

**UNITED STATES PATENT AND TRADEMARK OFFICE**

---

**BEFORE THE PATENT TRIAL AND APPEAL BOARD**

---

CISCO SYSTEMS, INC.,  
Petitioner,

---

IPR2026-00211  
Patent No. 12,279,116

---

**PETITION FOR *INTER PARTES* REVIEW  
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

**TABLE OF CONTENTS**

Table of Contents .....	2
Petitioner’s Exhibit List .....	4
I. Introduction.....	7
II. Grounds for Standing.....	9
III. Note.....	10
IV. Summary of the ’116 patent .....	10
V. Effective priority date .....	10
VI. Prosecution history .....	11
VII. Level of ordinary skill in the art.....	11
VIII. Background.....	11
IX. Claim Construction.....	12
X. Relief requested and reasons therefore.....	12
XI. Identification of how the claims are unpatentable.....	12
A. Challenged Claims .....	12
XII. Statutory grounds.....	13
A. Statement of Material Facts.....	14
B. Ground 1 .....	17
1. Abuan .....	17
2. Independent Claim 1 .....	19
3. Independent Claim 55 .....	64
C. Ground 2.....	66
1. Eisenberg.....	66
2. Reasons to Combine .....	68

3.	Dependent claim 15 .....	69
D.	Ground 3.....	70
1.	Beilis .....	70
2.	Reasons to Combine .....	72
3.	Dependent claim 21 .....	73
E.	Ground 4.....	74
1.	Guzman .....	74
2.	Reasons to Combine .....	75
3.	Dependent Claims 9-10.....	78
4.	Independent Claim 28 and Dependent Claims 29-41, 43-47, and 49-54.....	83
F.	Ground 5.....	86
1.	Dependent Claim 42 .....	86
G.	Ground 6.....	86
1.	Dependent Claim 48 .....	86
XIII.	Conclusion .....	86
XIV.	Mandatory notices .....	88
A.	Real party-in-interest.....	88
B.	Related matters .....	88
C.	Lead and back-up counsel and service information.....	88
XV.	Claims Appendix .....	90
	Certificate of Word Count.....	99
	Certificate of Service .....	100

**PETITIONER’S EXHIBIT LIST**

Ex.1001	U.S. 12,279,116
Ex.1002	Prosecution History of U.S. 12,279,116
Ex.1003	Declaration of Bruce McNair under 37 C.F.R. § 1.68
Ex.1004	<i>Curriculum Vitae</i> of Bruce McNair
Ex.1005	U.S. 2011/0249077 to Abuan et al.
Ex.1006	U.S. 8,396,490 to Platt et al.
Ex.1007	U.S. 2012/0092438 to Guzman Suarez et al.
Ex.1008	U.S. 5,689,641 to Ludwig et al.
Ex.1009	D. E. Knuth, Selected Papers on Computer Science (Cambridge Univ. Press), 1996
Ex.1010	B. W. Boehm, Software Engineering Economics (Prentice-Hall, Inc.), 1981
Ex.1011	U.S. 5,339,431
Ex.1012	U.S. 5,615,400
Ex.1013	U.S. 5,946,486
Ex.1014	U.S. 6,243,764
Ex.1015	U.S. 5,802,367
Ex.1016	U.S. 5,218,697
Ex.1017	U.S. 5,498,003
Ex.1018	U.S. 6,163,858
Ex.1019	U.S. 2010/0066807 to Eisenberg et al.
Ex.1020	Plaintiff’s Disclosure of Asserted Claims and Infringement Contentions, <i>Damaka, Inc. v. Cisco Systems, Inc.</i> , No. 2:25-cv-00593 (E.D. Tex. Nov. 13, 2025).
Ex.1021	U.S. 2008/0178264
Ex.1022	U.S. 2005/0235044

Ex.1023	U.S. 2006/0037064
Ex.1024	U.S. 2008/0201438
Ex.1025	U.S. 2012/0170722
Ex.1026	U.S. 2005/0265322
Ex.1027	U.S. 6,314,402
Ex.1028	U.S. 2005/0060183
Ex.1029	U.S. 2009/0193394
Ex.1030	U.S. 2008/0022327
Ex.1031	U.S. 2002/0199019
Ex.1032	U.S. 6,101,607
Ex.1033	U.S. 6,311,283
Ex.1034	U.S. 6,295,048
Ex.1035	U.S. 2009/0310023
Ex.1036	U.S. 2008/0127183
Ex.1037	U.S. 7,574,202
Ex.1038	U.S. 7,461,144
Ex.1039	U.S. 2007/0185998
Ex.1040	U.S. 6,820,235
Ex.1041	RFC 3264
Ex.1042	U.S. 9,058,655
Ex.1043	Apple Newsroom, “Apple Announces iPhone 2.0 Software Beta,” 2008 (retrieved from <a href="https://www.apple.com/newsroom/2008/03/06Apple-Announces-iPhone-2-0-Software-Beta/">https://www.apple.com/newsroom/2008/03/06Apple-Announces-iPhone-2-0-Software-Beta/</a> )
Ex.1044	The Big Blog, “Skype now available for Android phones,” 2010 (retrieved from <a href="https://web.archive.org/web/20101006000452/http://blogs.skype.com/en/2010/10/android.html">https://web.archive.org/web/20101006000452/http://blogs.skype.com/en/2010/10/android.html</a> )
Ex.1045	U.S. 2012/0019610
Ex.1046	U.S. 2013/0290982 to Beilis et al.

