NOTE: The VMC must use Vend Amount to update the credit on the screen. The Reader must fill this field with the used amount for the transaction.

Vend Denied (06H) Z1

Z1 : VEND DENIED
Approval denied for the patron's selection. Do not dispense any products.

Vend (13H / 63H)	Vend Request (00H) Y1	Item Price Y2-Y5	Item Number Y6-Y7

Level 03 (EXPANDED CURRENCY MODE) Readers

Y1 : VEND REQUEST
The patron has made a selection. The VMC is requesting vend approval from the payment media reader before dispensing the product.

Y2-Y5 : Item Price – scaled
The price of the selected product.

Y6-Y7 : Item Number
The item number of the selected product. This number is defined by the

manufacturer, and set to FFFFh for undefined or not implemented.

Reader Response:

Vend Approved (05H) Z1	Vend Amount Z2-Z5

Level 03 (EXPANDED CURRENCY MODE) Readers

Z1 : VEND APPROVED
Allow the selected product to be dispensed.

Z2-Z5 : Vend Amount - scaled
This is the amount deducted from the user's payment media or account. This may not match the amount specified in the VEND REQUEST command; it may be surcharged or discounted.
FFFFFFFFh - an electronic token was used.

NOTE: The VMC must use Vend Amount to update the credit on the screen. The Reader must fill this field with the used amount for the transaction.

7.4.6 VEND - Cancel

Vend (13H / 63H)	Vend Cancel (01H) Y1

Y1 : VEND CANCEL
This command can be issued by the VMC to cancel a VEND REQUEST command before a VEND APPROVED/DENIED has been sent by the payment media reader. The payment media reader will respond to VEND CANCEL with a VEND DENIED and return to the Session Idle state.

Reader response:

Vend Denied (06H) Z1

See paragraph 7.4.5 for explanation.

7.4.7 VEND - Success

Vend (13H / 63H)	Vend Success (02H) Y1	Item Number Y2-Y3

Y1 : VEND SUCCESS
The selected product has been successfully dispensed.

Y2-Y3 : Item number
The item number of the selected product. This number is defined by the manufacturer, and set to FFFFh for undefined or not implemented.

NOTE A reset between VEND APPROVED and VEND SUCCESS shall be interpreted as a VEND SUCCESS.

Reader response:

No Data response

7.4.8 VEND - Failure

Vend (13H / 63H)	Vend Failure (03H) Y1

Y1 : VEND FAILURE
A vend has been attempted at the VMC but a problem has been detected and the vend has failed. The product was not dispensed. Funds should be refunded to user's account.

Reader response:

No Data response

Vend failure sequence

In order to ensure that a reader refunds after a Vend Failure command, the VMC must send at least a single Poll command to obtain the reader possible answers:

ACK	Refund Complete
MALFUNCTION ERROR code 1100yyyy	Refund error-internal reader credit lost
SILENCE	Refund in progress. VMC must repoll reader until ACK or Malfunction error answer for maximum NON Response time.

7.4.9 SESSION COMPLETE

Vend (13H / 63H)	Session Complete (04H) Y1

Y1 : SESSION COMPLETE
This tells the payment media reader that the session is complete and to return to the Enabled state. SESSION COMPLETE is part of a command/response sequence that requires an END SESSION response from the reader.

Reader response:

End Session (07H) Z1

Z1 : END SESSION
This command is issued in response to a SESSION COMPLETE command. The

END SESSION response indicates the reader has returned to the Enabled state. If "END SESSION" is not received by the VMC within a the maximum application non-response time, the VMC must issue a "RESET" command.

7.4.10 CASH SALE

Vend (13H / 63H)	Cash Sale (05H) Y1	Item Price Y2-Y3	Item Number Y4-Y5

Level 01 / 02 / 03 Readers

Y1 : CASH SALE
A cash sale (cash only or cash and cashless) has been successfully completed by the VMC.

Y2-Y3 : Item Price – scaled
The price of the selected product or cash portion of the price.

Y4-Y5 : Item Number
The item number of the selected product. This number is defined by the manufacturer, and set to FFFFh for undefined or not implemented.

Note: This command is issued for cash auditing applications and is sent to the payment media reader if the SETUP/CONFIGURATION bit (b3) is enabled anytime a valid cash transaction is completed via a coin mechanism or bill validator.

Reader response:

No Data response

Vend (13H)	Cash Sale (05H) Y1	Item Price Y2-Y5	Item Number Y6-Y7	Item Currency Y8-Y9

Level 03 (EXPANDED CURRENCY MODE) Readers

Y1 : **CASH SALE**
A cash sale (cash only or cash and cashless) has been successfully completed by the VMC.

Y2-Y5 : Item Price – scaled
The price of the selected product or cash portion of the price.

Y6-Y7 : Item Number
The item number of the selected product. This number is defined by the manufacturer, and set to FFFFh for undefined or not implemented.

Y8-Y9 : Item Currency
The currency for the item price used during the vend. This value may be converted within the reader to the readers balancing currency. The item currency is sent using the numeric code as defined in ISO 4217 (see Appendix A1). The value is configured as packed BCD with the leading digit a 1 (one). For example, the code for the US dollar would be 1840 (Z10 = 18 and Z11 = 40). and for the Euro is 1978 (Z10 = 19 and Z11 = 78).

Note: This command is issued for cash auditing applications and is sent to the payment media reader if the SETUP/CONFIGURATION bit (b3) is enabled anytime a valid cash transaction is completed via a coin mechanism or bill validator.

Reader response:

No Data response

7.4.11 Negative Vend Request

Vend (13H / 63H)	Neg.Vend Request (06H) Y1	Item Value Y2-Y3	Item Number Y4-Y5

Level 03 Reader

Y1 : NEGATIVE VEND REQUEST
The patron has inserted an item. The VMC is requesting negative vend approval from the payment media reader before accepting the returned product.

Y2-Y3 : Item value – scaled
The value of the inserted product (16 Bit).

Y4-Y5 : Item Number
The item number of the inserted product. This number is defined by the manufacturer, and set to FFFFh for undefined or not implemented.

Reader response:

Vend Approved (05H) Z1	Vend Amount Z2-Z3

Level 03 (EXPANDED CURRENCY MODE disabled) Readers

Z1 : VEND APPROVED
Allow the returned product to be accepted, i.e. this means, the reader will be able to credit the value to the patrons card, when a vend success will follow the approved.

Z2-Z3 : Vend Amount – scaled
This is the amount of credit, which will be added to the user's payment media or account. This may not match the amount specified in the NEGATIVE VEND REQUEST command; it may be surcharged or discounted.
FFFFh - an electronic token will be credited.

Vend (13H / 63H)	Neg.Vend Request (06H) Y1	Item Value Y2-Y5	Item Number Y6-Y7

Level 03 (EXPANDED CURRENCY MODE) Readers

Y1 : NEGATIVE VEND REQUEST
The patron has inserted an item. The VMC is requesting negative vend approval from the payment media reader before accepting the returned product.

Y2-Y5 : Item value – scaled
The value of the inserted product.

Y6-Y7 : Item Number
The item number of the inserted product. This number is defined by the manufacturer, and set to FFFFh for undefined or not implemented.

Reader response:

Vend Approved (05H) Z1	Vend Amount Z2-Z5

Level 03 (EXPANDED CURRENCY MODE) Readers

Z1 : VEND APPROVED
Allow the returned product to be accepted, i.e. this means, the reader will be able to credit the value to the patrons card, when a vend success will follow the approved.

Z2-Z5 : Vend Amount – scaled
This is the amount of credit, which will be added to the user's payment media or account. This may not match the amount specified in the NEGATIVE VEND REQUEST command; it may be surcharged or discounted.

FFFFFFFFh - an electronic token will be credited.

Vend Denied (06H) Z1

Z1 : VEND DENIED
Approval denied for the returned product. Do not accept the product or return it if possible.

Note: This command is used in the uninterruptable vend sequence like the normal REQUEST VEND and is followed by the normal responses VEND APPROVED or VEND DENIED, for the reader to confirm the credit update possibility and the final VEND SUCCESS or VEND FAILURE command to update the patron's credit.

Designers of cashless devices must pay special attention in implementing this command, especially for non locking readers. Credit should only be generated on the media upon final reception of VEND SUCCESS to avoid unwanted credit in the system.

Designers of both the VMC and the readers have to deal with fault conditions of such a system carefully. A normal sequence description is added to the example vend sessions with hints to different application features.

7.4.12 READER - Disable

Reader (14H / 64H)	Disable (00H) Y1

Y1 : READER DISABLE
This informs the payment media reader that it has been disabled, i.e. it should no longer accept a patron's payment media for the purpose of vending. Vending activities may be re-enabled using the READER ENABLE command. The payment media reader should retain all SETUP information.

NOTE Any transaction in progress will not be affected and should continue to its normal completion.

Reader response:

No Data response

7.4.13 READER - Enable

Reader (14H / 64H)	Enable (01H) Y1

Y1 : READER ENABLE
This informs the payment media reader that is has been enabled, i.e. it should now accept a patron's payment media for vending purposes. This command must be issued to a reader in the Disabled state to enable vending operations.

Reader response:

No Data response

7.4.14 READER - Cancel

Reader (14H / 64H)	Cancel (02H) Y1

Y1 : READER CANCEL
This command is issued to abort payment media reader activities which occur in the Enabled state. It is the first part of a command/response sequence which requires a CANCELLED response from the reader.

Reader response:

Cancelled (08H) Z1

Z1 : CANCELLED
This is the reader's response to the READER CANCEL command from the VMC. This command comprises a command/response sequence. Its use is only appropriate in the Enabled state.

7.4.15 DATA ENTRY – Response (Key Entries)

The purpose of the overall Data Entry request / response sequence is to allow the machine user to enter data (i.e., a card validation number) using the selection buttons on the vending machine.

The DATA ENTRY request / response sequence can occur in the Enabled state only. It is the responsibility of the reader to enforce this rule.

Depending on the type of data being entered, it is a higher level system decision on whether or not the data is displayed on either the vending machine or card reader. If the data is not displayed (a recommendation for certain types of sensitive data) the vending machine or card reader display can still be optionally used to indicate a prompt and/or representation of the data entered for user feedback (i.e., asterisks *****). **Please see additional information on the vending machine's display usage for Data Entry in the DATA ENTRY REQUEST Response (12H) description in the 7.4.4 POLL section.**

The DATA ENTRY RESPONSE key entries are sent to the reader as they are pressed.

Depending on the user's speed of entry and vending machine controller cycle time, the data may

be sent either as a digit at a time, a sub group of digits, or the entire length of digits as specified in the Z2 Data Entry Length byte in the DATA ENTRY REQUEST response. For example, if the Data Entry Length is 6 digits, but only 2 are initially (and quickly) entered, the vending machine controller will send the 2 that are available via the DATA ENTRY RESPONSE Y2-Y9 command. The balance will be sent via other DATA ENTRY RESPONSE Y2-Y9 commands when available.

It is up to the reader to merge the received DATA ENTRY RESPONSE data and optionally update the display as required. The session is ended after the VMC sends the final DATA ENTRY RESPONSE data (no SESSION COMPLETE command is required). Note that the VMC display will remain available to the reader for the amount of time requested in the previous DISPLAY REQUEST response.

If the data entry process is cancelled by the VMC for any reason, the VMC will send the DATA ENTRY RESPONSE with all data bytes (Y2-Y9) set to FFh. This will terminate the DATA ENTRY REQUEST and return the reader to the Enabled state.

For ease of command message processing, the Data Entry Data has been fixed at 8 characters (Y2-Y9). Unused bytes must be sent as 00h to pad out the entire command to byte Y9.

Reader (14H / 64H)	Data Entry Response (03H) Y1	Data Entry Data Y2-Y9

Level 03 Readers (if option enabled)

Y1 : DATA ENTRY RESPONSE
The VMC is providing a DATA ENTRY RESPONSE to the reader.

Y2-Y9 : DATA ENTRY DATA
Data should be in ASCII, one character per byte. Data should be left justified (first character / key in Y2, second in Y3, etc.). The number of data bytes must equal eight (8) and unused data bytes must be sent as 00h.

If the data entry process is cancelled by the VMC for any reason, the VMC will send this message with all DATA ENTRY data bytes set to FFh.

Note: The reader must translate the VMC key information into the appropriate key needed for the application

Reader response:

No Data response

Note: If the reader has additional display information to send to the VMC following the DATA ENTRY RESPONSE, it should send it via a DISPLAY REQUEST response to one of the next POLL commands from the VMC.

7.4.16 REVALUE - Request (Level 02 / 03 Readers)

Revalue (15H / 65H)	Revalue Request (00H) Y1	Revalue Amount Y2-Y3

Level 02 / 03 Readers

Y1 : REVALUE REQUEST (Level 02 Readers)
A balance in the VMC account because coins or bills were accepted or some balance is left after a vend. With this command the VMC tries to transfer the balance to the payment media.

Y2-Y3 : Revalue amount - scaled.
The revalue amount should not exceed the revalue limit value given by the command REVALUE LIMIT REQUEST.

Revalue (15H / 65H)	Revalue Request (00H) Y1	Revalue Amount Y2-Y5

Level 03 (EXPANDED CURRENCY MODE) Readers

Y1 : REVALUE REQUEST (Level 03 Readers)
A balance in the VMC account because coins or bills were accepted or some balance is left after a vend. With this command the VMC tries to transfer the balance to the payment media.

Y2-Y5 : Revalue Amount - scaled.
The revalue amount should not exceed the revalue limit value given by the command REVALUE LIMIT REQUEST.

Reader response:

Revalue Approved (0DH) Z1

Level 02 / 03 Readers

Z1 : REVALUE APPROVED (Level 02 / 03 Readers)
A balance is in the VMC account because coins or bills were accepted or some balance is left after a vend. The VMC has issued a REVALUE REQUEST to the payment media reader to transfer the balance to the payment media. The payment media reader accepted the request and added its value to the payment media balance. The reader then responds with a REVALUE APPROVED, so the VMC

may clear the account.

Revalue Denied (0EH) Z1

Level 02 Readers

Z1 : REVALUE DENIED (Level 02 / 03 Readers)
A balance is in the VMC account because coins or bills were accepted or some balance is left after a vend. The VMC has issued a REVALUE REQUEST to the payment media reader to transfer the balance to the payment media. The payment media reader does not accept the request and responds with a REVALUE DENIED, so the VMC has to pay out change. It is a quite common situation if there is no payment media inserted at this moment.

7.4.17 REVALUE - Limit Request (Level 02 / 03 Readers)

Revalue (15H / 65H)	Revalue Limit Request (01H) Y1

Level 02 / 03 Readers

Note: If revaluing, follow the BEGIN SESSION with this command.

Y1 : REVALUE LIMIT REQUEST (Level 02 Readers)
In a configuration with a bill and/or coin acceptor and payment media reader connected to a VMC, the VMC must know the maximum amount the payment media reader eventually will accept by a REVALUE REQUEST. Especially if the bill acceptor accepts a wide range of bills. Otherwise the VMC may be confronted by the situation where it accepted a high value bill and is unable to pay back cash or revalue it to a payment media. (see also below)

Reader response:

Revalue Limit Amount (0FH) Z1	Revalue Limit Amount Z2-Z3

Level 02 / 03 (EXPANDED CURRENCY MODE disabled) Readers

Z1 : REVALUE LIMIT AMOUNT (Level 02 / 03 Readers)

The patron intends to revalue the payment media with a bill of some value. The VMC must know what kind of bills to accept, so it will issue a REVALUE LIMIT REQUEST to get the amount the payment media reader will accept. The payment media reader will respond with the scaled value, calculated with the maximum allowed payment media balance minus the current balance of the payment media. The payment media reader responds with REVALUE DENIED if there is no payment media available upon this request.

Z2-Z3 : Revalue limit value - scaled.

Reader response:

Revalue Limit Amount (0FH)	Revalue Limit Amount
Z1	Z2-Z5

Level 03 (EXPANDED CURRENCY MODE) Readers

Z1 : REVALUE LIMIT AMOUNT (Level 03 Readers)
The patron intends to revalue the payment media with a bill of some value. The VMC must know what kind of bills to accept, so it will issue a REVALUE LIMIT REQUEST to get the amount the payment media reader will accept. The payment media reader will respond with the scaled value, calculated with the maximum allowed payment media balance minus the current balance of the payment media. The payment media reader responds with REVALUE DENIED if there is no payment media available upon this request.

Z2-Z5 : Revalue Limit Value - scaled.

7.4.18 EXPANSION - Request ID

	Request ID (00H)	Manufacturer Code	Serial Number	Model Number	Software Version
Expansion (17H / 67H)	Y1	Y2-Y4	Y5-Y16	Y17-Y28	Y29-Y30

Y1 : REQUEST ID
The VMC is requesting payment media reader identification information. The information included above (Y2-Y30) provides the payment media reader with VMC identification information.

Y2-Y4 : Manufacturer Code - ASCII
Identification code for the equipment supplier. Currently defined codes are listed in the EVA document entitled "The Data Transfer Standard EVA-DTS" document, the Audit Data Dictionary section, chapter 4,

"Manufacturer Codes".

Y5-Y16 : Serial Number - ASCII
Factory assigned serial number.

Y17-Y28 : Model Number - ASCII
Manufacturer assigned model number.

Y29-Y30 : Software Version - packed BCD
Current software version.

Reader response:

Peripheral ID (09H)	Manufacture Code	Serial Number	Model Number	Software Version
Z1	Z2-Z4	Z5-Z16	Z17-Z28	Z29-Z30

Level 01 / 02 / 03 Readers (If VMC indicates Level 01 or 02)

Peripheral ID (09H)	Manufacture Code	Serial Number	Model Number	Software Version	Optional Feature Bits
Z1	Z2-Z4	Z5-Z16	Z17-Z28	Z29-Z30	Z31-Z34

Level 03 Readers (If VMC indicates Level 03)
See paragraph 7.4.4 for a detailed explanation of this response.

7.4.19 EXPANSION - Read User File (Level 02 Readers)

Obsolete Command – Do not use for new designs!! (Use EXPANSION - Diagnostics)

	Read User File (01H)	Number of User File
Expansion (17H / 67H)	Y1	Y2

Level 02 Readers

Y1= READ USER FILE
The VMC request's the user file. The length of the file is variable with a maximum length of 32 bytes. The contents of the data are defined by the VMC manufacturer. If the payment media reader does support this command it will respond with USER FILE DATA.

Y2= Number of User File.
The File identification number. The number and size of the data files are defined

by the payment media reader manufacturer. The maximum number of user files are FFh.

Reader response:

User Data File (10H) Z1	Number of User File Z2	Length of User File Z3	User Data Z4-Zn

Z1 : USER FILE DATA (only level 02 readers)
The VMC requires user data and has issued a EXPANSION - READ USER FILE to the payment media reader.

Z2 : Number of User File.
The File identification number. The number and size of data files are defined by the payment media reader manufacturer. The maximum number of user files are FFh.

Z3 : Length of user file
The length of the user file. The maximum length of the user file is 32 bytes. If the user file don't exists the length will be set to 00h.

Z4-Zn : Data defined by the VMC manufacturer.

7.4.20 EXPANSION - Write User File (Level 02 Readers)

Obsolete Command – Do not use for new designs!! (Use EXPANSION - Diagnostics)

Expansion (17H / 67H)	Write User File (02H) Y1	Number of User File Y2	Length of User File Y3	User Data Y4-Yn

Y1 : WRITE USER FILE
The VMC request's to write the user file. The length of the file is variable with a maximum length of 32 bytes. The contents of the data are defined by the VMC manufacturer. If the command is supported but the payment media reader is unable to write the payment media (writing problem or data too long) it will respond with MALFUNCTION/ERROR.

Y2 : Number of User File.
The File identification number. The number and size of data files are defined by the payment media reader manufacturer. The maximum number of user files are FFh.

Y3 : Length of user file

The length of the user file. The maximum length of the user file is 32 bytes.

Y4-Yn : Data defined by the VMC manufacturer.

Reader response:

No Data response

7.4.21 EXPANSION - Write Time/Date File (Level 02/03 readers)

Expansion (17H / 67H)	Write Time/ Date File (03H) Y1	Time Date Y2-Y11

Y1 : WRITE TIME/DATE FILE
The VMC requests to write the Time/Date file.

Y2- Y11: Time/Date to synchronize the card reader real time clock. The date bytes are BCD encoded.

Y2 = Years (Range: 00..99)
Y3 = Months (Range: 01..12)
Y4 = Days (Range: 01..31)
Y5 = Hours (Range: 00..23)
Y6 = Minutes (Range: 00..59)
Y7 = Seconds (Range: 00..59)
Y8 = Day of Week (Range: 01..07, Monday = 1..Sunday = 7)
Y9 = Week Number (Range: 01..53)
Y10 = Summertime (Range: 00..01, Summertime = 1)
Y11 = Holiday (Range: 00..01, Holiday = 1)

If any item of the time/date is not supported use FFH instead.

7.4.22 EXPANSION – Enable Options (Level 03 readers)

Expansion (17H / 67H)	Optional Feature Bit Enable (04H) Y1	Optional Feature Bits Y2-Y5

Level 03 Readers

Y1 : OPTIONAL FEATURE BIT ENABLE
The VMC can enable which level 3 features it desires.

Y2 - Y5: Individual expanded feature bits as sent by reader in response to the 17H-00H EXPANSION REQUEST ID command. To enable a feature, a bit is set to one.

