UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC.,
Petitioner,

v.

KAIFI LLC,
Patent Owner.

IPR2025-00626
U.S. Patent No. 11,082,518

DECLARATION OF HENRY HOUH, UNDER 37 C.F.R. § 1.68 IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

I. INTRODUCTION 5

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE 8

III. LEVEL OF ORDINARY SKILL IN THE ART 12

IV. RELEVANT LEGAL STANDARDS 13

V. TECHNICAL BACKGROUND 15

A. Smart Home Automation 15

B. Relation Servers 16

C. Data Storage and Format 17

VI. THE ’518 PATENT 19

VII. CLAIM CONSTRUCTION 21

VIII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE 22

A. Ground 1: Claims 1-20 are obvious under §103 over Ermis. 23

1. Summary of Ermis 23

2. Detailed Analysis of the Claims 28

a) Claim 1 28

b) Claim 2 63

c) Claim 3 68

d) Claim 4 70

e) Claim 5 75

f) Claim 6 82

g) Claim 7 84

h) Claim 8 89

i) Claim 9 90

j) Claim 10 92

k) Claim 11 95

l) Claim 12 99

m) Claim 13 100

n) Claim 14 104

o) Claim 15 105

p) Claim 16 107

q) Claim 17 109

r) Claim 18 111

s) Claim 19 114

t) Claim 20 118

B. Ground 2: Claims 1-20 is obvious under 35 U.S.C. § 103 over Ermis in view of Nakano..... 120

1. Summary of Nakano 120

2. Reasons to Combine Ermis and Nakano..... 124

3. Detailed Analysis of the Claims 131

a) Claim 1 131

b) Claims 2-14 136

c) Claim 15 ... 136

d) Claims 16-20 ... 140

IX. DECLARATION ... 141

I, Henry H. Houh, Ph.D., do hereby declare as follows:

I. INTRODUCTION

1. I am making this declaration at the request of Amazon.com, Inc. in the matter of the *Inter Partes* Review of U.S. Patent No. 11,082,518 ("the '518 patent") to Jeong et al.

2. I am being compensated for my work in this matter at my standard hourly rate. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this proceeding. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

3. I have been asked to provide my opinions regarding whether the subject matter of claims 1-20 ("the Challenged Claims") of the '518 patent would have been obvious to a person having ordinary skill in the art ("POSITA") at the time of the alleged invention, in light of the prior art. It is my opinion that the Challenged Claims would have been obvious to a POSITA.

4. In the preparation of this declaration, I have studied:

a. the '518 patent, Ex.1001;

b. the prosecution history of the '518 patent ("'518 File History"), Ex.1002;

c. U.S. Patent Application Pub. 2012/0079091 to Ermis et al. ("Ermis"), Ex.1005; and

d. U.S. Patent No. 9,720,391 to Nakano et al. ("Nakano"), Ex.1023;

5. In forming the opinions expressed below, I have considered: the documents listed above; the relevant legal standards, including the standard for obviousness; and my own knowledge and experience based upon my work in the fields of digital networking and data processing as described below, as well as portions of the following additional materials:

a. U.S. Patent Application Pub. 2004/0131079 to Hedge et al. ("Hedge"), Ex.1006;

b. U.S. Patent Application Pub. 2003/0177283 to Hamilton et al. ("Hamilton"), Ex.1007;

c. U.S. Patent Application Pub. 2008/0256081 to Bui et al. ("Bui"), Ex.1008;

d. U.S. Patent Application Pub. 2004/0243580 to Markki et al. ("Markki"), Ex.1009;

e. U.S. Patent Application Pub. 2006/0070004 to Miller et al. ("Miller"), Ex.1010;

f. U.S. Patent Application Pub. 2007/0036461 to Atherton et al. ("Atherton"), Ex.1011;

g. U.S. Patent Application Pub. 2014/0359283 to Lee et al. ("Lee"), Ex.1013;

h. U.S. Patent No. 8,914,360 to Gailloux et al. ("Gailloux"), Ex.1014;

i. U.S. Patent Application Pub. 2014/0365964 to Forsblom et al. ("Forsblom"), Ex.1015;

j. U.S. Patent Application Pub. 2014/0325081 to Heinrich et al. ("Heinrich"), Ex.1016;

k. U.S. Patent No. 5,526,523 to Straub et al. ("Straub"), Ex.1024;

l. U.S. Patent No. 5,951,639 to MacInnis ("MacInnis"), Ex.1025;

m. U.S. Patent Application Pub. 2003/0177285 to Hunt et al. ("Hunt"), Ex.1026;

n. U.S. Patent Application Pub. 2012/0291068 to Khushoo et al. ("Khushoo"), Ex.1027;

o. Harper, Richard, "Inside the Smart Home," Springer, 2003 ("Harper"), Ex.1028;

p. U.S. Patent Application Pub. 2016/0085222 to Yabe et al. ("Yabe"), Ex.1029;

q. Schaefer, Robbie et al. "Profile Processing and Evolution for Smart Environments," 2006 ("Schaefer"), Ex.1030;

r. U.S. Patent No. RE44,534 to Li et al. ("Li"), Ex.1031;

s. U.S. Patent No. 8,868,424 to Moore et al. ("Moore"), Ex.1032;

t. U.S. Patent Application Pub. 2014/0236630 to Murata ("Murata"), Ex.1033;

u. U.S. Patent Application Pub. 2013/0041793 to Kashima et al. ("Kashima"), Ex.1034;

v. U.S. Patent Application Pub. 2014/0143419 to Vyatkin et al. ("Vyatkin"), Ex.1035;

w. U.S. Patent Application Pub. 2009/0319683 to Martin ("Martin"), Ex.1036;

x. U.S. Patent Application Pub. 2014/0040182 to Gilder et al. ("Gilder"), Ex.1037;

y. U.S. Patent Application Pub. 2005/0172280 to Ziegler et al. ("Ziegler"), Ex.1038; and

z. U.S. Patent Application Pub. 2005/0066325 to Mori et al. ("Mori"), Ex.1039.

6. Unless otherwise noted, all **emphasis** in any quoted material has been added. Claim terms are *italicized*.

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

7. My complete qualifications and professional experience are described in my *Curriculum Vitae*, a copy of which can be found in Ex.1004. The following is a brief summary of my relevant qualifications and professional experience.

8. I received a Ph.D. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology ("MIT") in 1998. Beforehand, I received a Master of Science degree in Electrical Engineering and Computer Science in 1991, a Bachelor of Science degree in Electrical Engineering and Computer Science in 1989, and a Bachelor of Science degree in Physics in 1990, all from MIT.

9. I am currently self-employed as an independent technical consultant. I am also president of a company that provides supplemental science, technology, engineering, and mathematics ("STEM") education to children of all ages.

10. I have early experience with connecting devices to a network. During my university studies, I focused on communications and data networking. As part of my doctoral research at MIT from 1991-1998, I worked as a research assistant in the Telemedia Network Systems ("TNS") group at the Laboratory for Computer Science. The TNS group built a high-speed gigabit ATM network and applications which ran over the network, such as remote video capture (including audio), processing, and display on computer terminals. Our high-speed data network carried multimedia data including video and audio data. Our system was operational by early to mid-1993, and some of our group (including myself) presented a paper on our system in September 1993 at the 18th Conference on Local Computer Networks held in Minneapolis. The cameras in our network were directly connected to the network and commanded across the network from software running on our computer workstations, and this may have been one of the first examples, if not the first, of a camera device directly connected to the network. As another demonstration, our group also connected a radio controller for a remote controlled toy car to the serial port of a computer, allowing the toy car, on which we mounted a video camera, to be driven from the computer and over the network. This toy car device received commands transmitted over a network from a remote computer, and video data from the toy car was transmitted wirelessly then over a computer network back to the user controller. On occasion, we allowed users visiting our web site to drive the toy car from their remote computer while they watched the video on their computer. The video stream was encoded by TNS-designed hardware, streamed over the TNS-designed network, and displayed using TNS-designed software.

11. I was also an early adopter of home automation, starting around 2015, with installing smart lights, a video doorbell, network garage door openers, cooking appliances, and numerous Amazon Alexa and Google Home devices, through which many home devices can be controlled.

12. I have been awarded several United States patents, and I have several patent applications pending including the following examples:

- U.S. Patent No. 7,975,296, "Automated security threat testing of web pages";
- U.S. Patent No. 7,877,736, "Computer language interpretation and optimization for server testing";
- U.S. Patent No. 7,801,910, "Method and apparatus for timed tagging of media content";
- U.S. Patent 7,590,542, "Method of generating test scripts using a voice-capable markup language";
- U.S. Patent No. 6,967,963, "Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data";
- U.S. Patent Application Publication No. 20070106685, "Method and apparatus for updating speech recognition databases and reindexing audio and video content using the same";
- U.S. Patent Application Publication No. 20070106693, "Methods and apparatus for providing virtual media channels based on media search";

- U.S. Patent Application Publication No. 20070106760, "Methods and apparatus for dynamic presentation of advertising, factual, and informational content using enhanced metadata in search-driven media applications";
- U.S. Patent Application Publication No. 20070112837, "Method and apparatus for timed tagging of media content";
- U.S. Patent Application Publication No. 20070118873, "Methods and apparatus for merging media content"; and
- U.S. Patent Application Publication No. 20090222442, "User-directed navigation of multimedia search results."

13. I have reviewed the '518 patent, and relevant excerpts of the prosecution history of the '518 patent. Based on my experience and education, I believe that I am qualified to offer opinions as to the knowledge and level of skill of one of ordinary skill in the art at the time of the invention of the '518 patent (which I further describe below).

III. LEVEL OF ORDINARY SKILL IN THE ART

14. I understand there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including (1) the levels of education and experience of persons working in the field at the time of the invention; (2) the sophistication of the technology; (3) the types of problems encountered in the field;

and (4) the prior art solutions to those problems.

15. I have been informed that the earliest alleged priority date for the ’518 patent is November 3, 2014.

16. A POSITA in the field of the ’518 patent, as of its earliest possible priority date in November 2014, would have been someone knowledgeable and familiar with electronic device communication that are pertinent to the ’518 patent. That person would have a bachelor’s degree in electrical engineering, computer engineering, computer science, or equivalent training, and approximately two years of experience working in the field of electronic device communication. Lack of work experience can be remedied by additional education, and vice versa.

17. For purposes of this Declaration, in general, and unless otherwise noted, my statements and opinions, such as those regarding my experience and what a POSITA would have understood or known generally (and specifically related to the references I consulted herein), reflect the knowledge that existed in the relevant field as of the alleged priority date of the ’518 patent (i.e., November 3, 2014). Unless otherwise stated, when I provide my understanding and analysis below, it is consistent with the level of a POSITA prior to the alleged priority date of the ’518 patent.

IV. RELEVANT LEGAL STANDARDS

18. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the ’518 patent, I am relying on certain basic legal principles that Amazon’s counsel has explained to me. These principles are discussed below.

19. I understand that prior art to the ’518 patent includes patents and printed publications in the relevant art that predate the priority date of the ’518 patent.

20. I have been informed by Amazon’s counsel that a claimed invention is unpatentable under 35 U.S.C. § 103 if the differences between the claimed invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA. I have also been informed by Amazon’s counsel that the obviousness analysis considers factual inquiries, including the level of ordinary skill in the art, the scope and content of the prior art, and the differences between the prior art and the claimed subject matter.

21. I have been further informed by Amazon’s counsel that there are several recognized rationales for combining references or modifying a reference to show obviousness of claimed subject matter. These rationales include: (a) combining prior art elements according to known methods to yield predictable results; (b) simple substitution of one known element for another to obtain predictable results; (c) use of a known technique to improve a similar device (method, or product) in the same way; (d) applying a known technique to a known device (method, or product) ready for improvement to yield predictable results; (e) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and (f) some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify the prior art or to combine prior art teachings to arrive at the claimed invention.

V. TECHNICAL BACKGROUND

A. Smart Home Automation

22. The concept of interconnecting multiple devices associated with a user to automate various tasks was commonplace long before the priority date of the ’518 patent. This concept was routinely employed in the context of smart home automation. Khushoo, [0001]. “Most homes have numerous home devices, such as electrical systems (e.g., light switches, radios, etc.), mechanical systems (e.g., windows, doors, door locks, etc.), communication systems (e.g., a security system, a local area network (LAN), etc.), and entertainment systems (e.g., televisions, home theater systems, etc.). **A smart home (or connected home) brings all of these home devices together**. By wiring controls of each home device to one point, commonly known as a control system overlay or control point, the home devices can be integrated and made interoperable. Such smart home automation technologies are commercially available.” Khushoo, [0001].

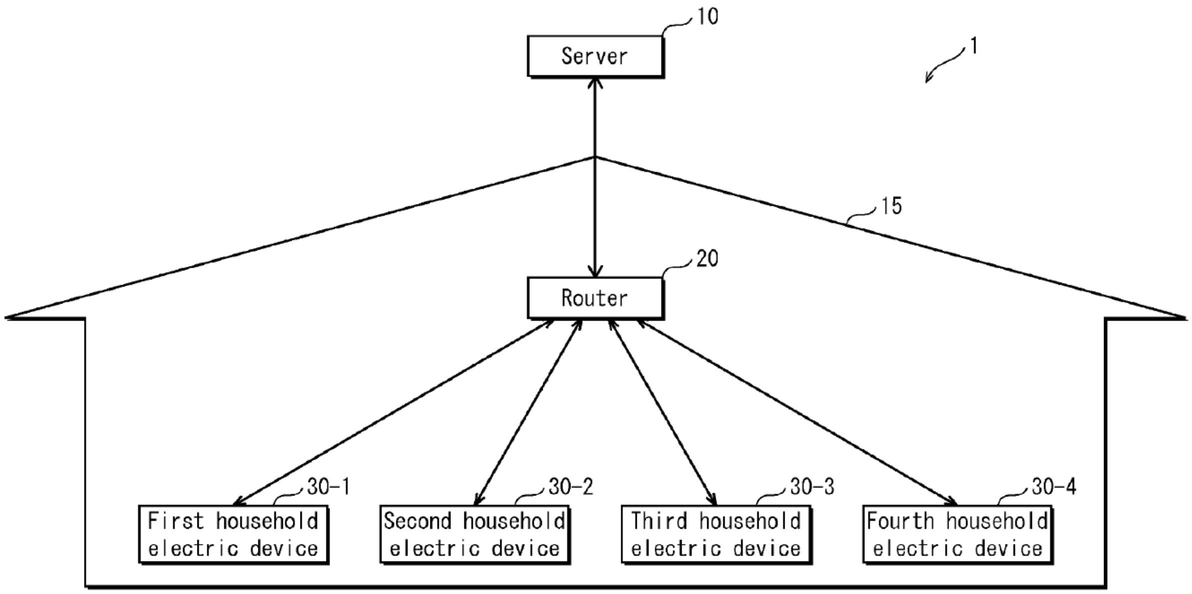
23. Smart homes were described in publications as early as 2003, with indications that the concept of a smart home was first introduced as early as 1984. *See* Harper, 1-4.

24. Smart home automation includes a network of multiple devices. For example, a smart home is described as "a network, run by a server that operates all of the functions of the house." Harper, 2. While the concept of coordinating networked objects to perform tasks is common in the context of smart home automation, it also applies to other non-home-related contexts, such as smart capabilities within a user's car. *See* Ermis, Figs. 23-24.

B. Relation Servers

25. It was well known that communication and tasks performed by devices in a network were managed by a server. For example, multiple "household electric devices" within a smart home are shown managed by a server 2010 by Nakano, below. Nakano, 2:20-23.

FIG. 1



Nakano, Fig. 1

26. In other prior art, multiple "home devices 150" are controlled by a "control point 130 and backend server 160" which "may be combined in a single device." Khushoo, [0029]. In a typical smart home, "a server…operates all of the functions of the house." Harper, 2; *see also* Yabe, Abstract, [0035]; Schaefer, 3 (managing "profiles" of devices and users). The '518 patent refers to such a server as a "relation server." '518 patent, 1:23-29, Fig. 1.

C. Data Storage and Format

27. The server operating and controlling multiple connected devices stores information related to those devices, including data showing that the devices are part of the network, associated with the server, and associated with one another. For example, data related to connected devices may be stored as "entity profiles." Ermis, [0041]; *see also* Nakano, Fig. 3 ("device information table"); Yabe, Abstract ("A controller stores an apparatus-room association table"). Data identifying "interactions between the" devices is also stored. Ermis, [0042].

28. A smart home server also stores data related to tasks performed by the multiple connected devices. For example, as explained in greater detail below, Ermis describes storing data flows defining tasks performed by multiple devices. Ermis, [0128] (describing adding "data flows to the interaction flow database 1308"). Nakano refers to tasks performed by multiple devices in a smart network as "cooperative processes." Nakano, 1:17-18. Other prior art describes similar data related to tasks performed by multiple devices in a network. *See, e.g.*, Harper, 24 (performing "task[s]" such as "control[ling] room temperature, water heat, ventilation and lighting"); Yabe, [0072] (server stores a "schedule table" when tasks are performed); Schaefer, 7-8 (performing multiple tasks in advance of a user returning home)

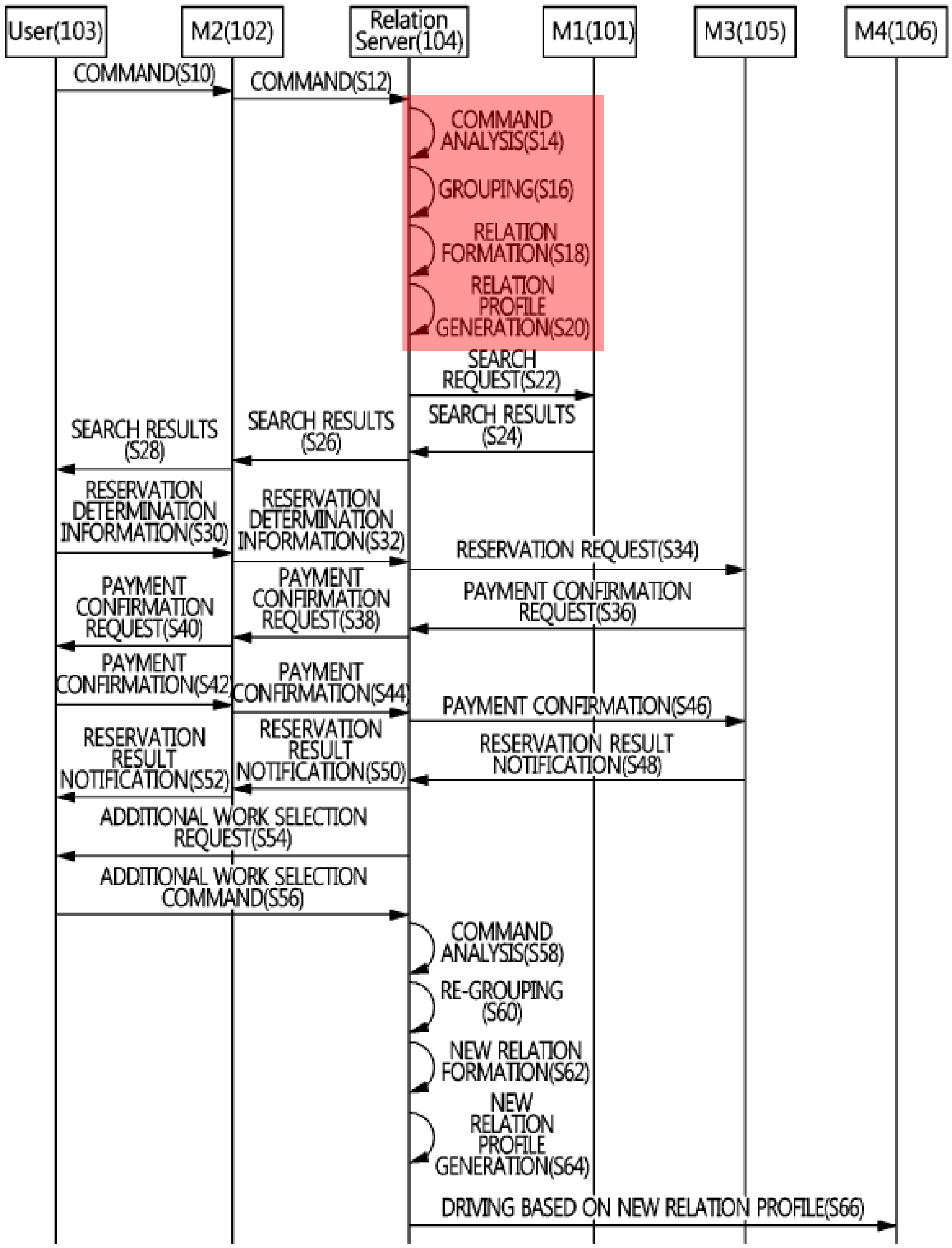
29. The data stored on a smart home server may be stored in a variety of formats. For example, data may be stored in various text-based table structures. *See* Nakano, Figs. 3-4, 2:61-64; Yabe, [0009]-[0010]; Ermis, [0070] ("XQUERY" for querying an xml database).

30. A POSITA would have recognized that various terms are used to refer to data, such as data stored in a server for automating tasks. These terms include "parameter," "variable," "information," "value," "metadata," "specification," "field," "characteristic," "protocol," etc. *See, e.g.*, Ermis, [0089]; Hedge, [0038]; Hamilton, 29 (Appendix A); Bui, Abstract; Markki, [0175]; Miller, [0007]; Schaefer, 6. A POSITA would have recognized that these terms are interchangeable.

VI. THE '518 PATENT

31. The '518 patent relates to automating tasks performed by multiple devices using a "relation server." '518 patent, (54) (Title), Abstract. The relation server "manag[es] relations between machines" to "execute a command" from a user. '518 patent, Abstract. At a high level, the '518 patent describes a relation server that receives a command from a user, analyzes the command to determine which machines are necessary to execute the command, and controls those machines to execute the command. The '518 patent provides an example "scenario" with Figure 2, shown below, in which "the first machine 101 is a movie search engine, the second machine 102 is a smart phone, the third machine is a machine having a movie reservation function." '518 patent, 8:62-9:3.

32. In response to a user command, the relation server "generate[s] a capability set required to execute the command," and identifies or "group[s] the machines 101, 102, 105, and 106 which will execute the command." '518 patent, 9:13-18.



’518 patent, Fig. 2 (annotated)

33. Based on the analysis of the relation server, the relation server then transmits various commands to the selected machines to perform the requested command. *See* '518 patent, Fig. 2 (steps S22-S66).

34. The user then provides another command to the relation server "to make a reservation (e.g. the selected movie, the selected movie theater, the movie start time, the number of tickets to be reserved, etc.)." '518 patent, 9:63-10:3.

35. According to the '518 patent, the list of processes performed by devices to perform the requested command is referred to as a "task processing schedule" parameter. *See* '518 patent, 6:58-7:27 (Table 2) (listing "Process 1" through "Process n"). Each process may have a "Process start time." '518 patent, 7:8.

36. The '518 patent uses the terms "parameter" and "sub-parameter" to broadly refer to lists or categories of data associated with users or machines, as well as data not including lists, values such as machine IDs and group IDs, etc. *See* '518 patent, 4:32-58 (Table 1), 6:58-7:27 (Table 2).

37. As explained below, each of these concepts were well known before the priority date of the '518 patent.

VII. CLAIM CONSTRUCTION

38. It is my understanding that in order to properly evaluate the '518 patent, the terms of the claims must first be interpreted. It is my understanding that for the purposes of this *inter partes* review, the claims are to be given their ordinary and customary meaning as would have been understood by a POSITA in light of the specification and prosecution history, unless the inventor has set forth a special meaning for a term. I have also been informed that claim terms only need to be construed to the extent necessary to resolve the obviousness inquiry. I have reviewed the entirety of the ’518 patent, as well as its prosecution history. It is my opinion that for purposes of applying the prior art presented herein to evaluate the patentability of the Challenged Claims, no claim terms require express construction. The ’518 patent specification provides definitions for multiple claim terms, which are addressed in the mapping below as needed (“*relation profile*” in [1.2.2], “*capability set parameter*” in [4.0], “*sub-parameter*” in [8.0], “*input parameter sub-parameter*” in [11.1], “*operating system parameter*” in [15.0], and “*interface parameter*” in [16.0]). My analysis below demonstrates that these claim terms, as defined, are rendered obvious by the prior art.

VIII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

39. I have been asked to provide my opinion as to whether the Challenged Claims of the ’518 patent would have been obvious in view of the prior art. The discussion in this Declaration provides a detailed analysis of how the asserted prior art references teach each limitation of the Challenged Claims.

40. As part of my analysis, I have considered, and discuss in detail, the scope and content of the prior art and any differences between the alleged invention and the prior art.

41. As described in detail below, it is my opinion that the alleged invention recited in the Challenged Claims would have been obvious in view of the teachings of the asserted prior art and the knowledge of a POSITA before the time of the alleged invention of the ’518 patent. I apply the prior art as follows:

Grounds	Claim(s)	Basis
#1	1-20	§103 **Ermis**
#2	1-20	§103 **Ermis** and **Nakano**

A. Ground 1: Claims 1-20 are obvious under §103 over Ermis.

1. Summary of Ermis

42. Like the ’518 patent, Ermis relates to a system that executes commands from a user by identifying the machines necessary to execute the command and controlling the machines. In more detail, Ermis describes a “system 12,” shown in Figure 1 below. Ermis, [0039].

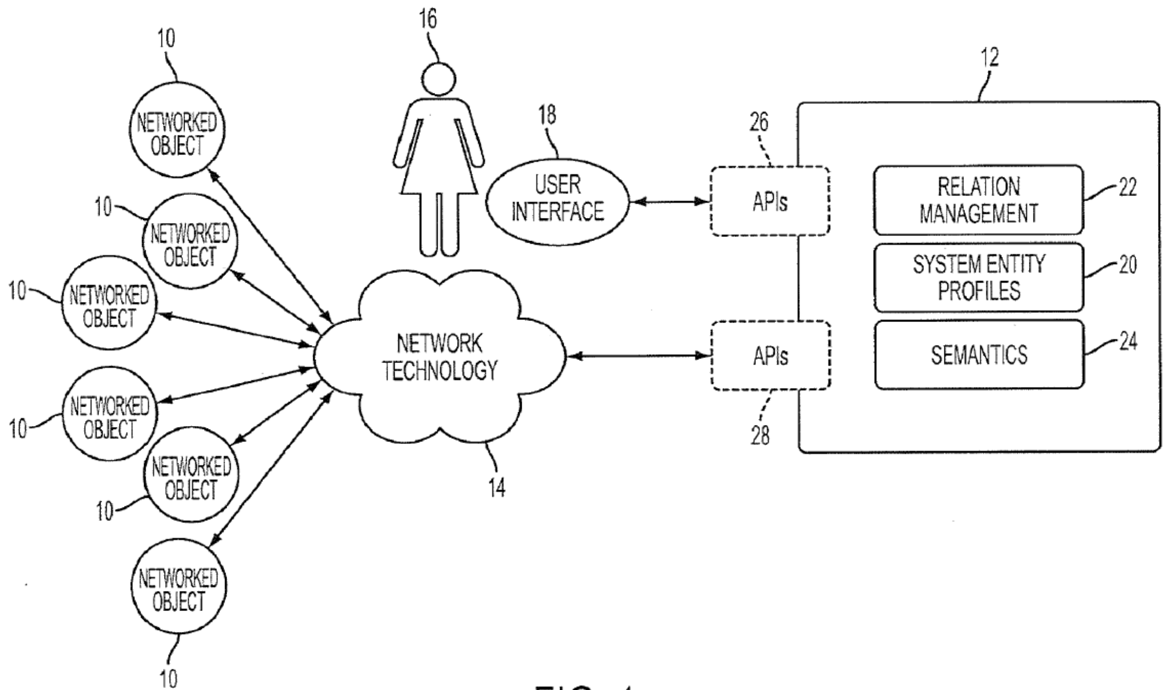


FIG. 1

Ermis, Fig. 1

43. The system 12 is a server. *See* Ermis, [0040] ("system 12…can, for example, be implemented, at least in part, as server-based software."). The system 12 communicates with multiple "networked objects 10," which may include "consumer electronics, digitally tagged objects, computer devices, mobiles, sensors, buildings, vehicles or even companies, brands, services and physical locations." Ermis, [0039]-[0040].