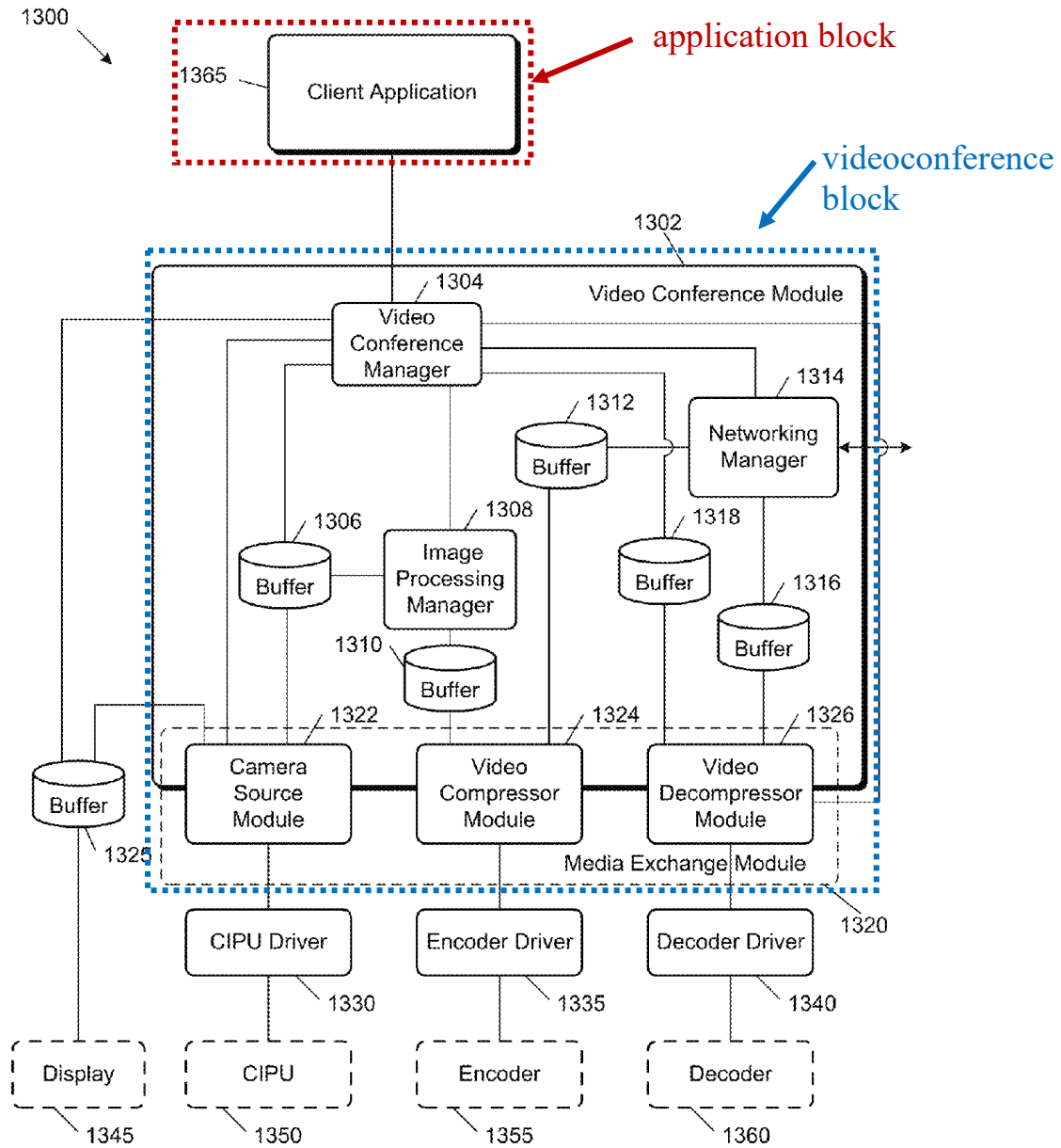
Ex.1047	Chart comparing claims 1-27 with claims 28-54 in redline
Ex.1048	iOS Reference Library, "Core Media Framework Reference," 2010 (retrieved from <a href="https://web.archive.org/web/20100906024157/http://developer.apple.com/iphone/library/documentation/CoreMedia/Reference/CoreMediaFramework/index.html#//apple_ref/doc/uid/TP40009756">https://web.archive.org/web/20100906024157/http://developer.apple.com/iphone/library/documentation/CoreMedia/Reference/CoreMediaFramework/index.html#//apple_ref/doc/uid/TP40009756</a> )

## **I. Introduction**

Cisco Systems, Inc. (“Petitioner”) respectfully requests that the Board review and cancel as unpatentable under 35 U.S.C. §103 claims 1-55 (hereinafter, the “Challenged Claims”) of U.S. 12,279,116 (the “’116 patent,” Ex.1001).

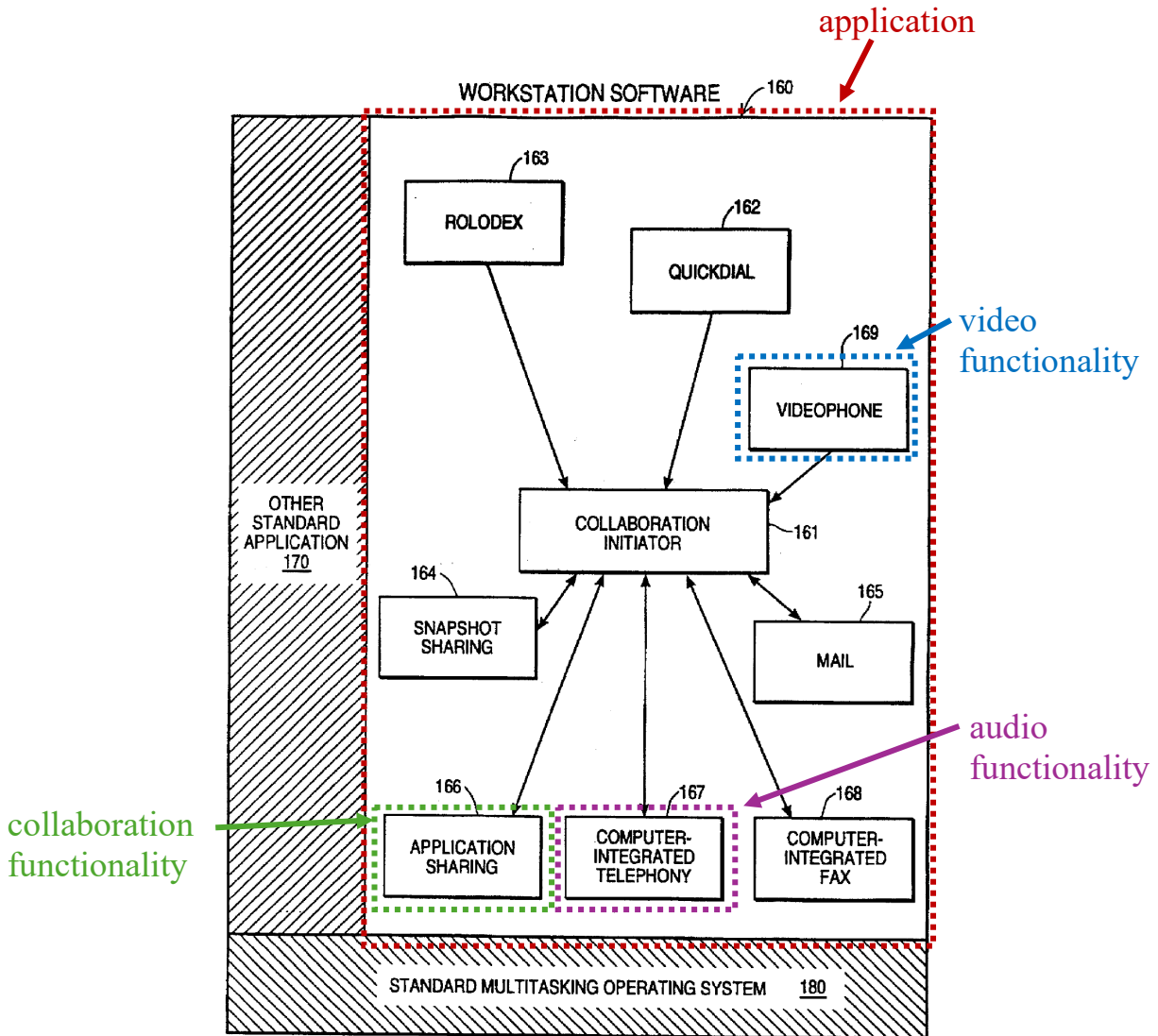
The ’116 patent is directed to “improving functionality in software applications.” Ex.1001, Abstract. The patent acknowledges that one known way to improve software was to rework an existing application to add “desired functionality” by incorporating additional executable instructions that “provide such additional functionality.” Ex.1001, 3:41-48. The claimed innovation does the same thing, except that the added instructions are characterized as being a “function block” that is added to a “superblock.” Ex.1001, 4:29-38. The superblock “may be an application,” while the function block provides capabilities for “audio, video, collaboration..., and/or other functions.” Ex.1001, 2:62-64, 4:38-43.

Organizing computer software into blocks of code was not new. The prior art figure below (dating from 2010) shows an application block designed to use video functions—specifically, videoconference functionality—provided by a separate block. Ex.1005, FIG. 13.