Bits should be sent in descending order, i.e. bit 31 is sent first and bit 0 is sent last. **All features are disabled after a reset**.

b0 - File Transport Layer supported
b1 - 0 = 16 bit monetary format, 1 = 32 bit monetary format
b2 - multi currency / multi lingual
b3 - negative vend
b4 - data entry
b5 to b31 not used (should be set to 0)

Note: If 32 bit monetary format (b1) and or multi currency / multi lingual (b2) options are enabled, this condition will be known as **EXPANDED CURRENCY MODE** in the rest of the document.

7.4.23 EXPANSION – FTL REQ TO RCV

Expansion (17H / 67H)	FTL (FAH)	REQ TO RCV
	Y1	Y2-Y6

Level 03 Readers (if File Transport Layer option enabled)

The VMC is requesting to receive data from the reader whose destination address will always be 10H or 60H. Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 : **FTL** REQ TO RCV
The VMC is requesting to receive data from the reader.

Y2 : **FTL** Destination Address (Reader = 10H / 60H as defined in Section 2.6.

Y3 : **FTL** Source Address
The source address of the command as defined in Section 2.6.

Y4 : **FTL** File ID
The type of information desired as defined in Section 2.6.

Y5 : **FTL** Maximum Length
The total number of blocks in the file as defined in Section 2.6.

Y6 : **FTL** Control
Data transfer control information as defined in Section 2.6.

Reader response:

Two responses are possible from the reader, either the SEND BLOCK (1DH) which transmits the initial (or only) part of the data or the RETRY / DENY (1CH). Note that the response can either be immediate or delayed.

FTL (1DH) SEND BLOCK Z1	SEND BLOCK Information Z2-Z34

Z1 : 1DH response which indicates SEND BLOCK

Z2 : Destination address of data as defined in Section 2.6

Z3 : Block # of data as defined in Section 2.6

Z4-Z34: Data (maximum of 31 bytes)

or

FTL (1CH) RETRY / DENY Z1	RETRY / DENY Information Z2-Z4

Z1 : 1CH response which indicates RETRY / DENY

Z2 : Destination address of response as defined in Section 2.6

Z3 : Source address of response (10H / 60H) as defined in Section 2.6

Z4 : Retry delay

7.4.24 EXPANSION – FTL RETRY / DENY

Expansion (17H)	FTL (FBH) Y1	RETRY / DENY Y2-Y4

Level 03 Readers (if File Transport Layer option enabled)

The VMC is retrying, denying, or aborting a data transfer to/from the reader whose destination address will always be 10H or 60H. Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 : **FTL** RETRY / DENY

The VMC is requesting to retry, deny, or abort a data transfer.

Y2 : **FTL** Destination Address (Reader = 10H / 60H)
The destination address of the command as defined in Section 2.6.

Y3 : **FTL** Source Address
The source address of the command as defined in Section 2.6.

Y4 : **FTL** Retry Delay
The time delay required of the sender as defined in Section 2.6.

Reader response:

None

7.4.25 EXPANSION – FTL SEND BLOCK

Expansion (17H / 67H)	FTL (FCH) Y1	SEND BLOCK Y2-Y34

Level 03 Readers (if File Transport Layer option enabled)

The VMC is sending data to the reader whose destination address will always be 10H or 60H. Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 : **FTL** SEND BLOCK
The VMC is requesting to send data.

Y2 : **FTL** Destination Address (Reader = 10H / 60H)
The destination address of the command / data as defined in Section 2.6.

Y3 : **FTL** Block #
The block # of data as defined in Section 2.6

Y4-Y34 **FTL** Data (maximum of 31 bytes)
The actual data block as defined in Section 2.6.

Reader response:

None

7.4.26 EXPANSION – FTL OK TO SEND

Expansion	FTL	OK TO SEND

(17H / 67H)	(FDH) Y1	 Y2-Y3

Level 03 Readers (if File Transport Layer option enabled)

The VMC is indicating that it is OK for the reader to transfer data. The destination address will always be the reader 10H or 60H. Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 : **FTL** OK TO SEND
The VMC is indicating it is OK to send data.

Y2 : **FTL** Destination Address (Reader = 10H / 60H)
The destination address of the command / data as defined in Section 2.6.

Y3 : **FTL** Source Address
The source address of the command as defined in Section 2.6.

Reader response:

One response is possible from the reader which transmits the initial (or only) part of the data.

Note that the response can either be immediate or delayed.

FTL (1DH) SEND BLOCK Z1	SEND BLOCK Information Z2-Z34

Z1 : 1DH response which indicates SEND BLOCK

Z2 : Destination address of data as defined in Section 2.6

Z3 : Block # of data as defined in Section 2.6

Z4-Z34: Data (maximum of 31 bytes)

7.4.27 EXPANSION – FTL REQ TO SEND

Expansion (17H / 67H)	FTL (FEH) Y1	REQ TO SEND Y2-Y6

Level 03 Readers (if File Transport Layer option enabled)

The VMC is requesting to send data to the reader whose destination address will always be 10H or 60H. Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 : **FTL** REQ TO SEND
The VMC is requesting to send data to the reader.

Y2 : **FTL** Destination Address (Reader = 10H / 60H)
The destination address of the command as defined in Section 2.6.

Y3 : **FTL** Source Address
The source address of the command as defined in Section 2.6.

Y4 : **FTL** File ID
The type of information desired as defined in Section 2.6.

Y5 : **FTL** Maximum Length
The total number of blocks in the file as defined in Section 2.6.

Y6 : **FTL** Control
Data transfer control information as defined in Section 2.6.

Reader response:

Two responses are possible from the reader, either the OK TO SEND (1EH) which allows the data transfer to start or the RETRY / DENY (1CH). Note that the response can either be immediate or delayed.

FTL (1EH) OK TO SEND Z1	OK TO SEND Information Z2-Z3

Z1 : 1EH response which indicates OK TO SEND

Z2 : Destination address of response as defined in Section 2.6

Z3 : Source address of response (10H / 60H) as defined in Section 2.6

or

FTL (1CH) RETRY / DENY Z1	RETRY / DENY Information Z2-Z4

Z1 : 1CH response which indicates RETRY / DENY

Z2 : Destination address of response as defined in Section 2.6

Z3 : Source address of response (10H / 60H) as defined in Section 2.6

Z4 : Retry delay

7.4.28 EXPANSION - Diagnostics

Expansion (17H / 67H)	Diagnostics (FFH) Y1	User Defined Data Y2-Yn

Y1 : DIAGNOSTICS.
Device manufacturer specific instruction for implementing various manufacturing or test modes.

Y2-Yn : User Defined Data.
The data portion of this command is defined by the manufacturer and is not part of this document.

Reader response:

Diagnostics Response (FFH) Z1	User Defined Z2-Zn

Z1 : DIAGNOSTICS RESPONSE

Z2-Zn : User Defined Data.
The data portion of this response is defined by the manufacturer and is not part of this document.

7.5 Cashless Device Non-Response Time

The default maximum non-response time for a cashless device is 5 seconds. This is the maximum time for which a cashless device will not respond to a command or a POLL with ACK, NAK or a message. The "Application Maximum Response Time" reported in byte Z7 of the Reader Configuration Data (7.4.2) supersedes this default value if Z7 is greater.

7.6 Cashless Device Power Requirements

The current draw for any cashless device must fall within the following limits. All measurements are at the minimum VMC Voltage Output.

Idle mode = 300 mA. (avg.) continuous

Transport or Read/Write cycle = 1.5 A @ 50% maximum duty cycle up to 5 seconds.

7.7 Example Vend Sessions

EXAMPLE VEND SESSION #1
(Valid Single Vend)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
VEND REQUEST	→		
	←	ACK	(Vend)
POLL	→		
	←	VEND APPROVED	
ACK	→		
VEND SUCCESS	→		
	←	ACK	(Session Idle)
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

EXAMPLE VEND SESSION #2
(Valid Multiple Vend)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
VEND REQUEST	→		
	←	ACK	(Vend)
POLL	→		
	←	VEND APPROVED	
ACK	→		
VEND SUCCESS	→		
	←	ACK	(Session Idle)
VEND REQUEST	→		
	←	ACK	(Vend)
POLL	→		
	←	VEND APPROVED	
ACK	→		
VEND SUCCESS	→		
	←	ACK	(Session Idle)
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

EXAMPLE VEND SESSION #3
(Session cancelled by user with reader return button)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
	User pushes reader RETURN button		
POLL	→		
	←	SESSION CANCEL	
ACK	→		
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

EXAMPLE VEND SESSION #4a
(Session cancelled by user via coin mechanism escrow return button before product was selected)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
	User pushes coin mech. escrow return		
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

EXAMPLE VEND SESSION #4b
(Session cancelled by user via coin mechanism escrow return button after product was selected)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
VEND REQUEST	→		
	←	ACK	(Vend)

User pushes coin mech. escrow return

CANCEL VEND	→		
	←	ACK	
POLL	→		
	←	VEND DENIED	(Session Idle)
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

EXAMPLE VEND SESSION #5
(VMC Failure/product not dispensed
Refund positive)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
VEND REQUEST	→		
	←	ACK	(Vend)
Reader deducts purchase price from payment media			
POLL	→		
	←	VEND APPROVED	
VMC fails to dispense product			
VEND FAILURE	→		
	←	ACK	
POLL	→		
	←	Silence during the refund operation	
POLL	→		
	←	ACK	C
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

EXAMPLE VEND SESSION #5A
(VMC Failure/product not dispensed
Refund fail)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
VEND REQUEST	→		
	←	ACK	(Vend)
Reader deducts purchase price from payment media			
POLL	→		
	←	VEND APPROVED	
VMC fails to dispense product			
VEND FAILURE	→		
	←	ACK	
POLL	→		
	←	Silence during the refund operation	
POLL	→		
	←	MALFUNCTION ERROR code	
		1100yyyy=refund fail	(Level 02 / 03)
		ACK	(Level 01)
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

EXAMPLE VEND SESSION #6
(Vend denied by reader)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
VEND REQUEST	→		
	←	ACK	(Vend)
Insufficient funds or payment media/account error			
POLL	→		
	←	VEND DENIED	(Session Idle)
VMC makes no attempt to dispense product			
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

EXAMPLE VEND SESSION #7
(Command Out of Sequence Error)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
VEND REQUEST	→		
	←	ACK	(Vend)
EXPANSION ID REQUEST	→		
	←	ACK	
POLL	→		
	←	COMMAND OUT OF SEQUENCE	(Session Idle)
ACK	→		
RESET	→	{**Mandatory**}	
	←	ACK	
			(Inactive)

EXAMPLE VEND SESSION #8a
(Reader busy for longer than max. non response time)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
VEND REQUEST	→		
	←	ACK	(Vend)
POLL	→		
		[silence…]	(Reader busy)
POLLs (numerous)	→		
	←	[silence…]	(continued POLLs w/ no response)
POLL	→		
	←	ACK	(restart Non-Response timer)
POLLs (numerous)	→		
	←	[silence…]	(continued POLLs w/ no response)
POLL	→		
	←	[silence…]	(Reader almost finished)
POLL	→		
	←	VEND APPROVED	(Reader ready)
ACK	→		
VEND SUCCESS	→		
	←	ACK	(Session Idle)
VEND REQUEST	→		
	←	ACK	(Vend)
POLL	→		
	←	VEND APPROVED	
ACK	→		
VEND SUCCESS	→		
	←	ACK	(Session Idle)
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

EXAMPLE VEND SESSION #8b
(Reader busy for shorter than max. non response time)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
VEND REQUEST	→		
	←	ACK	(Vend)
POLL	→		
	←	[silence...]	(Reader busy)
POLLs (numerous)	→		
	←	[silence...]	(Continued POLLs w/ no response)
POLL	→		
	←	[silence...]	(Reader almost finished)
POLL	→		
	←	VEND APPROVED	(Reader ready)
ACK	→		
VEND SUCCESS	→		
	←	ACK	(Session Idle)
VEND REQUEST	→		
	←	ACK	(Vend)
POLL	→		
	←	VEND APPROVED	
ACK	→		
VEND SUCCESS	→		
	←	ACK	(Session Idle)
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

NOTE

If the peripheral omits to respond within the maximum non-response time, it is considered to be off-line.

EXAMPLE VEND SESSION #8c
(No Response, Reader busy at Vend Request.)

Controller		Cashless Device	State/ Comment
POLL	→		
	←	BEGIN SESSION	
ACK	→		
VEND REQUEST	→		
	←	[silence…]	Reader busy. The reader may not send the response within the t-response(max) timeout or hasn't received the command completely due to line breakdown
VEND REQUEST	→		VMC repeats the command: As the VMC isn't sure, that the slave has received the command free of errors it repeats it. The command itself is not yet performed by the reader as long the ACK hasn't been sent.
	←	[silence…]	Reader busy
VEND REQUEST	→		
	←	ACK	(Vend) The reader will now perform the command. The response isn't available at the moment, thus the VEND REQUEST is only acked
POLL (numerous)	→		VMC polls the reader to obtain the data in VEND APPROVED
	←	ACK	The reader may send a ACK or [silence] to each POLL
POLL	→		
	←	VEND APPROVED	The response to the VEND REQUEST is now available. It must be sent within the time defined by the APPLICATION MAXIMUM RESPONSE TIME. This is measured from the ACK following the VEND REQUEST.
ACK	→		

EXAMPLE VEND SESSION #9
(Pre-approved authorization aborted by coin mechanism escrow return button before BEGIN SESSION)

Controller		Cashless Device	State
User swipes payment media			
			(Enabled)
POLL	→		
	←	ACK	
READER CANCEL	→		
	←	ACK	
(If applicable, reader aborts HOST communications, ejects payment media, etc...)			
POLL	→		
	←	CANCELLED	

EXAMPLE VEND SESSION #10
(Single Negative Vend)

Controller		Cashless Device	State
POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		
User inserted a payment media, and inserted then a product, which was detected valid, or pressed a selection button to identify the desired product which will be inserted later on			
NEGATIVE VEND REQUEST	→		
	←	ACK	(Vend)
POLL	→		
	←	VEND APPROVED	
The payment reader is able to add the desired value to the credit			
ACK	→		
The product is now fully accepted from the machine or the user has finally finished insertion of a valid product			
VEND SUCCESS	→		
	←	ACK	(Session Idle)
The payment media reader has added the credit			
SESSION COMPLETE	→		
	←	ACK	
POLL	→		
	←	END SESSION	(Enabled)

Normally, can or bottle return-vendors may check the product first, before the patron inserts his card. It is up to the VMC, to delay the negative vend request, until the session idle state is reached. In many return-vendors, from this state, the product is already fully accepted. Therefore, there is no need for the further sequences, this means, vend accepted, vend success will follow each other immediately.

If the payment media reader is not able to update the credit, there will be two conditions:

- The return vendor is able to escrow the product after the vend denied. In this case the session complete is sent, the product is return and the credit remains unchanged.
- The return vendor is not able to escrow the product after vend denied. In this case, session complete should be sent and there should be an update credit within the system (VMC), which could be returned by other means (i.e. return coins, tokens, etc).

If a return vendor is able to escrow the product again, this vendor normally accepts the product finally only a vend accepted was sent. In this case there may happen some fault condition which allows no final acceptance of the product. The return vendor then closes the session with vend failed instead of vend success, indicating to the reader not to update the system credit, or, if the payment media is no longer present, request re-insertion of the media.

EXAMPLE DATA ENTRY SESSION #1
(Three key Data Entry w/ Prompt & Asterisks for Entries)

Controller		Cashless Device	State
		Previously Enabled	Enabled
POLL	→		
	←	DATA ENTRY REQUEST + DISPLAY REQUEST (prompt)	
ACK	→		
		User pushes Selection Key 1	
DATA ENTRY RESPONSE (Key 1)	→		
	←	ACK	
POLL	→		
	←	DISPLAY REQUEST (prompt + *)	
ACK	→		
		User pushes Selection Key 2	
DATA ENTRY RESPONSE (Key 2)	→		
	←	ACK	
POLL	→		
	←	DISPLAY REQUEST (prompt + **)	
ACK	→		
		User pushes Selection Key 3	
DATA ENTRY RESPONSE (Key 3)	→		
	←	ACK	(Enabled)
POLL	→		
	←	DISPLAY REQUEST (prompt + *** or "Entry OK")	
ACK	→		

Note: After Display Request Time expires, VMC regains control of display

POLL	→		
	←	BEGIN SESSION	(Session Idle)
ACK	→		

EXAMPLE DATA ENTRY SESSION #2
(Data Entry with Reader Cancel)

Controller		Cashless Device	State
		Previously Enabled	Enabled
POLL	→		
	←	DATA ENTRY REQUEST + DISPLAY REQUEST (prompt)	
ACK	→		
		User pushes (valid) Selection Key	
DATA ENTRY RESPONSE (Key 1)	→		
	←	ACK	
POLL	→		
	←	DISPLAY REQUEST (prompt + *)	
ACK	→		
		User pushes (invalid) Selection Key	
DATA ENTRY RESPONSE (Key 2)	→		
	←	ACK	
POLL	→		
	←	DATA ENTRY CANCEL	
ACK	→		(Enabled)
POLL	→		
	←	DISPLAY REQUEST ("Error")	
ACK	→		
		After VMC Display Request Time expires, regains control of display	

Note that the above scenario is only an example and it may not be prudent to cancel a session after the first wrong entry. (Someone could fraudulently obtain a password by trying the maximum of selection keys at each position.)

EXAMPLE DATA ENTRY SESSION #3
(Data Entry with VMC Cancel)

Controller		Cashless Device	State
		Previously Enabled	Enabled
POLL	→		
	←	DATA ENTRY REQUEST + DISPLAY REQUEST (prompt)	
ACK	→		
		User pushes Selection Key	
DATA ENTRY RESPONSE (Key 1)	→		
	←	ACK	
POLL	→		
	←	DISPLAY REQUEST (prompt + *)	
ACK	→		
		User walks away & VMC times out	
DATA ENTRY RESPONSE (FF's)	→		
	←	ACK	(Enabled)
POLL	→		
	←	DISPLAY REQUEST ("Try Again")	
ACK	→		
		After VMC Display Request Time expires, regains control of display	

(this page intentionally left blank)

Section 8
Communications Gateway VMC/Peripheral Communication Specifications

8.1 Introduction

This section defines the communications bytes sent and received between a Communications Gateway (Comms Gateway) and the VMC. The Comms Gateway address is 00011xxxB (18H).

Unless otherwise stated, all information is assumed to be in a binary format.

After the VMC has issued a command, the Comms Gateway must respond with a reply. The reply may be an ACK or a detailed message response. If the command format expects a response, the Comms Gateway may: 1) respond with an ACK, to acknowledge receiving the command, and send the response later as a response to a POLL, or 2) immediately respond with the expected message.

The Comms Gateway response to a command from the VMC may be an ACK, a single message, or if there is more data to send it may be a multi message reply, up to the MDB maximum of 36 bytes.

The following command / response set has been defined to provide a means to transfer vending information system data from the VMC to the Comms Gateway in one of two ways;

1) Entire DTS files (including DXS, ST, SD1, G85, SE, and DXE records) are transferred using the file transport layer (FTL) of MDB.

2) Activity “Reports” are sent from the VMC to the Comms Gateway every time something happens in the vending system, it is then the Comms Gateways responsibility to store and assemble the DTS file. (DXS, ST, SD1, G85, SE and DXE data are not sent.) Obviously, a combination of these two methods can be designed to meet specific needs also.

8.2 VMC Commands

VMC Cmd	Code	VMC Data	Comm Gateway response
RESET	18H		00H - Just RESET (1)
SETUP	19H	Feature level (1) Scale factor (1) Decimal places (1)	01H - Comms Gateway Config (1) Feature level (1) Max. App. Resp. (2)
POLL	1AH		00H - Just RESET (1)
			01H - Comms Gateway Config (1) Feature level (1) Max. App. Resp. (2)
			02H - Request transmit (1)
			03H - Data transmitted (1)
			04H - Error (1) Error code (n)
			05H - DTS Event Acknowledge (1)
			06H - Peripheral ID: (1) Mfg. code (3) Serial number (12) Model number (12) Software ver. (2) Opt. features (4)
			07H - Radio Signal Strength (2)
			1BH - FTL REQ to RCV (option) (1)
			1CH - FTL RETRY / DENY (option) (1)
			1DH - FTL SEND BLOCK (option) (1)
			1EH - FTL OK to SEND (option) (1)
			1FH - FTL REQ to SEND (option) (1)
			FFH - Diagnostics (n)
REPORT	1BH	Type = 01, Transaction (1) Transaction Type (1) Selection (Row/Col.) (2) Price (2) Cash in, Coin tubes (2) Cash in, Cashbox (2) Cash in, Bills (2) Value in, Cashless #1 (2) Value in, Cashless #2 (2) Revalue to Cashless #1 (2) Revalue to Cashless #2 (2) Cash out (2) Discount Amount (2) Surcharge Amount (2) User Group # (1) Price List (1) Date (4) Time (2)	

		Type = 02, DTS Event (1) DTS Event Code (10) Date (4) Time (2) Duration (4) Activity (1) Terminal ID (12)	05-DTS Event Acknowledge (1)
		Type = 03, Asset ID (1) Asset Type = 0n (1) Manufacture Code (3) Serial Number (12) Model Number (12) Software Version (2) Type = 03, Asset ID (1) Asset Type = 8n (1) Asset Number (20)	
		Type = 04, Currency ID (1) VMC Currency Code (2) VMC Currency (1) VMC Decimal Point (1)	
		Type = 05, Product ID (1) Product Identification (20) Selection Presence (1)	
CONTROL	1CH	00H - Disable (1)	
		01H - Enable (1)	
		02H - Transmit (1)	
EXPANSION	1FH	00H - Identification	06H - Peripheral ID: (1) Mfg. code (3) Serial number (12) Model number (12) Software ver. (2) Opt. features (4)
		01H - Feature enable (1) Features enabled (4)	
		FAH - FTL (option) REQ TO RCV	1DH - SEND BLOCK 1CH - RETRY / DENY
		FBH - FTL (option) RETRY / DENY	No Data
		FCH - FTL (option) SEND BLOCK	No Data
		FDH - FTL (option) OK TO SEND	1DH - SEND BLOCK
		FEH - FTL (option) REQ TO SEND	1EH - OK TO SEND 1CH - RETRY/DENY
		FFH - Diagnostics (n)	FFH - Diagnostics (n)

8.3 Communications Gateway Command Format

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
RESET	18H	No data	Z1

This command is the vehicle that the VMC should use to tell the Comms Gateway that it should perform its initialization procedure. With the exception of the ACK response, it should abort all communication and revert to the internally stored operational parameters.