44. The user 16 sends a task request to the system 12 via a "user interface[] 18" which may be the user's "mobile device[]." Ermis, [0039].

45. After receiving a task request, Ermis's system 12 analyzes the capabilities of the networked objects 10 and determines which capabilities of which networked objects 10 are needed to execute the command. *See* Ermis, [0066] (describing "[a] service required capabilities database which describes what a service would require to be meaningfully consumed"). Then, the system 12 controls those networked objects 10 to execute the command. *See* Ermis, [0045]-[0046] (issuing commands to networked objects including a TV and video recorder to execute a user task request).

46. Ermis provides multiple examples of task requests, explaining the steps of identifying and controlling networked objects to fulfil them.

47. In one example, a user requests that the system 12 record a television broadcast. "[T]he user 16 can for example send a task request…to the TV's system entity (operative within system 12) requesting the system 12 to record a certain TV program." Ermis, [0045]. "The system entity in system 12 associated with the TV will accept and acknowledge this instruction." Ermis, [0045] If, however, the TV is not capable of executing the command itself, the system 12 may identify "a friendship relationship with the system entity of a video recorder," which "actually execute[s] the job." Ermis, [0046]. Then, "the user 16 will receive a confirmation from the TV's system entity (again via API 26 and user interface 18) that the requested task will be performed, and later on that it has been successfully completed." Ermis, [0046].

48. In another example, illustrated in Figure 17, below, the system 12 recommends that the lights in the user's house be turned off when the user leaves the house.

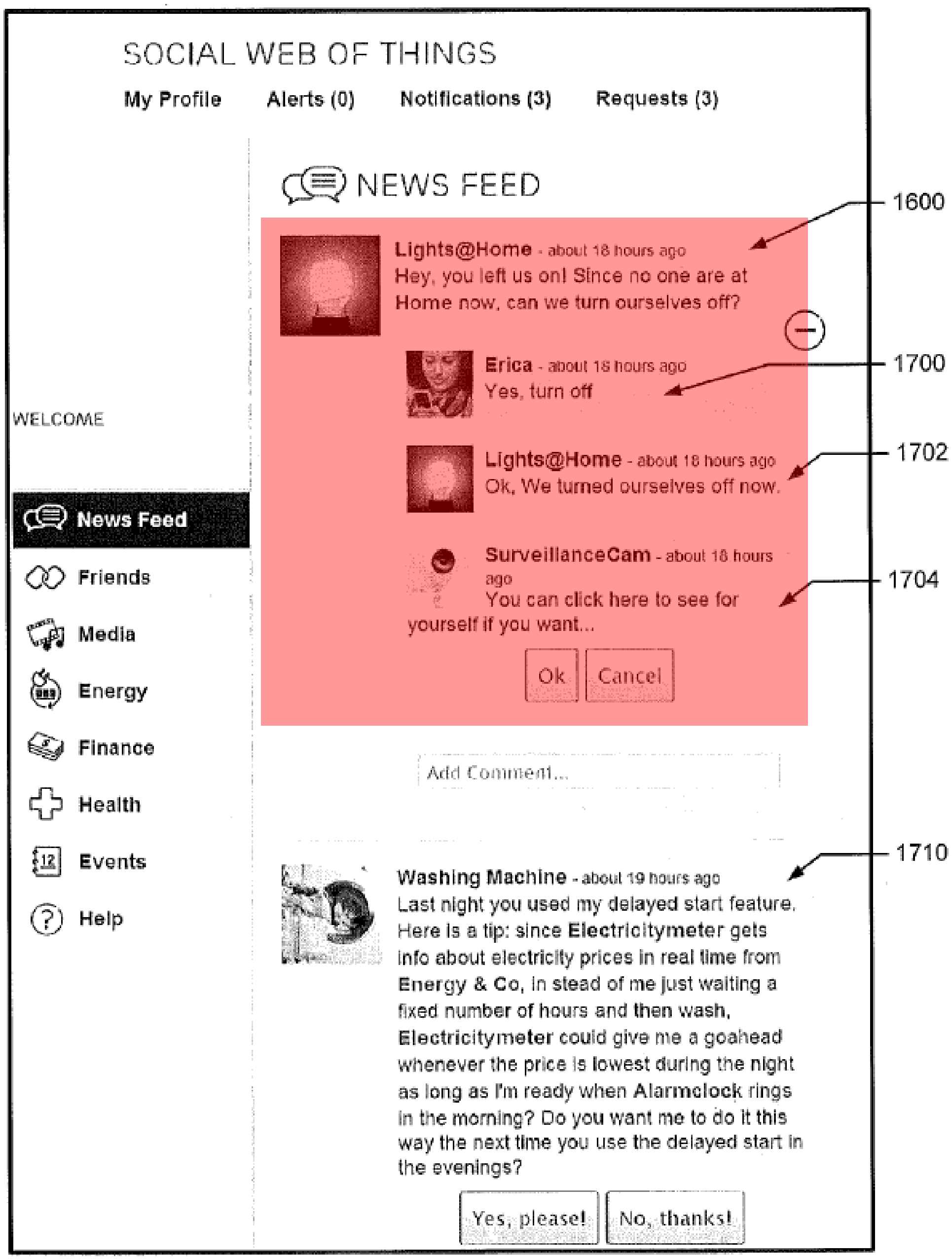


FIG. 17

Ermis, Fig. 17 (annotated)

49. When the user approves the recommendation, the system 12 establishes "a data flow…between the light controller and a surveillance camera" so that after the lights are turned off, the user can "observe a room which may include the controlled lights." Ermis, [0110].

50. Thus, Ermis describes multiple scenarios in which a system 12 coordinates multiple networked objects to execute a task request from a user.

2. Detailed Analysis of the Claims

a) Claim 1

[1.0] ***A system, comprising:***

51. To the extent the preamble is limiting, Ermis discloses or renders it obvious.

52. Ermis describes "**systems**, methods, devices and software (computer programs) for management of, and interaction with, networked objects." Ermis, [0038]. Ermis provides a "high level architectural view of the system…with respect to FIGS. 1 and 2," shown below. Ermis, [0039].

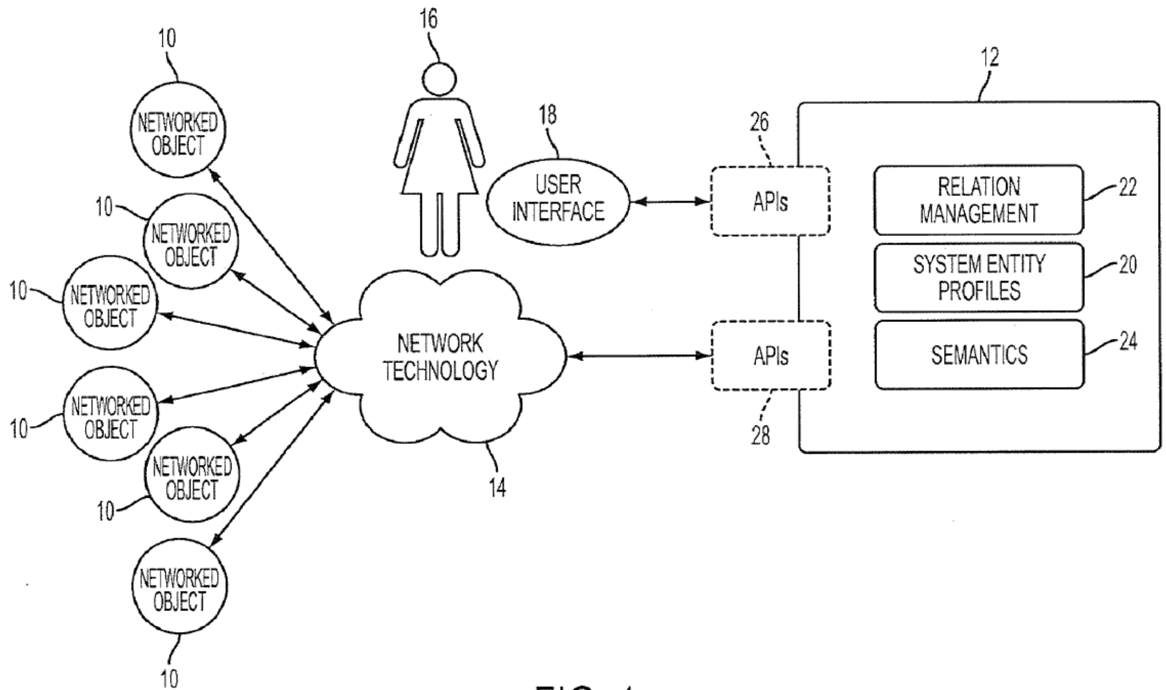


FIG. 1

Ermis, Fig. 1

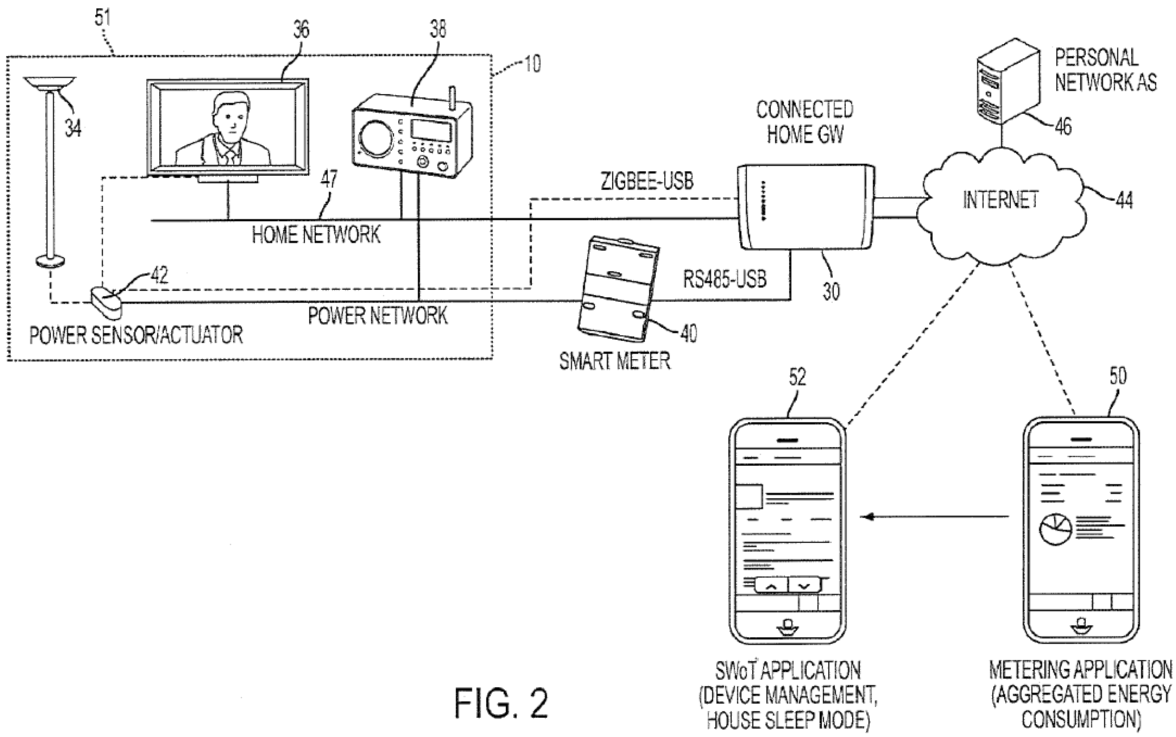


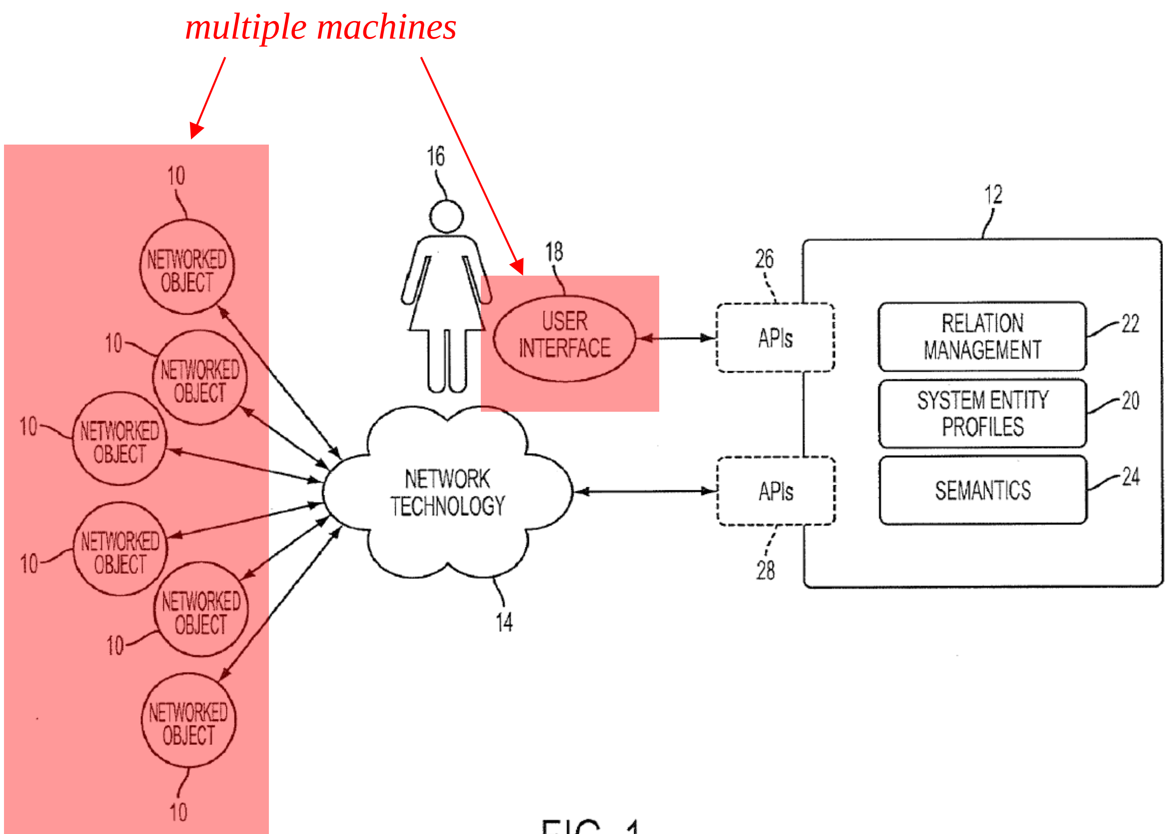
FIG. 2

Ermis, Fig. 2

53. Thus, Ermis renders obvious *a system* ("system[]…for management of, and interaction with, networked objects").

[1.1] ***multiple machines; and***

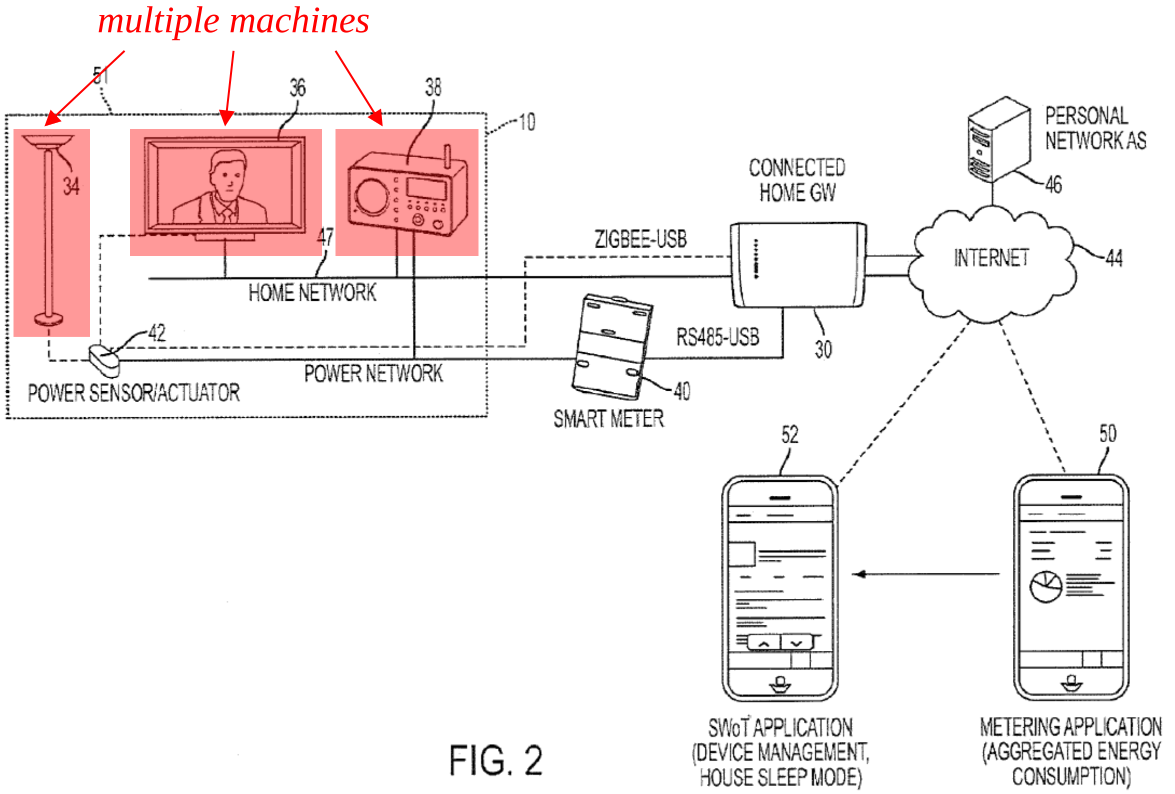
54. Ermis's system includes multiple networked objects 10 and a user interface 18, as shown in Figure 1, below.



Ermis, Fig. 1 (annotated)

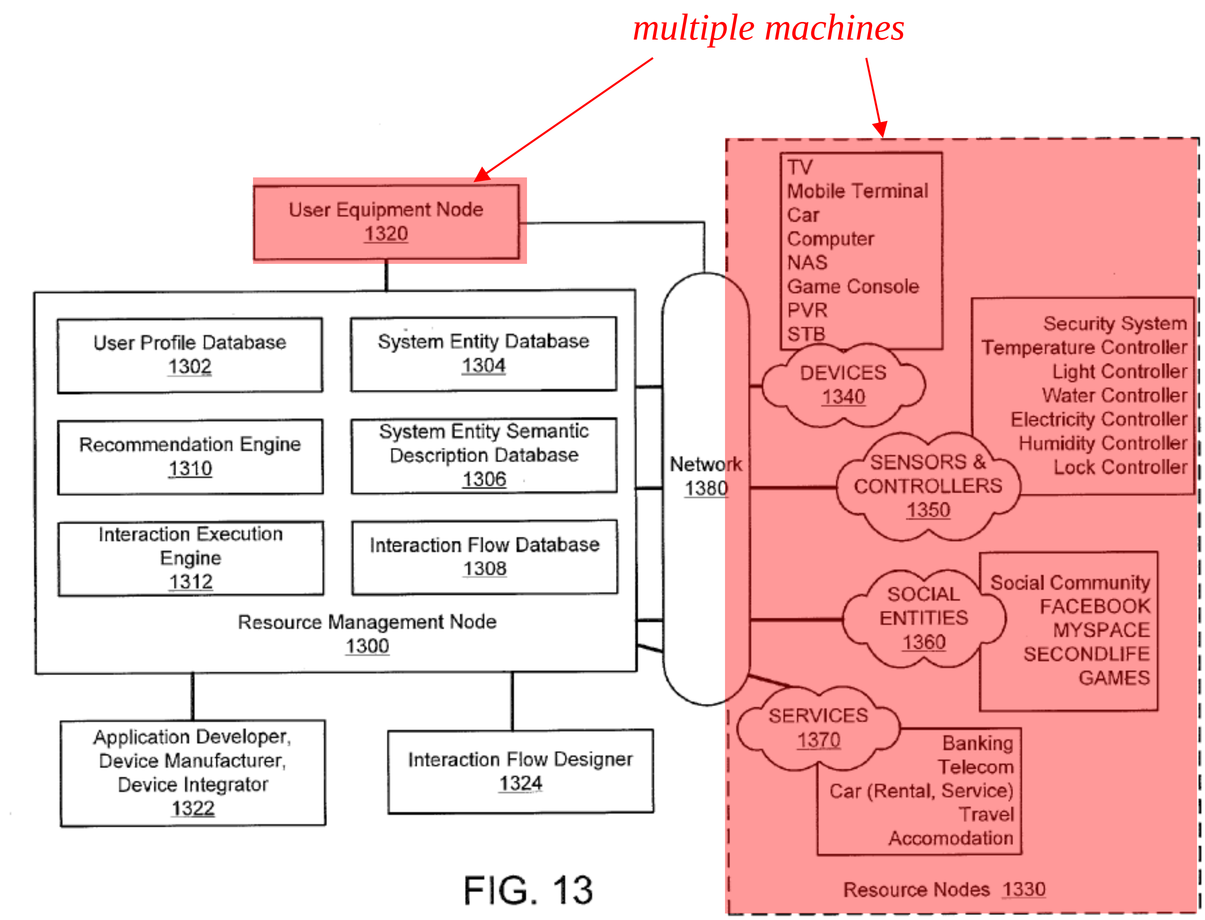
55. Ermis explains that in Figure 1, "a plurality of networked objects 10 is connected to the system 12." Ermis, [0039]. "Networked objects 10 include, but are not limited to: **consumer electronics**, digitally tagged objects, **computer devices**, **mobiles**, **sensors**, buildings, **vehicles** or even companies, brands, services and physical locations." Ermis, [0040]. Any of these networked objects 10 are *machines* as claimed.

56. Ermis also provides a more specific example of networked objects in Figure 2, below.



Ermis, Fig. 2 (annotated)

57. For Figure 2, Ermis explains that "the networked objects 10 include various power consuming devices including, e.g., **a light, 34**, **a television 36**, and **a radio 38**." Ermis, [0048]. Any of these devices are also *machines*. Ermis also uses the term "resource node" to describe the networked objects. *See, e.g.*, Ermis, [0096] ("resource nodes 1330"), Fig. 13.



Ermis, Fig. 13 (annotated)

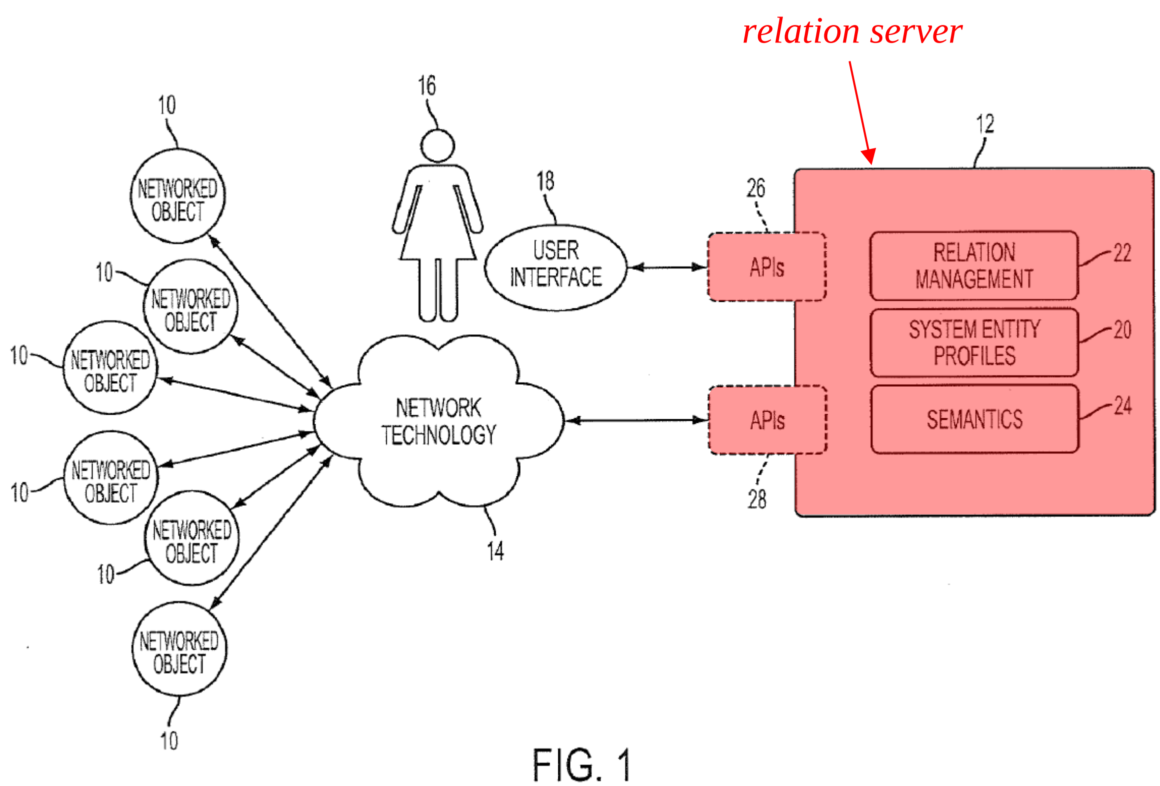
58. Therefore, Ermis renders obvious *multiple machines* (multiple "networked objects 10" and "user interface 19" and/or "resource nodes 1330" and "user equipment node 1320").

[1.2.1] *a relation server*

59. Ermis renders obvious *a relation server*. In Figure 1, Ermis's system 10 includes a "system 12 which manages the networked objects 10" and "can, for example, be implemented, at least in part, as **server-based software**." Ermis,

[0040].

60. The system 12, alone or in combination with the APIs 26 and 28 (highlighted below), renders obvious a *relation server* because it manages the relationships between the networked objects 10. Ermis, [0042] ("The system 12 also includes a **relation management function 22** which **coordinates the interactions between the networked objects 10**.").



Ermis, Fig. 1 (annotated)

61. In Figure 13, Ermis "depicts another system architecture that includes a resource management node 1300 that manages a plurality of resource nodes 1330." Ermis, [0096]. The system in Figure 13 is a more detailed example of the system in Figure 1, and Ermis explains that any feature in one embodiment may be used in combination with any feature of another embodiment. *See, e.g.*, Ermis, [0085] ("[E]ach feature or element can be used…in various combinations with or without other features and elements disclosed herein.").

62. Ermis's "resource management node 1300…manages a plurality of resource nodes 1330," and alternatively or in combination with the system 12, also renders obvious *a relation server*. Ermis, [0096].