**Ex.1005, Fig. 13 (annotated).**

Nor was the modular design of software applications in any way “new” even in 2010. The figure below, dating from 1993, shows the modular design of an application with discrete blocks providing audio, video, and collaboration functionality. Ex.1008, FIG. 20.



Ex.1008, Fig. 20 (annotated).

Because the Challenged Claims merely describe concepts and techniques that were already known, Petitioner asks the Board to institute trial and find the claims unpatentable.

## II. Grounds for Standing

Petitioner certifies that the '116 patent is eligible for IPR. Petitioner is not

barred or estopped from requesting this IPR.

### **III. Note**

Petitioner cites to exhibits' original page numbers. **Emphasis** in quoted material has been added. Claim terms are presented in *italics*.

### **IV. Summary of the '116 patent**

The '116 patent purports to describe an improved “manner in which functionality is accessed in certain environments, such as mobile device environments.” Ex.1001, 1:58-60. The improvement consists of a modular approach to software using a “superblock application” and a “function block” that provides “one or more functions (e.g., capabilities)” to the superblock. Ex.1001, 4:35-38. The function block comprises one or more blocks of executable instructions. Ex.1001, 5:1-4. The superblock may access the function block instructions by calling an application programming interface (API). Ex.1001, 6:27-35.

### **V. Effective priority date**

The '116 patent claims the benefit of a provisional application filed July 16, 2013. Petitioner's cited and relied-upon prior art predates July 2013. Accordingly, Petitioner has not undertaken a priority date analysis. Petitioner does not waive any right or opportunity it may have to dispute the priority date of the '116 patent in this or another forum where the issue is relevant.

## **VI. Prosecution history**

The '116 patent application was filed on Feb. 5, 2024 with only one claim. Ex.1002, 566, 598. The examiner rejected the claim for statutory double-patenting. Ex.1003, 274-280. In response, the applicant replaced claim 1 via amendment and added new claims 2-55. Ex.1002, 261-268. The Applicant filed a terminal disclaimer after an Examiner-initiated Examiner interview. Ex.1002, 22, 253-254. The examiner then allowed the claims, stating that the elements corresponding to limitations [1.1]-[1.3] of claim 1 as labeled below (and the other independent claims) were not clearly taught or fairly suggested by the prior art of record. Ex.1002, 19-20.

## **VII. Level of ordinary skill in the art**

A Person of Ordinary Skill in The Art (“POSITA”) in July 2013 would have had a working knowledge of the software programming art that is pertinent to the '116 Patent, including the use of software modules and application programming interfaces (APIs). A POSITA would have had a bachelor’s degree in computer science, computer engineering, or an equivalent, and three years of professional experience relating to computer programming. Lack of professional experience can be remedied by additional education, and vice versa. Ex.1003, ¶19.

## **VIII. Background**

Mr. McNair describes background information that would have been known

to a POSITA. Ex.1003, ¶¶25-38.

**IX. Claim Construction**

For the purposes of this proceeding, Petitioner proposes that no terms require express construction. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). Ex.1003, ¶¶39-40.

**X. Relief requested and reasons therefore**

Petitioner asks that the Board institute a trial for *inter partes* review and cancel the Challenged Claims in view of the analysis below.

**XI. Identification of how the claims are unpatentable**

**A. Challenged Claims**

Petitioner challenges claims 1-55.

Patent Owner has asserted claims 1, 9-13, 16-24, 26-29, 37-40, 43-51, and 53-54 in the co-pending litigation. *See* Ex.1020, 2. A finding that the Challenged Claims 1-55 are unpatentable in this proceeding will resolve the parties' dispute in the co-pending litigation and obviate any need for a trial regarding the '116 patent, substantially reducing the time and expense of that litigation for all parties.

**XII. Statutory grounds**

<b>Grounds</b>	<b>Claims</b>	<b>Basis</b>
#1	1-8, 11-20, 22-27, 55	35 U.S.C. § 103 over Abuan
#2	15	35 U.S.C. § 103 over Abuan and Eisenberg
#3	21	35 U.S.C. § 103 over Abuan and Beilis
#4	9-10, 28-41, 43-47, 49-54	35 U.S.C. § 103 over Abuan and Guzman
#5	42	35 U.S.C. § 103 over Abuan, Guzman, and Eisenberg
#6	48	35 U.S.C. § 103 over Abuan, Guzman, and Beilis

U.S. 2011/0249077 to Abuan et al. (Ex.1005, “**Abuan**”) published on October 13, 2011.

U.S. 2012/0092438 to Guzman Suarez et al. (Ex.1007, “**Guzman**”) was filed on October 18, 2010, and published on April 19, 2012.

U.S. 2010/0066807 to Eisenberg (Ex.1019, “**Eisenberg**”) published on March 18, 2010.

U.S. 2013/0290982 to Beilis et al. (Ex.1046, “**Beilis**”) was filed on April 30, 2012, and published on October 31, 2013.

Abuan, Guzman, Eisenberg, and Beilis are prior art under 35 U.S.C. § 102(a)(2).

Petitioner’s § 103 obviousness grounds rely on the combined teachings of the references and not on a physical incorporation of elements. *See In re Mouttet*,

---

686 F.3d 1322, 1332 (Fed. Cir. 2012).

Petitioner also cites below to additional prior art as evidence of the background knowledge of a POSITA and to provide contemporaneous context to support Petitioner’s assertions regarding what a POSITA would have understood from the prior art in the grounds. *See Yeda Research v. Mylan Pharm. Inc.*, 906 F.3d 1031, 1041-1042 (Fed. Cir. 2018) (affirming the use of “supporting evidence relied upon to support the challenge”); 37 C.F.R. § 42.104(b); *see also K/S HIMPP v. Hear-Wear Techs., LLC*, 751 F.3d 1362, 1365-66 (Fed. Cir. 2014); *Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, 1363 (Fed. Cir. 2016).

**A. Statement of Material Facts**

1. Allowing unrestricted and unlimited use of a server-provided feature would create the undesirable risk the server becoming overwhelmed and failing, either intentionally (in a denial-of-service attack) or unintentionally. Ex.1037, 19:5-10 (“denial of service attack (also, DoS attack) is an attack...that causes a loss of service to users, typically...by consuming the bandwidth of the victim network or overloading the computational resources of the victim system”); Ex.1038, 2:30-39 (“denial of service (DOS) attacks, where...the server at the target IP address is overwhelmed by fake requests for service, and is either unable to service ‘real’ requests for service, or crashes due to overload”); Ex.1039, [0002] (“In a denial of service attack, one or more attackers may send packets towards one

or more servers, overwhelming the servers so that they are no longer able to serve their legitimate clients.”).

2. As of the priority date, it was common for a server to require a user login to access the server’s services. Ex.1040, Fig. 6 (showing a “typical login screen” asking for “an account name 293 and a password 295”); Ex.1021, [0024] (“A user login typically includes a user name and password.”); Ex.1022, [0028] (describing “operations typically performed” during login for “authentication and authorization”); Ex.1023, [0002] (describing the “well known” login process in which “a server computer is responsible for granting access to a remote user based on a valid combination of userID and password”).