Z1 = 00 JUST RESET

Indicates the Comms Gateway has been reset internally or on command from the VMC.

The following initialization sequence is recommended. It should be used after "power up", after issuing the RESET command, or after issuing the Bus Reset (pulling the transmit line "active" for a minimum of 100 mS).

POLL – 18H
To obtain "JUST RESET" response

SETUP – 19H
To obtain Comms Gateway level and configuration information

EXPANSION IDENTIFICATION – 1F 00H
To obtain additional identification information and options

EXPANSION FEATURE ENABLE – 1F 01H
To enable desired options

CONTROL / ENABLE – 1CH / 01H
To enable / alert the Comms Gateway to start collecting data and / or monitoring for REPORT commands situations.

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
SETUP	19H	Y1 - Y3	Z1 - Z4

Y1 = VMC feature level

Indicates the highest Comms Gateway feature level that the VMC supports. Currently the highest feature level is 03, with no requirement to support previous (obsolete) levels 1 and 2.)

Y2 = Scale factor

The multiplier used to scale all monetary values transferred between the VMC and the Comms Gateway.

Y3 = Decimal places

The number of decimal places used to communicate monetary values between the VMC and the Comms Gateway.

Z1 = 01 COMMS GATEWAY CONFIGURATION

The Comms Gateway is responding to a SETUP command. This response includes the following data;

Z2 = Comms Gateway feature level

The feature level of the Comms Gateway. Currently the highest feature level is 03, with no requirement to support previous (obsolete) levels 1 and 2.)

Z3 - Z4 = Application maximum response time

The maximum length of time, in seconds, that an Comms Gateway may be unable to respond to any commands. This includes the time communicating over an external network. The VMC should continue POLLing the Comms Gateway during this time in an attempt to re-synchronize communications earlier. When the Comms Gateway is ready to communicate over the bus again, it should respond to the next POLL with COMPLETE (if communicating externally) or ACK. This time essentially replaces the standard MDB non-response time, as such it's default value is equal to the defined non-response time (5 seconds).

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
POLL	1AH	No data	Z1 - Zn

The POLL command is used by the VMC to obtain information from the Comms Gateway. This information may include setup information, activity requests, or error conditions. An ACK response indicates that no error states exist and either no information request is pending or pending information is not yet ready for transmission.

In addition to an ACK, the VMC may receive the following POLL responses from the Comms Gateway.

Z1 = 00 JUST RESET

Indicates the Comms Gateway has been reset internally or on command from the VMC.

Z1 = 01 COMMS GATEWAY CONFIGURATION

The Comms Gateway is responding to a SETUP command. This response includes the following data;

Z2 = Comms Gateway feature level

The feature level of the Comms Gateway. Currently the highest feature level is 03, with no requirement to support previous (obsolete) levels 1 and 2.)

Z3 - Z4 = Application maximum response time

The maximum length of time, in seconds, that an Comms Gateway may be unable to respond to any commands. This includes the time communicating over an external network. The VMC should continue POLLing the Comms Gateway during this time in an attempt to re-synchronize communications earlier. When the Comms Gateway is ready to communicate over the bus again, it should respond to the next POLL with COMPLETE (if communicating externally) or ACK. This time essentially replaces the standard MDB non-response time, as such it's default value is equal to the defined non-response time (5 seconds).

Z1 = 02 REQUEST TO TRANSMIT

The Comms Gateway is requesting permission to transmit data to an external collection device. This is done to control the bus power supply. The Comms Gateway should continue sending this response to each POLL until permission to transmit has been granted or the need to transmit goes away.

Z1 = 03 DATA TRANSMITTED

The Comms Gateway is finished transmitting to an external collect device.

Z1 = 04 ERROR

The Comms Gateway has developed some type of detectable error. The error codes will be sent continuously, or until the error is resolved.

Z2 – Zn = Error code

The error codes are ASCII strings taken from the EVA DTS Communications fault list.

Z1 = 05 DTS EVENT ACKNOWLEDGE

The Comms Gateway has recognized that a DTS Event has occurred and must act accordingly. The specific actions will be defined by the Comms Gateway operational specifications.

Z1 = 06H PERIPHERAL ID

Comms Gateway is sending peripheral ID information. This response includes the following data;

Z2 - Z4 = Manufacturer code

Identification code for the equipment supplier. Sent as ASCII characters. Blanks (20H) are acceptable.

Z5 - Z16 = Serial number

Factory assigned serial number sent as numeric ASCII characters. All bytes must be sent. Zeros (30H) and blanks (20H) are acceptable.

Z17 - Z28 = Model number ASCII.

Manufacturer assigned model number sent as ASCII characters. All bytes must be sent. Zeros (30H) and blanks (20H) are acceptable.

Z29 - Z30 = Software version

Current software version sent as packed BCD.

Z31 - Z34 = Optional Features

Each of the 32 bits indicate an optional features availability. If the bit is set the feature is available. Currently defined options are:

b0: File transport layer support
b1: Verbose mode: See REPORT command
b2 - b31: Future use, must be set to 0.

Z1 = 07H RADIO SIGNAL STRENGTH

The Comms Gateway is reporting its signal strength from the network. This response includes the following data;

Z2 = Signal Strength

The level of radio signal strength detected by the Comms Gateway. This is a binary number from 00H to 64H (100%) representing the percentage of expected signal. This can be sent after every POLL, or as needed due to changes in the signal.

Note that all FTL responses below are defined in Section 2.6. For the Comms Gateway, the source address will always be the Comms Gateway (18H) as defined in Section 2.3.

Z1 = 1BH REQ TO RCV (File Transport Layer)

The Comms Gateway is requesting to receive data from a device or VMC.

Z2 = Destination address of response
Z3 = Source address of response (18H)
Z4 = File ID
Z5 = Maximum length
Z6 = Control

Z1 = 1CH RETRY/DENY (File Transport Layer)

The Comms Gateway is requesting a device or VMC to retry or deny the last FTL command.

Z2 = Destination address of response
Z3 = Source address of response (18H)
Z4 = Retry delay

Z1 = 1DH SEND BLOCK (File Transport Layer)

The Comms Gateway is sending a block of data (maximum of 31 bytes) to a device or VMC.

Z2 = Destination address of data
Z3 = Block #
Z4-Z34 = Data (maximum of 31 bytes)

Z1 = 1EH OK TO SEND (File Transport Layer)

The Comms Gateway is indicating that it is OK for a device or VMC to send it data.

Z2 = Destination address of response
Z3 = Source address of response (18H)

Z1 = 1F REQ TO SEND (File Transport Layer)

The Comms Gateway is requesting to send data to a device or VMC.

Z2 = Destination address of response
Z3 = Source address of response (18H)
Z4 = File ID
Z5 = Maximum length
Z6 = Control

Z1 = FFH DIAGNOSTICS

The Comms Gateway is responding to a EXPANSION/DIAGNOSTICS command. This response includes the following data;

Z2 - Zn User defined data

Device manufacturer specific responses after receiving manufacturing or test instructions. Z1 - Zn implies that any number of bytes may be used for the response data from the Comms Gateway.

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
REPORT	1BH	Y1 – Ynn	No data

The REPORT command is used by the VMC to pass activity information to the Comms Gateway. If the "Verbose mode" is enabled via the EXPANSION / FEATURE ENABLE command, this command must be sent immediately following the completion of any activity it is describing. The activities may include; a transaction, a DTS defined event, an asset identification, currency identification, or product identification.

The intent of this command is to provide information so that the Comms Gateway can create a Data Transfer Standard file. All of the following fields show their corresponding DTS fields for reference, for further detail refer to the Data Transfer Standard.

If the "Verbose mode" is disabled, only the "DTS Event" report type records must be sent. This mode uses the FTL to transfer the complete DTS files and the DTS Event report types to alert the VMC of any alarm conditions.

Since reports data may vary, any field that is not relevant, or not known, should be populated with 00H's. All cash values are scaled and decimal adjusted using the data provided in the SETUP command.

Y1 = Type: The type of activity that is being reported, includes one of the following:

01H	Transaction
02H	DTS Event
03H	Asset ID
04H	Currency ID
05H	Product ID

If Y1 = 01H then the following "Transaction" data fields have been identified to be included:

Y2 = Transaction Type

This field defines the type of transaction that the following data describes. The defined transaction types include;

01H	Paid Vend
02H	Token Vend
03H	Free Vend
04H	Test Vend
05H	Revalue
06H	Negative Vend
07H	Vendless*
08H	Manual / Service

* The end of a "Vendless" transaction is defined by the VMC manufacturer, for example an escrow request, a failed vend, etc.

Y3 – Y4 = Item Number

This is the binary field used to link REPORT type 01 to REPORT type 05. It is an item number, 0000H through FFFFH of the selected product involved in the most recent transaction. This number is defined by the manufacturer.

Y5 – Y6 = Price (PA102)

The established price of the product involved in the most recent transaction. The established price is the price before any adjustments i.e. discounts surcharges, etc.

Y7 – Y8 = Cash in, Coin Tubes (CA303/CA307 or CA1001/CA1002)

The value of cash deposited into the coin tubes since the completion of the previous transaction.

Y9 – Y10 = Cash in, Cashbox (CA302/CA306)

The value of cash deposited into the cashbox since the completion of the previous transaction.

Y11 – Y12 = Cash in, Bills (CA304/CA308)

The value of cash deposited into the bill stacker since the completion of the previous transactions.

Y13 – Y14 = Value in, Cashless Device #1 (DA201/DA203)

The value removed from the media in cashless device #1 since the completion of the previous transaction.

Y15 – Y16 = Value in, Cashless Device #2 (DB201/DB203)

The value removed from the media in cashless device #2 since the completion of the previous transaction.

Y17 – Y18 = Revalue to Cashless Device #1 (DA401/DA402)

The value returned to the media in cashless device #1 since the completion of the previous transaction.

Y19 – Y20 = Revalue to Cashless Device #2 (DB401/DB402)

The value returned to the media in cashless device #2 since the completion of the previous transaction.

Y21 – Y22 = Cash out (CA401/CA403 or CA402/CA404)

The total value of the cash dispensed from the system since the completion of the previous transaction.

Y23 – Y24 = Discount Amount (CA701/CA702)

The value of any discounts awarded since the completion of the previous vend.

Y25 – Y26 = Surcharge Amount (CA705/CA706)

The value of any surcharges collected since the completion of the previous vend.

Y27 = User Group # (DA701 or DB701)

The user group number that the transaction is associated with.

Y28 = Price List (LA101)

The price list that the transaction is associated with

Y29 – Y32 = Date (PA501)

The date of the transaction. This data is sent as BCD in the following sequence YYYY/MM/DD. For example, 17 March 2002 would be 20H 02H 03H 17H. If the date is not known these bytes are filled with 99Hs.

Y33 – Y34 = Time (PA502)

The time of the transaction. This data is sent as BCD , 24 hour format, in the following sequence HHMM. For example, 6:30 PM would be 18H 30H. If the time is not known these bytes are filled with 99Hs.

If Y1 = 02H then the following "DTS Event" data fields have been identified to be included:

Y2 – Y11 = DTS Event Code (EA101 or EA201 or EA701)

This is an alpha-numeric ASCII code defining the event being reported. The codes are list in the EVA DTS manual. In addition to the standard DTS event codes, an interrogation event is reported as "EA3" and a power outage event is reported as "EA7".

Y12 – Y15 = Date (EA102)

The date of the event. This data is sent as BCD in the following sequence YYYY/MM/DD. For example, 17 March 2002 would be 20H 02H 03H 17H. If the date is not known these bytes are filled with 99Hs.

Y16 – Y17 = Time (EA103)

The time of the event. This data is sent as BCD in the following sequence HH/MM. For example, 6:30 PM would be 18H 30H. If the time is not known these bytes are filled with 99Hs.

Y18 – Y21 = Duration (EA206)

The duration of the event in total minutes. This data is sent as binary. For example, 4 hours and 15 minutes would be 00H 00H 00H FFH.

Y22 = Activity (EA205)

The current status of the events activity. This field is equal to 00H if the event is inactive (or not reset for "EA3") or 01H if the event is active (or reset for "EA3").

Z1 = 05 DTS EVENT ACKNOWLEDGE

The Comms Gateway has recognized that a possible alarm situation has occurred and must act accordingly. The specific actions will be defined by the Comms Gateway operational specifications.

If Y1 = 03H then the following "Asset ID" data fields have been identified to be included:

Y2 = Asset Type

The following code pairs have been defined to represent the type of equipment asset that is being communicated.

Code	Equipment type	DTS header (αα)
01H / 81H	Audit Module / Data Carrier (DC) Identification	AM1
02H / 82H	Bill Validator Identification	BA1
03H / 83H	Changer Identification	CA1
04H / 84H	Control Board Identification	CB1
05H / 85H	Cashless #1 Identification	DA1
06H / 86H	Cashless #2 Identification	DB1
07H / 87H	Machine Identification	ID1

If Y2 has the MSB = 0 (i.e. Y2 = 01H) then the following asset data fields have been identified to be included:

Y3 – Y5 = Manufacturer code (αα101, first 3 characters)

Identification code for the equipment supplier. Sent as ASCII characters. Blanks (20H) are acceptable.

Y6 - Y17 = Serial number ($\alpha\alpha$101, 4th through 15th characters)

Factory assigned serial number sent as numeric ASCII characters. All bytes must be sent. Zeros (30H) and blanks (20H) are acceptable.

Y18 - Y29 = Model number ($\alpha\alpha$102)

Manufacturer assigned model number sent as ASCII characters. All bytes must be sent. Zeros (30H) and blanks (20H) are acceptable.

Y30 - Y31 = Software version (or Build Standard) ($\alpha\alpha$103)

Current software version sent as packed BCD.

If Y2 has the MSB = 1 (i.e. Y2 = 81H) then the following asset data fields have been identified to be included:

Y2 – Y21 = Asset Number ($\alpha\alpha$105 or $\alpha\alpha$106)

The asset number of the equipment. This is a reference number used for tracking purposes, separate from the serial number. It is usually programmed by the equipment operator.

If Y1 = 04H then the following "Currency ID" data fields have been identified to be included:

Y2 – Y3 = VMC's Country / Currency Code (ID402)

The packed BCD Country / Currency code of the VMC can be sent in two different forms depending on the value of the left most BCD digit.

If the left most digit is a 0, the International Telephone Code is used to indicate the country that the changer is set-up for. For example, the USA code is 00 01H (Z2 = 00 and Z3 = 01).

If the left most digit is a 1, the latest version of the ISO 4217 numeric currency code is used. For example, the code for the US dollar is 18 40H (Z2 = 18 and Z3 = 40) and for the Euro is 1978 (Z2 = 19 and Z3 = 78).

All new designs after July, 2000 must use the ISO 4217 numeric currency codes.

Y4 = VMC's Coin Scaling Factor / Currency Description (ID403)

The multiplier used to scale all monetary values transferred between the VMC and the vending machines monetary system.

Y5 = VMC's Decimal Point (ID401)

The number of digits to the right of the decimal point. This field is used in countries whose currency requires a number of digits to the right of the decimal point other than 2.

If Y1 = 05H then the following "Product ID" data fields have been identified to be included:

Y2 – Y3 = Item Number

This is the binary field used to link REPORT type 01 to REPORT type 05. This number is defined by the manufacturer.

Y4 – Y9 = Product Number (PA101)

This is the ASCII representation of the Item Number that should be included in the DTS file. All bytes must be sent, leading blanks (20H) are acceptable.

Y10 – Y29 = Product Identification (PA103)

The ASCII product identification that should identify the product itself, as in a name (chips/crisps) or an ID number / bar code. All bytes must be sent, leading blanks (20H) are acceptable.

Y30 = Selection Presence Status (PA107)

This field is set to 00H if a vend mechanism (motor, solenoid, etc.) is present for this selection. This field is set to 01H if a vend mechanism is not present.

An example of a 01H being sent would be if the vend mechanism was present previously, and something occurred so that it is not being currently detected (i.e., removed, broken wire, etc.). It is **not** intended to indicate that a product is not available for vending (i.e., sold out).

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
CONTROL	1CH	Y1	No data

This command is the vehicle that the VMC uses to control the Comms Gateway's use of an external collection device. For example when it should, or should not, transmit through the external collection device. The information is identified by one of the following subcommands;

Y1 = 00 Disabled

No external transmissions will be granted and no REPORT commands will be sent.

Y1 = 01 Enabled

External transmissions may be requested and REPORT commands will be sent.

Y1 = 02 Transmit

Permission to transmit and / or receive data is granted, or a transmission session is requested. A DATA TRANSMITTED response to a POLL must be sent when the transmission session is complete.

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
EXPANSION/ IDENTIFICATION	1FH/00H	Y1	Z1 - Z34

Y1 = 00H IDENTIFICATION subcommand

The VMC is requesting Comms Gateway identification information for asset tracking and optional feature purposes.

Z1 = 06H PERIPHERAL ID

Comms Gateway is sending peripheral ID information. This response includes the following data;

Z2 - Z4 = Manufacturer code

Identification code for the equipment supplier. Sent as ASCII characters. Blanks (20H) are acceptable.

Z5 - Z16 = Serial number

Factory assigned serial number sent as numeric ASCII characters. All bytes must be sent. Zeros (30H) and blanks (20H) are acceptable.

Z17 - Z28 = Model number ASCII.

Manufacturer assigned model number sent as ASCII characters. All bytes must be sent. Zeros (30H) and blanks (20H) are acceptable.

Z29 - Z30 = Software version

Current software version sent as packed BCD.

Z31 - Z34 = Optional Features

Each of the 32 bits indicate an optional features availability. If the bit is set the feature is available. Currently defined options include:

b0:	File transport layer support.
b1:	Verbose mode: See REPORT command
b2 - b31:	Future use, must be set to 0.

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
EXPANSION/ FEATURE ENABLE	1FH/01H	Y1 - Y5	No data

Y1 = 01H FEATURE ENABLE subcommand

This command is used to enable each of the optional features defined in Z32-Z35 of the PERIPHERAL ID response. The VMC should send the EXPANSION /IDENTIFICATION command, receive the PERIPHERAL ID response, perform a logical OR with the optional features it wants to enable, and return the resulting enabled features back to the Comms Gateway by setting a bit to 1 for each respective optional feature enabled. All optional features are disabled after reset.

Y2 - Y5 = Optional features enabled

Each of the 32 bits indicates an optional features state. If the bit is set the feature is enabled.

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
EXPANSION COMMAND	0FH FAH **FTL** REQ TO RCV	Y1-Y5	Z1 - Zn (immediate or POLLed)

The VMC is requesting to receive data from the Comms Gateway whose destination address will always be (18H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (18H)
Y2 = Source address of command
Y3 = File ID
Y4 = Maximum length
Y5 = Control

Z1 = 1DH which indicates SEND BLOCK
Z2 = Destination address of data
Z3 = Block #
Z4 - Z34 = Data (maximum of 31 bytes)

or

Z1 = 1CH which indicates RETRY / DENY
Z2 = Destination address of response
Z3 = Source address of response (18H)
Z4 = Retry delay

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
EXPANSION COMMAND	0FH FBH **FTL** RETRY / DENY	Y1-Y3	None

The VMC is retrying, denying, or aborting a data transfer to/from the Comms Gateway whose destination address will always be (18H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (18H)
Y2 = Source address of command
Y3 = Retry delay

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
EXPANSION COMMAND	0FH FCH **FTL** SEND BLOCK	Y1-Y33	None

The VMC is sending data to the Comms Gateway whose destination address will always be (18H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command & data (18H)
Y2 = Block #
Y3 - Y33 = Data (maximum of 31 bytes)

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
EXPANSION COMMAND	0FH FDH **FTL** OK TO SEND	Y1-Y2	Z1-Z34 (immediate or POLLed)

The VMC is indicating that it is OK for the Comms Gateway to transfer data. The destination address will always be the Comms Gateway (18H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (18H)
Y2 = Source address of command

Z1 = 1DH which indicates SEND BLOCK
Z2 = Destination address of data
Z3 = Source address of data
Z4 - Z34 = Data (maximum of 31 bytes)

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
EXPANSION COMMAND	0FH FEH **FTL** REQ TO SEND	Y1-Y5	Z1 - Zn (immediate or POLLed)

The VMC is requesting to send data to the Comms Gateway whose destination address will always be (18H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (18H)
Y2 = Source address of command
Y3 = File ID
Y4 = Maximum length
Y5 = Control

Z1 = 1EH which indicates OK TO SEND
Z2 = Destination address of response
Z3 = Source address of response (18H)

or

Z1 = 1CH which indicates RETRY / DENY
Z2 = Destination address of response
Z3 = Source address of response (18H)
Z4 = Retry delay

VMC Command	Code/Sub-code	VMC Data	Comms Gateway Response
EXPANSION/ DIAGNOSTICS	1FH/FFH	Y1 - Yn	Z1 - Zn

Y1 = FFH DIAGNOSTICS subcommand

Device manufacturer specific instruction for implementing various manufacturing or test modes.