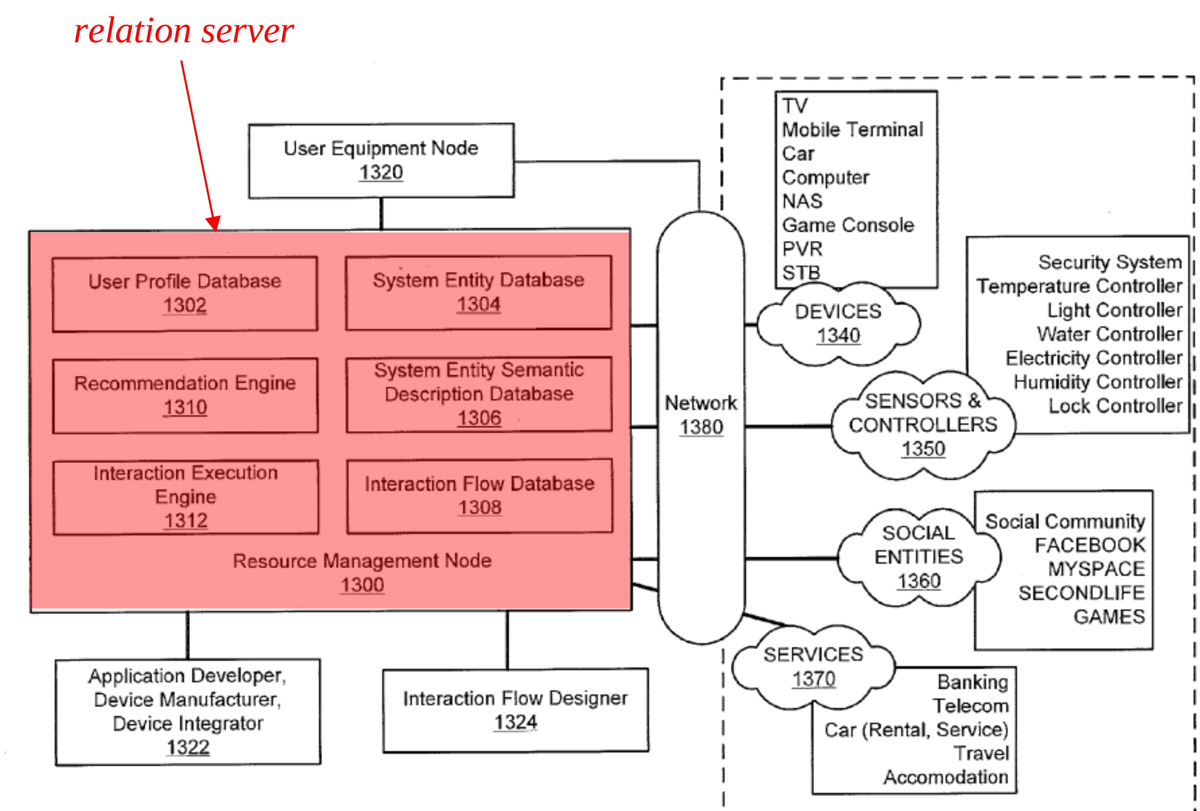


FIG. 13

Ermis, Fig. 13 (annotated)

63. Ermis explains that its "example embodiments" together "provide a number of advantages and benefits relative to existing resource management software," such as "streamline[d] interactions between users and the networked resource nodes." Ermis, [0007]. In light of those advantages, Ermis expressly instructs a POSITA to combine the features of its example embodiments when reviewing the detailed description. Ermis, [0012] ("it is intended that all embodiments disclosed herein can be implemented separately or combined in any way and/or combination"), [0085] ("each feature or element can be used…in various combinations with or without other features and elements disclosed herein."). Accordingly, a POSITA would have been motivated to combine (and would have been expected success in combining) elements from Ermis's various examples to achieve Ermis's stated advantages.

64. Thus, Ermis describes *a relation server* (system 12 alone or in combination with APIs 26 and 28 and/or resource management node 1300).

[1.2.2] ***for storing a relation profile***

65. The '518 patent defines a *relation profile* as "data representing relations formed to perform a task." '518 patent, 6:54-56.

66. **First**, Ermis describes forming relationships between the networked objects/resource nodes and associated data flows, which renders obvious *a relation profile*. Ermis describes forming "relationship[s] between…at least two resource nodes." Ermis, [0009]; *see also* Ermis, [0042] ("establish[ing] relationships between system entities 10”). These “relationships” are “**exercise[d]…in performance of various task requests**.” Ermis, [0044]. Ermis provides multiple examples of relationships between machines to perform various tasks.

67. In one example shown in Figures 16 and 17, a relationship is described “between [a] light controller and a surveillance camera” to turn off the lights in a house when the user is away and show surveillance camera footage to confirm that the lights were turned off. Ermis, [0110]. Specifically, the system “determines that one or more lights have been left on” and “communicat[es] a message to the user,” shown in Figure 17, below. Ermis, [0109]. If “the user has selected the illustrated ‘yes, turn off’ button,” it “serves as a triggering condition that causes…the light controller to turn off the lights.” Ermis, [0110]. After the lights have been turned off, the system “communicates another message to the user…that queries whether the user wants to view a picture/video from the security camera to observe a room which may include the controlled lights.” Ermis, [0110].

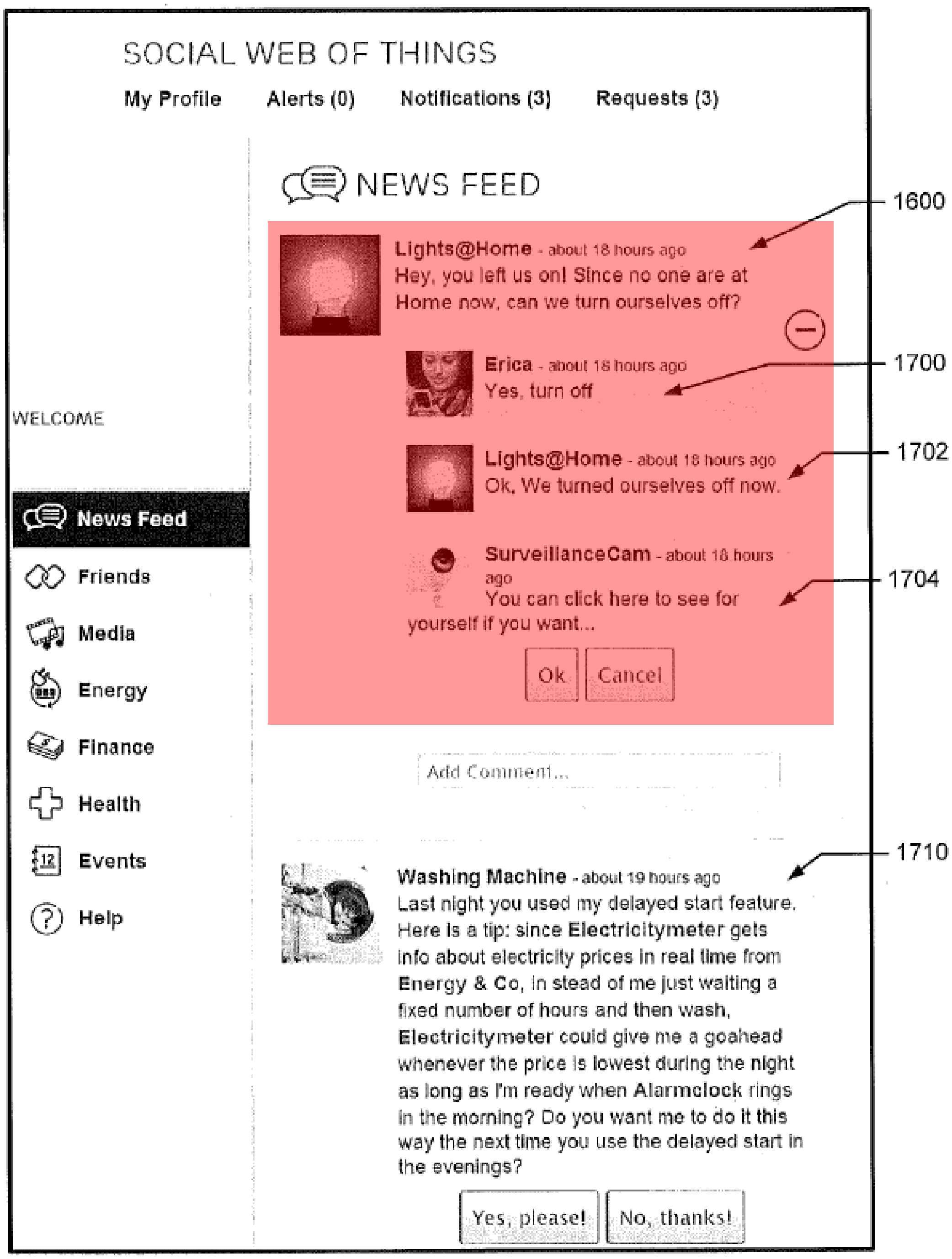


FIG. 17

Ermis, Fig. 17 (annotated)

68. This example in Figure 17 shows data representing relations formed between the light controller and surveillance camera to perform a task of turning off the lights when the user has left the house and then displaying surveillance camera footage, which renders obvious *a relation profile*.

69. In another example shown in Figures 17 and 18, a relationship is described between a "washing machine," an "electricity controller," and an "alarm clock" to schedule the start a washing machine while the user sleeps based on "the price of electricity and a planned wake-up alarm time." Ermis, [0111]. "[W]hen the triggering criteria is satisfied (e.g., price of electricity has dropped below the define[d] threshold value and/or less than a defined length of time remains before the wake-up time)," the system "responds by turning on the washing machine." Ermis, [0113].

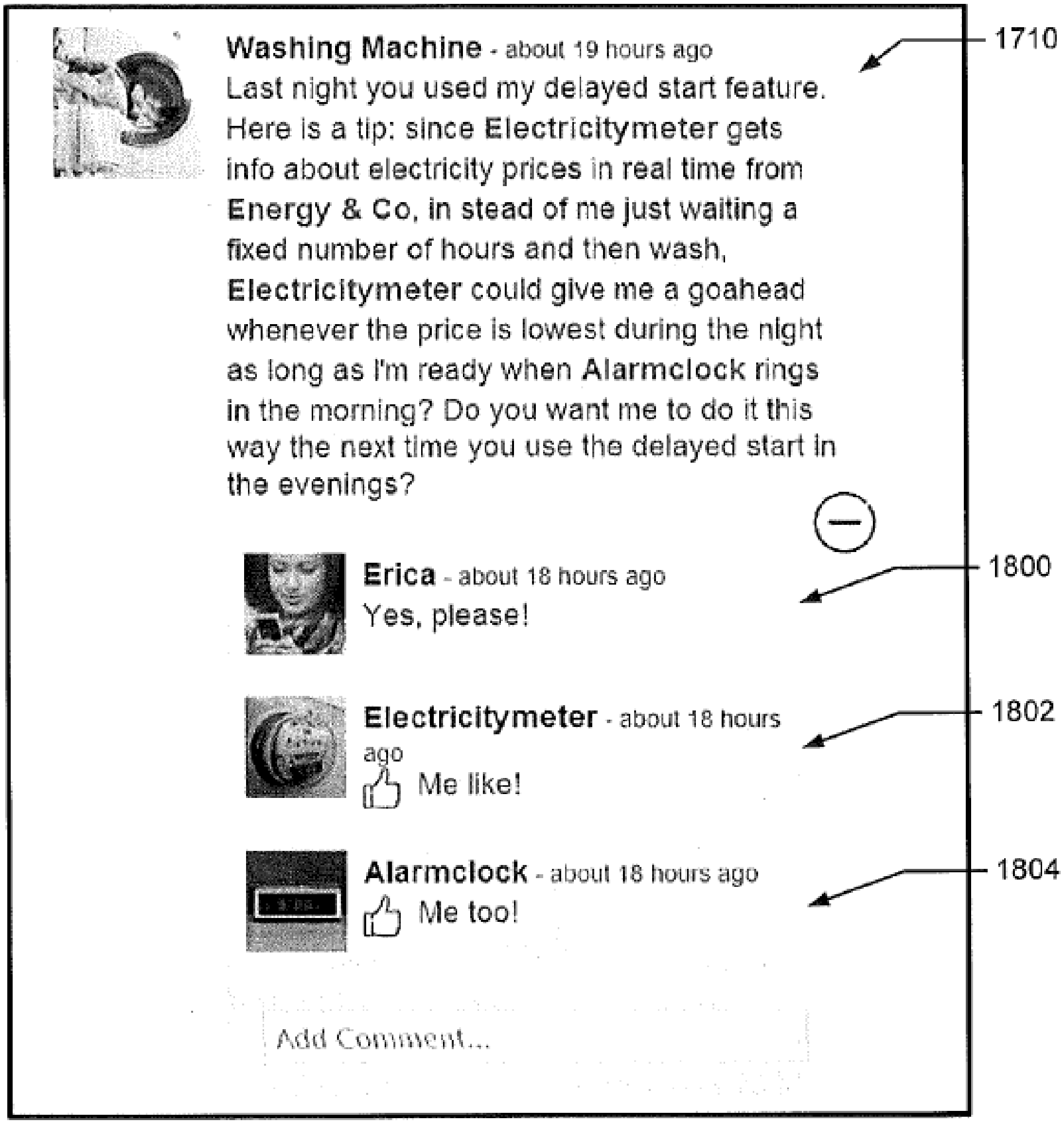


FIG. 18

Ermis, Fig. 18

70. This example in Figure 18 shows data representing relations formed between the washing machine, the electricity controller, and the alarm clock to perform the task of starting the washing machine based on data received from the electricity controller and alarm clock, which also renders obvious *a relation profile*.

71. In another example, a new relationship is created between a rental car and a user's phone to perform tasks such as streaming movies or music from the user's phone to the car speakers or sending GPS navigation guidance. In more detail, the rental car requests that a new relationship be formed with a "MediaServer" and "ThePhone" of a user, shown in Figure 24, below. After the relationship is established, it "enable[s] movies and music to be streamed from the media server device in the [phone] terminal through an entertainment system in the rental car." Ermis, [0126]. It also enables "driving routes and places that the user has stored in the map application to be used by the navigator to provide real-time driving instructions to the user." Ermis, [0126].

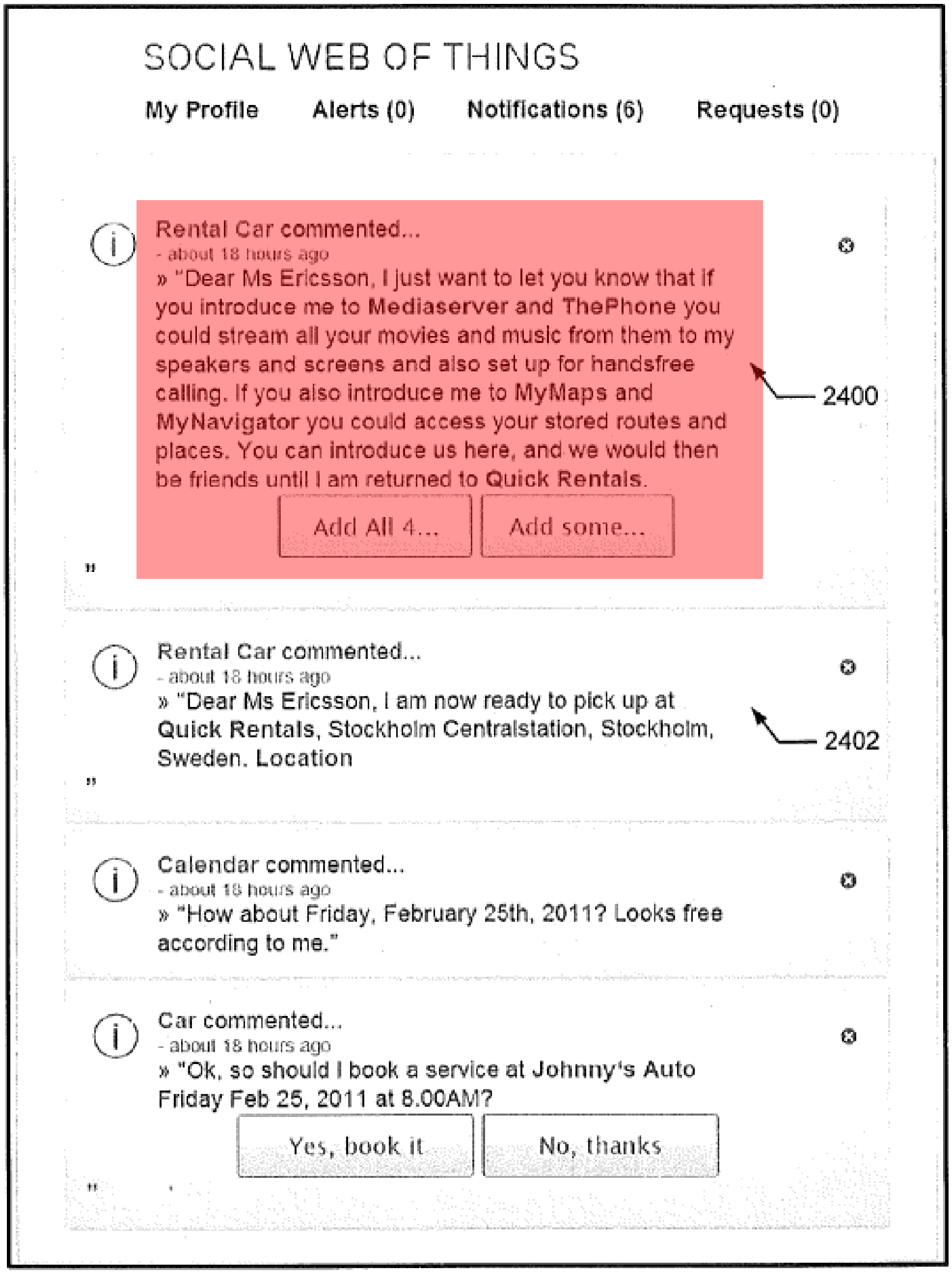


FIG. 24

Ermis, Fig. 24 (annotated)

72. This example in Figure 24 shows data representing a relation formed between the rental car and the user's phone to perform the tasks of streaming movies and music and providing GPS navigation directions, which also renders obvious *a relation profile*.

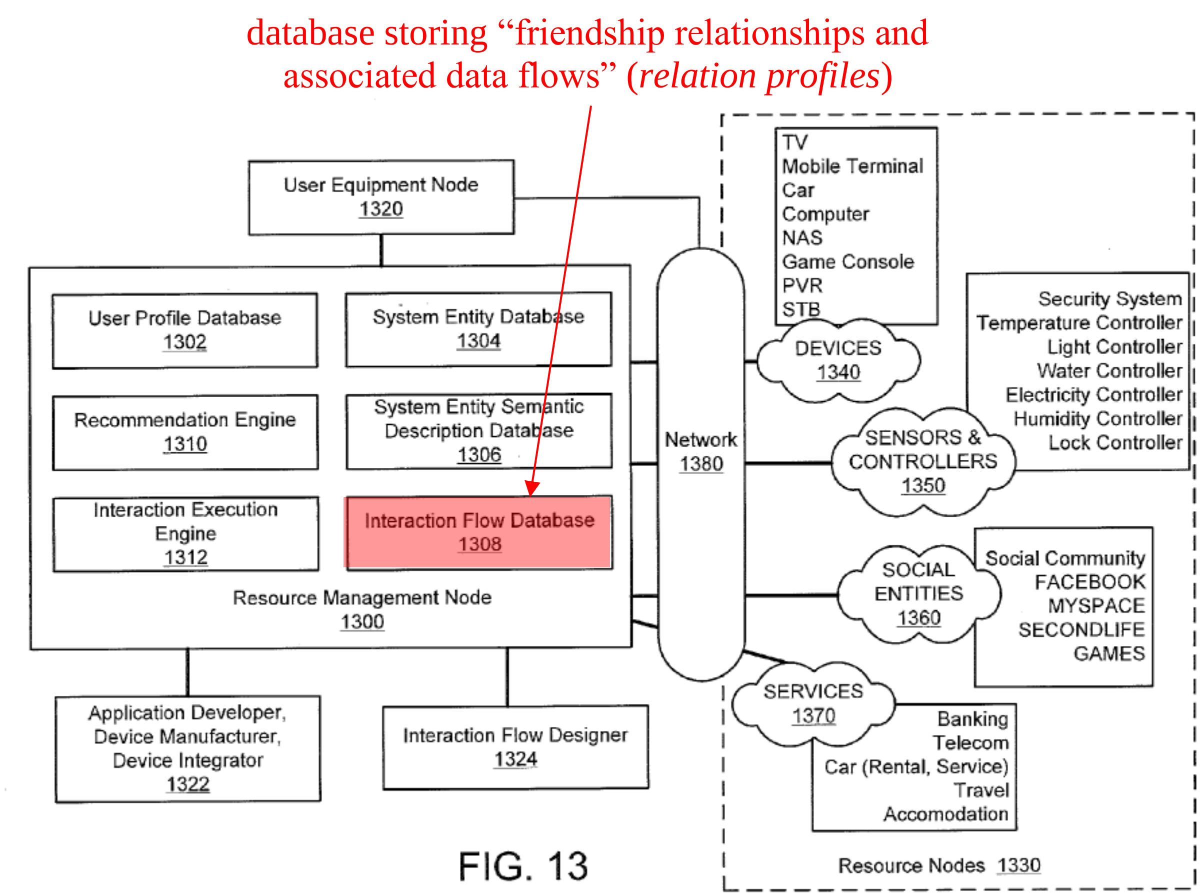
73. In addition to these examples explained above, Ermis provides multiple other examples of relation profiles between additional devices, which will be explained in more detail throughout. *See, e.g.*, Ermis, [0045]-[0046], [0053]-[0054] (relation formed between a TV and a video recorder to record a TV program), [0047] (forming new relationships between a network attached storage device with other machines in the network which use the storage device "as media source"), [0048] (relations between "a light, 34, a television 36, and a radio 38" and monitoring "power consumption" of the device and control them accordingly), [0074] (relationship between a user and a friend's storage device to "search for interesting media"), [0124] (relationship between a user's car and smart phone to inform the user of needed maintenance and schedule a service appointment). Because all of these relationships are generated to perform a task, *see* Ermis, [0044], any of these example relationships render obvious *a relation profile*.

74. **Second**, Ermis explains these relation profiles are stored within the relation server. The data indicative of any of the relationships described above is stored in an interaction flow database 1308, which is part of the resource management node 1300:

> The user can establish <u>**a friendship status**</u> between the identified

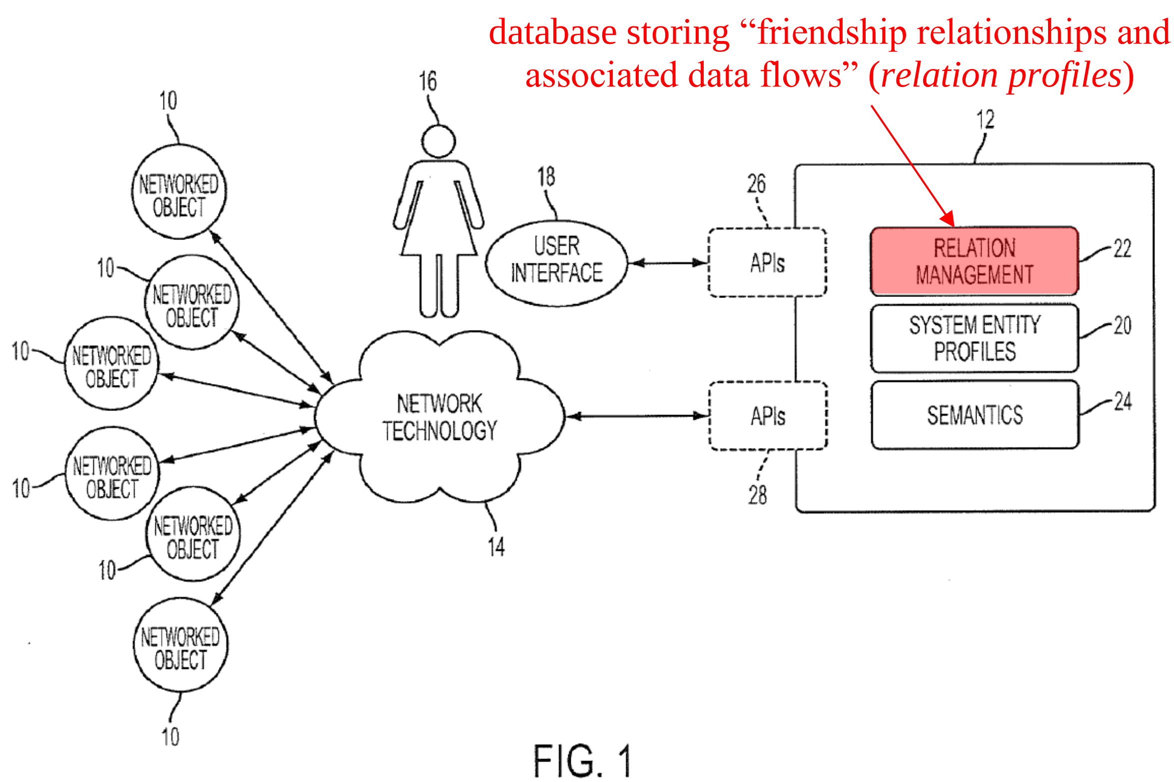
> resource nodes **by selecting the "Add All 4. . ." button**, or can select which of the resource nodes are to have friendship satisfy selecting the "Add some . . ." button. The interaction execution engine 1312 **adds <u>the selected friendship relationships and associated data flows to the interaction flow database 1308</u>**.

Ermis, [0128].



Ermis, Fig. 13 (annotated)

75. In the embodiment of Figure 1, relationship data is stored in a database in the "relation management function 22," which establishes and manages "**the relationship between networked objects**." Ermis, [0038]; *see also* Ermis, [0042] ("A relationship [between the networked objects 10] is established by the relation management function 22."), [0052] ("Relationship management function 22 of system 12 establishes and manages these different levels of relationship between networked objects 10 and users 16.").



Ermis, Fig. 1 (annotated)

76. Thus, Ermis renders obvious *a relation server* (system 12 alone or in combination with APIs 26 and 28 and/or resource management node 1300) *for storing a relation profile* ("friendship relationships and associated data flows" stored within interaction flow database 1308 within the relationship management function 1300 and/or stored within relationship management function 22 within the system 12).

[1.2.3] ***including a task processing schedule parameter which defines a sequence of performing multiple processes,***

77. Ermis describes "data flows" which render obvious *task processing schedule parameter[s]* because they "*define[] a sequence of performing multiple processes.*" *See* Ermis, Title.

78. As explained above at limitation [1.2.2], each of the relationships in Ermis are associated with a number of processes to perform a requested task, referred to as a data flow. *See supra* § VIII.A.2.a [1.2.2] (listing multiple examples of relationships formed between devices and the tasks performed by the devices within the relationship). These friendship relationships and associated data flows are stored in the interaction flow database 1308 in the embodiment in Figure 13 and/or the relationship management function 22 in the embodiment in Figure 1. *See, e.g.*, Ermis, [0128], [0038], [0042].

79. In Ermis, the data flows stored in interaction flow database 1308 contain information that "defines data flows that are permitted between identified ones of the resource nodes" and "further defines associated triggering criteria for **when identified ones of the data flows are to be performed**." Ermis, [0101]. "Each data flow can identify two or more resource nodes 1330 that are communicatively connected so that the data output of one resource node is provided as a data input to another resource node, and may connect the respective output and input interfaces of a plurality of resource nodes to provide **a serial daisy chain** or other connection architecture between the identified resource nodes." Ermis, [0101].

80. Ermis describes data flows as "functional data flows," Ermis, [0118], including a series of processes that "are to be **performed**," Ermis, Abstract, [0008], [0011], [0101]-[0102]. Specifically, "data flows [] define how applications and/or connected resource devices will communicate with each other **and <u>operate to provide defined functionality</u>**." Ermis, Abstract, [0119]. Ermis explains that "data flows [are]…created between…resource nodes 1330 **to provide increased functionality for a user**." Ermis, [0115]. In some cases, data flows may be recommended to a user, with the request including "an explanation of **the functional capabilities** that will be provided by the recommended data flow." Ermis, [0111].

81. As described in [1.2.2], "a data flow has been established between the light controller and a surveillance camera." Ermis, [0110]. After a user has been notified that lights have been left on more than a threshold time, the user selects the "yes, turn off" button on its user interface, which serves as a triggering event to cause the light controller to turn off the lights. The triggering event also causes the user to be prompted to view a picture/video from the security camera to observe a room which may include the controlled lights." Ermis, [0110]. If the user selects the illustrated "Ok" button, a further triggering event causes the security camera to output the picture/video for the user. Ermis, [0110].