3. A POSITA would have been familiar with voice-over-IP, which was a technique for exchange audio communications in real time. Ex.1026, [0085] (“well-known Voice-over-IP standards that are cornerstone technology for the transmission of real-time audio”); Ex.1027, 6:39-47 (“devices which perform these functions [such as analog-to-digital (A/D) conversion,...echo cancellation, and other types of signal conditioning, as well as a voice activity detector (VAD)...function for determining the temporal boundaries of a telephone caller's speech] are well-known in the art and are commercially available”); Ex.1028, [0026] (“audio (such as voice-over-IP)”).

4. A POSITA would have been familiar with instant messaging, which

---

was a technique for exchanging text-based messages in real time. Ex.1024, [0003] (“Instant messaging (“IM”) is a popular form of computer-based communication in which text-based messages are exchanged between users in real-time.”); Ex.1025, [0005] (a user “has access to a (text only) Instant Messaging client”).

5. The use of APIs to define standard, programmatic ways for an application to access functionality of a software module was well-known and commonly used. Ex.1029, [0087] (“APIs are well-known in the art”); Ex.1030, [0038] (“One of the primary purposes of an API is to provide a set of commonly-used functions”); Ex.1031, [0040] (“A well-known API”); Ex.1032, 5:32-48 (“An API is a well-known programming device.... [C]alls to API functions 60 are included in application program 56; the programmer is relieved of any task of having to incorporate or integrate the code itself of functions 60 in application program 56”).

6. A POSITA would have been familiar with device drivers for output devices and would have recognized that they commonly included APIs for reporting on an output device’s capabilities. Ex.1033, 6:30-32 (using “software (e.g., the device drivers) to determine the capabilities” of hardware resources); Ex.1034, 11:8-26 (describing device drivers determining a “display device’s maximum resolution”); Ex.1035, Abstract (“determine capabilities of graphics hardware from a device driver”), [0004] (“provid[ing] a set of application

programming interfaces (APIs) to ascertain device contexts and determine capabilities of graphics hardware from a device driver”); Ex.1036, [0033] (“device capabilities...may be exposed through one or more APIs”).

## **B. Ground 1**

### **1. Abuan**

Abuan generally relates to the field of communications software for mobile devices, and more particularly, to real-time video conferencing from mobile devices. Ex.1005, [0003]-[0004]. Abuan describes a modular approach to facilitating such conferencing features, with functionality organized into components such as a video processing and encoding module, media exchange module, video processing module, video conferencing and processing module, and many others. Ex.1005, [0043], [0099], [0056]. The modules’ functionalities are accessed via “a set of application program interfaces (APIs).” Ex.1005, [0044]. Ex.1003, ¶¶44-50.

Abuan is analogous art to the ’116 patent. First, Abuan is in the same field of endeavor as the ’116 patent, namely, the “manner in which functionality is accessed in certain environments, such as mobile device environments.” Ex.1001, 1:58-60; *see* Ex.1005, [0001]-[0002] (describing ways that “functionality” is accessed on “portable devices, such as smartphones”). Abuan and the ’116 patent also both generally relate to implementing functionality via software. *See*

Ex.1001, Abstract (“providing additional functionality to existing software”); Ex.1005, [0006]-[0007], [0105] (describing how instant messaging or voice-over-IP applications “use the video conferencing functions of the video conference module”). Abuan and the ’116 patent both describe at length features for facilitating videoconferencing. Ex.1001, Abstract, 4:43-51 (purpose of a function block is to provide functions including “audio, video,...conferencing, meetings, and/or other functions”); Ex.1005, [0099] (describing a “video conference module...for performing a variety of video conferencing functions”). Ex.1003, ¶50.

Second, Abuan is also reasonably pertinent to a problem allegedly addressed by the ’116 patent. The ’116 patent describes purported problems associated with using multiple applications on a mobile device, potentially requiring a user to “switch[] back and forth between the video window of [a] call and [a] superblock application.” Ex.1001, 4:22-26. The ’116 patent’s proposed solution is to provide functionality—such as communication functions—to a superblock application. Ex.1001, 4:43-61. Abuan similarly describes how communication functions are provided to a client application, which may be a “stand-alone application” or may itself be further “integrated into another application.” Ex.1005, [0105]. Ex.1003, ¶50.

## 2. Independent Claim 1 and dependent claims

**[1.0] A method for providing a real-time communication session over the internet for a superblock application intended for use on a computing device, the method comprising:**

Like the '116 patent, Abuan illustrates “a method for managing a video conference between a first device and a second device.” Ex.1005, Abstract. More specifically, Abuan’s teachings pertain to the transmission of “captured images to one or more devices **during a real-time communication session** between the users of” computing devices such as mobile devices. Ex.1005, [0003]; *see also* Ex.1005, [0033]-[0034] (describing “a real-time communication session between the users of the devices” and explaining that “[o]ne example of a real-time communication session that involves the transmission of the captured video images is a video conference”). Ex.1003, ¶¶51-52.

Figure 7 of Abuan, below, illustrates one example sequence for commencing a video conference between two mobile devices. With reference to Figure 7, Abuan explains that “one device (i.e., the device 705) requests a video conference and another device (i.e., the device 720) responds to such request.” Ex.1005, [0108]; Ex.1003, ¶53.

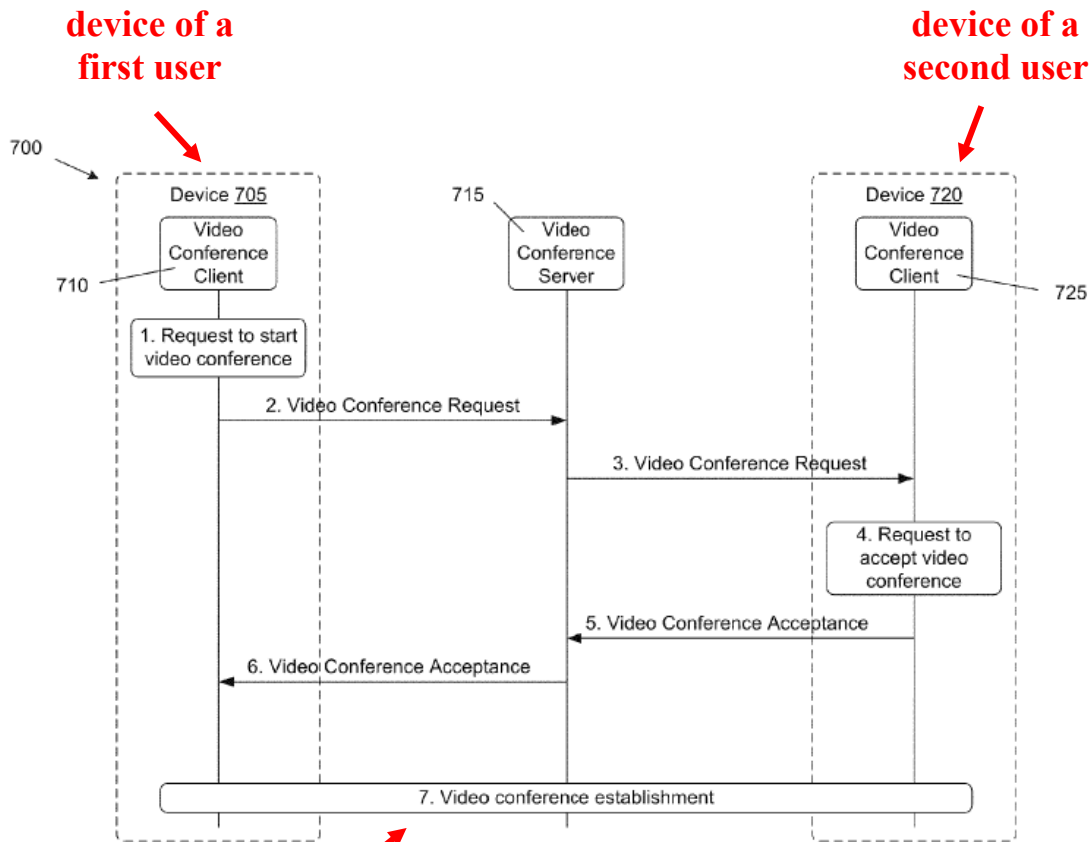


Figure 7

**Real-time communication session  
(video conference)**

**Ex.1005, Fig. 7 (annotated).**

Further, Abuan illustrates a video conference client (*superblock application intended for use on a computing device*) such as video conference client 710 (operating on device 705) and video conference client 725 (operating on device 720). Ex.1003, ¶54.