Y2 - Yn = User defined data

The data portion of this command is defined by the manufacturer and is not part of this document.

Z1 = FFH DIAGNOSTICS

The Comms Gateway is responding to a EXPANSION/DIAGNOSTICS command. This response includes the following data;

Z2 - Zn = User defined data

Device manufacturer specific responses after receiving manufacturing or test instructions. Z1 - Zn implies that any number of bytes may be used for the response data from the Comms Gateway.

8.4 Communications Gateway Non-Response Time

The maximum non-response time for a Comms Gateway is 5 seconds. This is the maximum time for which a Comms Gateway will not respond to a command with ACK, NAK, or a data message.

8.5 Communications Gateway Power Requirements

The current draw for any Comms Gateway must fall within the following limits. All measurements are at the minimum VMC Voltage Output.

Idle mode = 300 mA. (avg.) continuous

Active mode = 1.8 A continuous and up to 2.5 A (max) for an accumulated maximum of 10 seconds. The active power mode must be initiated by the REQUEST TO TRANSMIT followed by the CONTROL/TRANSMIT. The active power mode must be closed by sending the DATA TRANSMITTED. During this time the VMC will make its own decisions about which other peripherals will be disabled or not. This may result in the entire machine being disabled for normal vending.

8.6 Communications Gateway Examples

Event	Exchange
Power on Reset at VMC or JUST RESET received by VMC any other time	Reset sequence Enable sequence
Communications Gateway is triggered to send a file	Request sequence Transmit sequence
VMC is triggered to send a file	Dump sequence Transmit sequence
DTS Event situation occurs at VMC	DTS Event sequence Request sequence Transmit sequence
Error situation is detected at Comms Gateway	Error sequence
Every vend completion	Vend sequence

Reset sequence **VMC**		**Comms Gateway**	**Comments**
RESET (18)	→		Reset command
	←	ACK	
POLL (1A)	→		Must be sent once reset, internal or external
	←	JUST RESET (00)	
ACK	→		
SETUP (19…)	→		Establish operation configuration
	←	CONFIG. (01…)	
ACK	→		
EXPANSION/ID (1F/00…)	→		Send asset information
	←	PERIPHERAL ID (06…)	Get asset information
ACK	→		
EXPANSION/FEATURE ENABLE (1F/01…)	→		Enable additional feature if necessary
	←	ACK	
Enable sequence **VMC**		**Comms Gateway**	**Comments**
CONTROL/ENABLE (1C01)	→		Enable command
	←	ACK	
Disable sequence **VMC**		**Comms Gateway**	**Comments**
CONTROL/DISABLE (1C00)	→		Disable command
	←	ACK	

Request sequence VMC		Comms Gateway	Comments
File transfer done		**using the MDB**	**file transport layer**

Dump sequence VMC		Comms Gateway	Comments
File transfer done		**using the MDB**	**file transport layer**

Transmit sequence VMC		Comms Gateway	Comments
POLL (1A)	→		
	←	Request to transmit (02)	
ACK	→		
CONTROL/ TRANSMIT (1C/02)	→		
	←	ACK	
POLL (1A)			
		ACK	
		.	Continue POLLing until …
		.	
POLL (1A)	→		
	←	Data transmitted (03)	
ACK	→		

DTS Event sequence VMC		Comms Gateway	Comments
REPORT (1B / 02…)	→		
	←	ACK	
		.	Repeat until recognized
		.	
REPORT (1B /02…)	→		
	←	DTS EVENT ACKNOWLEDGE (05)	

Error sequence VMC		Comms Gateway	Comments
POLL (1A)	→		Sent continuously, or until the error is resolved
	←	ERROR (06)	
ACK	→		

Activity sequence VMC		Comms Gateway	Comments
REPORT (1B…)	→		Sent every activity
	←	ACK	

Section 9

Universal Satellite Device (USD)
VMC/Peripheral Communication Specifications

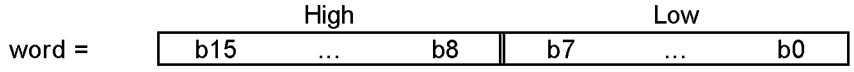
9.1 Introduction

An MDB Universal Satellite Device (USD) is a vending device which lacks customary credit acceptance peripherals. As such, a USD must rely on a host vending machine controller (VMC) to establish credit sufficient to perform a vend. The specification herein describes a protocol by which a USD and a VMC exchange messages and credit via the MDB bus.

9.1.1 Definitions

This section defines the non-response and application response time, base addresses, and the communication bytes sent by the MDB Universal Satellite Device (USD) and a Vending Machine Controller.

- The default maximum non-response time of the USD is 5 seconds.
- The default maximum application response time of the USD is 5 seconds.
- Three consecutive USD base addresses are defined to allow multiple USDs to operate simultaneously from a single VMC
- As defined in Section 2.3, the USD Base addresses are as follows: 01000xxxB (40H), 01001xxxB (48H), and 01010xxxB (50H).
- The specification defined herein assumes a USD base address of 40H in all examples. It should be understood that differing USD base addresses (48H and 50H) will follow the same command format.
- Multi-message responses to a single command are supported. Message length is subject to the 36 byte limit imposed by the MDB standard.
- Unless stated otherwise, all byte information contained herein is assumed to be in a binary format.
- Y_n represents bytes transmitted by the VMC, and Z_n are bytes transmitted by the USD.
- When words are referenced, they consist of two bytes with the higher order byte first.



9.2 USD Summary

This section is a summary of the USD command set and an overview of the modes of operation.

9.2.1 Command Summary

Command	Hex Code	Description
RESET	40	Command for USD to self-reset.
SETUP	41	Command to configure USD to VMC requirements.
POLL	42	Command to request for USD activity status.
VEND	43	Command for vend approve / deny.
FUNDS	44	Command to send funds available or to set prices.
CONTROL	45	Command to enable/disable USD.
EXPANSION	47	Command to allow addition of features and enhancements.

9.2.2 Overview

The USD Command set described herein allows USDs' to be controlled under the following three modes of operation. The USD's mode of operation is determined by the USD's configuration byte [1] and the sequence of commands the VMC uses.

Mode One VMC is used to select items to be vended from the USD and the VMC contains all pricing information. The USD receives vend requests from the VMC and reports vend success or failure.

Mode Two The USD or the VMC may select items to be vended. The USD may have special requirements for price and/or selection ID display. In this case, the USD may issue a **FUNDS** request to retrieve this information. The USD must then issue a **VEND** request to gain approval from the VMC before a vend can take place.

Mode Three The USD selects items to be vended and has its own pricing information. The USD must issue an vend request to the VMC and gain approval before a vend can take place.

9.3 Command Protocol

This section contains the complete command set relating to the USD.

9.3.1 RESET

[1] Configuration byte refers to byte Z31 of the sequence Z31 through Z34 of the expansion 07 command. Please refer to page 9.12 for more information on how this byte influences the USD's mode of operation.

Command	Code	VMC Data	USD Response data
RESET	40	No data bytes.	ACK

The **RESET** command is the vehicle that the VMC should use to instruct the USD to return to its default (power on) operating mode. The USD should respond to a reset command with an ACK to acknowledge receipt of the reset command. The USD must not accept any vend requests until the VMC issued setup command sequence has been completed.

The USD must also respond to the VMC issued "master reset" which resets all MDB peripheral devices. The VMC causes a master reset by transmitting a continuous break condition for a minimum of 100 milliseconds.

To ensure proper initialization, the USD should issue a "just reset" (see **POLL** response **00**) whenever it's pricing or configuration has changed.

9.3.2 SETUP

Command	Code	VMC Data	USD Response Data
SETUP	41	5 bytes: Y1-Y5	7 bytes: 04 + Z1 - Z6

The **SETUP** command is the vehicle that the VMC should use to configure the USD for feature level, credit scaling factor, display decimal place, and maximum vend approve/deny time. The USD responds to this command by returning it's feature level, highest vend price (divided by the scaling factor), selection configuration, and maximum application response time.

Alternatively, if the USD is not prepared to render a full response to the **SETUP** command, it may reply with an ACK. If this occurs, the USD must transmit it's setup data later, in response to a **POLL** command (see **POLL** command, response **04**). Until the **SETUP** command has been received by the USD, and the USD has correspondingly returned it's own setup data to the VMC, all vend requests will be disallowed.

Data sequence transmitted by the VMC to the USD during SETUP

VMC Data	Meaning or interpretation
Y1 =	VMC Feature level, Indicates current feature level of the VMC. Currently defined level is one.[2]
Y2 - Y3 =	Scaling factor 2 bytes (word). All transactions with the USD must be evenly divisible by this number.
Y4 =	Decimal place (02=US). Indicates the position of the decimal

[2] Feature level of the VMC is sent to allow the USD to arbitrate command compatibility with the VMC.

	place on the USD's optional credit display
Y5 =	VMC maximum approve / deny time in seconds, FF = 255 seconds.

Data sequence transmitted by the USD to the VMC during SETUP

USD Response	Meaning or interpretation
04 + Z1 =	USD Feature level, indicates current feature level of the USD. Currently defined level is one.[3]
Z2 - Z3 =	Maximum price on USD in 2 bytes (word). Indicates the highest priced item on the USD.[4] USD should return FF FFh if it does not have internal pricing capability.
Z4 - Z5 =	Item number, defined by the manufacturer configuration (Binary).
Z6 =	USD maximum application response time in seconds, FF = 255 seconds.

[3] Feature level of the USD is sent to allow the VMC to arbitrate command compatibility with the USD. The USD may opt to send this data later in response to a POLL.
[4] The maximum price on the USD is returned to the VMC so this price can be used in the computation of maximum credit acceptance.

9.3.3 POLL

Command	Code	USD response Data	USD Response Description
POLL	42	00	USD has just been reset, or wishes to be reset by the VMC.
		01 + 4 bytes Z1- Z4	Vend request, USD requests approval to vend a specified item from VMC.
		02	Vend or home success, requested vend or home was successful.
		03 + 4 bytes Z1 - Z4	Vend or home fail, requested vend or home has failed. Reason for failure is returned.
		04 + 6 bytes Z1 - Z6	USD configuration and setup data.
		05 + 2 bytes Z1 - Z2	USD item price request.
		06 + 2 bytes Z1 - Z2	USD Error codes.
		07 + 34 bytes Z1 - Z34	USD Peripheral ID string.
		08 + 4 bytes Z1 - Z4	USD Status response.
		09 + *n* bytes Z1 - *Zn*	USD multiple data block transfer response.
		0A + n bytes Z1 - Zn	USD single data block response
		1B + 5 bytes Z2 - Z6	FTL REQ TO RCV response
		1C + 3 bytes Z2 - Z4	FTL RETRY / DENY response
		1D + n bytes Z2 - Zn	FTL SEND BLOCK response
		1E + 2 bytes Z2 - Z3	FTL OK TO SEND response
		1F + 5 bytes Z2 - Z6	FTL REQ TO SEND response
		FF + Z1 - *Zn*	USD Diagnostic response.

The **POLL** command is used by the VMC to obtain status information from the USD. The same command is used by the USD to indicate a reset, request a vend, indicate vend success, indicate the reason for a vend failure, request the price of an item, send configuration and/or error data, return the USD's peripheral identification string, control the transmission and reception of data blocks, return a status and/or diagnostic response.

The USD responds to the **POLL** command with either an ACK, or a multi-byte response if there is more information to convey.

Data sequence transmitted by the USD to the VMC after a *Reset Request*

USD Response	Meaning or interpretation
00	The 00 response indicates that the USD has just been reset or wishes to be reset[5].

Data sequence transmitted by the USD to the VMC for a V*end Request*

USD Response	Meaning or interpretation
01 + Z1 - Z2 =	Selection in 2 bytes. Indicates the product to be vended by item number, defined by the manufacturer, as part of a vend request.
Z3 - Z4 =	Scaled product price in 2 bytes (word). Indicates the price of the product to be vended divided by the scaling factor. A price of FFFF is transmitted if the USD does not contain price information.

Data sequence transmitted by the USD to the VMC after a V*end or Home success*

USD Response	Meaning or interpretation
02	Indicates that the requested vend or home was successful.

Data sequence transmitted by the USD to the VMC after a V*end or Home Fail*

USD Response	Meaning or interpretation
03 + Z1 - Z2 =	USD item number, defined by the manufacturer.
Z3 - Z4 =	Bits: b0 = Selection sold out. b1 = Selection motor / actuator jam. b2 = Non-existent motor / actuator. b3 = Invalid selection range[6]. b4 = Health safety error. b5 - b15 = Not defined.

[5] The VMC is expected to reconcile whether the USD is transmitting a 00 in confirmation of a VMC issued reset that has just occurred, or as an unsolicited request to be reset. The context of the VMC's prior communication activity should be used in making this assessment.

Data sequence transmitted by the USD to the VMC if *SETUP response delayed*

USD Response	Meaning or interpretation
04 + Z1 =	USD Feature level, Indicates current feature level of the USD. The currently defined level is one.[7]
Z2 - Z3 =	Maximum price on USD 2 bytes (word). Indicates the highest priced item on the USD.[8] USD should return FF FFh if it does not have internal pricing capability.
Z4 - Z5 =	Item number, defined by the manufacturer.
Z6 =	USD maximum application response time in seconds, FF = 255 seconds.

Data sequence transmitted by the USD if the *USD needs pricing information*

USD Response	Meaning or interpretation
05 + Z1 - Z2 =	Item number, defined by the manufacturer.

Data sequence transmitted by the USD if the *USD has a failure to report to VMC*

USD Response	Meaning or Interpretation
06 + Z1 - Z2 =	Bits: b0 = Health Safety violation. b1 = Home or Chute sensor failure b2 = Keypad or Selection switch failure b3 - b15 = Not defined.

Data sequence transmitted by the USD for peripheral ID

[6] This error code is included to identify actuators that may not be present within the initially defined row and column configuration. See bytes Z4 and Z5 of the USD's setup response. This is typical in a snack machine implementation where some trays may not be populated with a full complement of motors and/or actuators.

[7] Feature level of the USD is sent to allow the VMC to arbitrate command compatibility with the USD. The USD may have elected to transmit this setup data in fulfillment of an earlier **SETUP** command.

[8] The maximum price on the USD is returned to the VMC so this price can be used in the computation of maximum credit acceptance.

USD Response	Meaning or Interpretation
07 + Z1 - Z4 =	Manufacturer ID Code.
Z5 - Z16 =	USD Serial Number.
Z17 - Z28 =	USD Model Number.
Z29 - Z30 =	USD Software Version.
Z31 - Z34 =	Optional feature bits.

Data sequence transmitted by the USD to the VMC after a Status request

USD Response	Meaning or interpretation
08 + Z1 - Z2 =	Item number, defined by the manufacturer.
Z3 - Z4 =	Bits: b0 = Selection sold out. b1 = Selection motor / actuator jam. b2 = Non-existent motor / actuator. b3 = Invalid selection range. b4 = Health safety error. b5 - b15 = Not defined.

Data sequence transmitted by the USD to the VMC after a USD data transfer command

USD Response	Meaning or interpretation
09 + Z1 =	Z1 = 00 USD requests to receive data block Z2 from VMC Z1 = 01 USD requests to send Z2 data block(s) to VMC Z1 = 02 USD data block response where: Z2 = data block number Z3 - *Zn* = contents of data block
Z2 =	Z2 = Block number USD requests to receive if Z1 = 00 Z2 = Number of blocks the USD requests to send if Z1 = 01 Z2 = Block number the USD is sending if Z1 = 02.
Z3 - *Zn* =	Contents of data block sent by USD to VMC if Z1 = 02

Data sequence transmitted by the USD to the VMC to send a single block of data

USD Response	Meaning or interpretation
0A + Z1 - Z*n*=	Z1 -Z*n* = Arbitrary data to be received by the VMC. The number "*n*" must be less than 35 per MDB standards

Data sequence transmitted by the USD to the VMC after an File Transport Layer (FTL) REQ TO RCV command

USD Response	Meaning or interpretation
Z1=1B + Z2 - Z6	The USD is requesting to receive data from a device or VMC Z2 = Destination address of response Z3 = Source address of response (40H, 48H, 50H) Z4 = File ID Z5 = Maximum length Z6 = Control

Data sequence transmitted by the USD to the VMC after an File Transport Layer (FTL) RETRY / DENY command

USD Response	Meaning or interpretation
Z1=1C + Z2 - Z4	The USD is requesting a device or VMC to retry or deny the last FTL command. Z2 = Destination address of response Z3 = Source address of response (40H, 48H, 50H) Z4 = Retry delay

Data sequence transmitted by the USD to the VMC after an File Transport Layer (FTL) SEND BLOCK command

USD Response	Meaning or interpretation
Z1=1D + Z2 - Z34	The USD is sending a block of data (maximum of 31 bytes) to a device or VMC. Z2 = Destination address of response Z3 = Block # Z4 - Z34 = Data (maximum of 31 bytes)

Data sequence transmitted by the USD to the VMC after an File Transport Layer (FTL) OK TO SEND command

USD Response	Meaning or interpretation
Z1=1E + Z2 - Z3	The USD is indicating that it is OK for the device or VMC to send it data. Z2 = Destination address of response Z3 = Source address of response (40H, 48H, 50H)

Data sequence transmitted by the USD to the VMC after an File Transport Layer (FTL) REQ TO SEND command

USD Response	Meaning or interpretation
Z1=1F + Z2 - Z6	The USD is requesting to send data to a device or VMC. Z2 = Destination address of response Z3 = Source address of response (40H, 48H, 50H) Z4 = File ID Z5 = Maximum length Z6 = Control

Data sequence transmitted by the USD to the VMC after a diagnostic command

USD Response	Meaning or interpretation
FF + Z1 - Z*n* =	Diagnostic response.

9.3.4 VEND

Command	Code	Sub-Cmd	VMC Data	Response Data
VEND	43	00	none	none
	43	01	none	none
	43	02	2 bytes Y1-Y2	none
	43	03	2 bytes Y1-Y2	none
	43	04	2 bytes Y1-Y2	5 bytes: 08 + Z1 - Z4

The **VEND** command is the vehicle that the VMC uses to signal vend approval or disapproval in response to a USD issued vend request (**POLL** response 01). The VEND command can also be used by the VMC to initiate a vend, home a selection, or query the status of a selection on the USD.

Sub Cmd:	Meaning or interpretation
00 =	Requested vend approved.
01 =	Requested vend disapproved.
02 =	Vend specified Item number, defined by the manufacturer.
03 =	Home specified Item number, defined by the manufacturer.
04 =	Request status of specified Item number, defined by the manufacturer.

Data sequence transmitted by the USD to the VMC after a Status request

USD Response	Meaning or interpretation
08 + Z1 - Z2 =	Item number, defined by the manufacturer.
Z3 - Z4 =	Bits: b0 = Selection sold out. b1 = Selection motor / actuator jam. b2 = Non-existent motor / actuator. b3 = Invalid selection range. b4 = Health safety error. b5 - b15 = Not defined.

9.3.5 FUNDS

Command	Code	Sub-Cmd	VMC Data	Response Data
FUNDS	44	00	2 bytes: Y1-Y2	none
	44	01	6 bytes: Y1-Y6	none

The **FUNDS** command is the vehicle the VMC should use to specify the funds available for vending. The **FUNDS** 00 command is issued by the VMC whenever the level of credit changes. Typically, the USD would display the credit information returned by a **FUNDS** 00 command on a credit display. The **FUNDS** 01 is issued by the VMC in response to an item price request (**POLL** response 05) by the USD.

Sub-Cmd	Meaning or interpretation

00 + Y1 - Y2 =	Funds available in 2 bytes (word), scaled by the coin scaling factor.

Sub Cmd	Meaning or interpretation
01 + Y1 - Y2 =	Item number, defined by the manufacturer.
Y3 - Y4 =	Selection price in 2 bytes (word) scaled by coin scaling factor.
Y5 - Y6 =	Alphanumeric selection identifier 2 bytes (word), or FFFF if not available.[9]

9.3.6 CONTROL

Command	Code	Sub-Cmd	VMC Data	Response Data
CONTROL	45	00	none	none
	45	01	none	none

This command is the vehicle the VMC should use to enable or disable the USD.

Sub-Cmd	Meaning or interpretation
00	Disable USD.
01	Enable USD.

9.3.7 EXPANSION

Command	Code	Sub-Cmd	VMC Data	Response Data
EXPANSION	47	00	None	07 + Z1 - Z34 Peripheral ID string and feature bits.
	47	01	Y1 – Y4	none
	47	02	Y1	none
	47	03	Y1 - Y*n*	none
	47	04	Y1	09 + Z1 + Z2 - Z*n*
	47	05	Y1 - Y*n*	none
	47	FA	Y1 - Y5	1D + Z2 - Z34 or 1C + Z2 - Z4
	47	FB	Y1 - Y3	none

[9] Alpha-numeric selection identifier is provided to the USD for display purposes only.