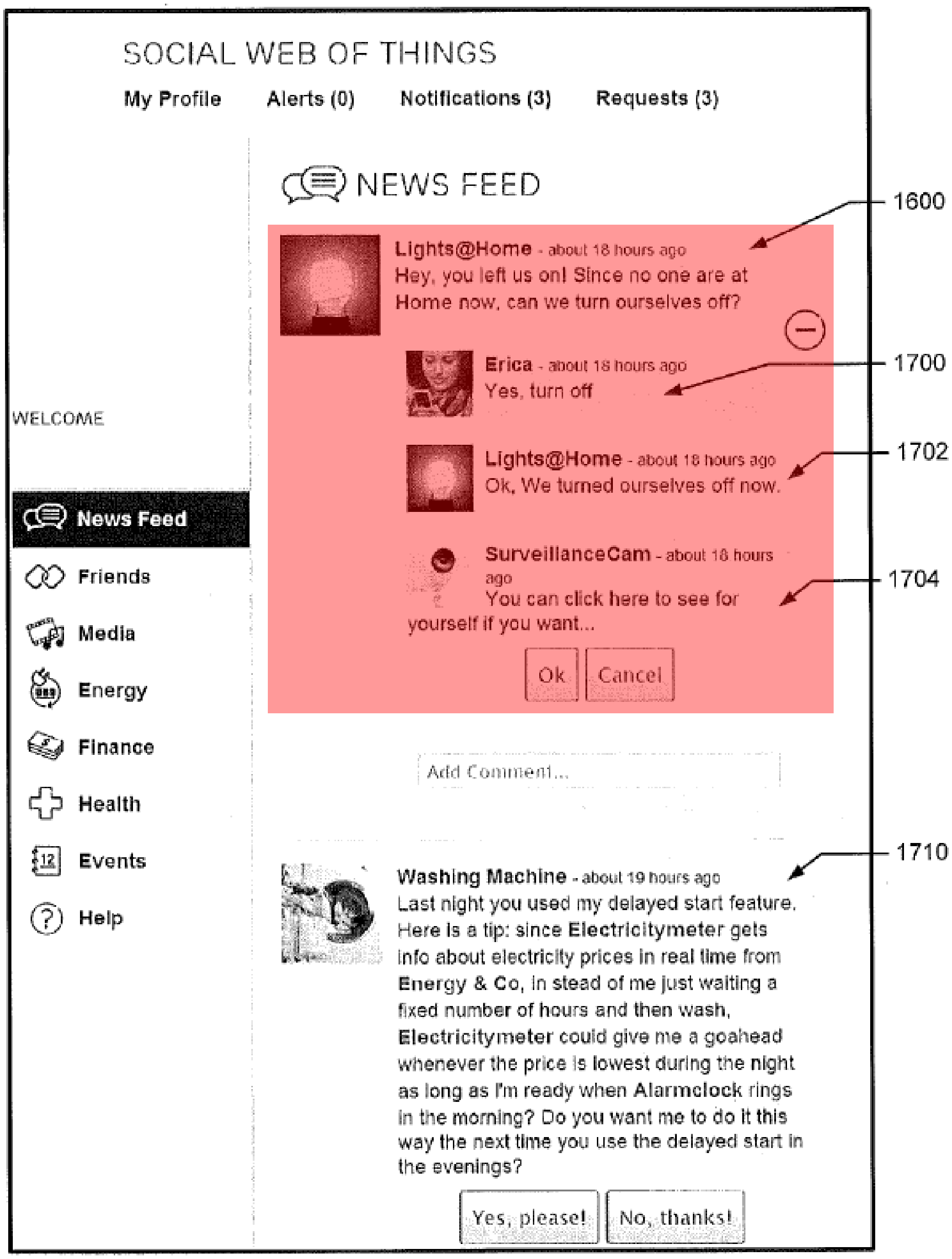


FIG. 17

Ermis, Fig. 17 (annotated)

82. The data flow performed by the light controller and a surveillance camera, including triggering criteria for turning off the lights by the light controller and displaying surveillance footage, renders obvious *a task processing schedule parameter which defines a sequence of performing multiple processes*.

83. In the example in Figures 17 and 18 involving "the washing machine, the electricity controller, the energy company controller interface, and the alarm clock," Ermis describes the multiple processes to be performed by these devices as a data flow that is "authorize[d] and establish[ed]." Ermis, [0110], [0108]. Ermis explains that this data flow includes the following processes to be performed:

> [i] the price of electricity can be obtained in real-time from the energy company controller interface,
>
> [ii] …the electricity controller can control the start time of the washing machine, and
>
> [iii] …the electricity controller can use the price of electricity and a planned wake-up alarm time defined in the alarm clock to make decisions for when the washing machine should be started.

Ermis, [0111].

84. Ermis provides another example of a data flow between the user interface 18, TV, and video recorder to perform a user command to record a TV program. *See* Ermis, [0045]-[0046]. Ermis describes the following sequence of performing multiple processes:

i. The user 16 sends a "task request…via user interface 18" to the system 12 "to record a certain TV program." Ermis, [0045].

ii. The system 12 "relays the command to the system entity of the video recorder." Ermis, [0045].

iii. The TV's system entity sends a "confirmation" to the user 16 "that the requested task will be performed." Ermis, [0046].

iv. The video recorder records the TV program. Ermis, [0046].

v. The TV's system entity sends confirmation to the user 16 that the task "has been successfully completed." Ermis, [0046].

85. In many cases, a data flow described by Ermis includes a triggering condition that causes the data flow to be executed. *See*, *e.g.*, [0101]-[0102] (data flows and "associated triggering criteria"), [0105], [0110]-[0113]. These triggering conditions are stored as part of the data flows. *See* Ermis, [0008] ("The interaction flow database contains information defining data flows that are permitted between identified ones of the resource nodes and associated triggering criteria defining when identified ones of the data flows are to be performed.").

86. All of these examples above illustrate that a data flow, associated with relationships between multiple devices, include a sequence of multiple processes that are performed by the multiple devices. Therefore, Ermis's data flow *defines a sequence of performing multiple processes* and renders obvious *a task processing schedule parameter*.

87. As explained above at limitation [1.2.2], Ermis's data flow is included within the *relation profile* ("friendship relationships and associated data flows," Ermis, [0128]). Ermis's "relationships" between machines are "exercise[d]…in performance of various task requests." Ermis, [0044].

88. In more detail, Ermis confirms that its data flows and associated triggering criteria are part and parcel with the friendship relationships between devices because they are created together and stored together. *See* Ermis, [0128] ("[t]he interaction execution engine 1312 **adds the selected friendship relationships and associated data flows to the interaction flow database** 1308"); Fig. 24 (illustrating a request to establish new relationships that enable recommended data flows); [0101], [0105]. Ermis further explains that "[t]he interaction flow database 1308 contains (assembles) information that defines data flows that are permitted between identified ones of the resource nodes 1330." Ermis, [0101].

89. That Ermis's data flows are included with friendship relationships, and are therefore included within *relation profiles*, is also confirmed by Ermis's explanation that the "one or more further friendship requests" are sent "**with information identifying the functionality** that will be created for the user by the one or more further recommended data flows." Ermis, [0132].

90. Therefore, Ermis renders obvious *a relation profile* (friendship relationship and associated data flow) *including a task processing schedule parameter* (data flow) *which defines a sequence of performing multiple processes* (examples of multiple steps to perform various data flows, described above).

[1.2.4] ***and for requesting selected machines of the multiple machines to perform the multiple processes,***

91. Ermis explains that, after a data flow is established, the system 12 (*relation server*) requests the selected machines to perform the multiple processes.

92. Describing the execution of data flows, "[t]he interaction execution engine 1312 can…**control operations performed by the resource nodes** on the data." Ermis, [0101]-[0102].

93. In the example of turning off lights when the user is away, "**the interaction execution engine 1312**…signal[s] the light controller to **turn off the lights**" and later to "signal the **security camera to output the picture/video**…." Ermis, [0110].

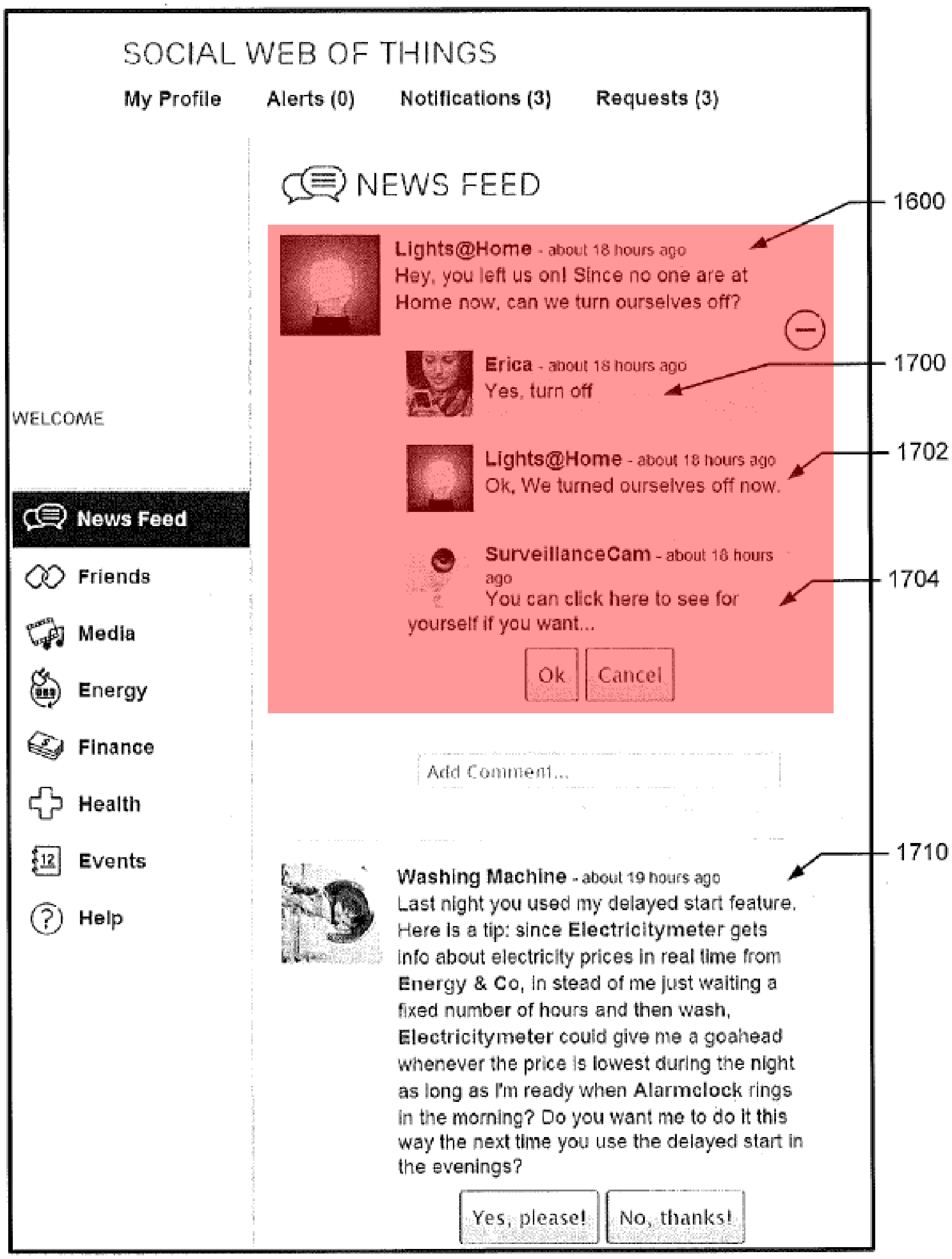


FIG. 17

Ermis, Fig. 17 (annotated)

94. In the example of turning on the washing machine when the price of electricity is low and before a wake-up time of an alarm clock, the "interaction execution engine 1312 first requests from the electricity meter to give it "a goahead whenever the price is lowest during the night as long as [the laundry] is ready when Alarmclock rings in the morning," Ermis, Fig. 17, and responds by **turning on the washing machine**." Ermis, [0113], [0102] ("The interaction execution engine 1312 can…control the flow of data between resource nodes and further control operations performed by the resource nodes on the data.").

95. In the example data flow for recording a TV program, Ermis explains that entities within the system 12 (e.g., "the TV's system entity (operative within system 12)," Ermis, [0045]) issue commands from the system 12 (*relation server*) to perform the requested task:

> In this case, the system entity of the TV can take responsibility for the request from the user 16 and relay the command to the system entity of the video recorder, which could, for example, be a representation of a physical device, a software functionality in the system, or a service provided via the network. The networked object video recorder 10 will actually execute the job, i.e., which is essentially subcontracted to it by the TV's system entity in system 12….

Ermis, [0046].

96. Therefore, the TV system entity, which is part of the system 12 (relation server), not the TV itself, requests that the video recorder record the TV program to complete the requested task. Specifically, Ermis explains that "the TV's system entity" is "**operative within system 12**." Ermis, [0045].

97. Thus, the system 12 issues the command to the video recorder, which may be "a physical device" separate from the system 12 itself. Ermis, [0046]; *see also* [0054] (describing "The TV (networked object 10) and/or its corresponding system entity in system 12").

98. In each of these examples, the user's interface 18 ("mobile phone") is included within the selected machines to perform the task because the relation server requests confirmation from the user to implement or perform data flows. *See* Ermis, [0050], [0109].

99. Thus, Ermis describes *requesting selected machines* (networked objects 10 (Ermis, Fig. 1)/resource nodes 1330 (Ermis, Fig. 13) of a friendship relationship for performing an associated data flow) *of the multiple machines* (networked objects 10/resource nodes 1330) *to perform the multiple processes* (system 12/resource management node 1300 instructs each identified machine to perform tasks to fulfil user request).

[1.3] ***wherein the relation server generates a new relation profile based on an intervention by a user in the relation profile,***

100. As explained at limitation [1.2.3] above, Ermis describes creating new relationships including new data flows (a *new relation profile*) based on user actions, such as in the example of establishing a relationship with a rental car in Figure 24. This creation of new relationships and associated data flows is *based on an intervention by a user*.

101. Specifically, the rental car in Figure 24 requests that the user allow new relationships to be generated between the rental car and "a media server device and a phone terminal" as well as "a map application, and a navigator application." Ermis, [0126]; *see* Ermis, Fig. 24.

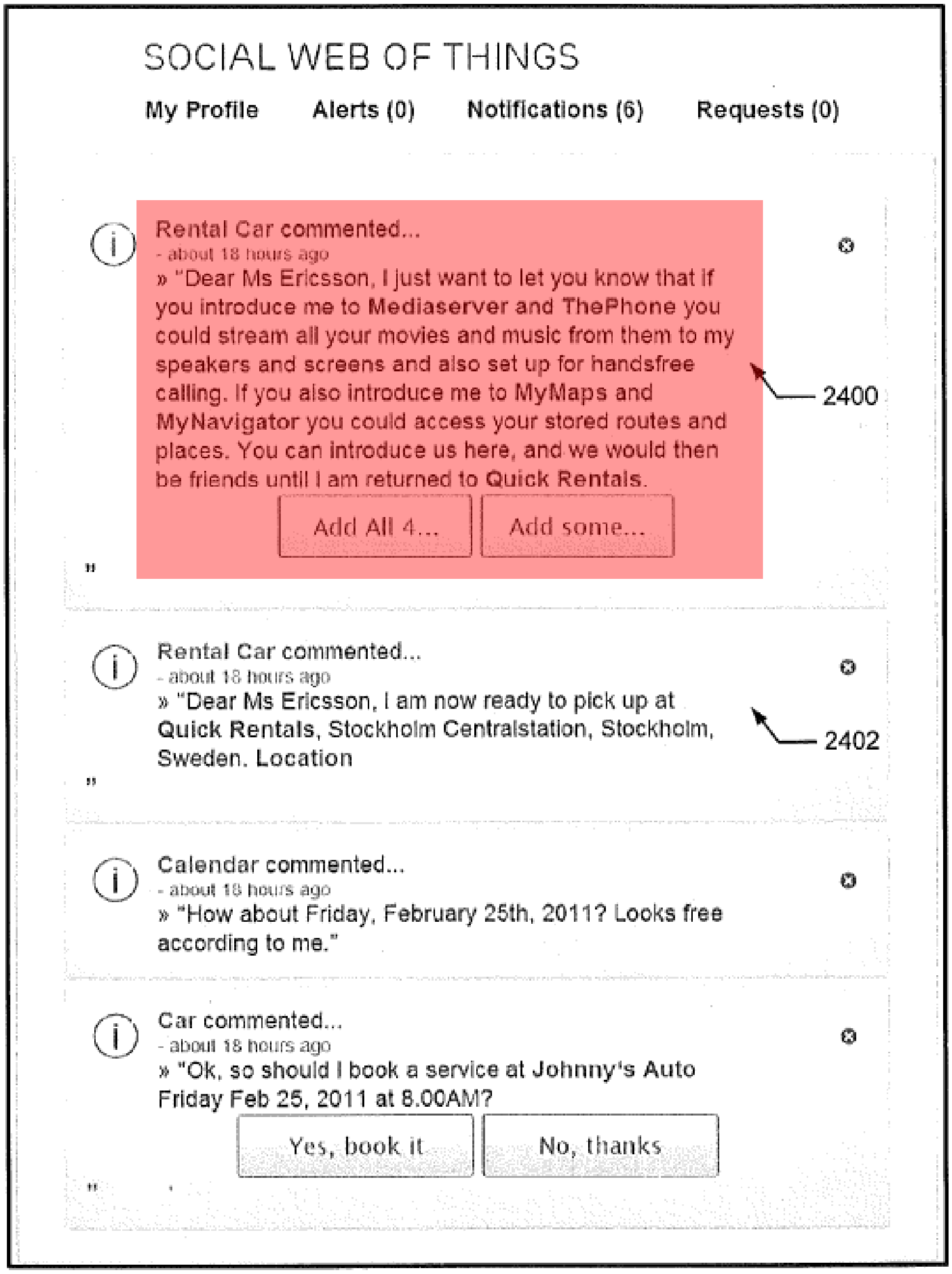


FIG. 24

Ermis, Fig. 24 (annotated)

102. These new relationships are added based on the user's selection of the "Add all 4…" button:

> The user can establish a friendship status between the identified resource nodes **by selecting the "Add All 4. . ." button**, or can select which of the resource nodes are to have friendship satisfy selecting the "Add some . . ." button. The interaction execution engine 1312 **adds <u>the selected friendship relationships</u> and associated data flows to the interaction flow database 1308**.

Ermis, [0128].

103. Thus, Ermis renders obvious *wherein the relation server* (interaction execution engine 1312 and/or interaction flow database 1308 within resource management node 1300) *generates a new relation profile* ("adds the selected friendship relationships") *based on an intervention by a user in the relation profile* (user selecting "Add All 4…" button).

104. In another example, Ermis describes establishing a new relation with a new machine in the event the machines currently associated with the user are unable to perform a task. Referencing "the previously described example with the TV and the video recorder (e.g., VCR)," Ermis explains a scenario in which "the VCR for some reason was unavailable or incapable of executing the requested task (e.g., recording a TV program)." Ermis, [0053]. "In this case, the TV (or more precisely the system entity in system 12 which corresponds to the TV) may be aware of other options for performing the requested task, e.g., other networked nodes that have the needed capability but with which it may or may not have a direct friend relationship." Ermis, [0053]. "[I]f the TV's owner/best friend (e.g., user/human 16 or networked object 10) has a friend that is connected to a system entity that can provide the functionality needed, the system 12 can send a message to that system entity requesting that, e.g., the requested program be recorded and stored." Ermis, [0053].

105. This collaboration with a new device to implement a new data flow also renders obvious *a new relation profile*. This new relation profile is also established in response to an intervention by a user because "the TV could be set to ask its owner if it is permitted to contact the owner's friend to ask for the needed favor." Ermis, [0054]. The owner/user approving this request to the friend renders obvious *an intervention by a user*.

106. Thus, Ermis renders obvious *wherein the relation server* (the relation management function 22 of the system 12/interaction flow database 1308 of resource management node 1300) *generates a new relation profile* (establishing a new friendship relationship and associated data flow to perform a task) *based on an intervention by a user in the relation profile* (user 16 approves recommended friendship relationship and associated data flow).

[1.4] ***wherein the task processing schedule parameter configures a start time of at least one process of the multiple processes, and***

107. Ermis provides multiple examples in which a data flow and associated triggering criteria (*task processing schedule parameter*) configures a start time, as claimed.

108. In the example data flow for turning off the lights, the process of turning off the lights starts when "one or more lights have been left on **more than a threshold time**." Ermis, [0109]. When lights have been left on more than a threshold time, a message is communicated "requesting the user's instructions for turning the lights off or leaving the lights on." Ermis, [0110]. When the user selects the "yes, turn off" button, the lights controller turns off the lights. Ermis, [0110]. Starting the process of turning off the lights after a threshold time (*start time*) renders obvious "*a start time of at least one process of the multiple processes*."

109. In the example data flow for starting a washing machine, the data flow configures a start time of the washing machine. "The recommendation engine 1310 has further determined that…the electricity controller can control **the start time** of the washing machine, and determined that the electricity controller can use the price of electricity and a planned wake-up alarm time defined in the alarm clock to make decisions when the washing machine should be started." Ermis, [0111]-[0112].

110. With this data flow enabled, "[t]he interaction execution engine 1312 determines when the triggering criteria is satisfied (e.g., price of electricity has dropped below the define[d] threshold value and/or less than a defined length of time remains before the wake-up time) and responds by turning on the washing machine." Ermis, [0113].

111. This data flow establishing the start time for a washing machine renders obvious wherein the task processing schedule parameter configures a start time of at least one process of the multiple processes.

112. In addition, with reference to Ermis's first example scenario in which a user requests that a TV program be recorded, the data flow (*task processing schedule parameter*) to record a TV program includes a start time at which the video recorder device should begin recording. *See* Ermis, [0046] ("the system entity of the TV can…relay the command to the system entity of the video recorder" to record the TV program). Ermis further explains that the "system entities profiles" include "data about their functionality, such as "**future planned (timer set) events**." Ermis, [0091]. "This enables functionality and interaction such as that exemplified by the TV and the video recorder scenario described above." Ermis, [0091]. Accordingly, the data flow including a request to record the TV program by Ermis's system 12 to the video recorder would include a "timer set" instruction, which also renders obvious *a start time of at least one process of the multiple processes*.

113. Thus, Ermis renders obvious *wherein the task processing schedule parameter* (data flow) *configures a start time of at least one process of the multiple* *processes* (determining a start time of a washing machine or a start time for recording a TV program).

[1.5] ***wherein the multiple machines includes at least one of a home appliance, a smart phone or a search engine.***

114. Ermis's machines include a TV, a video recorder, a "washing machine," an "alarm clock," and lights (*home appliance*). Ermis, [0046], [0109]-[0111], limitations [1.2] and [1.4].

115. Ermis also describes the "user's application interface 18" as a "mobile phone" which communicates with the server system 12, Ermis, [0047], and runs various applications, [0049]. Accordingly, a POSITA would have recognized this mobile phone as *a smart phone*.

116. Accordingly, Ermis renders obvious *a home appliance* (TV, video recorder, washing machine, alarm clock, lights), or *a smart phone* (mobile phone).

b) Claim 2

[2.0] ***The system of claim 1, wherein: a first machine of the multiple machines forwards a command received from the user to the relation server,***

117. In Ermis, a command from the user is received by the system 12 via the user interface 18, which may be the user's mobile phone. *See* limitation [1.5].

118. In more detail, Ermis explains that "the user 16 can for example send **a task request**, for example via **user interface 18** and API 26 **to the TV's system entity (operative within system 12)** requesting the system 12 to record a certain

TV program." Ermis, [0045].

119. The path of this task request (*a command*) is illustrated in Figure 1 from the user 16 to the system 12, below.

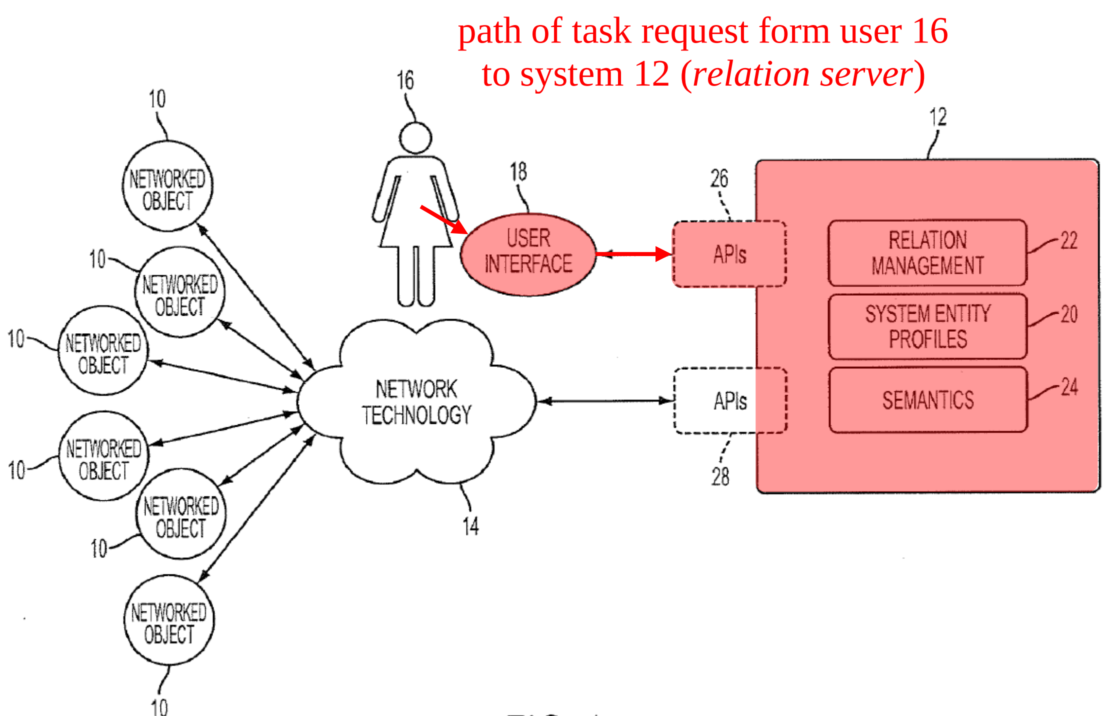


FIG. 1

Ermis, Fig. 1 (annotated)

120. As explained at limitation [1.1], the system 12 renders obvious the *relation server*.

121. The user interface 18 may be a user's mobile phone or other electronic device. For example, "[t]he users 16 can access the services provided by the system 12 via user interfaces 18 whose implementation which will vary depending upon, for example, the context and the **end user device** on which the particular user interface 18 is running, e.g., **mobile devices, computers, laptops, televisions or other devices**." Ermis, [0039]. Therefore, the end user device renders obvious *a first machine*.

122. Therefore, Ermis renders obvious *a first machine of the multiple machines* (end user device with user interface 18) *forwards a command* ("send[s] a task request") *received from the user* (from user 16) *to the relation server* (to the system 12).

[2.1] ***the relation server controls the selected machines for performing the multiple processes according to the command, and***

123. As explained at limitation [1.2.4] above, Ermis's system 12 controls the devices necessary for performing a user-requested task to perform the task. In the TV program recording example, Ermis explains that the system 12 (via the system entities for the TV and recorder) issues commands to the TV and video recorder (machines necessary to perform the task request) to record the TV program. *See* Ermis, [0045]-[0046]. In the examples of turning off lights and starting a washing machine, multiple processes are similarly performed according to a user command. *See* Ermis, Fig. 17 (turning off lights and displaying surveillance footage in response to user commands "Yes, turn off" and "Ok" button), Fig. 18 (user approving data flow).

124. These examples of performing multiple processes of a task request based on a user command are consistent with the ’518 patent’s description of commands. *See* ’518 patent, Fig. 2, 9:4-6 (commands S10 and S12 including “a movie recommendation command,” steps S30 and S32 selecting a movie, steps S42 and S44 confirming payment, etc.).

125. In addition, Ermis explains the data flow of a command from the system 12 to different machines required to perform a task:

> The interaction execution engine 1312 may, for example, provide a daisy chain data flow through three resource nodes, by **controlling** the first resource node to output data having certain characteristics (e.g. data format, data content, etc.) to an input interface of the second resource node, **control** the second resource node to operate on the data and to output data having certain characteristics (e.g. data format, data content, etc.) to an input interface of the third resource node. The interaction execution engine 1312 can similarly **initiate and control** the flow of data between resources nodes and further **control** operations performed by the resource nodes on the data.

Ermis, [0102].

126. Figure 13 “depicts another system architecture” providing more detail of the system 12 and networked objects 10. *See* Ermis, [0096], [0097] (“The user equipment node 1320 may correspond to the above-described user interface 18 of FIG. 1.”). Therefore, the system 12, including the interaction execution engine 1312, controls the networked objects 10 to perform the requested task.

127. Thus, Ermis describes *the relation server* (system 12) *controls the selected machines* ("control[s] operations performed by the resource nodes") *for performing the multiple processes according to the command* (to record a TV program in response to the user's task request).