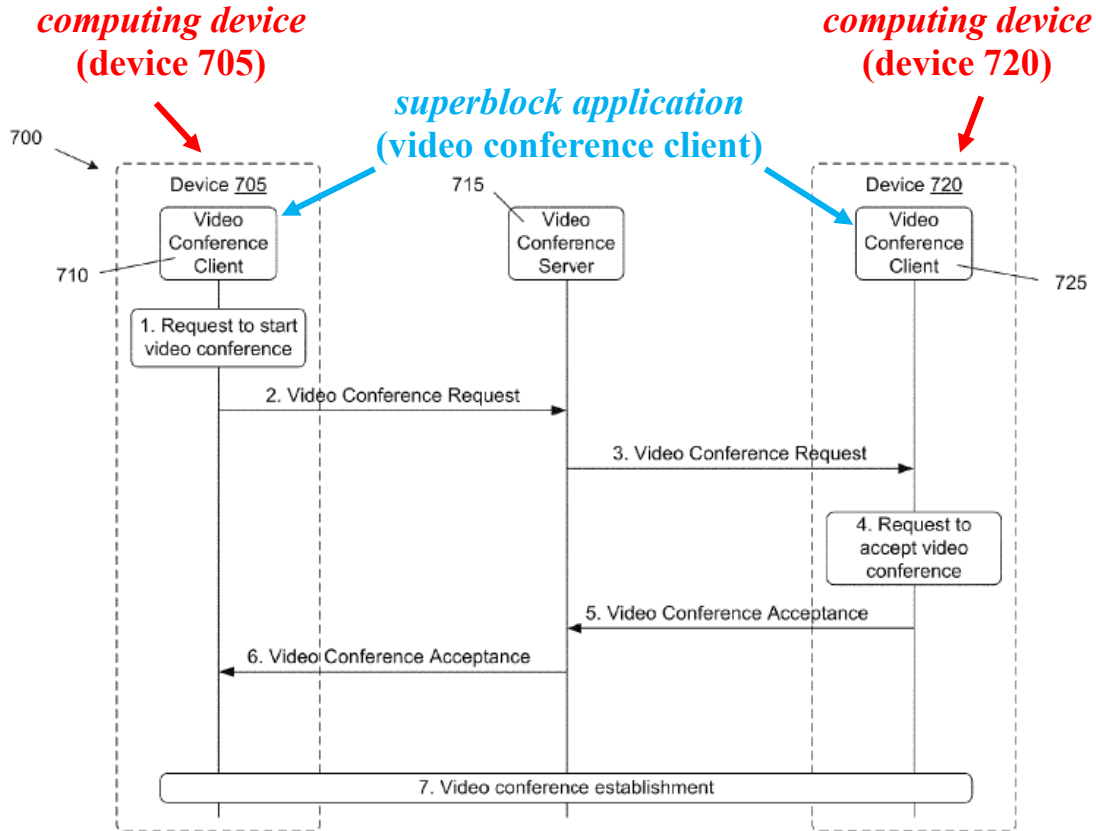


Figure 7

Ex.1005, Fig. 7.

Abuan explains that the video conference client for each respective user communicates using a network connection such as the Internet. Ex.1005, [0109] (“Each of the video conference clients 710 and 725 of some embodiments communicates with the video conference server 715 over a network (e.g.,...the Internet etc.)”). From the point of view of each user device, data for a video conference is sent to and received from the other user’s device over a network such

---

as the Internet. Ex.1005, [0106] (“[T]he network interface 650 is a communication interface that allows the video conference module 625 and the video conference client 645 to send data and receive data over a network,” such as sending “data (e.g., images captured by cameras of the dual camera mobile device) to another device on the Internet”); Ex.1003, ¶55.

In sum, and as detailed further in the analysis of each element of claim 1 below, Abuan discloses or renders obvious *providing a real-time communication session (establishing a real-time communication session such as a video conference) over the internet (communicating via the Internet) for a superblock application intended for use on a computing device (video conference client 710 (operating on device 705) and video conference client 725 (operating on device 720))*. Ex.1003, ¶56

**[1.1] providing a function block for use in adding additional functionality to a third party superblock application that has its own functionality and display window,**

“*superblock application*”: Abuan describes a video conferencing and processing module 1300 that includes a client application 1365 and various modules and components, including a video conference module 1302, that provide services to the client application. Ex.1005, [0170], [0183]. The client application is a “video conference client...that uses the video conferencing functions of the video conference module 1302.” Ex.1005, [0170]-[0171]. Abuan’s “applications are

implemented as software processes that are specified as a set of instructions recorded on a computer readable storage medium,” such as the memory of Abuan’s mobile device. Ex.1005, [0233]. The modules are comprised of software instructions, such as video conferencing instructions 1986, that are stored in memory. Ex.1005, [0233], [0254], Fig. 19; see also [0254] (explaining how an “API-implementing component” may be a “module”); Ex.1045, [0032] (“programmatic elements, often referred to as modules” “may include a program, a subroutine, a portion of a program, or a software component”). And Abuan contemplates that the client application 1365 is a third-party application, such as Skype. Ex.1005, [0171]. Thus, the video conferencing and processing module 1300 (including the client application 1365 and the various modules) corresponds to the claimed “*third party superblock application.*” Ex.1003, ¶¶57-58.



The superblock application and function block are annotated in Abuan's

Figures 13 and 14 below. Consistent with the '116 patent specification's examples, the function block is part of the superblock application. See, e.g., Ex.1001, Fig.3B; Ex.1003, ¶160.