	47	FC	Y1 - Y33	none
	47	FD	Y1 - Y2	1D + Z2 - Z34
	47	FE	Y1 - Y5	1E + Z2 - Z3 or 1C + Z2 - Z4
	47	FF	Diagnostics	Diagnostic response.

Data sequence transmitted by the USD to the VMC after an expansion 00 sub-command

USD Response	Meaning or Interpretation
07 + Z1 - Z3 =	Manufacturer ID Code.
Z4 - Z15 =	USD Serial Number.
Z16 - Z27 =	USD Model Number.
Z28 - Z30 =	USD Software Version.
Z31 - Z34 =	Optional feature bits: b0 = USD is capable of storing and controlling pricing. b1 = USD is capable of selecting items to vend. b2 = USD is capable of supporting the File Transport Layer. This support is defined in Section 2.6. b3 - b31 = Available for future use.

Sub-Command used by the VMC to enable optional feature bits on the USD

Sub-Cmd	Meaning or interpretation
01 + Y1 - Y4	Enable optional feature bits defined in Z31-Z34 above. Feature is enabled if bit is set to 1, all features are disabled after a reset.

Sub-Command used by the VMC to identify the number of data blocks it wishes to send to the USD

Sub-Cmd	Meaning or interpretation
02 + Y1	Number of data blocks the VMC has to send to the USD (Binary)

Sub-Command used by the VMC to transmit a data block to the USD (Y2-Y*n*) and to identify the current block number being transmitted (Y1)

Sub-Cmd	Meaning or interpretation
03 + Y1	Block number the VMC is transmitting to the USD

Y2 - Y*n*[10]	Data the VMC is transmitting to the USD

Sub-Command used by the VMC to request that the USD send or re-send data block number (Y1)

Sub-Cmd	Meaning or interpretation
04 + Y1	VMC requests USD to send block Y1

Sub-Command used by the VMC to send a single block of data to the USD

Sub-Cmd	Meaning or interpretation
05 + Y1 - Y*n*	VMC sends a single block of data consisting of Y1..Y*n*

Sub-Command used by the VMC for an FTL REQ TO RCV. The Z1- Zn response can be either immediate or delayed (POLLed).

Sub-Cmd	Meaning or interpretation
FA + Y1 - Y5	The VMC is requesting to receive data from the USD whose destination address will always be (40H, 48H, 50H). Note that all FTL Commands / Responses are defined in Section 2.6. Y1 = Destination address of command (40H,48H,50H) Y2 = Source address of command Y3 = File ID Y4 = Maximum length Y5 = Control
USD Response	**Meaning or interpretation**
Z1 - Z34 or Z1 - Z4	Z1 = 1DH which indicates SEND BLOCK Z2 = Destination address of data Z3 = Block # Z4 - Z34 = Data (maximum of 31 bytes) or Z1 = 1CH which indicates RETRY / DENY Z2 = Destination address of response Z3 = Source address of response (40H,48H,50H) Z4 = Retry delay

[10] The number "*n*" is limited by the MDB maximum message length of 36 bytes.

Sub-Command used by the VMC for an FTL RETRY / DENY.

Sub-Cmd	Meaning or interpretation
FB + Y1 - Y3	The VMC is retrying, denying, or aborting a data transfer to/from the USD whose destination address will always be (40H, 48H, 50H). Note that all FTL Commands / Responses are defined in Section 2.6. Y1 = Destination address of command (40H,48H,50H) Y2 = Source address of command Y3 = Retry delay

Sub-Command used by the VMC for an FTL SEND BLOCK.

Sub-Cmd	Meaning or interpretation
FC + Y1 - Y33	The VMC is sending data to the USD whose destination address will always be (40H, 48H, 50H). Note that all FTL Commands / Responses are defined in Section 2.6. Y1 = Destination address of command (40H,48H,50H) Y2 = Block # Y3 - Y33 = Data (maximum of 31 bytes)

Sub-Command used by the VMC for an FTL OK TO SEND. The Z1 to Z33 response can be either immediate or delayed (POLLed).

Sub-Cmd	Meaning or interpretation
FD + Y1 - Y2	The VMC is requesting to receive data from the USD whose destination address will always be (40H, 48H, 50H). Note that all FTL Commands / Responses are defined in Section 2.6. Y1 = Destination address of command (40H,48H,50H) Y2 = Source address of command
USD Response Z1 - Z34	**Meaning or Interpretation** Z1 = 1DH which indicates SEND BLOCK Z2 = Destination address of data Z3 = Source address of data Z4 - Z34 = Data (maximum of 31 bytes)

Sub-Command used by the VMC for an FTL REQ TO SEND. The Z1 - Zn response can be either immediate or delayed (POLLed).

Sub-Cmd	Meaning or interpretation
FE + Y1 - Y5	The VMC is requesting to send data to the USD whose destination address will always be (40H, 48H, 50H). Note that all FTL Commands / Responses are defined in Section 2.6.

	Y1 = Destination address of command (40H,48H,50H) Y2 = Source address of command Y3 = File ID Y4 = Maximum length Y5 = Control
USD Response Z1 - Z34 or Z1 - Z4	**Meaning or Interpretation** Z1 = 1EH which indicates OK TO SEND Z2 = Destination address of response Z3 = Source address of response (40H,48H,50H) or Z1 = 1CH which indicates RETRY / DENY Z2 = Destination address of response Z3 = Source address of response (40H,48H, 50H) Z4 = Retry delay

Data sequence transmitted by the USD to the VMC after a diagnostic command

USD Response	Meaning or interpretation
FF + Z1 - *Zn* =	Diagnostic response.

9.4 USD Power Requirements

This section defines the maximum power requirements for a USD.

USD peripherals may draw power from the MDB bus or from an integral power supply. In such cases where the USD will require power from the MDB bus, the current draw must remain within the following limits:

USD Mode	Current draw
Idle	200 mA (maximum continuous)
Vending/Homing	1.75 A (for up to 10 seconds)

9.5 Examples – Mode 1 / 2 / 3 Sessions

This section contains three examples of USD sessions in which each of the three modes of USD operation are demonstrated operation respectively.

9.5.1 MODE ONE

In this example session the VMC selects the item to vend and knows the vend price. The USD receives the vend command, attempts the vend, and reports if the attempted vend failed or was successful.

VMC	MDB Data	Explanation	USD
⇒	**43+02+01+03**	VMC requests to vend item from the USD.	
	<ACK>	USD acks vend request.	⇐
⇒	**42**	VMC polls the USD.	
	<ACK>	USD acks receipt of poll.	⇐
⇒	**42**	VMC polls the USD again .	
	02	USD responds: vend complete	⇐
⇒	**<ACK>**	VMC acks vend outcome.	

9.5.2 MODE TWO

In this example session the USD or the VMC can select items to vend but the USD may not be aware of the vend price of the item selected. If the USD needs the selected item price, it may request the item price from the VMC. The USD must then issue a **VEND** request, and wait for approval from the VMC before a vend is attempted. The VMC then approves or denies the requested vend and polls the USD for vend success or vend fail.

VMC	MDB Data	Explanation	USD
⇒	**42**	VMC polls the USD.	
	05+02+06	USD responds with pricing request for item in USD.	⇐
⇒	**<ACK>**	VMC acks the USD price request.	
⇒	**44+01+02+06+00+14 +FF+FF**	Using the Funds command the VMC sends a price of 20 coin factors for item in USD.	
	<ACK>	USD acks receipt of VMC price data.	⇐
⇒	**42**	VMC polls the USD.	
	01+02+06+FF+FF	USD responds with a request to vend item in USD at the VMC selected price.	⇐
⇒	**<ACK>**	VMC acks receipt of vend request.	
⇒	**43 + 00** or **01**	VMC approves or denies vend request.	
	<ACK>	USD acks receipt of approval or denial.	⇐
⇒	**42**	VMC polls the USD.	
	03+02+06+00+01	USD responds: vend fail, sold out.	⇐
⇒	**<ACK>**	VMC acks vend outcome.	

- The **FUNDS** command can be used by USD's which do not have internal prices but need pricing information for display purposes or for other reasons that are not required to complete a transaction.

9.5.3 MODE THREE

In this example session the USD selects the item to vend and is aware of the vend price of the item. The USD must issue a vend request and the VMC then approves or denies the requested vend. The VMC then polls the USD for vend success or vend fail.

VMC	MDB Data	Explanation	USD
⇒	**42**	VMC polls the USD.	
	01+03+02+00+1E	USD requests vend for item at in USD with price of 30 coin factors.	⇐
⇒	**<ACK>**	VMC acks the USD vend request.	
⇒	**43+ 00** or **01**	VMC approves or denies vend request.	
	<ACK>	USD acks receipt of approval or denial.	⇐
⇒	**42**	VMC polls the USD.	
	02	USD responds: vend complete	⇐
⇒	**<ACK>**	VMC acks vend outcome.	

9.6 Examples - Data Block Transfers

This section contains two examples in which data blocks are transferred between the VMC and the USD and vice versa.

9.6.1 Data Block Transfer from VMC to USD

In this example the VMC wishes to send two data blocks to the USD. To do this, the VMC uses the expansion 02 command to advise the USD of it's request to send data and also to identify the number of data blocks it wishes to send. In response, the USD uses a poll 09 to request the transmission of a data block with the block number enumerated as part of it's poll response. The VMC then uses a different expansion command (03) to send the data to the USD.

VMC	MDB Data	Explanation	USD
⇒	**47+02+02**	VMC issues a request to send two data blocks to the USD	
	<ACK>	USD acks receipt of the request	⇐
⇒	**42**	VMC polls the USD	
	09+00+01	USD responds with a request to receive data block number 01 from the VMC	⇐
⇒	**<ACK>**	VMC acks receipt of block number	
⇒	**47+03+01+21+22+23**	VMC transmits block number 01 containing data: 21, 22, and 23.	
	<ACK>	USD acks receipt of the data block	⇐
⇒	**42**	VMC polls the USD.	
	09+00+02	USD responds with a request to receive data block number 02 from the VMC.	⇐
⇒	**<ACK>**	VMC acks receipt of the block number.	
⇒	**47+03+02+24+25+26**	VMC transmits block number 02 containing data: 24, 25, and 26.	
	<ACK>	USD acks receipt of the data block	⇐

9.6.2 Data Block Transfer from USD to VMC

In this example the USD wishes to send two data blocks to the VMC. To do this, the USD makes use of the Poll 09 command to inform the VMC of it's request to send data and also to identify the number of data blocks it wishes to send. In response, the VMC uses expansion 04 command to request the transmission of a data block by the individual block number. The USD then uses the poll 09 response to send the data blocks to the VMC.

VMC	MDB Data	Explanation	USD
⇒	**42**	VMC polls the USD	
	09+01+02	USD responds with a request to send 2 data blocks to the VMC	⇐
⇒	**<ACK>**	VMC acks request to send data	
⇒	**47+04+01**	VMC responds with a request to receive data block number 01 from the USD	
	<ACK>	USD acks receipt of block number request	⇐
⇒	**42**	VMC polls the USD	
	09+02+01+55+56+57	USD responds by transmitting block number 01 containing data 55, 56, and 57.	⇐
⇒	**<ACK>**	VMC acks receipt of data	
⇒	**47+04+02**	VMC responds with a request to receive data block number 02 from the USD	
	<ACK>	USD acks receipt of block number request	⇐
⇒	**42**	VMC polls the USD	
	09+02+02+58+59+60	USD responds by transmitting block number 02 containing data 58, 59, and 60.	⇐
⇒	**<ACK>**	VMC acks receipt of data	

(this page intentionally left blank)

Section 10

Coin Hopper or Tube - Dispenser
VMC/Peripheral Communication Specifications

10.1 Introduction

This section defines the communication bytes sent and received by a coin dispensing device, which may be in the form of a hopper or tube device. As defined in Section 2.3, the dispenser's address is 01011xxxB (58H).

Unless stated otherwise, all information is assumed to be in a binary format

10.2 VMC Commands

Command	Hex Code	Description
RESET	58H	Command for dispenser to self-reset
SETUP	59H	Request for dispenser setup.
DISPENSER STATUS	5AH	Request for dispenser tube / hopper status and coin count.
POLL	5BH	Request for dispenser activity status.
MANUAL DISPENSE ENABLE	5CH	Signifies coin types allowable for coin dispensing. This command is followed by setup data. See command format section.
DISPENSE *	5DH	Command to dispense coins. Followed by coin type or value to dispense. See command format section.
PAYOUT *	5EH	Command to determine value of coins dispensed. Followed by payout status or value poll. See command format section.
EXPANSION *	5FH	Command to allow addition of features, File Transport Layer, and future enhancements. See command format section.

* **NOTE:** DISPENSE, PAYOUT, and EXPANSION commands are always followed by a "sub-command."

10.3 VMC Command Format

VMC Command	Code	VMC Data
RESET	58H	No data bytes

This command is the vehicle that the VMC should use to tell the dispenser that it should return to its default operating mode and initialize internal hardware systems. With the exception of the ACK response, it should abort all communication until otherwise instructed by the VMC.

The following initialization sequence is recommended. It should be used after "power up" or after issuing the Bus Reset (pulling the transmit line "active" for a minimum of 100 mS).

RESET – 58h
POLL – 5Bh
To obtain "JUST RESET" response

SETUP – 59h
To obtain dispenser level and configuration information

EXPANSION IDENTIFICATION – 5F 00h
To obtain additional dispenser information and options

EXPANSION FEATURE ENABLE – 5F 01h
To enable desired options

DISPENSER STATUS – 5Ah (Note 1)
To obtain dispenser status / change information

MANUAL DISPENSE ENABLE – 5Ch
To enable and disable manual coin pay-out if desired

No power above idle current can be drawn until after the first POLL following the RESET command. Also, the JUST RESET response to the POLL command must be delayed until any high current usage has been completed.

The dispenser must hold its response of the DISPENSER status until a valid current reading from the sensor system is achieved.

VMC Command	Code	Dispenser Response Data
SETUP	59H	26 bytes: Z1 - Z26

Z1 = Dispenser Feature Level - 1 byte

Indicates the feature level of the dispenser. This will distinguish the dispensers feature level to the VMC. Currently only level 1 is supported.

Z2 - Z3 = Country / Currency Code - 2 bytes

The packed BCD currency code of the dispenser is sent with the left most digit a 1. See Appendix A1 for the latest version of the ISO 4217 numeric currency code. For example, the code for the US dollar is 18 40H (Z2 = 18 and Z3 = 40) and for the Euro is 1978 (Z2 = 19 and Z3 = 78).

Z4 = Coin Scaling Factor - 1 byte

All dispensed coin values must be evenly divisible by this number. For example, this could be set to 05H for the USA nickel.

Z5 = Decimal Places - 1 byte

Indicates the number of decimal places on a credit display. For example, this could be set to 02H in the USA.

Z6 = Application Maximum Response Time (seconds) – 1 byte
The maximum length of time a dispenser will require to provide a response to any command from the VMC. The value reported here supercedes the dispenser's default NON-RESPONSE time defined in section 10.4 if the value reported here is greater.

Z7 – Z8 = Bit set, if coin disabled by dispenser (i.e. switch).

Z9 – Z10 = Bit set, if coin is self filling.

Z11 - Z26 = Coin Type Credit - 16 bytes

Indicates the value of coin types 0 to 15. Values must be sent in ascending order. This number is the coin's monetary value divided by the coin scaling factor. Unused coin types are sent as 00H. Unsent coin types are assumed to be zero. It is not necessary to send all coin types. Coin type credits sent as FFH are assumed to be vend tokens. That is, their value is assumed to be worth one vend.

The byte position in the 16 byte string indicates the coin type(s). For example, the first byte sent would indicate the value of coin type 0, the second byte sent would indicate the value of coin type 1, and so on. For example, the USA coin types may be; Coin type 0 = nickel, Coin type 1 = dime, Coin type 2 = quarter, Coin type 3 = dollar.

VMC Command	Code	Dispenser Response Data
DISPENSER STATUS	5AH	34 bytes: Z1 – Z34

Z1 - Z2 = Dispenser Full Status - 2 bytes

Indicates status of coin tube / hopper for coin types 0 to 15.

b15 b14 b13 b12 b11 b10 b9 b8 | b7 b6 b5 b4 b3 b2 b1 b0
Z1 Z2

A bit is set to indicate a full dispenser. For example, bit 7 = set would indicate the dispenser for coin type 7 is full.

Z3 – Z34 = Coin Count - 32 bytes

Indicates the greatest number of coins that the dispenser "knows" definitely are present in the coin tube / hopper. A word (2bytes) position in the 32 byte string indicates the number of coins in a tube / hopper for a particular coin type. For example, the first 2 bytes sent indicate the number of coins in a tube / hopper for coin type 0. Unsent bytes are assumed to be zero. For tube / hopper counts greater than 65535, counts should remain at 65535.

NOTE: If a dispenser can detect a tube or hopper jam, defective tube or hopper sensor, or other malfunction, it will indicate the tube / hopper is "bad" by sending a tube / hopper full status and a count of zero for the malfunctioning coin type.

VMC Command	Code	Dispenser Response Data
POLL	5BH	32 bytes: Z1 – Z32

Z1 - Z32 = Dispenser Activity - 32 bytes

Indicates the dispenser activity. If there is nothing to report, the dispenser should send only an ACK. Otherwise, the only valid responses are:

Coins Dispensed:

Z1 (10yzxxxx)

z	z=1 for manual dispense z=0 to report a non manual (automatic) dispense
y	y=1 For payout complete.
xxxx	The coin type dispensed (0 to 15)

Z2 - Z3	The number of coins dispensed.
Z4 – Z5	The number of coins in the dispenser.

Status:

(00000001) =	Escrow request[1] - An escrow lever activation has been detected. If a button is present and activated.
(00000010) =	Dispenser Payout Busy[2] - The dispenser is busy activating payout devices.
(00000011) =	Not Used
(00000100) =	Defective Dispenser Sensor[1] - The dispenser has detected one of the dispenser sensors behaving abnormally.
(00000101) =	Not Used
(00000110) =	Dispenser did not start[1].
(00000111) =	Dispenser Jam[1] - A dispenser payout attempt has resulted in jammed condition.
(00001000) =	ROM checksum error[1] - The dispensers internal checksum does not match the calculated checksum.
(00001001) =	Not Used
(00001010) =	Not Used
(00001011) =	Dispenser was "Just Reset"[1] - The dispenser has detected a Reset condition and has returned to its power-on idle condition.
(00001100) =	Not Used
(00001101) =	Not Used
(00001110) =	Not Used
(00001111) =	Filled key pressed[1] – The VMC should request a new DISPENSER STATUS.

NOTES: The dispenser may send several of one type activity, up to 16 bytes total. This will permit zeroing counters such as inventory and status.

1 Sent once each occurrence.
2 Sent once each POLL

File Transport Layer POLLed responses:

Note that all FTL responses are defined in Section 2.6. For the coin dispenser, the source address will always be the dispenser (58H) as defined in Section 2.3.

Z1

1B REQ TO RCV The coin dispenser is requesting to receive data from a device or VMC.

Z2 = Destination address of response
Z3 = Source address of response (58H)
Z4 = File ID
Z5 = Maximum length
Z6 = Control

1C RETRY/DENY The coin dispenser is requesting a device or VMC to retry or deny the last FTL command.

Z2 = Destination address of response
Z3 = Source address of response (58H)
Z4 = Retry delay

1D SEND BLOCK The coin dispenser is sending a block of data (maximum of 31 bytes) to a device or VMC.

Z2 = Destination address of data
Z3 = Block #
Z4-Z34 = Data (maximum of 31 bytes)

1E OK TO SEND The coin dispenser is indicating that it is OK for a device or VMC to send it data.

Z2 = Destination address of response
Z3 = Source address of response (58H)

1F REQ TO SEND The coin dispenser is requesting to send data to a device or VMC.

Z2 = Destination address of response
Z3 = Source address of response (58H)
Z4 = File ID
Z5 = Maximum length
Z6 = Control

VMC Command	Code	VMC Data
MANUAL DISPENSE ENABLE	5CH	2 bytes: Y1 – Y2

Y1 - Y2 = Manual Dispense Enable - 2 bytes

b15 b14 b13 b12 b11 b10 b9 b8 | b7 b6 b5 b4 b3 b2 b1 b0
Y1 Y2

A bit is set to indicate dispense enable. For example, bit 2 is set to enable dispensing of coin type 2. This command enables/disables manual dispensing using optional inventory switches. All manual dispensing switches are automatically disabled upon reset.

VMC Command	Code	Sub-command	VMC Data
DISPENSE COINS	5DH	00H	3 bytes: Y1 – Y3

b7 b6 b5 b4 b3 b2 b1 b0
Y1

Bits b3, b2, b1, b0 indicate coin type to be dispensed. Valid codes are 0H to FH to indicate coin types 0 to 15.

Bits b7, b6, b5, b4 = 0

Y2 - Y3 = Number of coins to be dispensed of coin type defined in Y1

There is no defined limit on how long the actual dispense takes since the command allows for up to 65535 coins to be paid out. The payout cycle begins when the dispenser ACKs the VMC's DISPENSE (5DH) command. This cycle typically lasts a minimum of 100 mS and ends when the dispenser stops dispensing the desired number of coins. VMCs should monitor the Dispenser Payout Busy and Dispenser Activity response to the POLL (5B) command to determine when the entire payout cycle is completed.