[2.2] ***the selected machine includes the first machine.***

128. As explained at limitation [1.2.2], Ermis's system 12 identifies all the machines necessary to perform a task. As explained at limitation [1.2.4], the selected machines include the user's interface 18 or mobile phone. In the examples of turning off lights and starting a washing machine, the data flows are implemented and performed after approval by the user. *See* Ermis, Figs. 16-18. In the example of recording a TV program, the system 12 receives the command from the user interface 18 and sends this command to the relevant machines (the TV and video recorder, Ermis, [0045]-[0046]). After the video recorder records the TV program, Ermis, [0046], "the user 16 will receive a confirmation from the TV's system entity (again via API 26 and **user interface 18**) that the requested task will be performed, and later on that it has been successfully completed." Ermis, [0046].

129. Because the user interface 18 (*the first machine*) is a machine that is necessary to complete the requested task of recording a TV program (i.e., necessary to receive and convey the confirmation that the task will be completed and later was completed from the system 12 to the user 16), the user interface 18 (*the first machine*) is part of the group of *selected machines*.

130. Thus, Ermis renders obvious *the selected machine*[s] (networked objects 10 necessary to complete a user request) *include*[] *the first machine* (user interface 18).

c) Claim 3

[3.0] ***The system of claim 2, wherein: the command includes a reservation or a payment.***

131. Ermis describes another example in which the user requests that a car (with smart capabilities and part of the networked objects 10) schedule a reservation for a maintenance appointment with a mechanic, which is *a reservation*, highlighted below. *See* Ermis, Fig. 23.

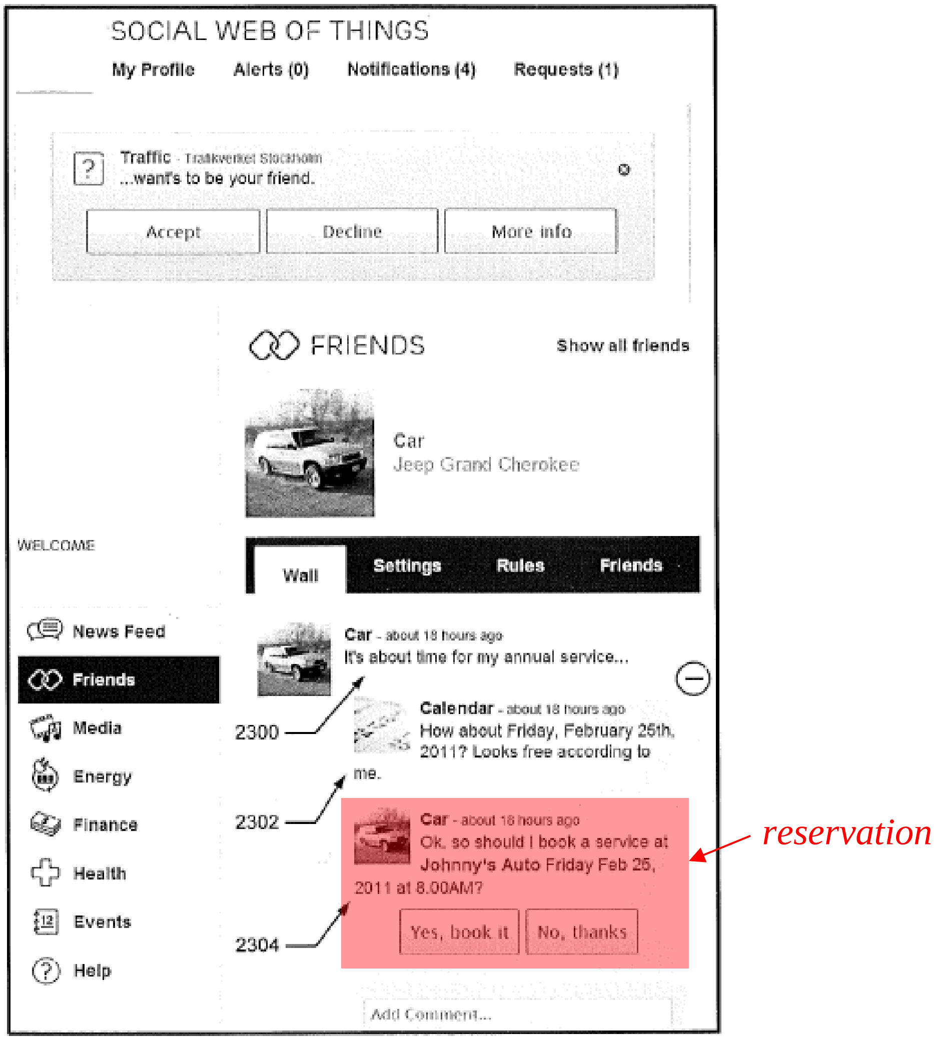


FIG. 23

Ermis, Fig. 23 (annotated)

132. "The interaction execution engine 1312 may…display an action message 2304 indicating that the car can **book the service** at the identified auto repair establishment at the indicated time of day and requesting the user's authorization to proceed with the **booking**." Ermis, [0125].

133. Ermis explains that when a data flow is performed, "the user 16 will receive a confirmation" both "that the requested task will be performed, and later on that it has been successfully completed." Ermis, [0046]. Therefore, if a user selects the "Yes, book it," button on the interface, *see* Ermis, Fig. 23, the system sends a confirmation that the maintenance service will be booked and that it has been successfully booked. Ermis explains that its system sends confirmation both when a task is scheduled to be performed as well as after it has been performed. *See* Ermis, [0046] ("the user 16 will receive a confirmation from the TV's system entity…that the requested task will be performed, and later on that it has been successfully completed.").

134. Thus, Ermis renders obvious *the command includes a reservation* (user requests that a car of the networked objects 10 schedule a maintenance appointment).

d) Claim 4

[4.0] ***The system of claim 2, wherein: the relation profile includes a capability set parameter, and the capability set parameter corresponds to information about capabilities required for performing the command.***

135. Consistent with the claim language, the '518 patent specification describes *a capability set parameter* as "includ[ing] information about capability parameters, which are required for relevant task [sic]." '518 patent, 7:33-36.

136. Ermis's system determines the required capabilities for a given data flow, which renders obvious *a capability set parameter* including *information* *about capabilities required for performing the command*. Specifically, "the network management system assigns the requested task to one of the plurality of networked objects which **has a capability to perform the requested task** and which has an established friendship relationship with the user." Ermis, [0085]. Therefore, the system first determines the capabilities required to perform a requested task and then assigns the task according to those determined capabilities.

137. Ermis's system also includes "[a] **service required capabilities database** which describes **what a service would require** to be meaningfully consumed." Ermis, [0066]. In other words, the system first identifies which capabilities of machines are required to perform any command/service. This information may stored as a parameter in a "service required capabilities database." Ermis, [0066]; *see also* [0071] (a "service may require certain capabilities on the service consumers"). This service required capabilities database is stored in the system 12. *See* Ex.1005, [0048] (system 12 is "implemented as a home gateway (GW) 30" in the embodiment in Figure 2), [0064]-[0066] ("service required capabilities database" is stored in "context management system 110, 112" within "the gateway 30").

138. Ermis also explains that after the capabilities for a particular service are identified, the system matches these capabilities with devices in the network that provide these capabilities. Specifically, a "Service Matchmaker 118 compares the **requirement of the services** in the Service Registry 114 with device capability and other context information of the PN, and the matched service and device pairs are stored in the context information of the PN." Ermis, [0072].

139. Ermis provides multiple examples of identifying capability set parameters for various tasks. In Ermis's example of starting the washing machine, the system first "**identifies capabilities** of **the washing machine**, **the electricity controller**, **the energy company controller interface**, and **the alarm clock**." Ermis, [0111].

140. In more detail, the system determines "the **functional capabilities that will be provided by the recommended data flow** if accepted by the user." Ermis, [0111]. For example, "[t]he recommendation engine 1310 has further determined[:]

> [1.] that the price of electricity **can** be obtained in real-time from the energy company controller interface,
>
> [2.] that the electricity controller **can** control the start time of the washing machine,
>
> [3.] …that the electricity controller **can** use the price of electricity and a planned wake-up alarm time defined in the alarm clock to make decisions for when the washing machine should be started."

Ermis, [0111]. Each of these functions that "can" be performed by the various machines (obtaining the price of electricity, controlling the start time of the washing machine, etc.) are capabilities of the devices that are needed to perform the requested task of turning off the washing machine when the price of electricity is lowest and before an alarm clock sounds. The data identifying these various capabilities, therefore, renders obvious *a capability set parameter*, as claimed. Other prior art at the time shows determining the capabilities required to perform a task and assigning these capabilities to a parameter (i.e., a *capability set parameter*) was commonplace and consistent with Ermis's disclosure. For example, Li describes "constructing a machine using a plurality of devices selected from a group of devices" in response to "an instruction from a user." Li, Abstract. Based on the instruction, the system "determine[s] one or more capabilities required for the machine" and "search[es] in the group for devices substantially matching at least one of the capabilities." Li, Abstract. These capabilities necessary to perform the requested user instruction are referred to with a parameter "required dynamic machine capabilities" including a list of capabilities A-H. *See* Li, Fig. 2A. Further, the data identifying the capabilities needed to a perform a particular task would be generated based on the metadata within each profile of the machine identifying the capabilities of the machine. *See infra*, claim 11; *e.g.*, Ermis, Abstract, [0008], [0011] ("metadata identifying capabilities of the resource nodes"), [0041] (The system entity profiles 20 may include, but are not limited to, information about…capability.”).

141. A POSITA would also have understood that the capability data described by Ermis corresponds to *a capability set parameter* as claimed based on the use of the term “parameter” in the ’518 patent, as well as the use of the terms parameter and metadata in the art at the time. Ermis’s disclosure of data corresponding to capabilities required to perform a command renders obvious *a capability set parameter*, such as the example parameters described by the ’518 patent. *See* Li, Abstract, Fig. 2A. Data is another term for parameter. *See* Bui, Abstract (describing “metadata or other suitable parameters”); Markki, Abstract (describing “metadata and/or other parameters”); Miller, [0007] (“The metadata includes parameters.”); Atherton, Abstract (describing “metadata as at least one parameter”); MacInnis, [0072]-[0073] (“the term ‘parameter value’ refers to the specific piece of data associated with a parameter”).

142. In Ermis, the data describing capabilities required to perform a data flow (*capability set parameter*) are included within the relationships between devices and associated data flows (*relation profile*). As explained at limitation [1.2.3] above, relationships between networked objects in Ermis include the data flows they enable because the relationships are created and stored “in performance of various task requests.” Ermis, [0044]; *see also* Ermis, [0128] (“[t]he interaction execution engine 1312 **adds the selected friendship relationships and associated**

data flows to the interaction flow database 1308”), Fig. 24 (illustrating a request to form a new relationships with multiple accompanying data flows). Because the data flows included within relationships between devices, the capabilities of the devices necessary to perform the data flows are also included within these relationships.

143. Further, Ermis explains that relationships between devices are created based on the functionality of the devices to perform tasks. *See* Ermis, [0087] (System entities “establish a level of friendship relation…**based on the…functionality**” of the devices.). Therefore, relationships between devices in Ermis include the capabilities of the devices required to perform a particular command and are stored in the interaction flow database 1308 and/or system entity database 1304. Ermis, [0044], [0128], [0098].

144. Therefore, Ermis renders obvious *the relation profile* (friendship relationship and associated data flow) *includes a capability set parameter* (data identifying capabilities of networked objects)*, and the capability set parameter corresponds to information about capabilities required for performing the command* (capability data identifying capabilities of networked objects of a particular friendship relationship and associated data flow to perform a task).

e) Claim 5

[5.0] ***The system of claim 2, wherein: the relation profile includes a group***

parameter, and the group parameter corresponds to identities of machines required for performing the command.

145. Ermis explains that multiple devices may be grouped together to perform a particular task request and the information identifying the group is included as part the corresponding friendship relationship and associated data flow (*relation profile*) in the system. For example, Ermis's "recommendation engine generates a recommended data flow between **at least two** of the resource nodes." Ermis, [0009]. The system groups the devices which execute the data flow by sending a "message" such as "a friendship request" to the relevant devices. Ermis, [0009]. "[T]he user's acceptance of the friendship request…add[s] **information** to the interaction flow database **that identifies a friend relationship between the at least two resource nodes**." Ermis, [0009].

146. The information identifying the at least two resource nodes that form the relationship renders obvious *a group parameter*. As explained by Ermis, these machines within the friend relationship were selected to perform "a recommended data flow," which shows that they are *identities of machines required for performing the command*. Ermis, [0009].

147. Ermis provides an example of a group parameter with reference to lights of a smart home. In Figure 14, below, the lights are grouped and labeled with the string "Lights@Home." Ermis, Fig. 14.

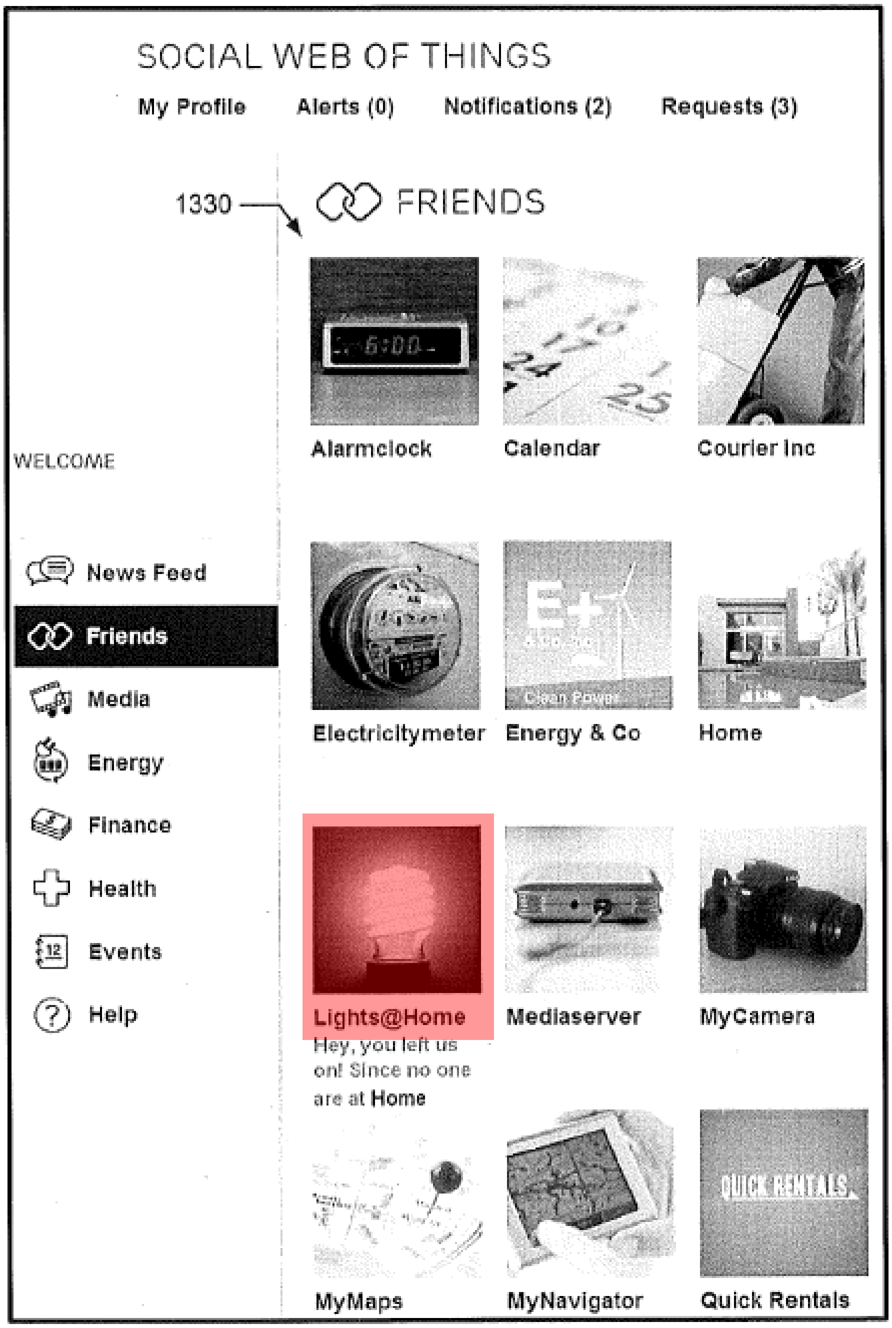


FIG. 14

Ermis, Fig. 14 (annotated)

148. As shown in Figure 14, and Figure 16 below, the system may recommend that a dataflow be implemented that detects when the homeowner leaves the house. As shown in Figure 16, below, the system sends the user 16 a message, as if from the "Lights@Home" group saying "Hey, you left us on! Since no one are [sic] at Home now, can we turn ourselves off?" Ermis, Fig. 16. The user 16 may then respond with yes or no.

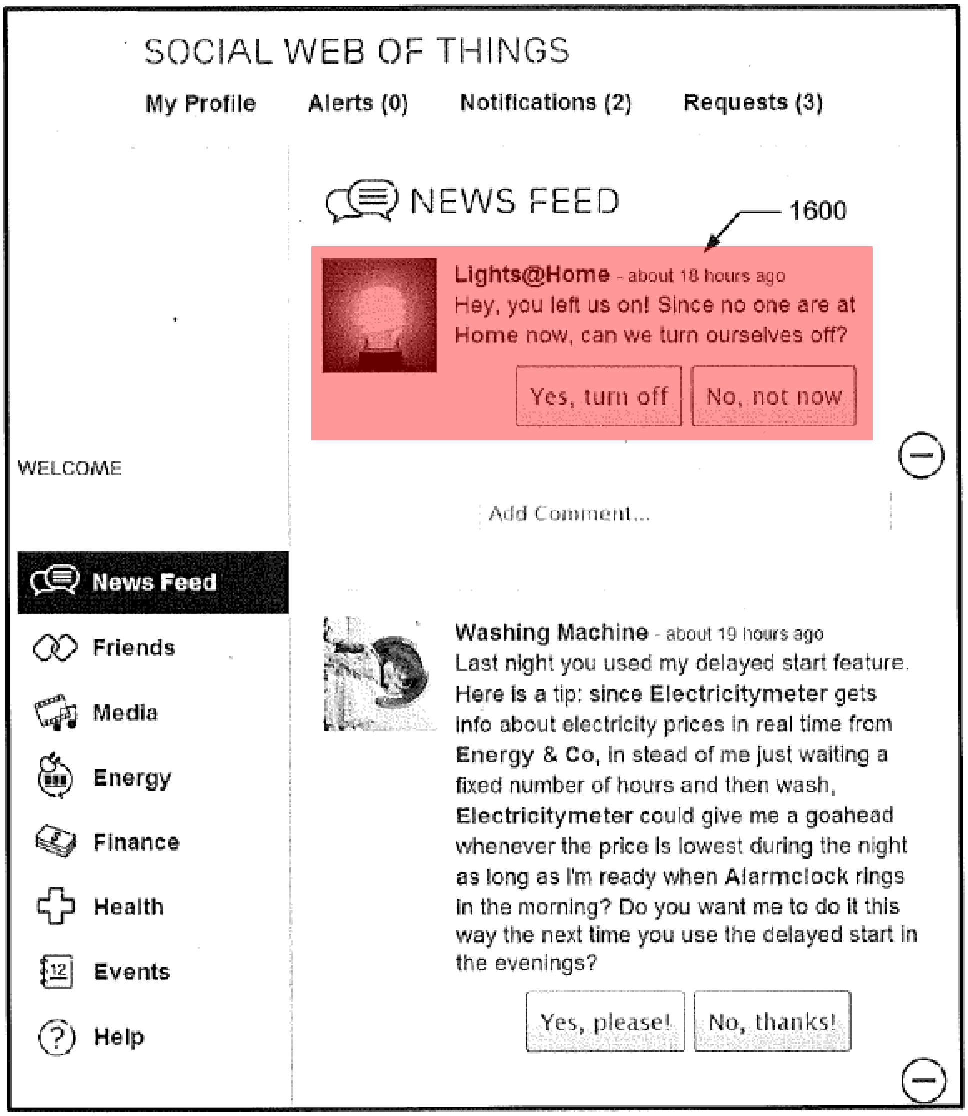


FIG. 16

Ermis, Fig. 16 (annotated)

149. Because the "Lights@Home" includes the multiple light devices in the smart home that will be turned off by the recommended data flow (the *command*, claimed), information identifying the friend relationship between the lights in the "Lights@Home" group renders obvious *a group parameter* which *corresponds to identities of machines required for performing the command*. In the example, the group may also include any additional devices that perform processes of the data flow, such as the surveillance camera. *See* Ermis, Fig. 17.

150. Ermis explains that the process of identifying which machines are needed to perform a command is a part of creating any data flow. For example, as explained at limitation [1.2.2], in the example of a user requesting that the system record a TV program, the system 12 identifies all the machines required to perform the command including the **user interface 18**, **TV**, and **video recorder**. *See* Ermis, [0045]-[0046]. Ermis's Figure 25 shows "methods and operations for managing resource nodes and data flows therebetween using friendship requests." Ermis, [0034], [0130]. As shown in Figure 25, below, step 2404 of the process of creating a data flow includes "identifying a friend relationship between the at least two resource nodes and the associated triggering criteria." Ermis, Fig. 25.

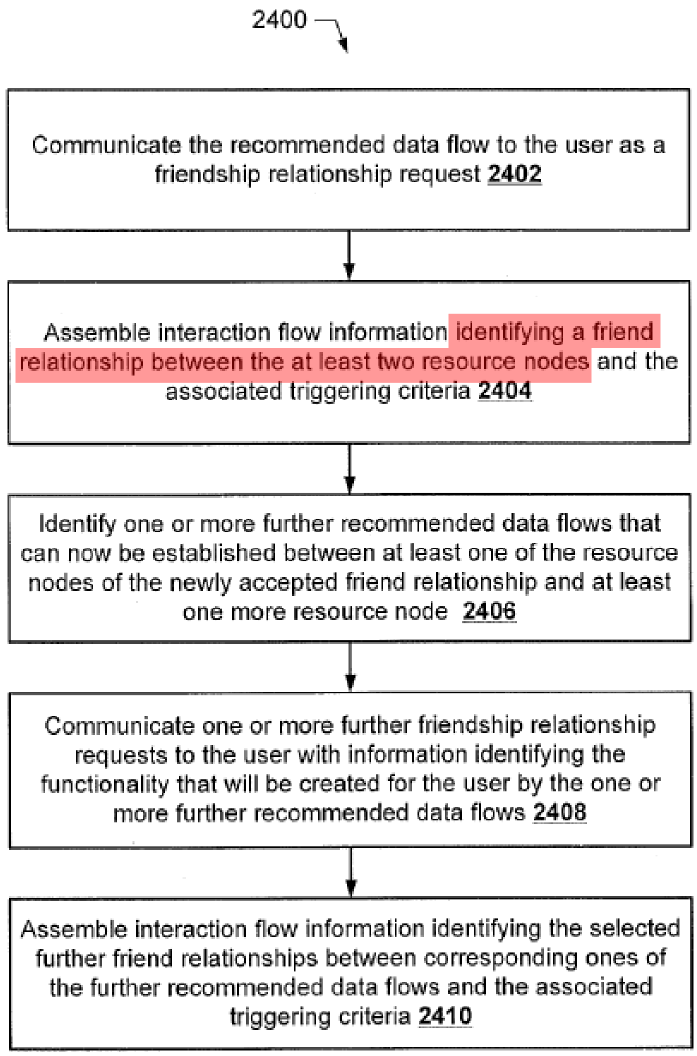


FIG. 25

Ermis, Fig. 25 (annotated)

151. The information identifying the machines in a relationship to perform a task (*group parameter*) is included in the friendship relationship and associated data flow (*relation profile*). *See* Ermis, [0009] (relationship data includes "information…that identifies [the] friend relationship between the at least two resource nodes.").

152. Thus, Ermis renders obvious *the relation profile* (friendship relationship and associated data flow) *includes a group parameter* (information identifying a friend relationship between at least two machines)*, and the group parameter corresponds to identities of machines required for performing the command* (machines of the friend relationship are those needed to perform a user-requested task).

f) Claim 6

[6.0] ***The system of claim 2, wherein: the relation profile includes a group identity parameter, and the group identity parameter corresponds to an identity of a group consisting of machines required for performing the command.***

153. Claim 5 recites "[t]he system of claim 2, wherein: the relation profile includes a group parameter, and the group parameter corresponds to identities of machines required for performing the command."

154. Claim 6 recites "[t]he system of claim 2, wherein: the relation profile includes a group **<u>identity</u>** parameter, and the group **<u>identity</u>** parameter corresponds to **~~identities~~** **<u>an identity of a group consisting</u>** of machines required for performing the command."

155. As explained for claim 5, Ermis provides an example of a name assigned to groups of devices required for a command as well. The group of lights required to turn off the lights when the user leaves the house is labeled "Lights@Home." Ermis, Fig. 16; *see also* Ermis, [0041] (describing identifying machines based on data corresponding to an **identity** of each machine), [0060] (describing "the **identity** of a device, i.e., a networked object 10").

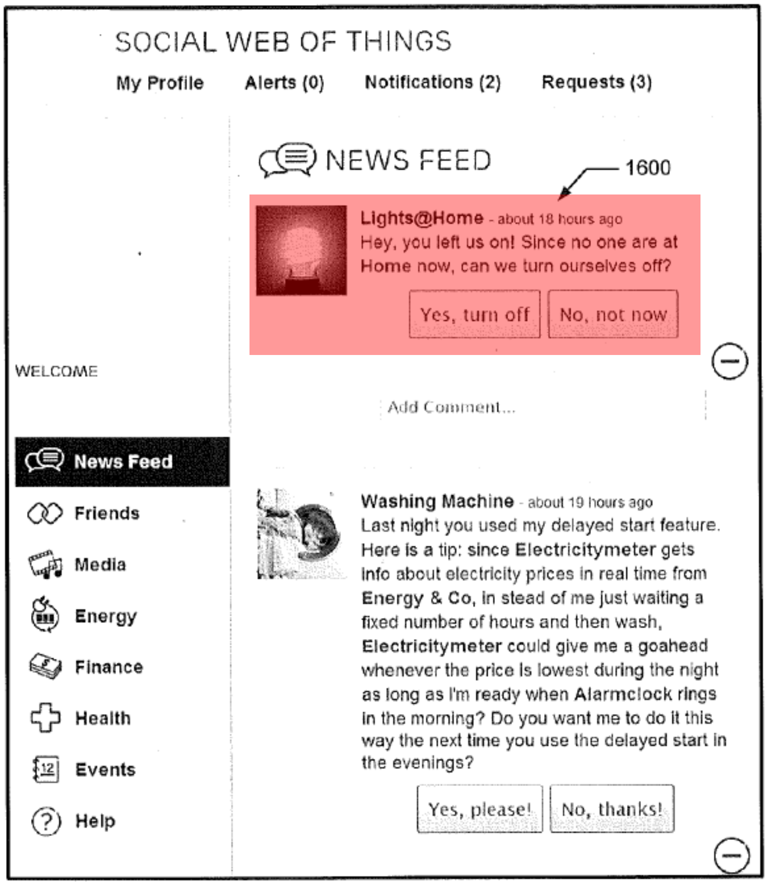


FIG. 16

Ermis, Fig. 16 (annotated)

156. Just as the information identifying the machines within a group (*group parameter*) is included within the friendship relationship and associated data flow (*relation profile*), the identify of that group (*group identity parameter*) is would also be included within the relationship data.

157. Thus, Ermis renders obvious *the relation profile* (friendship relationship and associated data flow) *includes a group identity parameter* (title identifying a friend relationship between at least two machines, such as "Lights@Home" title), *and the group identity parameter corresponds to an identity of a group consisting of machines required for performing the command* (title identifying machines of the friend relationship needed to perform a user-requested task).

g) Claim 7

[7.0] ***The system of claim 2, wherein: the relation profile includes a task description parameter, the task description parameter corresponds to information related to the command,***

158. As explained at claim 1, Ermis provides multiple examples of task requests and describes generating a series of processes, referred to as "data flows," to fulfil the task requests. *See supra* § VIII.A.2.a. These data flows render obvious *task processing schedule parameter*[s] because they *define a sequence of performing multiple processes*. *See, e.g.*, Ermis, [0045]-[0046] (describing a data flow for recording a TV program including the TV, a video recorder, and user device) (limitation [1.2.3]).