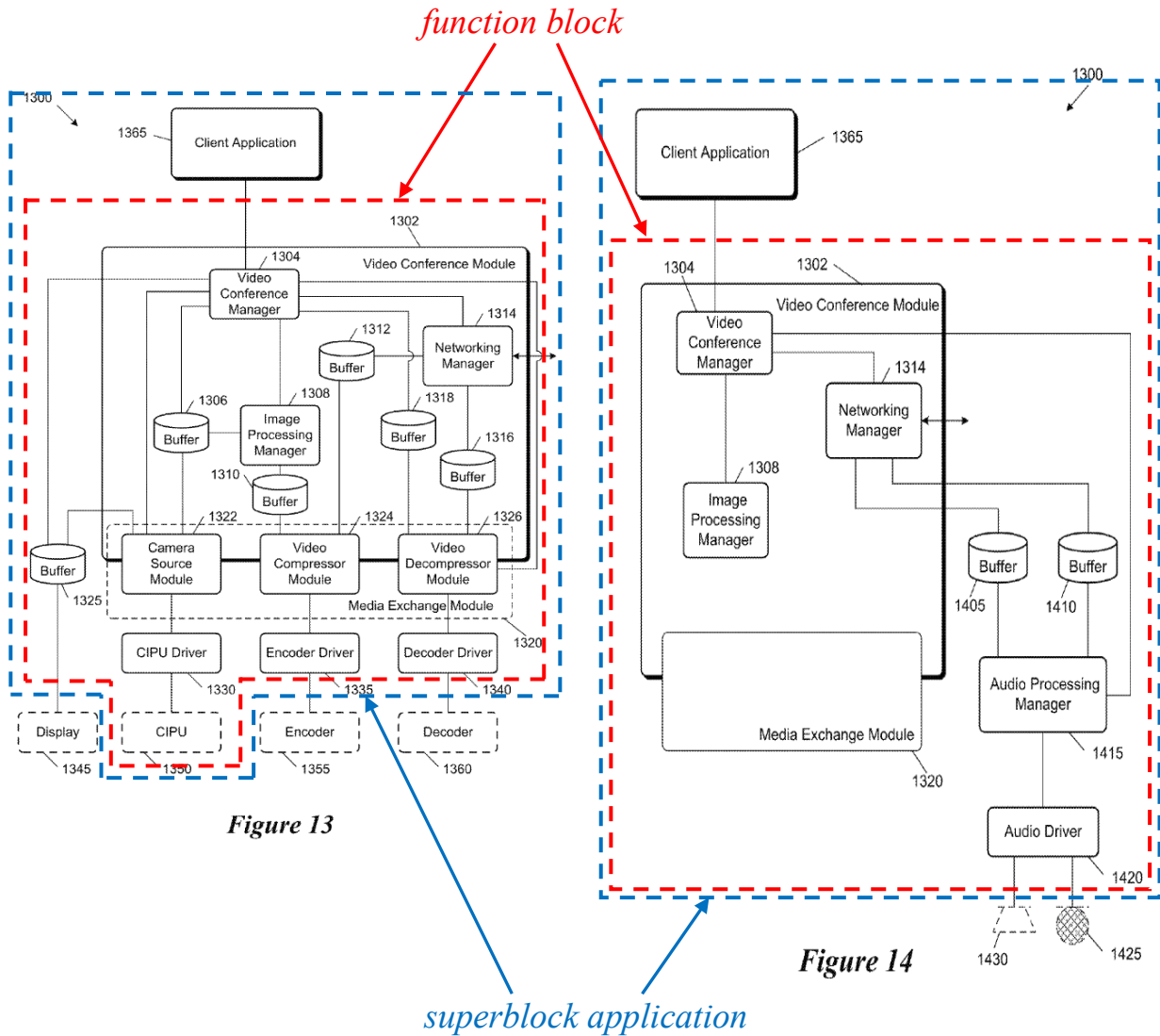


Figure 13

Figure 14

superblock application

Ex.1005, Figs. 13 and 14 (annotated).

Abuan explains that “client application 1365 [the superblock application]...uses the video conferencing functions [functionality] of the video conference module 1302 [the function block].” Ex.1005, [0171]. Further, Abuan describes the application employing the videoconferencing module and other modular components as “a stand-alone application,” such that it *has its own* stand-alone *functionality and display window*. Ex.1005, [0105], Figs. 8-11 (illustrating example user interfaces for videoconference functionality with display windows). A POSITA would have understood Abuan’s reference to a “stand-alone application” to mean that all the functionality of the application, including its module subcomponents (e.g., *the function block* functionality) is compiled into a single, unitary application. *See* Ex.1012, 1:26-29 (“Traditionally, an application's source files are compiled in object modules and then linked together with other object modules, generically called libraries, to form a complete stand-alone application.”). Thus, it would have been obvious to a POSITA for *the superblock applications* (Abuan’s client application operating on a users’ device) to include the module subcomponents provided as *a function block* and performed within the execution context of Abuan’s client application and not some other application. Ex.1003, ¶61.

For example, Abuan describes how an application accesses the functionality of the other software modules via defined application programming interfaces

(APIs). Ex.1005, [0235]-[0237]. This structure puts functionality in the API-implementing component (e.g., the function block) and not in the API-calling component (e.g., the superbloc application):

An API allows a developer of an API-calling component (which may be a third party developer) to **leverage specified features provided by an API-implementing component**.

Ex.1005, [0239].

It will be appreciated that the **API-implementing component 1810 may include additional functions, methods, classes, data structures, and/or other features that are not specified through the API 1820 and are not available to the API-calling component 1830**.

Ex.1005, [0247]; Ex.1003, ¶62.

The use of APIs to define standard, programmatic ways for an application to access functionality of a software module was well-known and commonly used.

Ex.1029, [0087] (“APIs are well-known in the art”); Ex.1030, [0038] (“One of the primary purposes of an API is to provide a set of commonly-used functions”);

Ex.1031, [0040] (“A well-known API”); Ex.1032, 5:32-48 (“An API is a well-known programming device.... [C]alls to API functions 60 are included in

application program 56; the programmer is relieved of any task of having to incorporate or integrate the code itself of functions 60 in application program 56”);

Ex.1003, ¶¶63-64.

Thus, Abuan renders obvious this limitation.

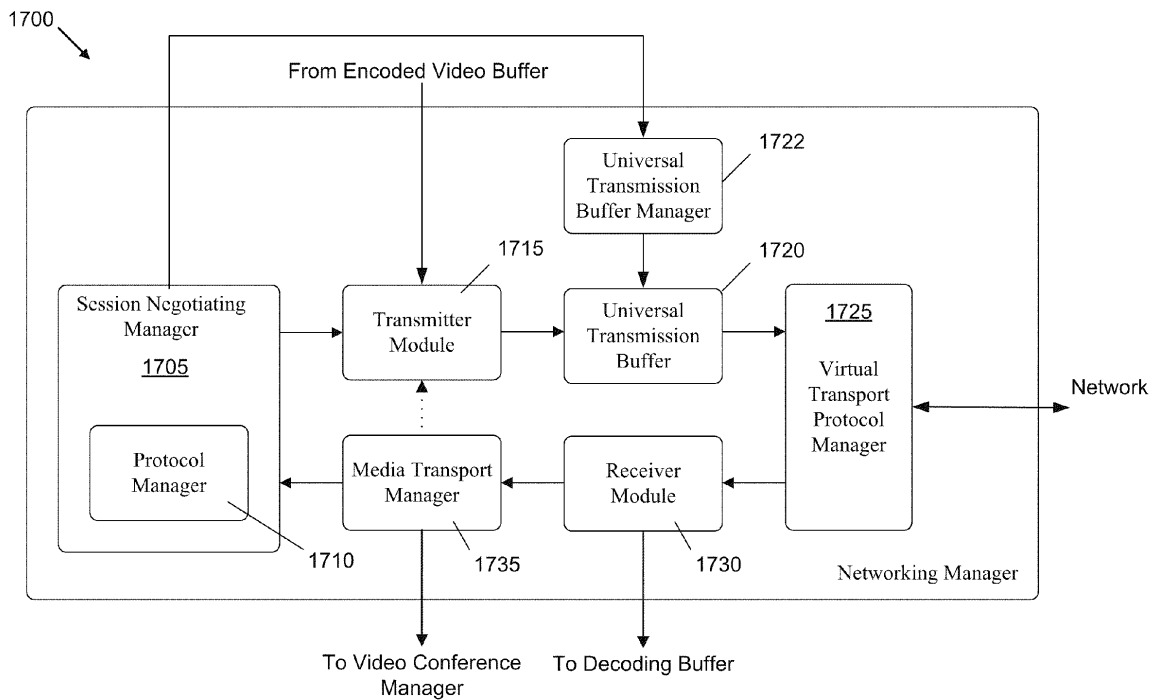
**[1.2] wherein the function block is configured to be compiled into the superblock application and is configured to add the additional functionality to provide the real-time communication session using one or more servers connected over the internet, and**