VMC Command	Code	Sub-command	VMC Data
DISPENSE VALUE	5DH	01H	2 bytes: Y1, Y2

Y1 – Y2 = Value of coins to be paid out.

Y1 and Y2 are defined as the value of coins and this value is expressed as the number of coin scaling factors that would sum to the value. For example, in a USA system using a scaling factor of 05, if the change to be paid out is 75 cents, then Y1 will equal fifteen. That is, the sum of fifteen nickels equal 75 cents. The coin dispenser will determine which actual denominations of coins will be paid out. In the 75 cent example, the coins may be 3 quarters; or, 7 dimes & 1 nickel; or, 2 quarters & 2 dimes & 1 nickel, etc. The actual coins dispensed and if the dispense is finished can be acquired via the PAYOUT STATUS (5E, 00) and PAYOUT VALUE POLL (5E, 01).

VMC Command	Code	Sub-command	VMC Data	Dispenser Response
PAYOUT STATUS	5E	00H	None	32 bytes: Z1-Z32

Z1 – Z32 = Number of each coin type paid out - 32 bytes (2 bytes per coin type).

This is the dispenser's response to the last VMC DISPENSE VALUE command (5DH sub command 01H). Bytes are sent in ascending order of coin types. A bytes position in the string indicates the coin type. That is, bytes one and two are the number of coins for coin type 1, bytes three and four are the number of coins for coin type two, and so on. Unsent bytes above the coin types dispensed are assumed to be zero.

The dispenser clears payout data after an ACK response from the VMC.

The VMC should compare the value of the coins paid out to the (5DH) DISPENSE VALUE command's Y2-Y3.

NOTES:

1) If the dispenser's payout is busy it will respond to the PAYOUT STATUS command with an ACK only.

2) If no coins have been paid out, at least one zero valued data byte must be sent.

3) There is no defined limit on how long the actual payout takes. See dispense command (5DH) for further details

VMC Command	Code	Sub-command	VMC Data	Dispenser Response Data
PAYOUT VALUE POLL	5EH	01H	None	2 bytes: Z1-Z2

Z1 – Z2 = Dispenser Payout Activity - 2 bytes

An interval value (scaled) which indicates the amount of paid out change since the previous PAYOUT VALUE POLL (or between the initial DISPENSE VALUE command (5DH sub command 01H) and the first PAYOUT VALUE POLL).

A 00H response indicates no coins were paid out since the previous PAYOUT VALUE POLL (or the initial DISPENSE VALUE command (5DH sub command 01H)).

An ACK only indicates that the change payout is finished. This should be followed by the PAYOUT STATUS command (5EH-00H) to obtain the complete payout data.

NOTE: The initial intent of this command is to determine the amount of change paid out so that the credit display can be decremented as coins are dispensed.

VMC Command	Code	Sub-Command	Dispenser Response Data
EXPANSION COMMAND	5FH	00H IDENTIFICATION	33 bytes: Z1 - Z33

Z1 - Z3 = Manufacturer Code - 3 bytes
Identification code for the equipment supplier. Sent as ASCII characters. Currently defined codes are listed in the **EVA** document entitled "***European Vending Association Data Transfer Standard***" (**EVA-DTS**), the Audit Data Lists section, sub-section 2, "Manufacturer Codes".

Z4 - Z15 = Serial Number - 12 bytes

Factory assigned serial number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable.

Z16 - Z27 = Model Number - 12 bytes

Manufacturer assigned model number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable.

Z28 - Z29 = Software Version - 2 bytes

Current software version. Must be sent in packed BCD.

Z30 - Z33 = Optional Features - 4 bytes

Each of the 32 bits indicate an optional features availability. If the bit is set the feature is available. Bits should be sent in descending order, i.e. bit 31 is sent first and bit 0 is sent last. Currently defined options are:

B0 - File Transport Layer (FTL) supported as defined in Section 2.6.

B1 - B31 Available for future use

VMC Command	Code	Sub-Command	VMC Data
EXPANSION COMMAND	5FH	01H FEATURE ENABLE	4 bytes: Y1 - Y4

This command is used to enable each of the optional features defined in Z30-Z33 above. To enable a feature a bit is set to one. **All optional features are disabled after reset.**

VMC Command	Code	Sub-command	VMC Data	Dispenser Response
EXPANSION COMMAND	5FH **FTL**	FAH REQ TO RCV	Y1-Y5	Z1 - Zn (immediate or POLLed)

The VMC is requesting to receive data from the dispenser whose destination address will always be (58H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (58H)
Y2 = Source address of command
Y3 = File ID
Y4 = Maximum length
Y5 = Control

Z1 = 1DH which indicates SEND BLOCK
Z2 = Destination address of data
Z3 = Block #
Z4 - Z34 = Data (maximum of 31 bytes)

or

Z1 = 1CH which indicates RETRY / DENY
Z2 = Destination address of response
Z3 = Source address of response (58H)
Z4 = Retry delay

VMC Command	Code	Sub-command	VMC Data	Dispenser Response
EXPANSION COMMAND	5FH **FTL**	FBH RETRY / DENY	Y1-Y3	None

The VMC is retrying, denying, or aborting a data transfer to/from the dispenser whose destination address will always be (58H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (58H)
Y2 = Source address of command
Y3 = Retry delay

VMC Command	Code	Sub-command	VMC Data	Dispenser Response
EXPANSION COMMAND	5FH	FCH **FTL** SEND BLOCK	Y1-Y33	None

The VMC is sending data to the dispenser whose destination address will always be (58H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command & data (58H)
Y2 = Block #
Y3 - Y33 = Data (maximum of 31 bytes)

VMC Command	Code	Sub-command	VMC Data	Dispenser Response
EXPANSION COMMAND	5FH	FDH **FTL** OK TO SEND	Y1-Y2	Z1-Z34 (immediate or POLLed)

The VMC is indicating that it is OK for the dispenser to transfer data. The destination address will always be the dispenser (58H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (58H)
Y2 = Source address of command

Z1 = 1DH which indicates SEND BLOCK
Z2 = Destination address of data
Z3 = Source address of data
Z4 - Z34 = Data (maximum of 31 bytes)

VMC Command	Code	Sub-command	VMC Data	Dispenser Response
EXPANSION COMMAND	5FH	FEH **FTL** REQ TO SEND	Y1-Y5	Z1 (immediate or POLLed)

The VMC is requesting to send data to the dispenser whose destination address will always be (58H). Note that all FTL Commands / Responses are defined in Section 2.6.

Y1 = Destination address of command (58H)
Y2 = Source address of command
Y3 = File ID
Y4 = Maximum length
Y5 = Control

Z1 = 1EH which indicates OK TO SEND
Z2 = Destination address of response
Z3 = Source address of response (58H)

or

Z1 = 1CH which indicates RETRY / DENY
Z2 = Destination address of response
Z3 = Source address of response (58H)
Z4 = Retry delay

VMC Command	Code	Sub-command	VMC Data	Dispenser Response
EXPANSION COMMAND	5FH	FFH DIAGNOSTICS	Y1-Yn	Z1-Zn

Y1 - Yn = Device manufacturer specific instruction for implementing various manufacturing or test modes. Y1 - Yn implies that any number of bytes can be used for the VMC data to the peripheral.

Z1 - Zn = Device manufacturer specific responses after receiving manufacturing or test instructions. Z1 - Zn implies that any number of bytes can be used for the Dispenser response data from the peripheral.

10.4 Dispenser Non-Response Time

The default maximum non-response time for the dispenser is 5 seconds. This is the maximum time for which a dispenser will not respond to a command or a POLL with ACK, NAK or a message. The "Application Maximum Response Time" reported in byte Z6 of the SETUP (10.3) supersedes this default value if Z6 is greater.

10.5 Dispenser Power Requirements

The current draw for any dispenser must fall within the following limits. All measurements are at the minimum VMC voltage output.

Idle mode = 200 mA. (max.) continuous

Coin payout = 2.5 A. (max.) for up to 15 seconds per coin dispensed. This is the maximum for all dispensers operating simultaneously in this unit.

Appendix 1

Currency Codes

A1.1 Information

The following **Tables of Codes for the Representation of Currencies and Funds** are provided by the Secretariat of ISO 4217 MA. It is provided here to be used for the MDB currency code information sent between the credit peripherals and the VMC.

Table A.1 Currency and Funds Code List (English alphabetical order by entity)

Table A.2 Funds Codes Registered with the Maintenance Agency

Table A.3 Codes for Historic Denominations of Currencies and Funds

A1.2 MDB/ICP Use

As stated in the individual credit device sections, the two byte, packed BCD country / currency code of the coin changer, bill validator, and card reader devices can be sent in two different forms depending on the value of the left most BCD digit .

If the left most digit is a 0, the International Telephone Code is used to indicate the country that the reader is set-up for.

For example, the USA telephone code is 001 which translates into the MDB code as **00 01**h (Zx = **00**h and Zy = **01**h).

If the left most digit is a 1, the latest version of the ISO 4217 numeric currency code is used as listed in this Appendix.

For example, the code for the US dollar is 840 which translates into the MDB code as **18 40**h (Zx = **18**h and Zy = **40**h).

The code for the Euro is 978 which translates into the MDB code as **1978**h (Zx = **19**h and Zy = **78**h).

FFFFh should be used if the country code in unknown (Zx = **FF**h and Zy = **FF**h).

Note that for level 3 cashless readers, it is mandatory to use the the ISO 4217 numeric currency code.

Table A.1 Currency and Funds Code List (English alphabetical order by entity)

ENTITY	Currency	Code		Decimal Position
		Alphabetic	Numeric	
AFGHANISTAN	Afghani	AFA	004	2
ALBANIA	Lek	ALL	008	2
ALGERIA	Algerian Dinar	DZD	012	2
AMERICAN SAMOA	US Dollar	USD	840	2
ANDORRA	Spanish Peseta	ESP	724	0
	French Franc	FRF	250	2
	Andorran Peseta	ADP	020	0
ANGOLA	New Kwanza	AON	024	2
	Kwanza Reajustado	AOR	982	2
ANGUILLA	East Caribbean Dollar	XCD	951	2
ANTARCTICA	No universal currency			
ANTIGUA AND BARBUDA	East Caribbean Dollar	XCD	951	2
ARGENTINA	Argentine Peso	ARS	032	2
ARMENIA	Armenian Dram	AMD	051	2
ARUBA	Aruban Guilder	AWG	533	2
AUSTRALIA	Australian Dollar	AUD	036	2
AUSTRIA	Schilling	ATS	040	2
AZERBAIJAN	Azerbaijanian Manat	AZM	031	2
BAHAMAS	Bahamian Dollar	BSD	044	2
BAHRAIN	Bahraini Dinar	BHD	048	3
BANGLADESH	Taka	BDT	050	2
BARBADOS	Barbados Dollar	BBD	052	2
BELARUS	Belarussian Ruble	BYB	112	0
	Belarussian Ruble	BYR	974	0
BELGIUM	Belgian Franc	BEF	056	0
BELIZE	Belize Dollar	BZD	084	2
BENIN	CFA Franc BCEAO+	XOF	952	0
BERMUDA	Bermudian Dollar (customarily known as Bermuda Dollar)	BMD	060	2
BHUTAN	Indian Rupee	INR	356	2
	Ngultrum	BTN	064	2

+ CFA Franc BCEAO; Responsible authority: Banque Centrale des Etats de

l'Afrique de l'Ouest.

Table A.1 (Continued)

ENTITY	Currency	Code Alphabetic	Code Numeric	Decimal Position
BOLIVIA	Boliviano	BOB	068	2
	Mvdol*	BOV	984	2
BOSNIA & HERZEGOVINA	Convertible Marks	BAM	977	2
BOTSWANA	Pula	BWP	072	2
BOUVET ISLAND	Norwegian Krone	NOK	578	2
BRAZIL	Brazilian Real	BRL	986	2
BRITISH INDIAN OCEAN TERRITORY	US Dollar	USD	840	2
BRUNEI DARUSSALAM	Brunei Dollar	BND	096	2
BULGARIA	Lev	BGL	100	2
	Bulgarian LEV	BGN	975	2
BURKINA FASO	CFA Franc BCEAO+	XOF	952	0
BURUNDI	Burundi Franc	BIF	108	0
CAMBODIA	Riel	KHR	116	2
CAMEROON	CFA Franc BEAC#	XAF	950	0
CANADA	Canadian Dollar	CAD	124	2
CAPE VERDE	Cape Verde Escudo	CVE	132	2
CAYMAN ISLANDS	Cayman Islands Dollar	KYD	136	2
CENTRAL AFRICAN REPUBLIC	CFA Franc BEAC#	XAF	950	0
CHAD	CFA Franc BEAC#	XAF	950	0
CHILE	Chilean Peso	CLP	152	0
	Unidades de fomento*	CLF	990	0
CHINA	Yuan Renminbi	CNY	156	2
CHRISTMAS ISLAND	Australian Dollar	AUD	036	2
COCOS (KEELING) ISLANDS	Australian Dollar	AUD	036	2
COLOMBIA	Colombian Peso	COP	170	2
COMOROS	Comoro Franc	KMF	174	0

+ CFA Franc BCEAO; Responsible authority: Banque Centrale des Etats de l'Afrique de l'Ouest.

CFA Franc BEAC; Responsible authority: Banque des Etats de l'Afrique Centrale.

- Funds code [See table A.2(E) for definitions of funds types].

Table A.1 (Continued)

ENTITY	Currency	Code	Decimal Position
		Alphabetic	Numeric	
CONGO	CFA Franc BEAC#	XAF	950	0
CONGO, THE DEMOCRATIC REPUBLIC OF	Franc Congolais	CDF	976	2
COOK ISLANDS	New Zealand Dollar	NZD	554	2
COSTA RICA	Costa Rican Colon	CRC	188	2
COTE D'IVOIRE	CFA Franc BCEAO+	XOF	952	0
CROATIA	Kuna	HRK	191	2
CUBA	Cuban Peso	CUP	192	2
CYPRUS	Cyprus Pound	CYP	196	2
CZECH REPUBLIC	Czech Koruna	CZK	203	2
DENMARK	Danish Krone	DKK	208	2
DJIBOUTI	Djibouti Franc	DJF	262	0
DOMINICA	East Caribbean Dollar	XCD	951	2
DOMINICAN REPUBLIC	Dominican Peso	DOP	214	2
EAST TIMOR	Timor Escudo	TPE	626	0
	Rupiah	IDR	360	2
ECUADOR	US Dollar	ESD	840	2
EGYPT	Egyptian Pound	EGP	818	2
EL SALVADOR	El Salvador Colon	SVC	222	2
EQUATORIAL GUINEA	CFA Franc BEAC#	XAF	950	0
ESTONIA	Kroon	EEK	233	2
ERITREA	Nakfa	ERN	232	2
ETHIOPIA	Ethiopian Birr	ETB	230	2
FAEROE ISLANDS	Danish Krone	DKK	208	2
FALKLAND ISLANDS (MALVINAS)	Falkland Islands Pound	FKP	238	2

CFA Franc BEAC; Responsible authority: Banque des Etats de l'Afrique Centrale.

+ CFA Franc BCEAO; Responsible authority: Banque Centrale des Etats de l'Afrique de l'Ouest.

* Funds code [see Table A.2 (E) for definitions of funds types].

Table A.1 (Continued)

ENTITY	Currency	Code	Decimal	
		Alphabetic	Numeric	Position
FIJI	Fiji Dollar	FJD	242	2
FINLAND	Markka	FIM	246	2
FRANCE	French Franc	FRF	250	2
FRENCH GUIANA	French Franc	FRF	250	2
FRENCH POLYNESIA	CFP Franc	XPF	953	0
FRENCH SOUTHERN TERRITORIES	French Franc	FRF	250	2
GABON	CFA Franc BEAC#	XAF	950	0
GAMBIA	Dalasi	GMD	270	2
GEORGIA	Lari	GEL	981	2
GERMANY	Deutsche Mark	DEM	276	2
GHANA	Cedi	GHC	288	2
GIBRALTAR	Gibraltar Pound	GIP	292	2
GREECE	Drachma	GRD	300	2
GREENLAND	Danish Krone	DKK	208	2
GRENADA	East Caribbean Dollar	XCD	951	2
GUADELOUPE	French Franc	FRF	250	2
GUAM	US Dollar	USD	840	2
GUATEMALA	Quetzal	GTQ	320	2
GUINEA	Guinea Franc	GNF	324	0
GUINEA-BISSAU	Guinea-Bissau Peso	GWP	624	2
	CFA Franc BCEAO+	XOF	952	0
GUYANA	Guyana Dollar	GYD	328	2
HAITI	Gourde	HTG	332	2
	US Dollar	USD	840	2
HEARD AND MCDONALD ISLANDS	Australian Dollar	AUD	036	2
HONDURAS	Lempira	HNL	340	2

CFA Franc BEAC; Responsible authority: Banque des Etats de l'Afrique Centrale.

+ CFA Franc BCEAO; Responsible authority: Banque Centrale des Etats de l'Afrique de l'Ouest.

Table A.1 (Continued)

ENTITY	Currency	Code	Decimal
		Alphabetic	Numeric	Point
HONG KONG	Hong Kong Dollar	HKD	344	2
HUNGARY	Forint	HUF	348	2
ICELAND	Iceland Krona	ISK	352	2
INDIA	Indian Rupee	INR	356	2
INDONESIA	Rupiah	IDR	360	2
INTERNATIONAL MONETARY FUND (IMF)**	SDR	XDR	960	N.A.
IRAN (ISLAMIC REPUBLIC OF)	Iranian Rial	IRR	364	2
IRAQ	Iraqi Dinar	IQD	368	3
IRELAND	Irish Pound	IEP	372	2
ISRAEL	New Israeli Sheqel*	ILS	376	2
ITALY	Italian Lira	ITL	380	0
JAMAICA	Jamaican Dollar	JMD	388	2
JAPAN	Yen	JPY	392	0
JORDAN	Jordanian Dinar	JOD	400	3
KAZAKHSTAN	Tenge	KZT	398	2
KENYA	Kenyan Shilling	KES	404	2
KIRIBATI	Australian Dollar	AUD	036	2
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	North Korean Won	KPW	408	2
KOREA, REPUBLIC OF	Won	KRW	410	0

* Currency name was effective 4th September 1985

** This entry is not derived fron ISO 3166, but is included here in alphabetic sequence for convenience.

Table A.1 (Continued)

ENTITY	Currency	Code		Decimal
		Alphabetic	Numeric	Point
KUWAIT	Kuwaiti Dinar	KWD	414	3
KYRGYZSTAN	Som	KGS	417	2
LAO PEOPLE'S DEMOCRATIC REPUBLIC	Kip	LAK	418	2
LATVIA	Latvian Lats	LVL	428	2
LEBANON	Lebanese Pound	LBP	422	2
LESOTHO	Rand	ZAR	710	2
	(financial Rand)*	ZAL	991	2
	Loti	LSL	426	2
LIBERIA	Liberian Dollar	LRD	430	2
LIBYAN ARAB JAMAHIRIYA	Libyan Dinar	LYD	434	3
LIECHTENSTEIN	Swiss Franc	CHF	756	2
LITHUANIA	Lithuanian Litas	LTL	440	2
LUXEMBOURG	Luxembourg Franc	LUF	442	0
MACAU	Pataca	MOP	446	2
MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF	Denar	MKD	807	2
MADAGASCAR	Malagasy Franc	MGF	450	0
MALAWI	Kwacha	MWK	454	2
MALAYSIA	Malaysian Ringgit	MYR	458	2
MALDIVES	Rufiyaa	MVR	462	2
MALI	CFA Franc BCEAO+	XOF	952	0
MALTA	Maltese Lira	MTL	470	2
MARSHALL ISLANDS	US Dollar	USD	840	2
MARTINIQUE	French Franc	FRF	250	2
MAURITANIA	Ouguiya	MRO	478	2
MAURITIUS	Mauritius Rupee	MUR	480	2
MEXICO	Mexican Peso	MXN	484	2
	Mexican Unidad de Inversion (UDI)*	MXV	979	2
MICRONESIA	US Dollar	USD	840	2
MOLDOVA, REPUBLIC OF	Moldovan Leu	MDL	498	2
MONACO	French Franc	FRF	250	2

* Funds code [See table A.2(E) for definitions of funds types].

\+ CFA Franc BCEAO; Responsible authority: Banque Centrale des Etats de l'Afrique de l'Ouest.

Table A.1 (Continued)

ENTITY	Currency	Code		Decimal Point
		Alphabetic	Numeric	
MONGOLIA	Tugrik	MNT	496	2
MONTSERRAT	East Caribbean Dollar	XCD	951	2
MOROCCO	Moroccan Dirham	MAD	504	2
MOZAMBIQUE	Metical	MZM	508	2
MYANMAR	Kyat	MMK	104	2
NAMIBIA	Rand	ZAR	710	2
	Namibia Dollar**	NAD	516	2
NAURU	Australian Dollar	AUD	036	2
NEPAL	Nepalese Rupee	NPR	524	2
NETHERLANDS	Netherlands Guilder	NLG	528	2
NETHERLANDS ANTILLES	Netherlands Antillian Guilder	ANG	532	2
NEW CALEDONIA	CFP Franc	XPF	953	0
NEW ZEALAND	New Zealand Dollar	NZD	554	2
NICARAGUA	Cordoba Oro	NIO	558	2
NIGER	CFA Franc BCEAO+	XOF	952	0
NIGERIA	Naira	NGN	566	2
NIUE	New Zealand Dollar	NZD	554	2
NORFOLK ISLAND	Australian Dollar	AUD	036	2
NORTHERN MARIANA ISLANDS	US Dollar	USD	840	2
NORWAY	Norwegian Krone	NOK	578	2
OMAN	Rial Omani	OMR	512	3
PAKISTAN	Pakistan Rupee	PKR	586	2
PALAU	US Dollar	USD	840	2

+ CFA Franc BCEAO; Responsible authority: Banque Centrale des Etats de l'Afrique de l'Ouest.