159. These data flows are stored in an "interaction flow database 1308" which "contains (assembles) **information** that defines data flows." Ermis, [0101]. This information that defines data flows renders obvious *a task description* *parameter* because it *corresponds to* ***information related to the command***.

160. As explained at claims 5 and 6 above, information related to data flow includes identifies of the machines involved in performing the data flow and may include information such as "a MAC address" or "explicit names." *See supra* § VIII.A.2.f-g; Ermis, [0060]. This information describing a data flow is later conveyed to the user in the form of a message. For example, "[t]he recommendation engine communicates **a message** to a user that **identifies the recommended data flow** for acceptance by the user." Ermis, [0009].

161. Examples of messages describing data flows are shown below in Figure 16, including data flows for turning off the lights in a house and starting a washing machine.

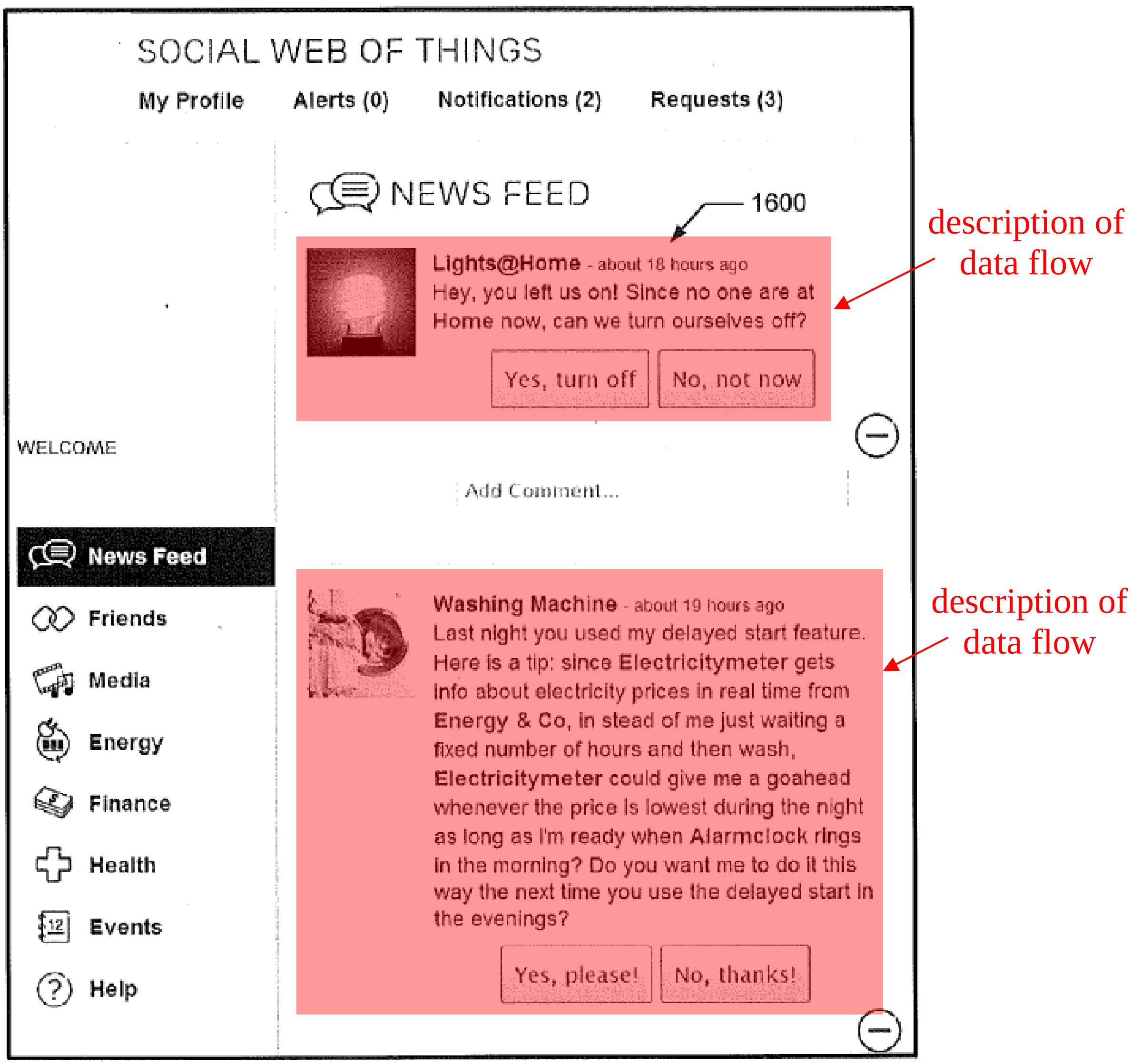


FIG. 16

Ermis, Fig. 16 (annotated)

162. Ermis explains that "a recommended data flow" is identified with "a natural language message 1710" that provides "an explanation of the functional capabilities that will be provided by the recommended data flow if accepted by the user." Ermis, [0111]; *see also* [0126].

163. In addition, because a data flow in Ermis is included within relationship data, information describing the task (task description parameter) is also a part of the relationship data.

164. Thus, Ermis renders obvious the relation profile (friendship relationship and associated data flow) includes a task description parameter, the task description parameter corresponds to information related to the command (information defining a data flow and messages to the user regarding the data flow).

[7.1] ***and the information is saved in a form of text.***

165. As explained at limitation [7.0], Ermis's data flows, including "information that defines data flows" (a *task description parameter*), are stored in an "interaction flow database 1308." Ermis, [0101], [0065]-[0069]. Ermis explains that the information stored in the system databases is used to fulfill "task requests which involve networked objects having at least a predetermined relationship level." Ermis, [0070]. To access this information, Ermis's system "quer[ies] the databases." Ermis, [0070].

166. These queries may "take various forms, for instance a SPARQL or XQUERY query." Ermis, [0070]. An XQUERY is used to query **text-based databases**. "XQuery" is "a query language **<u>for XML data sources</u>**." Forsblom,

[0030]; *see also* Lee, [0045] ("XQuery (a query language that is designed to process data **from XML files**…)"). Therefore, because Ermis's databases are queried with XQUERYs, the databases described by Ermis are XML databases. XML databases are text-based. *See* Gailloux, 5:1-3 ("**Text-based data**, such as **XML-structured data**, requires fewer resources…."); Forsblom, [0029] (describing "**text-based eXtensible Markup Language (XML)**"); Heinrich, [0032] (describing "**text-based XML** documents (XML—Extensible Markup Language)"). Therefore, information defining data flows (*a task description parameter*) stored in Ermis's XML databases is also text-based.

167. Because the databases described above including the *task description parameter*, are text-based, the *information* of the *task description parameter…is saved in a form of text*, as claimed.

168. That the task description parameter is saved in the form of text in Ermis system is confirmed by other disclosure as well. For example, *information related to* [a] *command* would include the identities of the machines required to perform the command, such as "a MAC address," or "explicit names." Ermis, [0060]. A MAC address and explicit name are text-based. This device identification data is included in the "system entity database 1304" which "contains (assembles) information that identifies communication (e.g., network 1380) addresses for the resource nodes 1330." Ermis, [0100]. Therefore, this data would be stored in the same format (as text) as opposed to being converted into some other non-text format for storage.

169. Therefore, Ermis renders obvious task description *information is saved in a form of text* (information in a text-based XML database and/or a "MAC address" or "explicit name[]" of machines needed to complete a task request).

h) Claim 8

[8.0] ***The system of claim 1, wherein: the task processing schedule parameter includes a sub-parameter configuring a start condition of at least one process of the multiple processes, or a sub-parameter configuring an allocated machine identity of at least one process of the multiple processes.***

170. The '518 patent explains that "the term 'sub-parameter' may denote a parameter including detailed items for the corresponding parameter." '518 patent, 4:29-31. Ermis renders obvious a sub-parameter as defined by the '518 patent.

171. As described at limitation [1.2.2] above, Ermis describes a task request in which the user requests that the washing machine start when the price of electricity is lowest but before the user's alarm clock goes off in the morning. *See* limitation [1.2.2].

172. This data flow (*task processing schedule parameter*) includes a *start condition* of the washing machine. Specifically, the start time of the washing machine is dependent on the lowest price of electricity during the night before the alarm clock goes off. *See* Ermis, [0111]-[0112]. This is described by Ermis as "the triggering criteria" which renders obvious *a start condition*. Ermis, [0112].

173. In addition, because this triggering criteria is part of the data flow for starting the washing machine, it also renders obvious a *sub-parameter*.

174. Therefore, Ermis renders obvious *the task processing schedule parameter* (washing machine data flow) *includes a sub-parameter configuring a start condition of at least one process of the multiple processes* (triggering criteria based on price of electricity and time of alarm clock).

175. In addition, Ermis explains that a MAC address is associated with each machine in its system. A MAC address is *an allocated machine identity*. It would have been obvious that the MAC address identifying the machines required to perform a process would be included in *a task processing schedule parameter*. Therefore, Ermis also renders obvious *the task processing schedule parameter* (data flow) *includes a sub-parameter configuring an allocated machine identity of at least one process of the multiple processes* (MAC address of machine configured to perform a process of a data flow).

i) Claim 9

[9.0] ***The system of claim 1, wherein: the relation server stores multiple machine profiles of the multiple machines.***

176. As illustrated in Figure 1, Ermis's system 12 (relation server) includes system entity profiles 20.

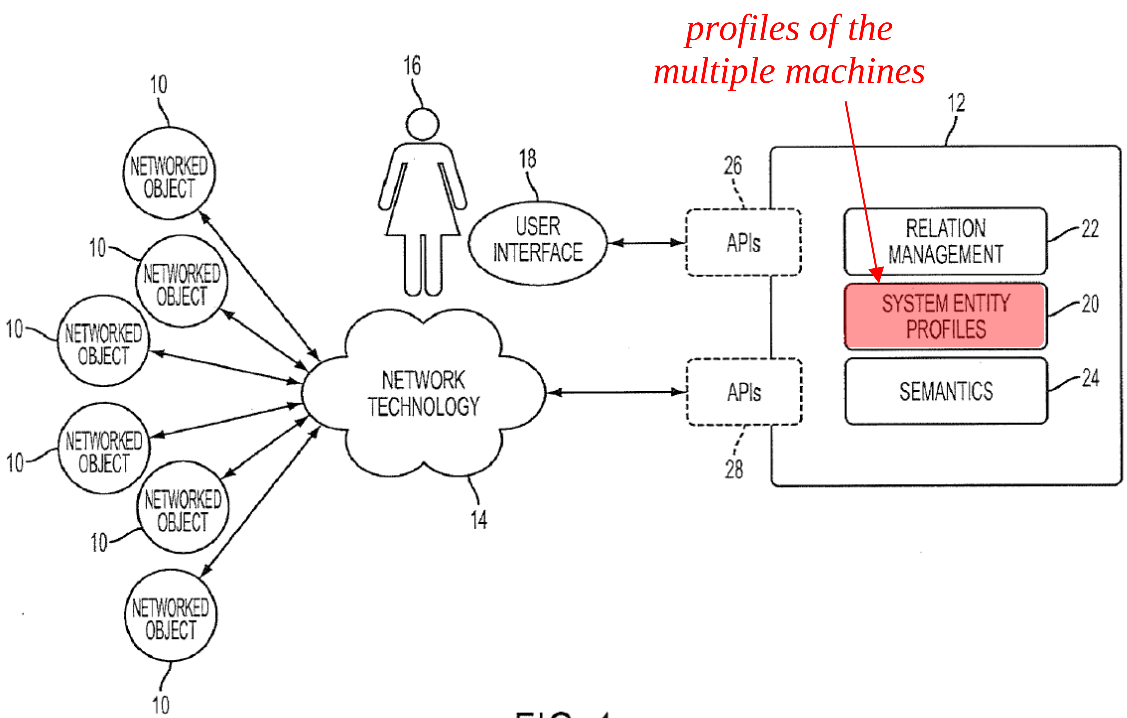


FIG. 1

Ermis, Fig. 1 (annotated)

177. Ermis explains that these profiles are created for each networked object 10 (*multiple machines*). Specifically, "**each networked object** is represented by **a unique and identifiable profile** that contains information about each object's system characteristics." Ermis, [0038]. "The networked objects 10 can also be represented and identified in the system 12 by unique profiles." Ermis, [0040]. In the Figure 13 embodiment, profiles are stored in the "system entity database 1304." Ermis, [0099].

178. Thus, Ermis renders obvious *the relation server* (system 12) *stores multiple machine profiles of the multiple machines* (stores a unique and identifiable profile for each networked object 10).

j) Claim 10

[10.0] ***The system of claim 9, wherein: at least one machine profile of the multiple machine profiles includes a status parameter, and***

179. The system entity profiles 20 described above at claim 9 include *a status parameter*. For example, "[e]ach system entity can be made aware of **its connected system entities profiles**, including but not limited to data about their functionality, dependencies, **current status** as well as previous and future planned (timer set) events, capabilities, mandate and responsibilities." Ermis, [0091].

180. In the user interface, "a 'Profile' element…enables the user 16 to access the system entity profiles 20." Ermis, [0050]. An example of user and machine profiles is shown in Figure 7B, below. *See* Ermis, [0074] ("By clicking on Bob's icon 204, Alice can further **check Bob's networked object information** as shown in FIG. 7(b).").

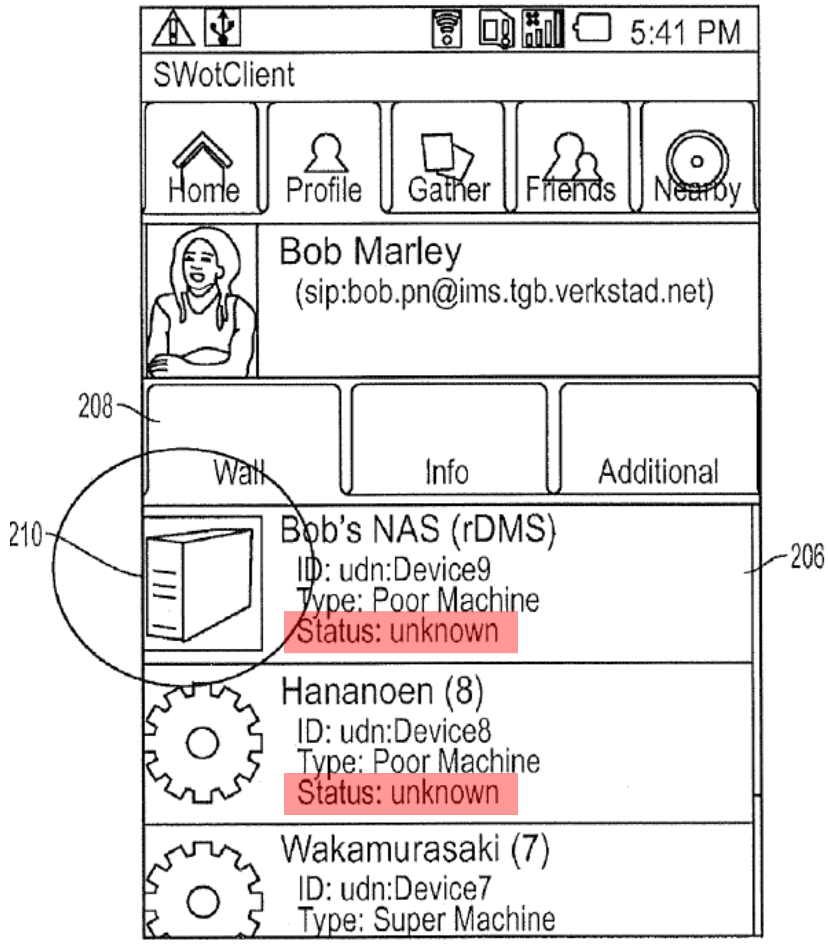


FIG. 7B

Ermis, Fig. 7B (annotated)

181. As shown highlighted above, each machine profile includes a status parameter. While each status parameter in the example of Figure 7B is set to "unknown," elsewhere, Ermis provides greater detail regarding this status information. For example, the system manages a "database, list or other data structure indicating **the current status of a particular device** or personal network." Ermis, [0069]. The status of a machine may include parameters such as "[c]urrent bandwidth and other situational parameters, such as ambient temperature, light level, physical orientation, etc." and can be "updated relatively often." Ermis, [0069]. These status parameters indicating the "current status of a particular device" renders obvious *a status parameter*. Ermis, [0069].

182. Thus, Ermis renders obvious *at least one machine profile of the multiple machine profiles* (unique and identifiable profile for a networked object 10) *includes a status parameter* (data "indicating the current status of a particular device" including a "situational parameter").

[10.1] ***the status parameter includes at least one sub-parameter of a current operation status sub-parameter, a current availability sub-parameter, a current process sub-parameter, a scheduled termination time of current process sub-parameter, a current execution function sub-parameter, or a scheduled termination time of currently executed function sub-parameter.***

183. Ermis describes data relating to the current operation status of the networked objects 10. For example, "[e]ach system entity can be made aware of its connected system entities profiles, including but not limited to data about their functionality, dependencies, **current status** as well as previous and future planned (timer set) events, capabilities, mandate and responsibilities." Ermis, [0091]. "[D]ata about" a networked object's "current status" renders obvious a *current operation status*. This information regarding the networked object's "current status" would be included within the status parameter of the system entity profile 20, described above for limitation [10.0].

184. Ermis also renders obvious the status parameter includes at least one sub-parameter of…current availability sub-parameter.

185. Ermis also describes a "database or list **indicating which devices and user-provided services are available** in particular user situations." Ermis, [0067]. Ermis provides an example in which "the VCR for some reason was unavailable." Ermis, [0053]. This availability data would also be included in the status parameter of the system entity profile and renders obvious a *current availability* for a networked object.

186. Thus, Ermis renders obvious *the status parameter* (data "indicating the current status of a particular device" including a "situational parameter") *includes at least one sub-parameter of a current operation status sub-parameter* (data corresponding to the "current status" of a "system entity"), *a current availability sub-parameter* (data corresponding to "which devices and user-provided services are available").

k) Claim 11

[11.0] ***The system of claim 9, wherein: at least one machine profile of the multiple machine profiles includes a capability parameter, and***

187. As explained at claim 9, Ermis explains that "each networked object is represented by **a unique and identifiable profile**." Ermis, [0038]. Ermis further explains that the profile for each networked object 10 "contains information about each object's system characteristics." Ermis, [0038]. This profile is described as a "system entity profile[] 20." Ermis, [0041]. "System entity information is assembled that identifies communication addresses for the resource nodes and **associated metadata identifying capabilities of the resource nodes**." Ermis, [0011].

188. "The system entity database 1304 contains information that identifies communication (e.g., network) addresses for each of the resource nodes 1330 of FIG. 14 and **metadata that identifies their respective capabilities** (e.g., functional capabilities, data input/output format(s), data content characteristics, performance characteristics, etc.)." Ermis, [0105].

189. When describing the networked objects 10, Ermis provides a list of some of the characteristics that may be included in a networked object's profile. Specifically, "[t]he system entity profiles 20 may include, but are not limited to, information about name, technical specifications, manufacturer, **<u>capability</u>**, location, history and other **<u>metadata</u>** associated with the respective networked object 10 or user 16." Ermis, [0041]; *see also* Ermis, [0111] ("the system entity database 1302…identifies **<u>capabilities</u>** of the washing machine, the electricity controller, the energy company controller interface, and the alarm clock."). A networked object's capability, included in "entity profile[] 20" is, therefore, considered "metadata," as described by Ermis, which renders obvious *a capability* *parameter* claimed.

190. Ermis's disclosure of data corresponding to capabilities of machines renders obvious *a capability parameter* as it is used in examples from the '518 patent. As explained at claim 4, above, metadata is another term for parameter. *See* Bui, Abstract; Markki, Abstract; Miller, [0007]; Atherton, Abstract.

191. Thus, Ermis renders obvious *at least one machine profile of the multiple machine profiles* ("system entity profile[] 20" of a networked object 10) *includes a capability parameter* (includes "metadata" including "information about…capability").

[11.1] ***the capability parameter includes at least one sub-parameter of a title of process sub-parameter, an input parameter sub-parameter, an output parameter sub-parameter, a process processing time sub-parameter, a process processing condition sub-parameter, or a function sub-parameter.***

192. The '518 patent explains that "[t]he input parameter may refer to information that is input in order to operate the corresponding machine." '518 patent, 5:13-17.

193. Ermis explains that the capability metadata (*capability parameter*) includes input and output format parameters: "The system entity database 1304 contains … metadata that identifies their respective capabilities (e.g., functional capabilities, **data input/output format(s),** data content characteristics, performance characteristics, etc.)." Ermis, [0105], [0041].

194. Therefore, Ermis renders obvious *the capability parameter includes at least one sub-parameter of…an input parameter sub-parameter* (an input technical specification of a resource node), *an output parameter sub-parameter* (an output technical specification of a resource node).

195. Ermis renders obvious *the capability parameter includes…a process processing condition sub-parameter*.

196. In addition, when describing forming data flows, after the system identifies "capabilities of the resource nodes," for example from the capability metadata described above at limitation [11.1], the system then also identifies "associated triggering criteria for when identified ones of the data flows are to be performed." Ermis, [0100]-[0101]. This "triggering criteria" associated with resource nodes would be included within the capability parameter of a system entity profile 20 for a networked object.

197. Therefore, Ermis renders obvious the capability parameter includes at least one sub-parameter of…a process processing condition sub-parameter (a "triggering criteria").

198. Ermis renders obvious the capability parameter includes…a function sub-parameter.

199. Ermis also explains that "[a]ny system entity associated with system 12 can be aware of **functionality** provided by networked objects 10 with whom it has a predetermined relationship or level of friendship, e.g., the **functionality** of networked objects 10." Ermis, [0053]. The functionality of the networked objects 10 would be included in that objects system entity profile 20, managed by the relation management function 22. *See* Ermis, [0052] (describing identifying friend networked objects and the "functionality…they provide").

200. Therefore, Ermis also renders obvious *the capability parameter* (metadata" including "information about…capability" of a networked object 10) *includes at least one sub-parameter of…a function sub-parameter* ("functionality" of a networked object).

l) Claim 12

[12.0] ***The system of claim 9, wherein: at least one machine profile of the multiple machine profiles includes a machine identity parameter.***

201. Ermis explains that each of its networked objects 10 includes an identity value. For example, "[r]egarding the identities of the various entities described above with respect to FIGS. 4 and 5, the identity of a device 96, i.e., a networked object 10, can be for example provided by the device manufacturer, e.g., **a MAC address**. However, device identities could also be provided in other ways, e.g., on an IP level by a home DNS service or **using explicit names, e.g., entered into the system by user configuration**." Ermis, [0060].

202. A MAC address or name entered by the user are *machine identity* *parameter*[s].

203. These machine identities are part of the networked object's profile. *See* Ermis, [0041] ("The system entity profiles 20 may include, but are not limited to, information about **name**.").

204. Thus, Ermis renders obvious *at least one machine profile of the multiple machine profiles* ("system entity profile[] 20" of a networked object 10) *includes a machine identity parameter* (networked object identity, such as MAC address or user-assigned name).

m) Claim 13

[13.0] ***The system of claim 9, wherein: at least one machine profile of the multiple machine profiles includes a user identity parameter, and the user identity parameter corresponds to an identity of a user being capable of using a machine corresponding to the at least one machine profile.***

205. As explained at claim 9, each of Ermis's networked objects 10 includes "a unique and identifiable profile that contains information about each object's system characteristics." Ermis, [0038].

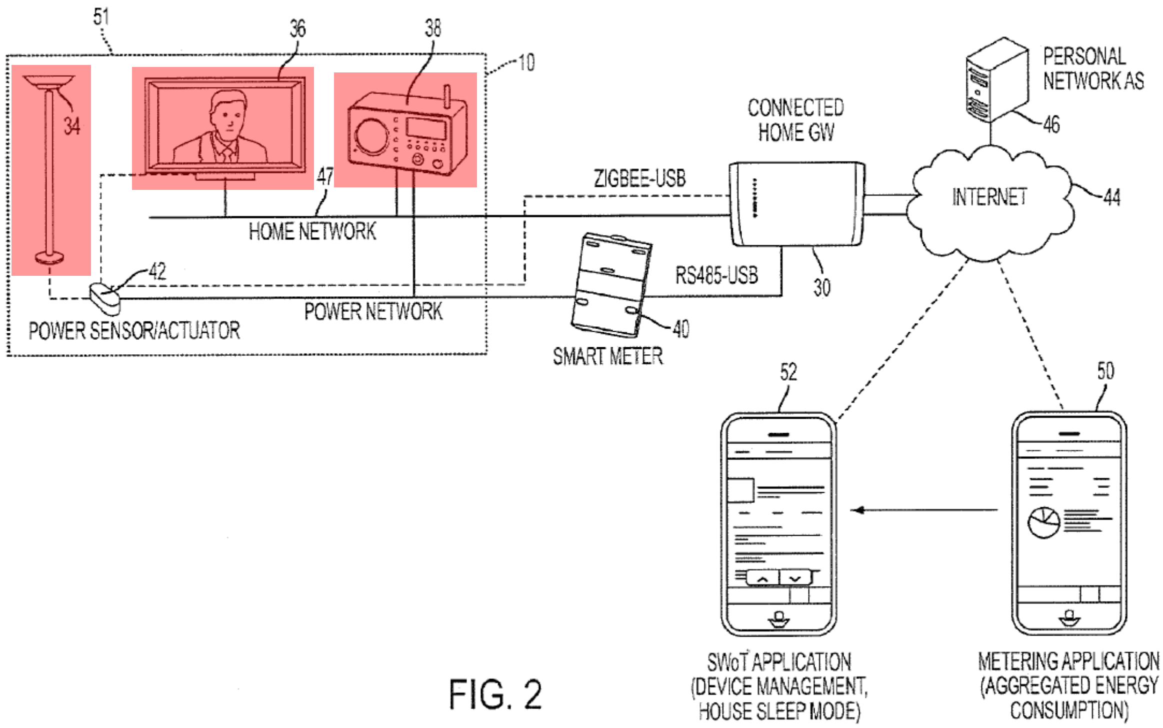
206. Ermis explains that the user 16 has an associated profile. For example, "[p]ersons that are using the system 12, i.e., the users 16, are represented and identified as entities in the system 12 by unique user profiles in the system." Ermis, [0040]. The user 16's profile is then connected to the profiles of any networked objects associated with the user 16. For example, "[t]he users connect their own system entity to system entities of networked objects." Ermis, [0087]. "The object's system entities connect to each other's profiles." Ermis, [0087]. In this way, a user can "connect[] his/her user profile to the device." Ermis, [0087].

207. As Ermis teaches, "[t]he objects' system entities connect to each other's profiles, i.e. establish a level of friendship relation as described above…as managed by a user (**authorized to do so and that has connected his/her user profile to the device**) at step 1204." Ermis, [0087]. As a result, "connected users and objects' system entities according to example embodiments may have reciprocal presence in each other's profiles." Ermis, [0090]. Thus Ermis teaches machine profiles that include a user identity parameter corresponding to a user capable of using the machine.

208. In addition, Ermis also refers to the application interface 18 as "the user's application interface 18, e.g.,…his or her mobile phone." Ermis, [0047]. The user "accept[s] (or reject[s]) the connection of the system entity of [a] new device" with the application interface 18. Ermis, [0047]. Therefore, the application interface 18 in Ermis is identified as being associated with the user 16. That the application interface 18 belongs to the user and that commands from the application interface 18 may be trusted as coming from the user 16 is "metadata associated with the" application interface 18, Ermis, [0041], which is stored within the machine profile of the application interface 18. This "metadata" associating the user 16 with the user interface 18 renders obvious *a user identity parameter*.

209. In addition, Ermis also explains that the user 16 also has an associated profile. For example, "[p]ersons that are using the system 12, i.e., the users 16, are represented and identified as entities in the system 12 by unique user profiles in the system." Ermis, [0040]. The user 16's profile is then connected to the profiles of any networked objects associated with the user 16. For example, "[t]he users connect their own system entity to system entities of networked objects." Ermis, [0087]. "The object's system entities connect to each other's profiles." Ermis, [0087]. In this way, a user can "connect[] his/her user profile to the device." Ermis, [0087].