As discussed above in [1.1], Abuan describes the application employing the videoconferencing module and other modular components as “a stand-alone application.” Ex.1005, [0105]. A POSITA would have understood Abuan’s reference to a “stand-alone application” to mean that all the functionality of the application, including its module subcomponents (e.g., *the function block* functionality) is contained in a single, unitary application. See Ex.1012, 1:26-29 (“Traditionally, an application's source files are compiled in object modules and then linked together with other object modules, generically called libraries, to form a complete stand-alone application.”); Ex.1003, ¶¶65-66. Thus, it would have been obvious to a POSITA that Abuan’s client application (*the superblock application*) includes the module subcomponents as *the function block configured to be compiled* therein and performed within the execution context of Abuan’s client application and not some other application. And Abuan explains that the API functions or methods discussed above in [1.1] are compiled into stand-alone applications. Ex.1005, [0239] (explaining that an “API can be specified in terms of a programming language that can be interpreted or compiled when an application is built”); Ex.1003, ¶66.

Further, Abuan explains that its computing devices, such as device 705 or device 720, include a networking manager as part of the video conference module (*function block*) reflected in Figure 13 and shown in further detail in Figure 17.

Abuan explains that “the networking manager 1700 manages network connections (e.g., connection establishment, connection monitoring, connection adjustments, connection tear down, etc.) between a dual camera mobile device on which it operates and a remote device in a video conference.” Ex.1005, [0208]; Ex.1003,

¶67.



**Figure 17**

**Ex.1005, Fig. 17.**

As reflected in Figure 17, above, the networking manager 1700 “includes a session negotiating manager 1705, a transmitter module 1715, a universal transmission buffer 1720, a universal transmission buffer manager 1722, a virtual transport protocol (VTP) manager 1725, a receiver module 1730, and a media transport manager 1735.” Ex.1005, [0209]. These elements of the video conference manager (*function block*) are *configured to add the additional functionality to provide the real-time communication session using one or more servers connected over the internet* because they allow for the establishment and management of a video conference. Ex.1005, [0208]. And as reflected in Figure 7, below, Abuan explains that a video conference is established between a first user and a second user using an intermediary video conference server. Ex.1005, [0109] (“**Each of the video conference clients 710 and 725 of some embodiments communicates with the video conference server 715 over a network...through a network interface** such as the network interface 650 described above.”); Ex.1003, ¶68.

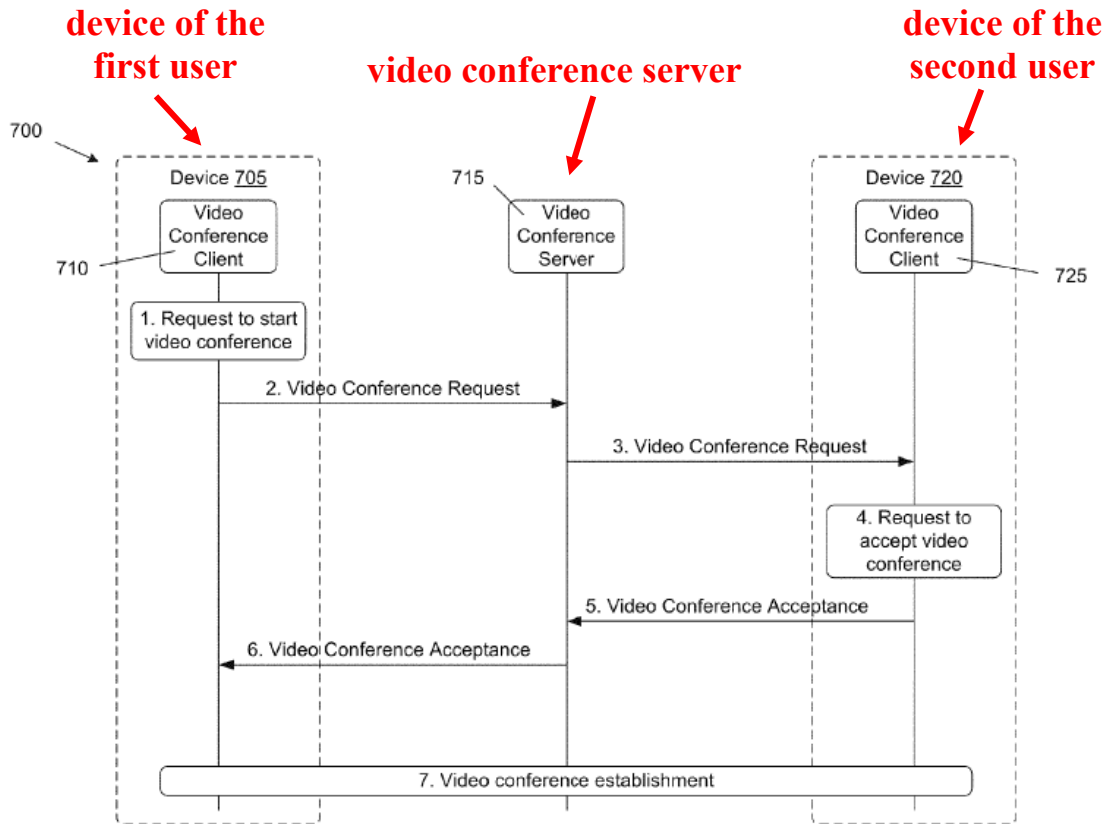


Figure 7

Ex.1005, Fig. 7.

Accordingly, because Abuan's video conference manager (*function block*) provides the networking manager 1700 which in turn manages communications to the video conference server (*configured to add the additional functionality to provide the real-time communication session using one or more servers connected over the internet*), Abuan discloses or renders obvious this limitation. Ex.1003, ¶69.

**[1.3] wherein the function block is configured to interact with the superblock application through one or more application programming interface (API) calls; and**

As explained above at [1.1], Abuan’s video conferencing and processing module 1300 (including the client application 1365 and the various modules) corresponds to the claimed “*superblock application.*” Ex.1003, ¶¶70-71.

As explained above at [1.1]-[1.2], Abuan’s various modules and components implementing functions provided to the client application 1365 (e.g., the video conference module 1302) correspond to the claimed “*function block.*” And as detailed in [1.1]-[1.2], Abuan describes how an application accesses the functionality of the other software modules via defined *application programming interface (APIs) calls*. Ex.1005, [0235]-[0237]; Material Fact #5, Material Fact #6; Ex.1003, ¶¶72-74.

Thus, Abuan discloses or renders obvious this limitation.