The lowest unit of recorded value for the Iraqi Dinar is the Dirham (1 Iraqi Dinar = 20 Dirhams).

** The Namibia Dollar becomes effective September 15th 1993

Table A.1 (Continued)

ENTITY	Currency	Code	Decimal
		Alphabetic	Numeric	Point
PANAMA	Balboa	PAB	590	2
	US Dollar	USD	840	2
PAPUA NEW GUINEA	Kina	PGK	598	2
PARAGUAY	Guarani	PYG	600	0
PERU	Nuevo Sol	PEN	604	2
PHILIPPINES	Philippine Peso	PHP	608	2
PITCAIRN	New Zealand Dollar	NZD	554	2
POLAND	Zloty	PLN	985	2
PORTUGAL	Portuguese Escudo	PTE	620	0
PUERTO RICO	US Dollar	USD	840	2
QATAR	Qatari Rial	QAR	634	2
REUNION	French Franc	FRF	250	2
ROMANIA	Leu	ROL	642	2
RUSSIAN FEDERATION	Russian Ruble	RUR	810	2
	Russian Ruble	RUB	643	2
RWANDA	Rwanda Franc	RWF	646	0
ST HELENA	St Helena Pound	SHP	654	2
ST KITTS - NEVIS	East Caribbean Dollar	XCD	951	2
SAINT LUCIA	East Caribbean Dollar	XCD	951	2
ST PIERRE AND MIQUELON	French Franc	FRF	250	2
SAINT VINCENT AND THE GRENADINES	East Caribbean Dollar	XCD	951	2
SAMOA	Tala	WST	882	2
SAN MARINO	Italian Lira	ITL	380	0
SAO TOME AND PRINCIPE	Dobra	STD	678	2
SAUDI ARABIA	Saudi Riyal	SAR	682	2

Table A.1 (Continued)

ENTITY	Currency	Code		Decimal
		Alphabetic	Numeric	Point
SENEGAL	CFA Franc BCEAO+	XOF	952	0
SEYCHELLES	Seychelles Rupee	SCR	690	2
SIERRA LEONE	Leone	SLL	694	2
SINGAPORE	Singapore Dollar	SGD	702	2
SLOVAKIA	Slovak Koruna	SKK	703	2
SLOVENIA	Tolar	SIT	705	2
SOLOMON ISLANDS	Solomon Islands Dollar	SBD	090	2
SOMALIA	Somali Shilling	SOS	706	2
SOUTH AFRICA	Rand	ZAR	710	2
SPAIN	Spanish Peseta	ESP	724	0
SRI LANKA	Sri Lanka Rupee	LKR	144	2
SUDAN	Sudanese Dinar	SDD	736	2
SURINAME	Surinam Guilder	SRG	740	2
SVALBARD AND JAN MAYEN ISLANDS	Norwegian Krone	NOK	578	2
SWAZILAND	Lilangeni	SZL	748	2
SWEDEN	Swedish Krona	SEK	752	2
SWITZERLAND	Swiss Franc	CHF	756	2
SYRIAN ARAB REPUBLIC	Syrian Pound	SYP	760	2
TAIWAN, PROVINCE OF CHINA	New Taiwan Dollar	TWD	901	2
TAJIKISTAN	Tajik Ruble	TJR	762	0

+ CFA Franc BCEAO; Responsible authority: Banque Centrale des Etats de l'Afrique de l'Ouest.

Table A.1 (Continued)

ENTITY	Currency	Code		Decimal
		Alphabetic	Numeric	Point
TANZANIA, UNITED REPUBLIC OF	Tanzanian Shilling	TZS	834	2
THAILAND	Baht	THB	764	2
TOGO	CFA Franc BCEAO+	XOF	952	0
TOKELAU	New Zealand Dollar	NZD	554	2
TONGA	Pa'anga	TOP	776	2
TRINIDAD AND TOBAGO	Trinidad and Tobago Dollar	TTD	780	2
TUNISIA	Tunisian Dinar	TND	788	3
TURKEY	Turkish Lira	TRL	792	0
TURKMENISTAN	Manat	TMM	795	2
TURKS AND CAICOS ISLANDS	US Dollar	USD	840	2
TUVALU	Australian Dollar	AUD	036	2
UGANDA	Uganda Shilling ++	UGX	800	0
UKRAINE	Hryvnia	UAH	980	2
UNITED ARAB EMIRATES	UAE Dirham	AED	784	2
UNITED KINGDOM	Pound Sterling	GBP	826	2
UNITED STATES	US Dollar	USD	840	2
	(Same day)*	USS	998	2
	(Next day)*	USN	997	2
UNITED STATES MINOR OUTLAYING ISLANDS	US Dollar	USD	840	2

+ CFA Franc BCEAO; Responsible authority: Banque Centrale des Etats de l'Afrique de l'Ouest.

++ The Uganda Shilling was denominated as from 18 May 1987.

* Funds code [See table A.2(E) for definitions of funds types].

Table A.1 (Continued)

ENTITY	Currency	Code		Decimal Point
		Alphabetic	Numeric	
URUGUAY	Peso Uruguayo	UYU	858	2
UZBEKISTAN	Uzbekistan Sum	UZS	860	2
VANUATU	Vatu	VUV	548	0
VATICAN CITY STATE (HOLY SEE)	Italian Lira	ITL	380	0
VENEZUELA	Bolivar	VEB	862	2
VIETNAM	Dong	VND	704	2
VIRGIN ISLANDS (BRITISH)	US Dollar	USD	840	2
VIRGIN ISLANDS (U.S.)	US Dollar	USD	840	2
WALLIS AND FUTUNA ISLANDS	CFP Franc	XPF	953	0
WESTERN SAHARA	Moroccan Dirham	MAD	504	2
YEMEN	Yemeni Rial	YER	886	2
YUGOSLAVIA	New Dinar	YUM	891	2
ZAMBIA	Kwacha	ZMK	894	2
ZIMBABWE	Zimbabwe Dollar	ZWD	716	2

Table A.1 (Continued)

ENTITYCurrency	Code	Decimal		
		Alphabetic	Numeric	Position
Entity not applicable	Gold	XAU	959	N.A.
	Bond Markets Units			
	European Composite Unit (EURCO)	XBA	955	N.A.
	European Monetary Unit (E.M.U.-6)***	XBB	956	N.A.
	European Unit of Account 9 (E.U.A.- 9)	XBC	957	N.A.
	European Unit of Account 17 (E.U.A.- 17)	XBD	958	N.A.
	Palladium	XPD	964	N.A.
	Platinum	XPT	962	N.A.
	Silver	XAG	961	N.A.

*** E.M.U.-6 is sometimes known as the European Currency Unit. This should not be confused with the settlement unit of the European Monetary Cooperation Fund (E.M.C.F.) which has the same name (see entry for 'European Monetary Cooperation Fund' in this table).

Table A.1 (Continued)

ENTITY	Currency	Code		Decimal
		Alphabetic	Numeric	Position
Entity not applicable	Special settlement currencies			
	UIC-Franc	XFU	Nil	N.A.
	Gold-Franc	XFO	Nil	N.A.
	Codes specifically reserved for testing purposes	XTS	963	N.A.
	The codes assigned for transactions where no currency is involved are:	XXX	999	N.A.
	euro*	EUR*	978	2

* On 1 January 1999, the euro will become the currency of those Member States of the European Union which adopt the single currency in accordance with the Treaty establishing the European Community. This code has been issued now so that technical preparations can be started. The code element "EU" has been reserved by the ISO 3166 Maintenance Agency for use within ISO 4217 where "R" has been appended to make an acceptable mnemonic code.

Table A.2 Funds Codes Registered with the Maintenance Agency

CURRENCY AUTHORITY	Currency	Fund Type	Code Alphabetic	Code Numeric	Decimal Position
BOLIVIA		Mvdol	BOV	984	2
CHILE		Unidades de Fomento	CLF	990	0
MEXICO		Mexican Unidad de Inversion (UDI)	MXV	979	2
UNITED STATES	US Dollar	Same day	USS	998	2
		Next day	USN	997	2

Definitions of the fund types listed above

BOV: For indexation purposes and denomination of certain finacial instruments (ex. treasury bills). The Mvdol is set daily by the Central Bank of Bolivia based upon the official USD/BOB rate.

CLF: This development unit has been approved by the Chilean government for use in insurance transactions (with effect from 10 April 1980).

ECV: A daily indexation mechanism set by the Ecuadorian Central Bank. The UVC is set according to the variation of the Consumer price Index (Urban), as compiled by the National Census and Statistics Institute (INEC).

MXV : The UDI is an inflation adjusted mechanism set by the Central Bank of Mexico according to the variation in the Mexican Consumer Price Index. The value of the UDI is expressed in terms of Mexican Pesos per UDI. It is used to denominate mortgage loans, some bank deposits with maturities of 3 month or more and Government bonds (UDIBONOS).

USN: "Next day" funds are immediately available for transfer in like funds, and subject to settlement, available the next business day for same day funds transfer or withdrawal in cash.

USS: "Same day" funds are immediately available for transfer today or for withdrawal in cash, subject to the settlement of the transaction through the payment mechanism used.

(USD designates the US Dollar, the currency designator when an accumlation of amounts contains more than one funds type.)

Table A.3 Codes for Historic Denomination of Currencies and Funds

ENTITY	Historic Currencies	Code	Numeric	WD
ALBANIA	Old Lek	ALK *	-	12/89
ANGOLA	Kwanza	AOK	-	03/91
ARGENTINA	Peso Argentino	ARP	-	07/85
	Austral	ARA	-	01/92
	Peso	ARY*	-	
1989/1990				
BELGIUM	Convertible Franc	BEC	993	03/90
	Financial Franc	BEL	992	03/90
BOLIVIA	Peso	BOP	-	02/87
BOSNIA & HERZEGOVINA	Dinar	BAD	070	09/97
BRAZIL	Cruzeiro	BRB	-	03/86
	Cruzado	BRC	-	02/89
	New Cruzado	BRN	-	03/90
	Cruzeiro	BRE	076	08/93
	Cruzeiro Real	BRR	987	07/94
BULGARIA	Lev A/62	BGK*	-	
1989/1990				
	Lev A/52	BGJ*	-	
1989/1990				
BURMA#	N/A	BUK	-	02/90
CHINA	Peoples Bank Dollar	CNX*	-	12/89
CROATIA	Dinar	HRD	-	01/95
CZECHOSLOVAKIA	Krona A/53	CSJ*	-	1989/1990
	Koruna	CSK	200	03/93
ECUADOR	Sucre	ECS	218	9/00
	Unidad del Valor constante (UVC)*	ECV	983	9/00
EQUATORIAL GUINEA	Ekwele	GQE	226	06/86

Ekwele	EQE*	-	12/89

* Non ISO code
# Change in country name

Table 3 (Continued)

ENTITY	Historic Currencies	Code	Numeric	WD
EUROPEAN MONETARY COOPERATION FUND (EMCF)**	European Currency Unit (E.C.U)	XEU	954	01/99
GERMAN DEMOCRATIC REPUBLIC	Mark der DDR	DDM	278	07/90 to 09/90
GEORGIA	Georgian Coupon	GEK	268	10/95
GUINEA	Syli	GNS	-	02/86
	Syli	GNE*	-	12/89
GUINEA BISSAU Between 1978-	Guinea Escudo	GWE	-	1981
ICELAND 1989/1990	Old Krona	ISJ*	-	
ISRAEL 1989/1990	Old Shekel	ILR*	-	
Between 1978-	Pound	ILP	-	1981
LESOTHO	Maloti	LSM	-	05/85
LAO	Kip Pot Pol	LAJ*	-	12/89
LATVIA	Latvian Ruble	LVR	-	12/94
LITHUANIA	Talonas	LTT	-	07/93
LUXEMBOURG	Convertible Franc	LUC	989	03/90
	Financial Franc	LUL	988	03/90
MALDIVES	Maldive Rupee	MVQ*	-	12/89
MALI	Mali Franc	MAF*	-	12/89
		MLF	446	11/84
MALTA	Maltese Pound	MTP	-	06/83
MEXICO	Mexican Peso	MXP	-	01/93

* Non ISO code

Table 3 (Continued)

ENTITY	Historic Currencies	Code	Numeric	WD
MOZAMBIQUE Between 1978-1981	Mozambique Escudo	MZE	-	
NICARAGUA	Cordoba	NIC	-	10/90
PERU	Sol	PES	-	02/86
	Inti	PEI	-	07.91
1989/1990	Sol	PEH*	-	
POLAND	Zloty	PLZ	616	01/97
ROMANIA 1989/1990	Leu A/52	ROK*	-	
SOUTH AFRICA	Financial Rand	ZAL	991	03/95
SOUTHERN RHODESIA# Between 1978-1981	Rhodesian Dollar	RHD	-	
SPAIN Between 1981-1983	Spanish Peseta ("A" Account)	ESA	996	
	(convertible Peseta Accounts)	ESB	995	12/94
SUDAN	Sudanese Pound	SDP	-	06/98
UNION OF SOVIET SOCIALIST REPUBLICS#	Rouble	SUR	-	12/90
YEMEN, DEMOCRATIC OF	Yemeni Dinar	YDD	720	09/91
UGANDA	Uganda Shilling	UGS	-	05/87
1989/1990	Old Shilling	UGW*	-	

UKRAINE	Karbovanet	UAK	804	09/96

* Non ISO code
# Change in country name.

Table 3 (Continued)

ENTITY	Historic Currencies	Code	Numeric	WD
URUGUAY	Old Uruguay Peso	UYN*	-	12/89
	Uruguayan Peso	UYP	-	03/93
VIETNAM 1989/1990	Old Dong	VNC*	-	
YUGOSLAVIA	New Yugoslavian Dinar	YUD	-	01/90
	Yugoslavian Dinar	YUN	890	11/95
ZAIRE	Zaire	ZRZ	-	02/94
	New Zaire	ZRN	180	06/99
ZIMBABWE	Rhodesian Dollar	ZWC*	-	12/89
ENTITY AND CURRENCY NOT APPLICABLE	RINET Funds Code	XRE	N/A	11/99

* Non ISO code

ANNEX

INFORMATION TO BE PROVIDED BY THOSE MAKING APPLICATION FOR THE ISSUE OF NEW CODES, AMENDMENTS AND DELETIONS.

Applications for additions or changes to the code lists are acceptable from any source. However, on order to ensure rapid processing by the Secretaries, the information required from applicants has been laid down as follows:

(a) Name of entity

(b) Name of currency

(c) The institution responsible for the currency (name and place of operation).

(d) Requirements:

(1) Whether currency or funds code: if funds code, give definition and proposed use;

(2) If new code, make proposal;

(3) If revision, state existing code and make proposal;

(4) If deletion, indicate code to be deleted;

(e) Reason for application;

(f) Evidence of support (currency code only);

(g) Date of implementation (indicate if special conditions of urgency apply);

(h) Application submitted by (name, address, telephone, telex numbers, etc, of applicant);

(i) Date of application.

Applications should be addressed to

Miss A M Wadsworth Tel. (0181) 996 7466 National
Secretariat for ISO4217MA +44 181 996 7466 International
BSI
389 Chiswick High Road Fax (0181) 996 7466 National
London +44 181 996 7466 International
W4 4AL United Kingdom

Appendix 2

Battery Operated Card Reader

A2.1 Special Application

The Battery Operated Card Reader described below is a special application of the MDB/ICP specification (non-standard) and is not sanctioned by NAMA. It is provided here to document an application that exists in use today.

A2.2 Extension to MDB/ICP – Card Reader Using Standby Feature

Some Vending machines use battery operated equipment. According to this feature, these machines and all devices used within these machines must provide a standby operating mode.

During standby operation - necessary for saving battery power while the machine is not in use - all devices shall consume a minimum standby current. Any device is equipped with some hardware wake-up mechanism. Both standby current and wake-up mechanism is to be defined in the device related hardware specification.

After wake-up, a device uses normal operating current, until a defined shutdown sequence is established and the device enters standby mode again.

The following specification shows the extensions and procedures for a normal MDB/ICP card reader and VMC-controller necessary to do wake-up and shut down sequences. The hardware specification related to wake-up is a separate BDTA-document. To understand the following details, it is necessary to know, that a separate bi-directional wake-up pin is applied to the card-reader. Pulling the wake-up line (from the card-reader while a card is inserted), both card-reader and VMC will be brought to normal operation mode.

A2.3 Extension to MDB/ICP – SETUP Config Data

SETUP (11H)	Config Data (00H) Y1	VMC Feature Level Y2	Columns on Display Y3	Rows on Display Y4	Display Info Y5

Y1 : Configuration data.
VMC is sending its configuration data to reader.

Y2 : VMC Feature Level.
Indicates the feature level of the VMC. The available feature levels are:

01 - The VMC is not capable or will not perform the advanced features as specified in Table 1: COMMANDS & RESPONSES following Section 7.3.2. The reader will not provide advanced information to the VMC, but can do the advanced features internally (transparently to the VMC). The reader has no revaluation capability.

02 - The VMC is capable and willing to perform the advanced features as specified in Table 1: COMMANDS & RESPONSES following Section 7.3.2. The reader will provide advanced information to the VMC (if possible) and will not do the advanced features internally.

03 - The VMC is able to support level 02, but also supports some or all of the optional features listed in the EXPANSION ID command (i.e., file transfer, 32 bit credit, multi-currency / language features, negative vend, and / or data entry).

81H: VMC is Level 01, but battery operated.
82H: VMC is Level 02, but battery operated.
83H: VMC is Level 03, but battery operated.

Y3 : Columns on Display. The number of columns on the display. Set to 00H if the display is not available to the reader.

Y4 : Rows on Display.
The number of rows on the display.

Y5 : Display Information - xxxxxyyy
xxxxx = Unused
yyy = Display type
000 : Numbers, upper case letters, blank and decimal point.
001 : Full ASCII
010-111: Unassigned

Reader Config Data (01H) Z1	Reader Feature Level Z2	Country / Currency Code High Z3	Country / Currency Code Low Z4	Scale Factor Z5	Decimal Places Z6	Application Maximum Response Time Z7	Miscellaneous Options Z8

Z1 : READER - Configuration data.
Indicates the payment media reader is responding to a SETUP - Configuration data request from the VMC.

Z2 : Reader Feature Level.
Indicates the feature level of the reader. Currently feature levels are:

01 - The reader is not capable or will not perform the advanced features as specified in Table 1: COMMANDS & RESPONSES following Section 7.3.2. The reader will not provide advanced information to the VMC, but can do the advanced features internally (transparently to the VMC). The reader has no revaluation capability.

02 - The reader is capable and willing to perform the advanced features as specified in Table 1: COMMANDS & RESPONSES following Section 7.3.2. The reader will provide advanced information to the VMC (if possible) and will not do the advanced features internally.

03 - The reader is able to support level 02, but also supports some or all of the optional features listed in the EXPANSION ID command (i.e., file transfer, 32 bit credit, multi-currency / language features, negative vend, and / or data entry).

80H: This bit is additionally set, if the reader is capable to work in battery operation mode and should be compared with the VMC against its own working mode. This is also done from the reader against the VMCs request in Y2.

Z3-Z4 : Country / Currency Code - packed BCD.
The packed BCD country / currency code of the changer can be sent in two different forms depending on the value of the left most BCD digit.

If the left most digit is a 0, the International Telephone Code is used to indicate the country that the changer is set-up for. For example, the USA code is 00 01H (Z3 = 00 and Z4 = 01).

If the left most digit is a 1, the latest version of the ISO 4217 numeric currency code is used (see Appendix A1). For example, the code for the US dollar is 18 40H (Z2 = 18 and Z3 = 40) and for the Euro is 1978 (Z3 = 19 and Z4 = 78). Use FFFFh if the country code in unknown.

For level 3 cashless readers, it is mandatory to use the ISO 4217 numeric currency code (see Appendix A1).

Z5 : Scale Factor.
The multiplier used to scale all monetary values transferred between the VMC and the reader.

Z6 : Decimal Places.
The number of decimal places used to communicate monetary values between the VMC and the payment media reader.

All pricing information sent between the VMC and the payment media reader is scaled using the scale factor and decimal places. This corresponds to:

$$\text{ActualPrice} = P \cdot X \cdot 10^{-Y}$$

where P is the scaled value send in the price bytes, and X is the scale factor, and Y is the number of decimal places. For example if there are 2 decimal places and the scale factor is 5, then a scaled price of 7 will mean an actual of 0.35.

Z7 : Application Maximum Response Time - seconds.
The maximum length of time a reader will require to provide a response to any command from the VMC. The value reported here supercedes the payment reader's default NON-RESPONSE time defined in section 7.5 if the value reported here is greater.