210. In another example, Ermis explains that multiple users may be associated with a networked object. For example, "[g]roups of users could be created among users living within the same household, or other attributes that the user has decided to share." Ermis, [0086], [0122]. The networked objects can be connected to "**a particular user** (or group of users, e.g., a family)." Ermis, [0048]. In this example, Ermis identifies multiple machines that may be associated with each member of the group of users or family living in the smart home, for example, "a light, 34, a television 36, and a radio 38." Ermis, [0048]. These devices are highlighted in Figure 2, below.



Ermis, Fig. 2 (annotated)

211. In this example in which the devices are associated with multiple users, the "metadata" of these devices would indicate this association with multiple users. Ermis, [0041]. This metadata is stored as part of the machine profile of each of the networked objects shown above. Ermis, [0041]. This metadata indicates that any of the users in the group or family described can use the devices (light 34, television 36, or radio 38). Ermis, [0048].

212. Therefore, Ermis explains that networked objects can be connected to "a **particular user**." Ermis, [0048].

213. Therefore, Ermis renders obvious *at least one machine profile of the* *multiple machine profiles* (the machine profile of the application interface 18) *includes a user identity parameter* (metadata indicating that the application interface 18 belongs to the user 16 because the user 16 "has connected his/her user profile to the device").

n) Claim 14

[14.0] ***The system of claim 9, wherein: at least one machine profile of the multiple machine profiles includes a group identity parameter, and the group identity parameter corresponds to an identity of a user group being capable of using a machine corresponding to the at least one machine profile.***

214. As explained at claim 13, Ermis describes associating a user profile with the profile of a networked object. *See, e.g.*, Ermis, [0048], [0087]. "[M]etadata" in the entity profile 20 of the networked object would indicate this association. Ermis, [0041].

215. As relevant to claim 14, Ermis's disclosure of connecting profiles of networked devices to particular users also describes groups of users being associated with a networked object. For example, "**<u>[g]roups of users could be created</u>** among users living within the same household, or other attributes that the user has decided to share." Ermis, [0086], [0122]. In further detail, Ermis describes these networked objects as "a set of networked objects with which a particular user (or **<u>group of users</u>**, e.g., **<u>a family</u>**) **<u>has a relationship</u>**." Ermis, [0048]; *see supra* § VIII.A.2.m.

216. Thus, Ermis renders obvious *at least one machine profile of the multiple machine profiles* (profile of the light 34, TV 36, or radio 38) *includes a group identity parameter* (metadata of these devices indicating that they are associated with a group of users)*, and the group identity parameter corresponds to an identity of a user group being capable of using a machine corresponding to the at least one machine profile* (each user of the "group of users, e.g., a family" "has a relationship" with the light 34, TV 36, or radio 38).

o) Claim 15

[15.0] ***The system of claim 8, wherein: at least one machine profile of the multiple machine profiles includes an operating system parameter, and the operating system parameter corresponds to a type of an operating system which is used by a machine corresponding to the at least one machine profile.***

217. The '518 patent explains that "[t]he machine operating system parameter may include sub-parameters related to a type and a version of an operating system." '518 patent, 2:10-12.

218. Ermis explains that the user 16's mobile device has an operating system. "More particularly, the storage/memory 704 may include **an operating system**." Ermis, [0076]. As previously explained, each device in Ermis's system is associated with a "system entity profile[] 20," which contains a "technical specification[]" of the device. Ermis, [0041]. An indication of the operating system is a "technical specification[]" of the user 16's mobile device, and therefore would be indicated with "metadata" in the profile of the device so as to enable intercommunication between devices. Ermis, [0041].

219. Ermis recognizes that various networked objects 10 include potentially differing operating systems and ensures that the inputs and outputs of the various devices are interoperable. *See* Ermis, [0043] ("using transcoding information that is defined in the system entity semantic description database 1306 to transcode data that is output by one of the resource nodes into a format that is compatible with the input of another one of the resource nodes **to provide compatible communications**."), [0115] (ensuring that "brand-dependent semantic interaction language[s]" are interoperable). These differing operating systems would similarly be indicated with "metadata" indicating "technical specifications" of these devices. Ermis, [0041].

220. Thus, Ermis renders obvious at least one machine profile of the multiple machine profiles (system entity profiles 20 of networked objects 10) includes an operating system parameter, and the operating system parameter corresponds to a type of an operating system which is used by a machine corresponding to the at least one machine profile (metadata indicating the operating system of a networked object 10 and whether its output is compatible with an input of another networked object 10).

p) Claim 16

[16.0] ***The system of claim 8, wherein: at least one machine profile of the multiple machine profiles includes an interface parameter, and the interface parameter corresponds to an interface protocol between the relation server and a machine corresponding to the at least one machine profile.***

221. The ’518 patent explains that “[t]he machine interface parameter may include sub-parameters related to interfaces and interface protocols between the machines, and interfaces and interface protocols between each machine and the relation server.” ’518 patent, 2:13-16.

222. Ermis explains that the networked objects 10 interface with the system 12 via APIs 26 and 28. *See* Ermis, [0043]. To do so, each networked object 10 includes interface data stored in its system entity profile 20.

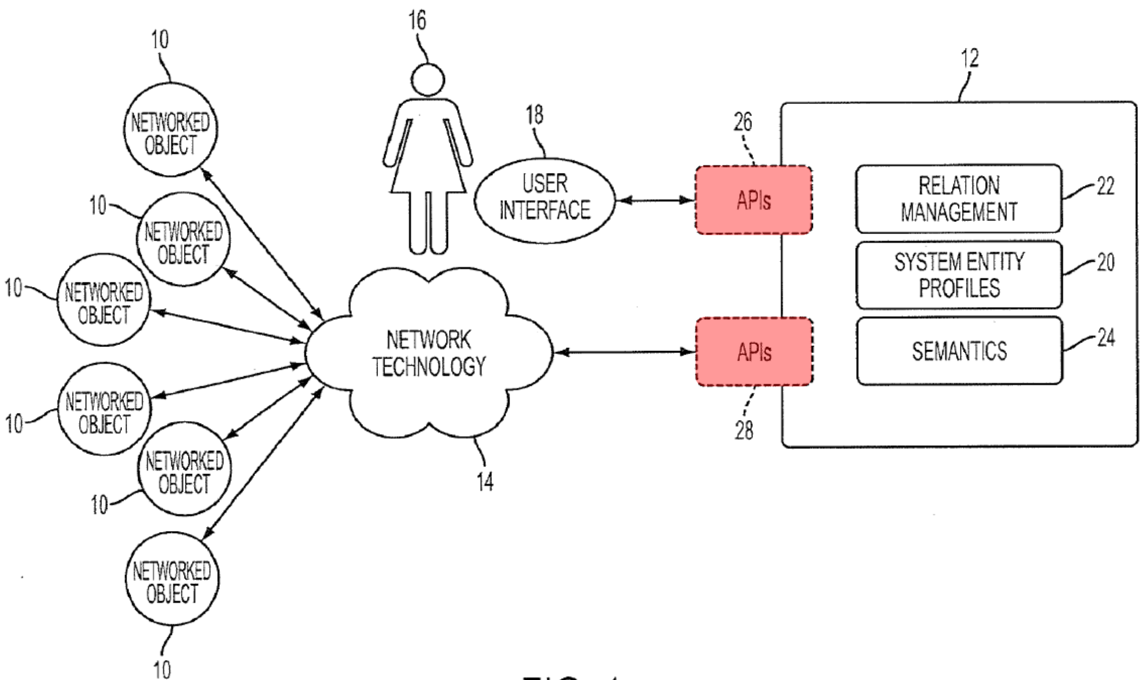
> The system entity database 1304 contains information that identifies communication (e.g., network) addresses for each of the resource nodes 1330 of FIG. 14 and **metadata that identifies their respective capabilities** (e.g., functional capabilities, **data input/output format(s)**, data content characteristics, performance characteristics, etc.).

Ermis, [0105].

223. A device’s “**metadata**” identifying “**data input/output format(s)**” *an interface parameter*. Further this data is included within the entity profiles for each device. In addition, this information would be included in its “technical specifications” which Ermis explains are part of “[t]he system entity profiles 20.”

Ermis, [0041].

224. In more detail, Ermis explains that "[t]he system 12 may interface with its objects 10 and users 16 through a set of **<u>Application Programming Interfaces (APIs) 26 and 28</u>**." Ermis, [0043]. These APIs 26 and 28 are highlighted in Figure 1, below.



Ermis, Fig. 1 (annotated)

225. As shown above, the APIs 26 and 28 are used to facilitate communication between the networked objects 10/user interface 18 and the system 12. *See* Ermis, [0044] ("The system 12 may interface with its objects 10 and users 16 through a set of Application Programming Interfaces (APIs) 26 and 28 in order to establish relationships using social mapping principles and also to exercise those relationships in performance of various task requests.”).

226. An API defines a set of rules and protocols that allow different devices to communicate. *See, e.g.*, Hedge, [0041] (An “API provides protocol configuration information.”); Hamilton, [0005]-[0018] (listing multiple “API[s]” that “provide a protocol” for a number of communication functions). Therefore, because the APIs 26 and 28 facilitate communication between the system 12 and the objects 10/interface 18, they *correspond[] to an interface protocol between the relation server* (system 12) *and a machine corresponding to the at least one machine profile* (networked objects 10 or application interface 18).

227. Thus, Ermis renders obvious *at least one machine profile of the multiple machine profiles includes an interface parameter* (system entity profile for network device includes technical specification metadata including interface data), *and the interface parameter corresponds to an interface protocol* (APIs 26 and 28 establish interface protocols) *between the relation server* (system 12) *and a machine corresponding to the at least one machine profile* (networked objects 10/interface 18 and system).

q) Claim 17

[17.0] ***The system of claim 1, wherein: the relation server saves information related to grouping the multiple machines.***

228. As explained at claim 5, in Ermis, "[a]ccording to example embodiments it is also possible to **aggregate multiple networked objects under a common profile**." Ermis, [0041]. Ermis provides an example in which a "group of location sensors may be identified in the system as one home location sensor." Ermis, [0041], [0047] ("group of system entities present in his or her Social Web of Things"). As explained above, the interaction flow database includes "information…that identifies a friend relationship between the at least two resource nodes." Ermis, [0009]. In the examples above, this information would be indicative of the machines required to perform commands (lights and surveillance camera in the turning off lights data flow example, electricitymeter, alarm clock, and washing machine in the washing machine example, etc.). Ermis, Figs. 16-18.

229. In Figure 1, Ermis further explains that this information is saved by the system 12 (*relation server*). Specifically, Ermis's system 12 includes a "relation management function 22" which manages "the relationship between networked objects." Ermis, [0038]. The system 12 also maintains "system entity profiles 20 for each networked object and user 16." Ermis, [0041]. This "relation management function 22…coordinates the interactions between the networked objects 10 by applying social management principles." Ermis, [0042]. In Figure 13, this information is saved in the interaction flow database 1308. Ermis, [0009].

230. Thus, Ermis renders obvious *the relation server* (system 12 and/or resource management node 1300) *saves information related to grouping the multiple machines* (stores information regarding the group of machines required to perform a task in the relation management function 22 and/or interaction flow database 1308).

r) Claim 18

[18.0] *The system of claim 1, wherein: the relation server generates a capability set required to execute a user command, and*

231. Ermis explains that after a request is received, the system "determine[s] device capabilities" of the various networked objects 10 "in response to [the] service request." Ermis, [0063]. "Determin[ing] device capabilities" needed "in response to a service request" renders obvious *generat*[ing] *a capability set required to execute a user command*.

232. This context management generating the capability set is performed by a component of the system 12. Figure 6 provides "a more detailed architectural schematic for implementing management of networked objects," such as the system 12. Ermis, [0019]; *see also* Ermis, [0085]. It shows that the system 12 includes a "context manager 110" which performs the context management described above, including "determin[ing] device capabilities." Ermis, [0063].

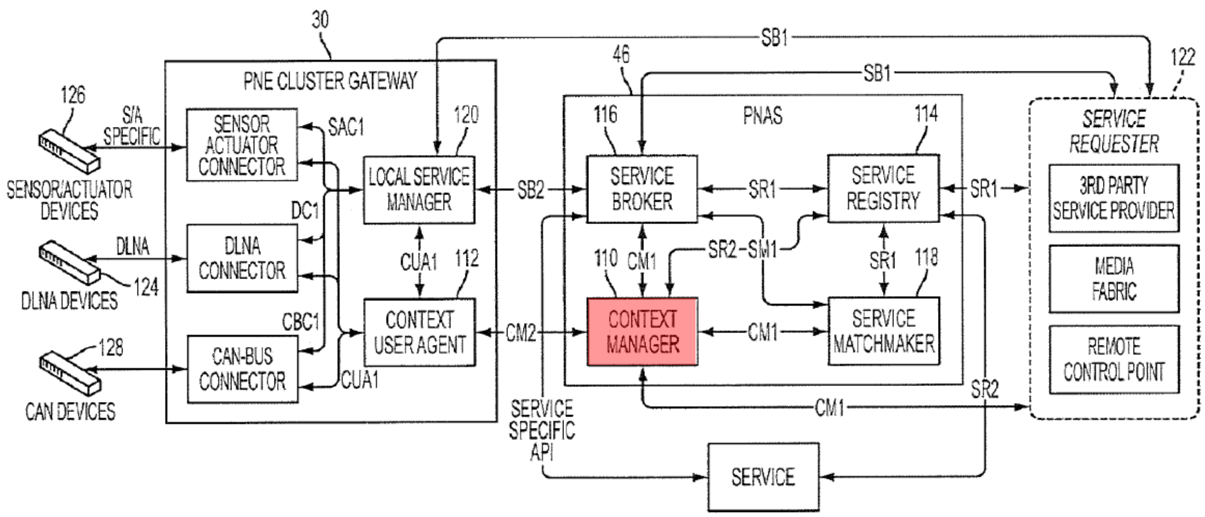


FIG. 6

Ermis, Fig. 6 (annotated)

233. Thus, Ermis renders obvious *the relation server* (context manager 110 of system 12) *generates a capability set required to execute a user command* (“determine[s] device capabilities…in response to a service request”).

[18.1] ***generates the task processing schedule parameter according to the capability set, a status parameter of at least one machine of the multiple machines, and a capability parameter of the at least one machine.***

234. Ermis further explains that generating the data flow is done according to the capabilities of the devices. In the example of recording a TV program, the system 12 first determines both that “the TV is in fact not capable of recording anything itself,” but “another networked object 10,” the “video recorder,” “can perform the requested service.” Ermis, [0045]-[0046]. Then, the system 12 generates a data flow in which the TV entity within the system 12 “relay[s] the command to the system entity of the video recorder." Ermis, [0046].

235. As explained at limitation [1.2.2], Ermis describes data flows which render obvious the *task processing schedule parameter* claimed. Ermis further explains that the system 12 generates the data flow. Specifically, "[t]he interaction flow database 1308 contains (**assembles**) **information that defines data flows**." Ermis, [0101].

236. Figure 13 "depicts another system architecture" providing more detail of the system 12 and networked objects 10. *See* Ermis, [0096], [0097] ("The user equipment node 1320 may correspond to the above-described user interface 18 of FIG. 1."), [0085] ("[E]ach feature or element can be used alone without the other features and elements of the embodiments or in various combinations with or without other features and elements disclosed herein."). Therefore, the system 12, including an interaction flow database 1308, assembles information that defines data flows.

237. Ermis's example of fulfilling a request from the user to record a TV program illustrates that the system 12 generates the data flow as well. When the user request is received, "the system entity of the TV can take responsibility for the request from the user 16 and relay the command to the system entity of the video recorder." Ermis, [0046]. These entities are "in system 12." Ermis, [0046]; *see also* Ermis, [0053].

238. Finally, as previously explained at claim 4, the capabilities of these devices are included in "**metadata identifying capabilities of the resource nodes**," which renders obvious a *capability set parameter* for the reasons described at claim 4. Ermis, [0008].

239. Thus, Ermis renders obvious *the relation server* (system 12)…*generates the task processing schedule parameter* (generates a data flow) *according to the capability set* (after determining the capabilities of the networked objects 10 based on the capability metadata).

s) Claim 19

[19.0] ***The system of claim 1, wherein: the intervention includes selecting some processes of the multiple processes.***

240. As explained at limitation [1.3] above, Ermis describes *generat[ing] a new relation profile based on an intervention by a user in the relation profile*. *See supra* § VIII.A.2.a [1.3]. Specifically, the rental car in Figure 24 requests that the user allow new relationships to be generated between the rental car and "a media server device and a phone terminal" as well as "a map application, and a navigator application." Ermis, [0126]; *see* Ermis, Fig. 24. As explained at limitation [1.3], the user's selection of the "Add All 4…" button is *an intervention by [the] user* that *generates a new relation profile*.

241. Ermis also illustrates an alternative option the user can select in which some but not all of the recommended data flows are added. *See* Ermis, Fig. 24, [0128] (User can "select[] the 'Add some…' button.").

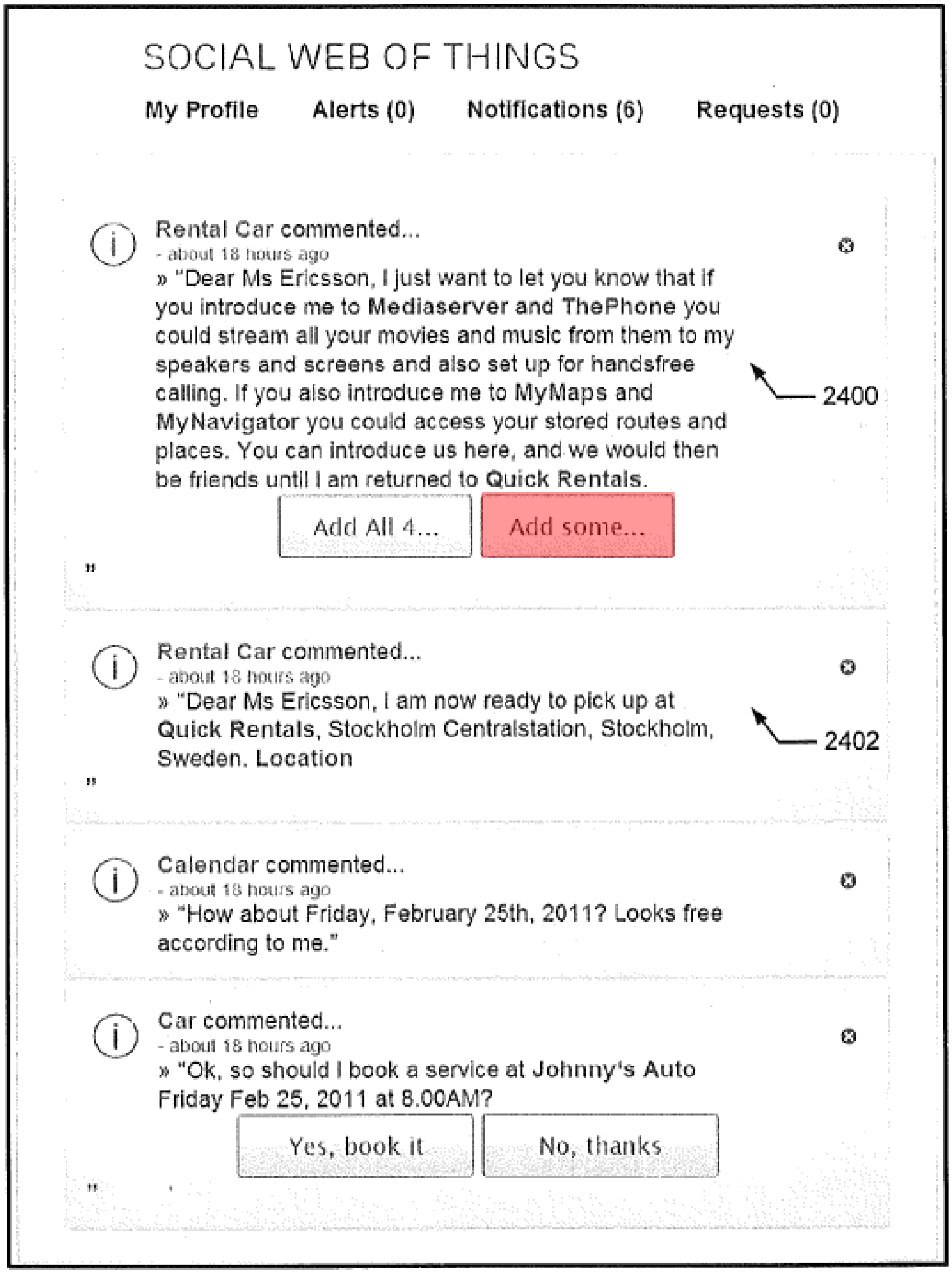


FIG. 24

Ermis, Fig. 24 (annotated)

242. If the user selects the "Add some…" button, "[t]he interaction execution engine 1312 adds **the selected** friendship relationships and associated data flows to the interaction flow database 1308." Ermis, [0128].

243. In addition to allowing the user to select which data flows will be created, Ermis also allows the user to further decide which processes within one data flow to implement and which processes not to implement, i.e., a user *intervention includes selecting some processes of the multiple processes*, such that the processes of a newly created data flow, such as the data flows described with reference to Figure 24 above, may be selected by the user.

244. For example, Figure 17 illustrates a scenario in which two processes of a data flow are recommended to the user: (1) turning off the lights while the user is away from the house, and then, (2) viewing the surveillance camera footage.

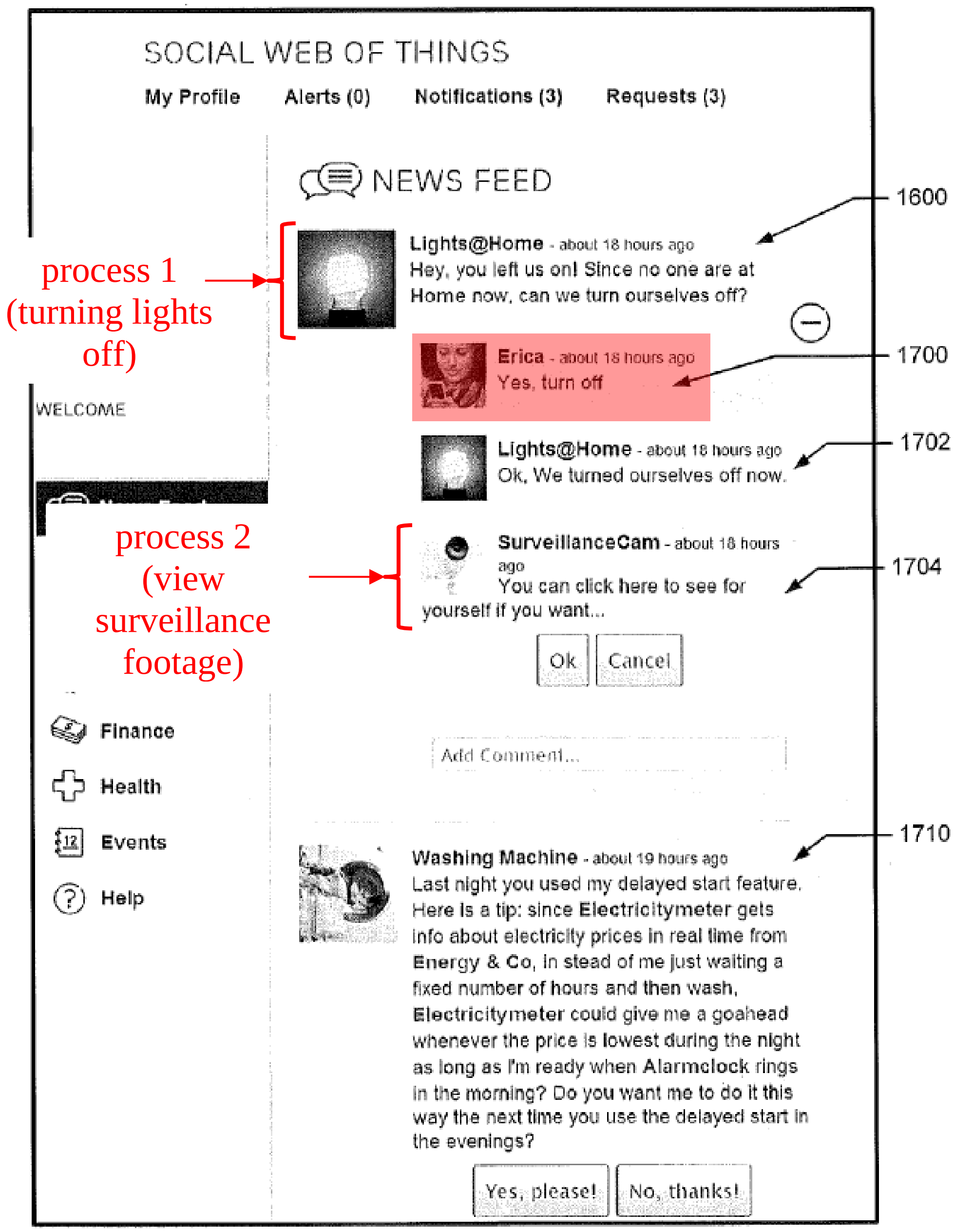


FIG. 17

Ermis, Fig. 17 (annotated)

245. As shown in Figure 17, the user has the option of approving or not approving each of the two processes in this data flow. In this example, the user approved process 1 of the recommended data flow (turning the lights off) at step 1700, highlighted above, but has not yet approved process 2 (viewing surveillance footage). The user approving process 1 renders obvious a user *intervention* that *includes selecting some processes of the multiple processes*.

t) Claim 20

[20.0] ***The system of claim 1, wherein: the intervention includes approving the multiple processes.***

246. As described at limitation [1.3], four new relationships and included data flows are recommended to the user from a rental car.

247. Specifically, the rental car in Figure 24 requests that the user allow new relationships to be generated between the rental car and "a media server device and a phone terminal" as well as "a map application, and a navigator application." Ermis, [0126]; *see* Ermis, Fig. 24.