Z8 : Miscellaneous Options - xxxxyyyy

xxxx:	Unused (must be set to 0)
yyyy:	Option bits
b0=0:	The payment media reader is NOT capable of restoring funds to the user's payment media or account. Do not request refunds.
b0=1:	The payment media reader is capable of restoring funds to the user's payment media or account. Refunds may be requested.
b1=0:	The payment media reader is NOT multivend capable. Terminate session after each vend.
b1=1:	The payment media reader is multivend capable. Multiple items may be purchased within a single session.
b2=0:	The payment media reader does NOT have a display.
b2=1:	The payment media reader does have its own display.
b3=0:	The payment media reader does NOT support the VEND/CASH SALE subcommand.
b3=1:	The payment media reader does support the VEND/CASH SALE subcommand.
b4-b7=0	Any future options must be covered by the EXPANSION COMMAND option bits.

Note: The following changes are the only changes to upgrade to battery operated readers:

If a VMC is battery operated, it signals the card reader with the flag 80H to work in battery operation mode. Within byte Z2 the reader also sets the flag to 80H to signal standby feature capability.

If only one of both is in standby capability, this results in an configuration error and the manufacturers should deal with handling of this condition. Assume that at least one device will not enter standby mode and therefore battery lifetime is dramatically reduced!

A2.4 VMC-Reader Operation Sequences

The VMC and the Reader should operate during battery mode in the following way:

After wake-up, the VMC starts with the normal sequences:

Reset
Setup/Config
MAX/MIN-price
Identify
Enable
Poll

During these sequences, the VMC has two possibilities to signal the Card-Reader, not to enter standby-mode again:

Pulling the wake-up pin to low level
Running poll sequences in continuos timing.

If neither the wake-up pin is driven low, nor any command is further sent to the card reader, the reader enters standby state after its Application Maximum Response Time (normally defined to 5 sec in ICP, but sent in byte Z7 of status response)

During card operation, the sequences continue normally with

Begin Session

Vend Request
Vend Accepted
Vend Success
Cancel Session/Session Complete

Whenever a cancel session or session complete command is received, the reader should stop all internal work after a defined timeout period (Application Maximum Response Time) is finished after the last command sequence and after the wake-up pin is not pulled low.

The VMC should stop polling after the cancel session or session complete command and additionally should no longer pull wake-up pin.

If even the reader or the VMC may wish any further communication (i.e. for additionally trailing display messages or multi vend purposes or etc.) the reader can use any non idle answer to the poll command (i.e. the display message) whereas the VMC can continue polling or pulling the wake-up pin.

Note that the wake-up pin may not be used from the reader to hold on operation, cause dynamic system consideration and of course holding more devices within the system in normal operation mode is not a good job.

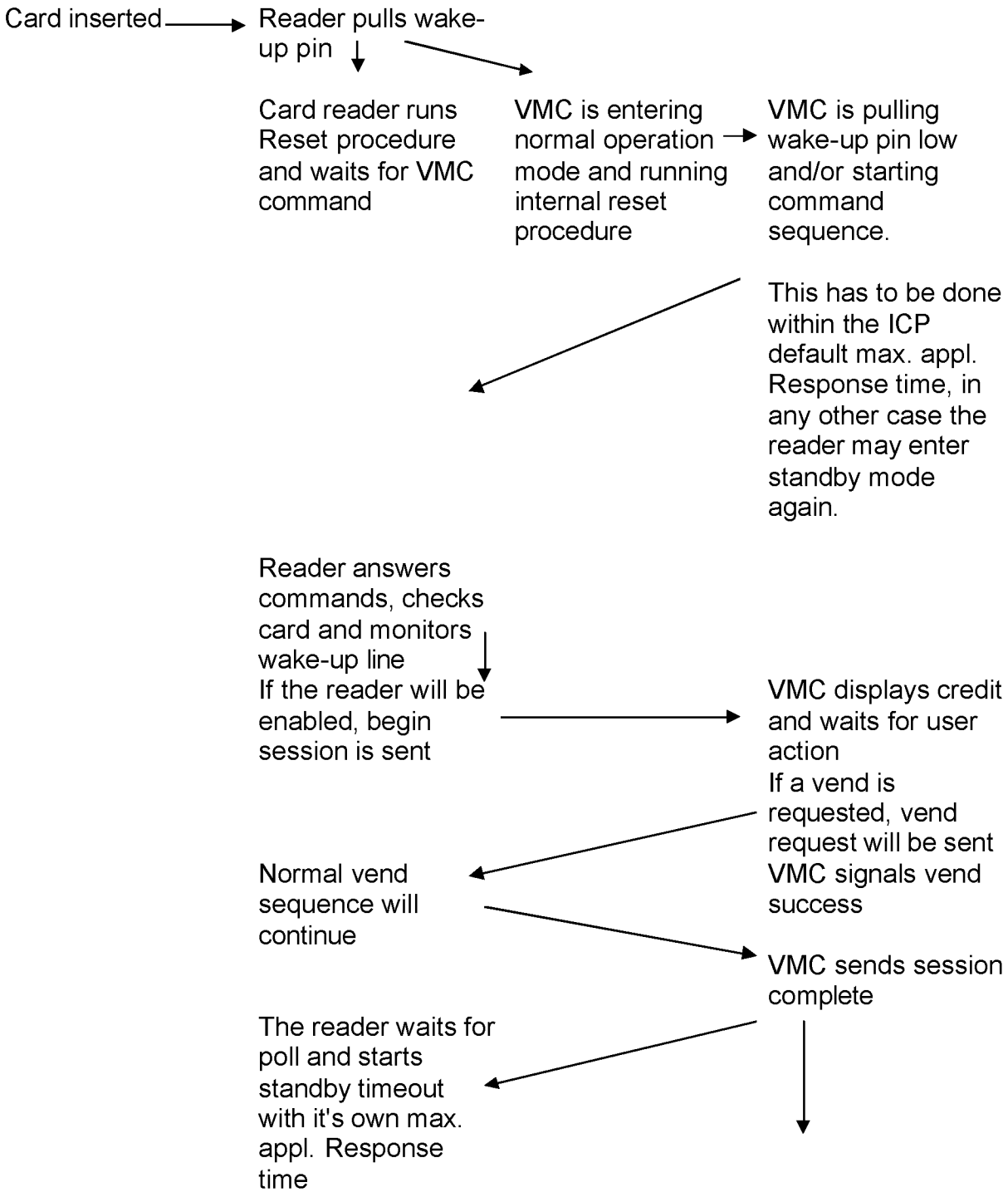
The reader should be in a power saving mode after this timeout period where power consumption is less than 10 uA.

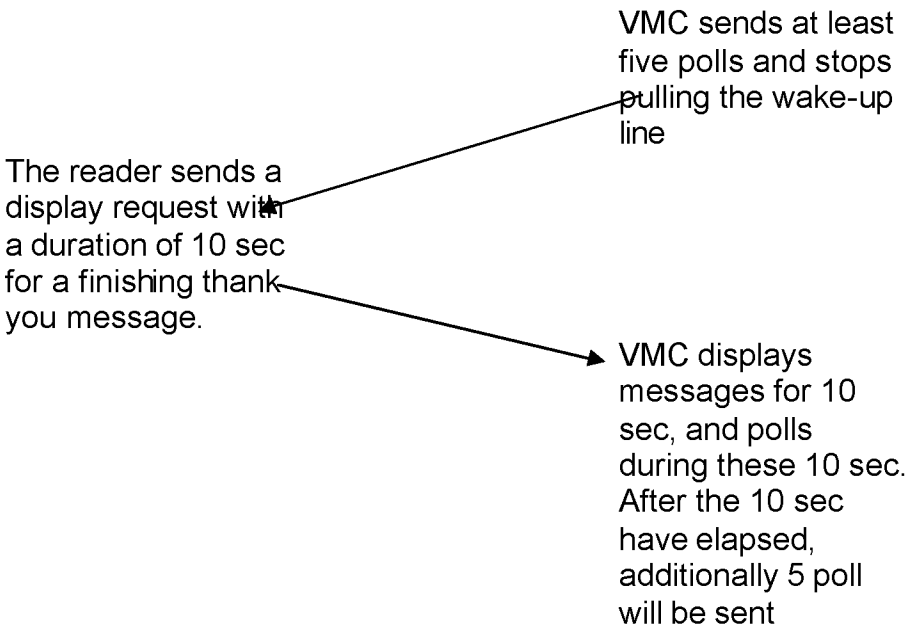
To allow the reader holding VMC operating, at least 5 poll have to be sent, after the cancel session or session complete. If any one of these polls is answered different with only a ACK, 5 polls have to be sent again. Note, that if a display message is sent, display time is added!

If the reader entered standby state, and a new card is inserted, the procedure starts a again.

Whenever during this next session, the reader should avoid all unnecessary work, i.e. display messages like „reader xyz, Software version 99.4711“ or „checking RAM“ and so on should be avoided. While in battery operation, the user has inserted a card and is waiting for display of his fund, to continue with a vend and is not interested in service related messages.

A2.5 Session Example



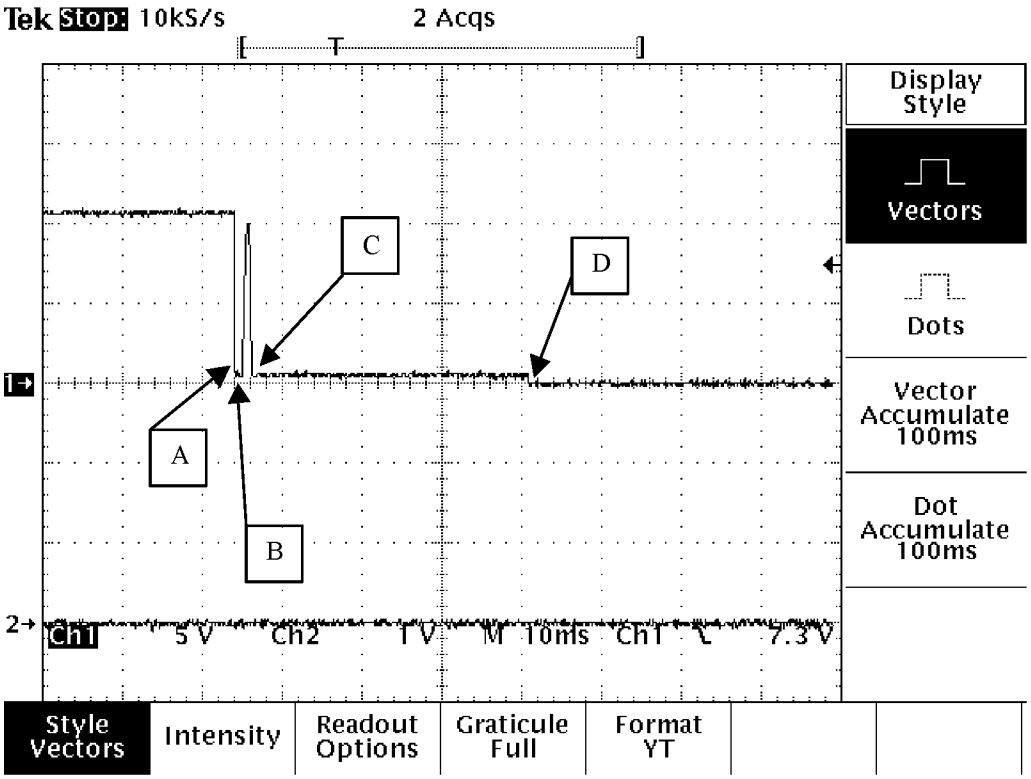


A2.6 Hardware Considerations

Hardware Considerations

Within this special battery operation, the pin 3 of MDB/ICP connector is used as a wake-up signal. Refer to special BDTA-hardware specification.

To show an example of the timing for this pin, refer to the following diagram, which gives an example of all special timing problems related to more than one wake-up condition.



Position A: mechanical switch on VMC is pulling pin 3 low (i.e. door switch)
Position B: mechanical switch is released
Position C: card reader has finished reset routines and pulls pin 3 low
Position D: VMC has finished reset routines and pulls pin 3 low too.

If a card is inserted first, pin 3 may be pulled low first at position B.
If VMC is waked up via other means, maybe card reader is waked up at position D first.

In any case, this is a good example to clarify different waveform conditions on pin 3. Please note that any device may release pin 3 after a short duration (<1ms) cause pin 3 should work as dynamically wake-up. Holding pin 3 permanently low may prevent other

devices from wake-up, i.e. after all devices ran into timeout and one is still holding pin 3, the other can no longer enter ready state (Note i.e. to door-switches etc.)

(this page intentionally left blank)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

24341 7590 11/01/2023
Morgan, Lewis & Bockius LLP (PA)
1400 Page Mill Road
Palo Alto, CA 94304-1124

EXAMINER	
HASSAN, AURANGZEB	
ART UNIT	PAPER NUMBER
2184	

DATE MAILED: 11/01/2023

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/443,802	07/27/2021	Paresh K. Patel	104402-5053-US	7874

TITLE OF INVENTION: DEVICE AND METHOD FOR PROVIDING EXTERNAL ACCESS TO MULTI-DROP BUS PERIPHERAL DEVICES

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0.00	$0.00	$480	02/01/2024

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 40% the amount of undiscounted fees, and micro entity fees are 20% the amount of undiscounted fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications. **Because electronic patent issuance may occur shortly after issue fee payment, any desired continuing application should preferably be filed prior to payment of this issue fee in order not to jeopardize copendency.**

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

24341 2590 11/01/2023
Morgan, Lewis & Bockius LLP (PA)
1400 Page Mill Road
Palo Alto, CA 94304-1124

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via EFS-Web or by facsimile to (571) 273-2885, on the date below.

(Typed or printed name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/443,802	07/27/2021	Paresh K. Patel	104402-5053-US	7874

TITLE OF INVENTION: DEVICE AND METHOD FOR PROVIDING EXTERNAL ACCESS TO MULTI-DROP BUS PERIPHERAL DEVICES

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0.00	$0.00	$480	02/01/2024

EXAMINER	ART UNIT	CLASS-SUBCLASS
HASSAN, AURANGZEB	2184	705-071000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

☐ Change of correspondence address (or Change of Correspondence Address form PTO/AIA/122 or PTO/SB/122) attached.

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/AIA/47 or PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 .....
2 .....
3 .....

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. Fees submitted: ☐ Issue Fee ☐ Publication Fee (if required)

4b. Method of Payment: *(Please first reapply any previously paid fee shown above)*

☐ Electronic Payment via Patent Center or EFS-Web ☐ Enclosed check ☐ Non-electronic payment by credit card (Attach form PTO-2038)

☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. .....

5. **Change in Entity Status** (from status indicated above)

☐ Applicant certifying micro entity status. See 37 CFR 1.29

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

☐ Applicant asserting small entity status. See 37 CFR 1.27

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

☐ Applicant changing to regular undiscounted fee status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature ..... Date .....

Typed or printed name ..... Registration No. .....

Petitioner Exhibit 1002-0989

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/443,802	07/27/2021	Paresh K. Patel	103402-5053-US	7874

24341 7590 11/01/2023

Morgan, Lewis & Bockius LLP (PA)
1400 Page Mill Road
Palo Alto, CA 94304-1124

EXAMINER	
HASSAN, AURANGZEB	

ART UNIT	PAPER NUMBER
2184	

DATE MAILED: 11/01/2023

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Corrected ***Notice of Allowability***	**Application No.** 17/443,802	**Applicant(s)** Patel, Paresh K.	
	Examiner AURANGZEB HASSAN	**Art Unit** 2184	**AIA (FITF) Status** Yes

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☑ This communication is responsive to IDS' 9/28/23 and 10/3/23.
   ☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3. ☑ The allowed claim(s) is/are 1-3,5-11 and 13-20 . As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information , please see **http://www.uspto.gov/patents/init_events/pph/index.jsp** or send an inquiry to **PPHfeedback@uspto.gov.**

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   Certified copies:

   a) ☐ All b) ☐ Some* c) ☐ None of the:

   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____ .

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☑ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date 9/28/23; 10/3/23.
3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material _____.
4. ☐ Interview Summary (PTO-413), Paper No./Mail Date. _____.
5. ☐ Examiner's Amendment/Comment
6. ☑ Examiner's Statement of Reasons for Allowance
7. ☐ Other _____.

/HENRY TSAI/
Supervisory Patent Examiner, Art Unit 2184

DETAILED ACTION

Notice of Pre-AIA or AIA Status

The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after allowance or after an Office action under *Ex Parte Quayle*, 25 USPQ 74, 453 O.G. 213 (Comm'r Pat. 1935). Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, prosecution in this application has been reopened pursuant to 37 CFR 1.114. Applicant's submission filed on 9/28/23 has been entered.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on 9/28/23 and 10/3/23 are being considered by the examiner.

Allowable Subject Matter

Claims 1 – 3, 5 – 11, and 13 – 20 are allowed.

The following is an examiner's statement of reasons for allowance: Applicant's submission of IDS' on 9/28/23 and 10/3/23 have been fully considered and therein the claim limitations are in allowable format.

The prior art fails to teach or suggest alone or in combination the limitations of the claims as a whole including a slave interface coupled to a machine controller of a machine's MDB further coupled to a host. Further including registering the device as a slave to the machine controller, registering a first peripheral as a slave to the device and receiving/validating a request from a mobile device to access the first peripheral where the configuration of slaves and master are handled to interface request handling therein. Prior art is further silent on modification to combine such features and functionality and is therefore deemed allowable.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to AURANGZEB HASSAN whose telephone number is (571)272-8625. The examiner can normally be reached on 7 AM to 3 PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at http://www.uspto.gov/interviewpractice.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Henry Tsai can be reached on 571-272-4176. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AH

/HENRY TSAI/
Supervisory Patent Examiner, Art Unit 2184

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 January 2017 (19.01.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/010936 A1

(51) International Patent Classification:
G06Q 20/32 (2012.01) *G06Q 40/02* (2012.01)

(21) International Application Number:
PCT/SG2016/050297

(22) International Filing Date:
27 June 2016 (27.06.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10201505585T 16 July 2015 (16.07.2015) SG

(71) Applicant: **TOUREGO GLOBAL PTE. LTD.** [SG/SG]; 141 Cecil Street, #08-07 Tung Ann Association Building, Singapore 069541 (SG).

(72) Inventor: **TAN, Tie Wee**; c/o 14, Robinson Road, #08-01A, Far East Finance Building, Singapore 048545 (SG).

(74) Agent: **YUSARN AUDREY**; 24 Raffles Place, #27-01, Clifford Centre, Singapore 048621 (SG).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

(54) Title: SYSTEM AND METHOD FOR FACILITATING REFUNDS

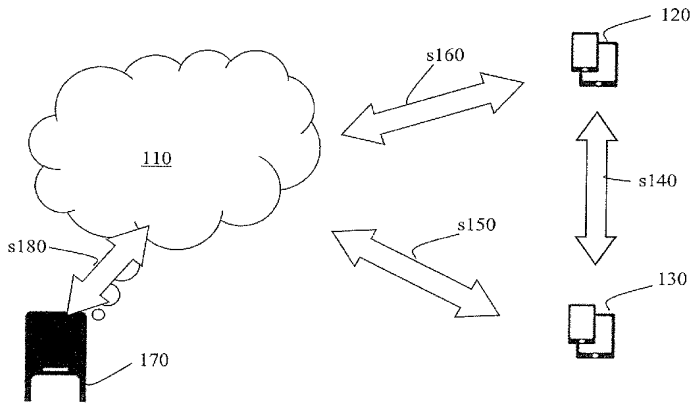


FIG. 1

(57) Abstract: A system and method for facilitating a refund to a user is disclosed. The system comprises a mobile device having means to communicate with a central processor for the generation of a refund account and a unique identifier; the unique identifier associated with the user; a computer device operable to access the unique identifier for verification upon receipt of a refund qualifying transaction associated with the user, the computer device further operable to send information relating to the refund qualifying transaction and generate a refund request upon successful verification; and the central processor operable to be in data communication with the computer device to receive and process the refund request, the central processor further configured to generate an electronic ticket to be sent to the mobile device for the generation of a refund upon successful process of the refund request.

WO 2017/010936 A1

SYSTEM AND METHOD FOR FACILITATING REFUNDS

FIELD OF THE INVENTION

The present invention relates to a system and method for facilitating refunds, such as but not limited to a tax refund for purchases made by tourists while travelling within their destination countries ("tourist refund").

BACKGROUND ART

The following discussion of the background to the invention is intended to facilitate an understanding of the present invention only. It should be appreciated that the discussion is not an acknowledgement or admission that any of the material referred to was published, known or part of the common general knowledge of the person skilled in the art in any jurisdiction as at the priority date of the invention.

In line with tax principle while at the same time attracting tourists, jurisdictions that levy a Value-Added Tax (VAT) or Goods and Services Tax (GST) or similar would allow individuals such as tourists to claim a tax refund on the goods purchased within a country. This tax refund would be administered upon the departure of the tourists from the country; provided certain criteria are met such as the goods are unopened, unused and exported out of the country.

The Revenue/Customs authorities of the country can administer tourist refund scheme(s), although it is increasingly popular for commercial contractors to be appointed to administer on their behalf. These commercial contractors typically work in cooperation with the Revenue/Customs authorities to ensure that the appropriate amount of tax refunds is paid out to the tourist(s). A service or commission fee will be charged which would form the bulk of the revenue model for the commercial contractor.

Two methods of administration of tourist refund schemes are currently being operated in the world. One is based on the authentication of paper forms and mandatory checks including physical checks at the point of departure by