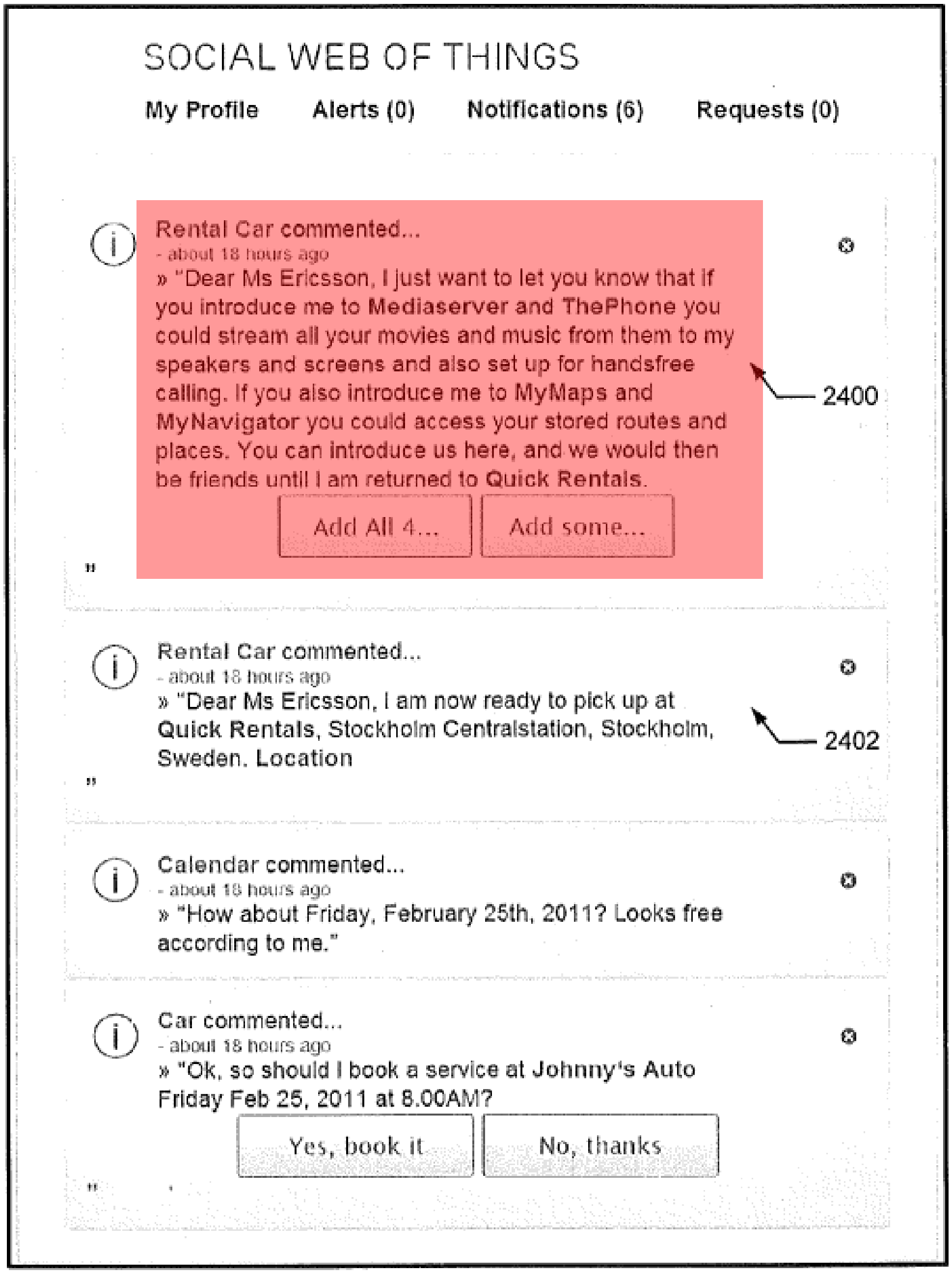


FIG. 24

Ermis, Fig. 24 (annotated)

248. These new relationships are added based on the user's selection of the "Add all 4…" button. Ermis, [0128]. In addition, these new data flows include multiple processes, such as "stream[ing]…movies and music" and "access[ing] stored routes and places." Ermis, Fig. 24.

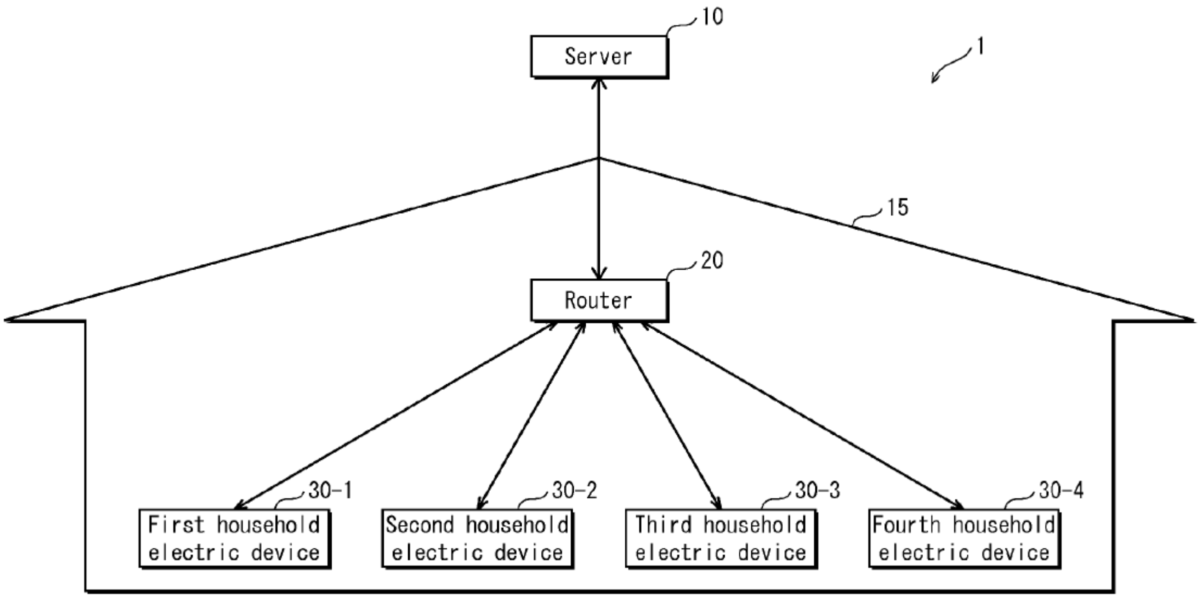
249. Thus, Ermis renders obvious *the intervention includes approving the multiple processes* (user selecting the "Add All 4…" button to implement recommended data flows).

B. Ground 2: Claims 1-20 is obvious under 35 U.S.C. § 103 over Ermis in view of Nakano.

1. Summary of Nakano

250. Like the '518 patent and Ermis, Nakano describes principles of home automation, titled "Coordination Processing Execution Method and Coordination Processing Execution System." Nakano, (54). Nakano describes multiple machines, referred to as "household electric devices" in communication with "a server 10," shown below. Nakano, 3:63-67.

FIG. 1



Nakano, Fig. 1

251. Figure 3 illustrates profile information stored for each of the household electric devices. For example, each device has a "device ID," a "product code," and the operating software running on the device indicated by "a version number." Nakano, 6:37-43. This information is stored in a "Device Information Table," shown below.

FIG. 3

	Device ID (Device name)	Product code (Product type)	Software version	Operation ID (Operation name)	Control command	Registration state
311	First household electric device	P-06D (Smartphone)	1.12	A1 (Power ON)	D201xxxx	1 (Registered)
				A2 (Alarm)	D205xxxx	
				:	:	
312	Second household electric device	CS-X252C (Air conditioner)	1.01	B1 (Power ON)	9203xxxx	1 (Registered)
				B2 (High-power mode operation)	9252xxxx	
				B3 (Power-saving mode operation)	9280xxxx	
				:	:	
313	Third household electric device	SR-SX102 (Rice cooker)	1.03	C1 (Power ON)	800Fxxxx	1 (Registered)
				C2 (Rice cooking start)	8020xxxx	
				:	:	
314	Fourth household electric device	NA-VX8200L (Washing machine)	1.10	D1 (Power ON)	9101xxxx	1 (Registered)
				D2 (Washing start)	9104xxxx	
				:	:	
315	Fifth household electric device	DMC-TZ40 (Digital camera)	1.00	E1 (Power ON)	A209xxxx	0 (Not registered)
				E2 (Image capture)	A285xxxx	
				:	:	
316	Sixth household electric device	HH-LC712A (Ceiling light)	1.01	F1 (Light up)	E942xxxx	0 (Not registered)
				F2 (Cinema mode)	EA94xxxx	
				F3 (Power-saving mode)	E548xxxx	
				F4 (ON timer)	E968xxxx	
317	Seventh household electric device	BG-849 (Air conditioner)	2.10	G1 (Power ON)	3924xxxx	0 (Not registered)
				G2 (ON timer)	8643xxxx	
				:	:	
318	Eighth household electric device	COF-63B (Coffee maker)	1.09	H1 (Power ON)	5829xxxx	0 (Not registered)
				H2 (Brewing start)	6487xxxx	
				H3 (Timer brewing)	935Exxxx	
				:	:	
319	Ninth household electric device	TH-P65VT60 (Television)	1.01	J1 (Power ON)	6385xxxx	0 (Not registered)
				:	:	
	:	:	:	:	:	:

Nakano, Fig. 3

252. Nakano's "product code is an identifier given to the household electric device by the manufacturer of the household electric device." Nakano, 5:63-65. "The software version is a number indicating a version of software (can also be firmware, middleware, or the like) for controlling the household electric device, and is given to the software by a software developer or the like." Nakano, 6:4-10. "Software…for controlling [a] household electric device" includes operating system software. Nakano, 6:4-10.

253. Nakano's system also uses information identifying various devices to form "cooperative processes" included in a "cooperative process composition table," shown in Figure 4, below. Nakano, 8:4-11.

FIG. 4

	First cooperative process		Second cooperative process		Third cooperative process		Fourth cooperative process		Fifth cooperative process		Sixth cooperative process		Seventh cooperative process		Eighth cooperative process		…
	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	…
First operation	1	A2	1	A2	1	A2	1	A2	1	A2	2	B1	6	F1	6	F1	…
Second operation	2	B3	2	B2	2	B3	7	G2	7	G2	6	F1	2	B3	2	B1	…
Third operation	3	C2	3	C2	3	C2	3	C2	3	C2					9	J1	…
Fourth operation			8	H2	4	D2	8	H2	4	D2							…

411 412

Nakano, Fig. 4

254. "[A] cooperative process refers to a group of operations linked together that are each executed by a different one of a plurality of household electric devices." Nakano, 3:46-49. Each cooperative process lists the "device ID of a household electric device executing the operation and operation ID identifying the operation." Nakano, 8:17-20.

2. Reasons to Combine Ermis and Nakano

255. A POSITA would have been motivated to combine the teachings of Ermis with Nakano. Before the priority date of the '518 patent, it would have been obvious, beneficial, and predictable to implement Nakano's teachings of using a cooperative process composition table to store information related to a task performed by multiple processes, as well as using product codes and software version numbers to indicate operating systems of devices. A POSITA would have recognized these techniques as well-known techniques to ensure proper operation and coordination between devices to perform various task requests.

a) Ermis and Nakano are Analogous art

256. One of ordinary skill in the art when considering the teachings of Ermis would have naturally considered the teachings of Nakano, because Nakano is analogous prior art pertaining to the same field of endeavor as both the '518 patent and Ermis, namely, "managing **relations** between machines…**to execute a command**." '518 patent, Abstract; Ermis, [0044] ("establish[ing] **relationships** using social mapping principles and…exercis[ing] those relationships in **performance of various task requests**."). Similarly, Nakano describes managing a "**cooperative process**" in which a "cooperative process execution system 1…control[s] each of one or more household electric devices **to execute an operation**." Nakano, 3:50-55. Accordingly, both Ermis and Nakano describe managing relations between multiple devices and controlling the multiple devices to execute processes of a task. *See* Ermis, [0101]-[0102] ("The interaction execution engine 1312 can…**control operations performed by the resource nodes** on the data."); Nakano, 3:50-55 ("**controlling each of one or more household electric devices to execute an operation**.").

257. Both Ermis's system and Nakano's system use a server to analyze commands and coordinate processes performed by devices. *See* Ermis, [0040] ("system 12…can, for example, be implemented, at least in part, as **server-based software**."); Nakano, 4:10-14 ("The server 10 executes cooperative processes by controlling household electric devices").

b) Motivation to Combine Ermis with Nakano

258. A POSITA would have been motivated to implement Nakano's teachings regarding storing data for cooperative processes in cooperative process composition tables with Ermis. A POSITA would have recognized that data related to multiple operations performed by multiple devices to execute a task would advantageously be stored as part of a cooperative process composition table, as described by Nakano.

259. Storing this data as a structured data table would advantageously allow for efficient querying using SQL or similar database languages, such as XML explicitly referenced by Ermis. *See* Ermis, [0070] (describing using an "XQUERY" to query databases in Ermis's system). This format would ensure that queries could easily retrieve all operations for a specific device, all devices involved in a specific task, or the sequence of operations to perform a specific task. It was known that a standardized data format ensures that data "may be easily retrieved and analyzed." Moore, 1:39-42; *see also* Murata, [0018] ("non-standardized data" is "not easily retrieved"). This format would additionally enforce constraints on the data stored, reducing errors, and ensuring that only valid operations are assigned to devices. *See* Kashima, [0127] ("a standard data format…reduce[s] input cost and input error"). In addition, implementing Nakano's cooperative process composition table would increase scalability and flexibility which would be particularly useful in Ermis's system which describes sharing popular data flows with different users. *See* Vyatkin, [0223] ("standardised data structures" for "interoperability" and "flexib[ility]"). Nakano's cooperative process composition table would also optimize performance of indexing and caching by making lookups more efficient. *See* Martin, [0047] ("standard data crawlers and indexing techniques for efficiency"). The consistent structure of Nakano's cooperative process composition table would also increase interoperability and integration by allowing data to be exported into common formats for implementation or modification in other systems. *See* Vyatkin, [0223]; Gilder, [0060] ("standards for 'data interoperability'" and "'exporting' data").

260. Accordingly, the Ermis-Nakano combination would have been the combination of prior art elements (performing multiple processes to execute a requested task, as described by Ermis and Nakano) according to known methods (storing the multiple processes performed by different devices to perform the requested task as a cooperative process including device IDs and operation IDs) to yield the predictable result of coordinating multiple networked devices to execute a task.

261. Regarding the use of product codes and version numbers, a POSITA would have been motivated to incorporate Nakano's teachings regarding identifying an operating system of a device using a product code and software version number into Ermis's system because Nakano's technique was a well-known method of indicating operating system characteristics, and quickly and easily identifying whether the device operating system is up to date and capable of performing tasks as expected.

262. Nakano explains that "when a software is updated from an older version to a newer version, a number greater than that given to the older version is given to the newer, updated version of the software." Nakano, 6:7-10. Because a device's operating system is assigned a software version number and iterated in this manner, a user, manufacturer, or service provider may easily determine whether the operating system of the device is up to date and is, therefore, in a condition capable to install various applications or additional software, perform particular tasks unique to a particular operating system version, and maintain security. *See* Ziegler, Abstract, [0006].

263. Implementing Nakano's teachings regarding an indication of an operating system by a product code and version number also ensures quick and simple identification of interoperability of a device with other devices. For example, applications and software typically list compatible operating systems and the designation of an operating system by its version number would ensure compatibility of the device with a particular application or software. *See* Mori, Abstract.

264. As explained above, Ermis explains that each device, or networked object, in its system includes an operating system. *See* Ermis, [0076] (describing "an operating system" in the "storage/memory 704" of a device). In addition, Ermis explains that each device includes a "system entity profile[] 20," which contains a "technical specification[]" of the device, including "metadata" of any suitable type. Ermis, [0041]. As explained above, a POSITA would have recognized that because each device includes an operating system and metadata describing characteristics of the device, this metadata would include an operating system parameter. *See supra* § VIII.A.2.o. However, to the extent Ermis does not explicitly describe a specific parameter indicative of the device operating system, a POSITA would have been motivated to implement Nakano's techniques indicating a device's operating system with a product code and version number that align with Ermis's system.

265. Both Ermis and Nakano list interoperability of devices as an objective. *See* Ermis, [0043] (describing "using transcoding information" for outputs of devices "**to provide compatible communications**" with other devices), [0115] (describing the need to ensure that "brand-dependent semantic interaction language[s]" are interoperable); Nakano, 3:58-62, 15:1-5 (ensuring interoperability between devices by ensuring that the devices "communicat[e] with the server" and are "mediated by the router 20")

266. A POSITA would have been motivated to combine the teachings of Ermis with Nakano to ensure that any user, such as an installer or maintenance worker for the system described in Ermis, may easily determine whether the operating system of a particular device is up to date and capable of performing its expected functions. For example, different operating system versions provide for installation of various applications or software or different versions of applications or software. *See* Mori, Abstract. A POSITA would have therefore been motivated to incorporate Nakano's use of product codes and software version numbers to improve Ermis in the same way.

267. Accordingly, the Ermis-Nakano combination would have been the combination of prior art elements (Ermis's networked objects including system entity profiles containing information regarding characteristics of the device) according to known methods (Ermis's method of including product code and software version information) to yield the predictable result of indicating an operating system of the device with a product code and version number.

c) Reasonable Expectation of Success

268. Not only would utilizing Nakano's teachings have been advantageous, a POSITA also would have found the combination of Ermis and Nakano to be predictable and would have expected to be successful in making the combination due to the compatibility and similarity of their teachings. Both Ermis and Nakano describe controlling interconnected machines to perform commands. Ermis, [0044] ("establish[ing] relationships using social mapping principles and…exercis[ing] those relationships in performance of various task requests."); Nakano, 3:50-55 (A "cooperative process execution system 1…control[s] each of one or more household electric devices to execute an operation.").

269. Both Ermis's system and Nakano's system use standard, commercially available devices. *See, e.g.*, Ermis, [0045]-[0046] (a TV and video recorder), [0111] (a washing machine, electricity controller, and alarm clock); Nakano, 3:63-4:9 (a smartphone, air conditioner, rice cooker, and a washing machine). Both use a server to control these devices. *See* Ermis, [0040] ("system 12…can, for example, be implemented, at least in part, as **server-based software**."); Nakano, 4:10-14 ("The server 10 executes cooperative processes by controlling household electric devices"). Based on these teachings, a POSITA would have been familiar with the aspects of Ermis's system and recognized its compatibility with Nakano's.

270. Accordingly, a POSITA would have been motivated and found it obvious to combine Ermis with Nakano in the manner proposed.

3. Detailed Analysis of the Claims

a) Claim 1

271. Limitations [1.0]-[1.2.2] and [1.2.4]-[1.5] are rendered obvious for the reasons described in Ground 1 above. *See supra* §§ VIII.A.2.a.

[1.2.3] ***including a task processing schedule parameter which defines a sequence of performing multiple processes,***

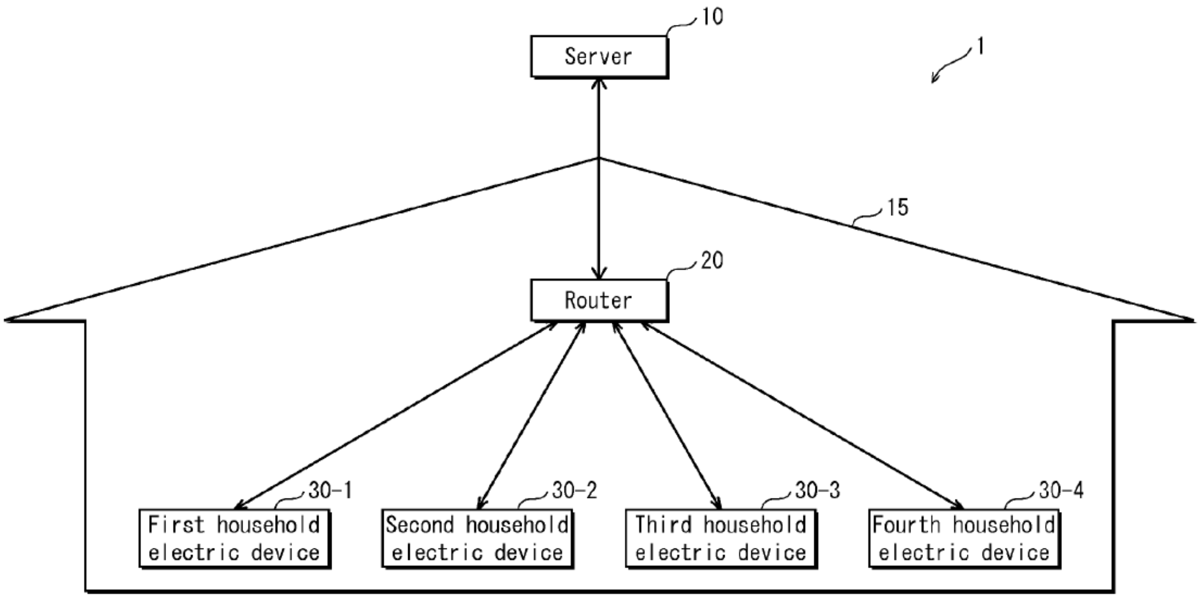
272. The combination of Ermis and Nakano also renders this limitation obvious.

273. As explained above, *see supra* § VIII.A.2.a [1.2.3], Ermis describes data flows which define a series of steps to be performed by multiple devices of a relationship to execute a requested task. For example, a washing machine, electricity controller, and alarm clock perform various tasks to determine a start time for a washing machine and to control the start of the washing machine. *See* Ermis, [0111]. Ermis provides several additional examples of data flows including a sequence of processes performed by multiple devices. *See, e.g.*, [0045]-[0046], [0053]-[0054] (steps to request that a TV program be performed, passing a command to a video recorder, recording the program, and sending confirmation to the user), [0048] (monitoring power consumption of "a light, 34, a television 36, and a radio 38" and controlling them accordingly), [0124] (identifying that a user's car is due for maintenance, notifying the user, and scheduling a service appointment).

274. Nakano further illustrates that these multiple processes to be performed would be included in a parameter stored in the system, e.g., *a task processing schedule parameter*.

275. Nakano describes methods for coordinating multiple "household electric devices that are registered to a server to execute **cooperative processes**." Nakano, Abstract. These electric devices 30 are shown in Figure 1 connected, via a router 20 to a server 10, below.

FIG. 1



Nakano, Fig. 1

276. "[A] cooperative process refers to **a group of operations** linked together that are each **executed by a different one of a plurality of household electric devices**." Nakano, 3:46-49. Cooperative processes are illustrated in Figure 4, below.

FIG. 4

task processing schedule parameter[s] …

multiple processes

	First cooperative process		Second cooperative process		Third cooperative process		Fourth cooperative process		Fifth cooperative process		Sixth cooperative process		Seventh cooperative process		Eighth cooperative process		…
	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	Device ID	Operation ID	…
First operation	1	A2	1	A2	1	A2	1	A2	1	A2	2	B1	6	F1	6	F1	…
Second operation	2	B3	2	B2	2	B3	7	G2	7	G2	6	F1	2	B3	2	B1	…
Third operation	3	C2	3	C2	3	C2	3	C2	3	C2					9	J1	…
Fourth operation			8	H2	4	D2	8	H2	4	D2							…

411 412

Nakano, Fig. 4

277. As illustrated above, each cooperative process lists the "device ID of a household electric device executing the operation and operation ID identifying the operation." Nakano, 8:17-20.

278. In Nakano, cooperative processes define a *sequence* of *multiple processes*. With reference to Figure 4, the "first operation" in each process "triggers the cooperative process" and the rest of the operations are referred to as "remaining processing." Nakano, 8:4-16. With reference to Figure 9, below, each operation is assigned to a "[s]pecification of time." Ermis, Fig. 9.

FIG. 9

sequence of performing multiple processes

Execution control number	Specification of time (Parameter)	Device ID	Operation ID
1	7:00	2	B2
	7:00	3	C2
2	6:55	2	B2
	7:05	3	C2
	7:10	4	D2
⋮	⋮	⋮	⋮

Nakano, Fig. 9

279. Because the time parameters of each operation are scheduled to occur one after another (e.g., 6:55, 7:05, and 7:10), a cooperative process in Nakano defines a *sequence* of *multiple processes*.

280. It would have been obvious to implement a "data flow," described by Ermis, as a "cooperative process," described by Nakano. As explained above, storing information regarding tasks to be performed by multiple devices as processes in a table including device IDs and operation IDs identifying the devices and processes to be performed in the order in which the processes are to be performed offers several advantages, including more efficient querying, ensuring data consistency so only valid device-operation associations exist, greater scalability and flexibility, providing streamlined lookups using device IDs and operation IDs, and greater interoperability. *See supra* § VIII.B.2 (Reasons to Combine Ermis and Nakano).

281. Thus, Nakano describes a cooperative process, which renders obvious *a task processing schedule parameter* because it includes a sequence of operations performed by various electric devices, which renders obvious *defin[ing] a sequence of performing multiple processes*.

b) Claims 2-14

282. Claims 2-14 are rendered obvious for the reasons described in Ground 1 above. *See* §§VII.A.2.b-n.

c) Claim 15

[15.0] ***The system of claim 8, wherein: at least one machine profile of the multiple machine profiles includes an operating system parameter, and the operating system parameter corresponds to a type of an operating system which is used by a machine corresponding to the at least one machine profile.***

283. As explained in Ground 1, above, Ermis renders obvious claim 15. Alternatively, the combination of Ermis and Nakano also renders obvious claim 15 as shown below.

284. Nakano describes data associated with various "household electric device[s].” Nakano, Fig. 3. This data includes a “[p]roduct code” and a “[s]oftware version,” highlighted below.

FIG. 3

	Device ID (Device name)	Product code (Product type)	Software version	Operation ID (Operation name)	Control command	Regist-ration state
311	First household electric device	P-06D (Smartphone)	1.12	A1 (Power ON)	D201xxxx	1 (Registered)
				A2 (Alarm)	D205xxxx	
				⋮	⋮	
312	Second household electric device	CS-X252C (Air conditioner)	1.01	B1 (Power ON)	9203xxxx	1 (Registered)
				B2 (High-power mode operation)	9252xxxx	
				B3 (Power-saving mode operation)	9280xxxx	
				⋮	⋮	
313	Third household electric device	SR-SX102 (Rice cooker)	1.03	C1 (Power ON)	800Fxxxx	1 (Registered)
				C2 (Rice cooking start)	8020xxxx	
				⋮	⋮	
314	Fourth household electric device	NA-VX8200L (Washing machine)	1.10	D1 (Power ON)	9101xxxx	1 (Registered)
				D2 (Washing start)	9104xxxx	
				⋮	⋮	
315	Fifth household electric device	DMC-TZ40 (Digital camera)	1.00	E1 (Power ON)	A209xxxx	0 (Not registered)
				E2 (Image capture)	A285xxxx	
				⋮	⋮	
316	Sixth household electric device	HH-LC712A (Ceiling light)	1.01	F1 (Light up)	E942xxxx	0 (Not registered)
				F2 (Cinema mode)	EA94xxxx	
				F3 (Power-saving mode)	E548xxxx	
				F4 (ON timer)	E968xxxx	
317	Seventh household electric device	BG-849 (Air conditioner)	2.10	G1 (Power ON)	3924xxxx	0 (Not registered)
				G2 (ON timer)	8643xxxx	
				⋮	⋮	
318	Eighth household electric device	COF-63B (Coffee maker)	1.09	H1 (Power ON)	5829xxxx	0 (Not registered)
				H2 (Brewing start)	6487xxxx	
				H3 (Timer brewing)	935Exxxx	
				⋮	⋮	
319	Ninth household electric device	TH-P65VT60 (Television)	1.01	J1 (Power ON)	6385xxxx	0 (Not registered)
				⋮	⋮	
	⋮	⋮	⋮	⋮	⋮	⋮

Nakano, Fig. 3 (annotated)

285. Nakano explains that "the device information table" of Figure 3 includes "(ii) a product code, [and] (iii) a software version." Nakano, 5:44-58. "The product code is an identifier given to the household electric device by the manufacturer of the household electric device." Nakano, 5:63-65. "The software version is a number indicating a version of software (can also be firmware, middleware, or the like) for controlling the household electric device, and is given to the software by a software developer or the like." Nakano, 6:4-10.

286. Nakano explains that this data is specifically included in the product's "device operation table." Nakano, 9:29-31. That the product code and software version listed in the operation table refers to an operating system parameter is confirmed by the types of devices described by Nakano, such as "a smartphone" which commonly includes an operating system. Nakano, 10:3-8.

287. While Ermis explains that its networked objects execute an "operating system," to the extent Ermis does not explicitly disclose a parameter indicative of these various operating systems of devices, it would have been obvious in view of the teachings of Nakano. Ermis, [0076]. Like the product code and software version parameters described by Nakano, an operating system of an electronic device was conventionally indicated by a product code and software version number. *See*, *e.g.*, Straub, 3:11-17 ("[A]pplication program[s] will **recognize the <u>operating system</u> by the <u>version number</u>**."); MacInnis, 4:42-47, Fig. 2 (typical product data includes "the manufacturer's model number of the terminal, [and] **the version number of the currently executing operating system**."); Hunt, [0009] (identifying "**operating systems** by using so-called **version numbers**."). In the combination, this product code and software version number data would be stored in the system entity profile of the device. *See* Ermis, [0041] (describing "metadata associated with the respective object 10" within its "system entity profile[] 20").

288. Thus, the combination of Ermis and Nakano renders obvious at least one machine profile of the multiple machine profiles (system entity profile 20, described by Ermis) includes an operating system parameter, and the operating system parameter corresponds to a type of an operating system which is used by a machine corresponding to the at least one machine profile (indication of product code and software version number, described by Nakano).

d) Claims 16-20

289. Claims 16-20 are rendered obvious for the reasons described in Ground 1 above. *See* §§VIII.A.2.p-t.

IX. DECLARATION

290. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code.

Dated: February 26, 2025

Henry H. Houh, Ph.D.