**[1.4] enabling establishment of the real-time communication session between the one or more servers and the function block compiled into the superblock application so that the function block can provide the real-time communication session to the superblock application.**

Abuan explains, with reference to Figure 7 above below, that a “video conference server 715 forwards (at operation 3) [a] video conference request [from device 705] to the video conference client 725 of the device 720” to establish a video conference between device 705 and device 720 (*enabling establishment of the real-time communication session*). Ex.1005, [0110]-[0111]. This

communication between the video conference client 725—operating on device 720—and the video conference server 715, is performed by the video conference module 1302 on that device (*between the one or more servers and the function block compiled into the superblock application*). More specifically, Abuan discloses that “[e]ach of the video conference clients 710 and 725...communicates with the video conference server 715 over a network (e.g., a cellular network, a local area network, a wireless network, a network of networks, the Internet etc.) through a network interface such as the network interface 650.” Ex.1005, [0109]. Abuan explains that the networking manager 1314 of the video conference module 1302 (or network layer 640 of the video conference module 625) “establishes the connections between the dual camera mobile device and the other device of the video conference at the start of the video conference.” Ex.1005, [0179]; *see also* [0104]; Ex.1005, [0156] (“A more detailed version of this video conference module [625] will be described below by reference to FIG. 13.”), [0171] (“the client application 1365 is the same as the video conference client 645 of FIG. 6.”). In this way, the video conferencing module 1302 (*the function block*) establishes the video conference session (*can provide the real-time communication session*) between the video conference client of device 705 (*the superblock application*) and device 720. Ex.1003, ¶75-77.

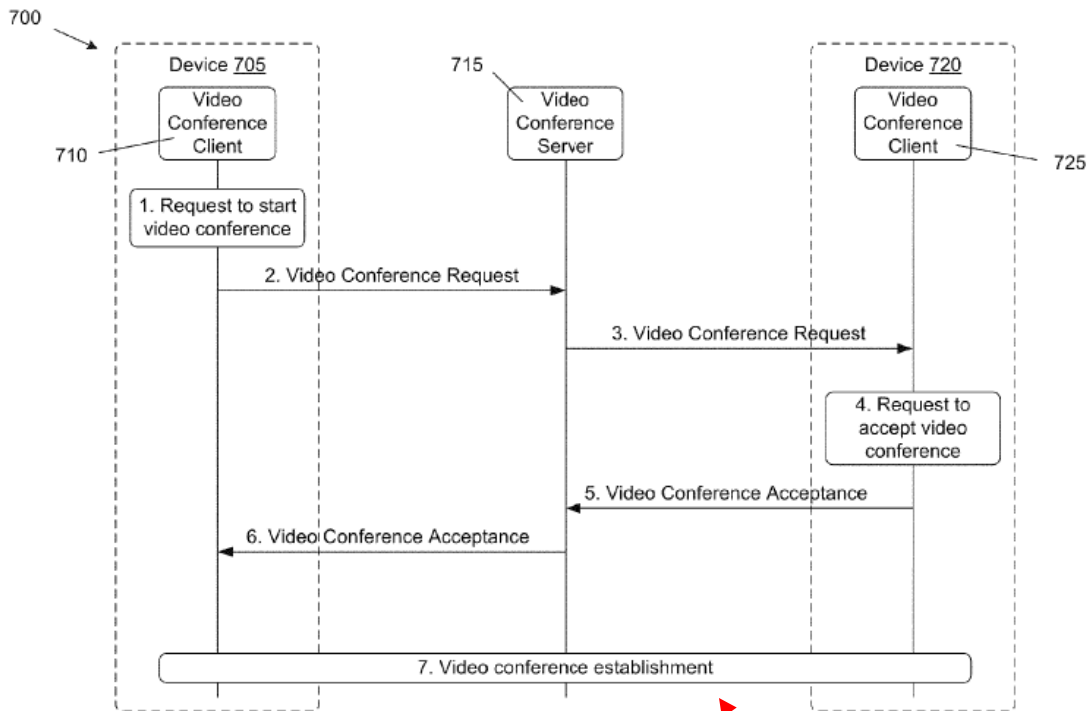


Figure 7

*establishment of the real-time communication session*

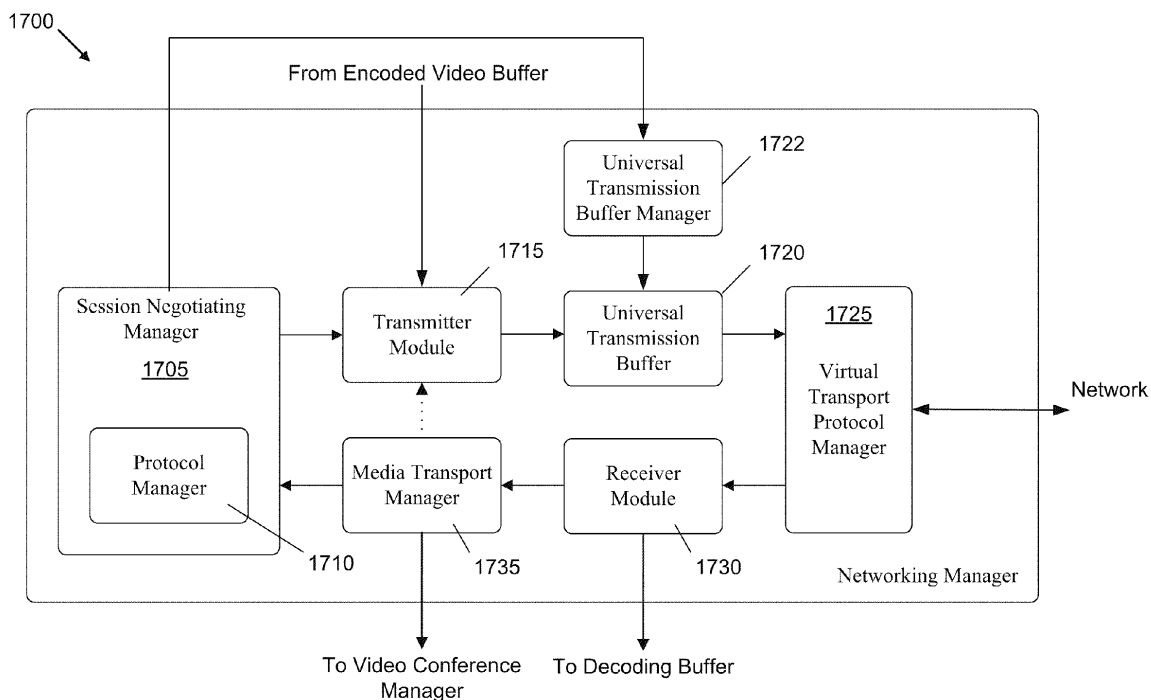
Ex.1005, Fig. 7.

Thus, Abuan renders obvious this limitation.

**[2.0] The method of claim 1 wherein enabling the establishment of the real-time communication session includes signaling communications between the one or more servers and the function block.**

Abuan explains that its computing devices, such as device 705 or device 720, include a networking manager as part of the video conference module (*function block*) reflected in Figure 13 and shown in further detail in Figure 17.

Abuan explains that “the networking manager 1700 manages network connections (e.g., connection establishment, connection monitoring, connection adjustments, connection tear down, etc.) between a dual camera mobile device on which it operates and a remote device in a video conference.” Ex.1005, [0208]; Ex.1003, ¶¶78-79.



**Figure 17**

**Ex.1005, Fig. 17.**

As discussed above in claim 1, Abuan explains that a video conference is established between a first user and a second user via messages (*signaling communications*) exchanged with an intermediary video conference server.

Ex.1003, ¶80.

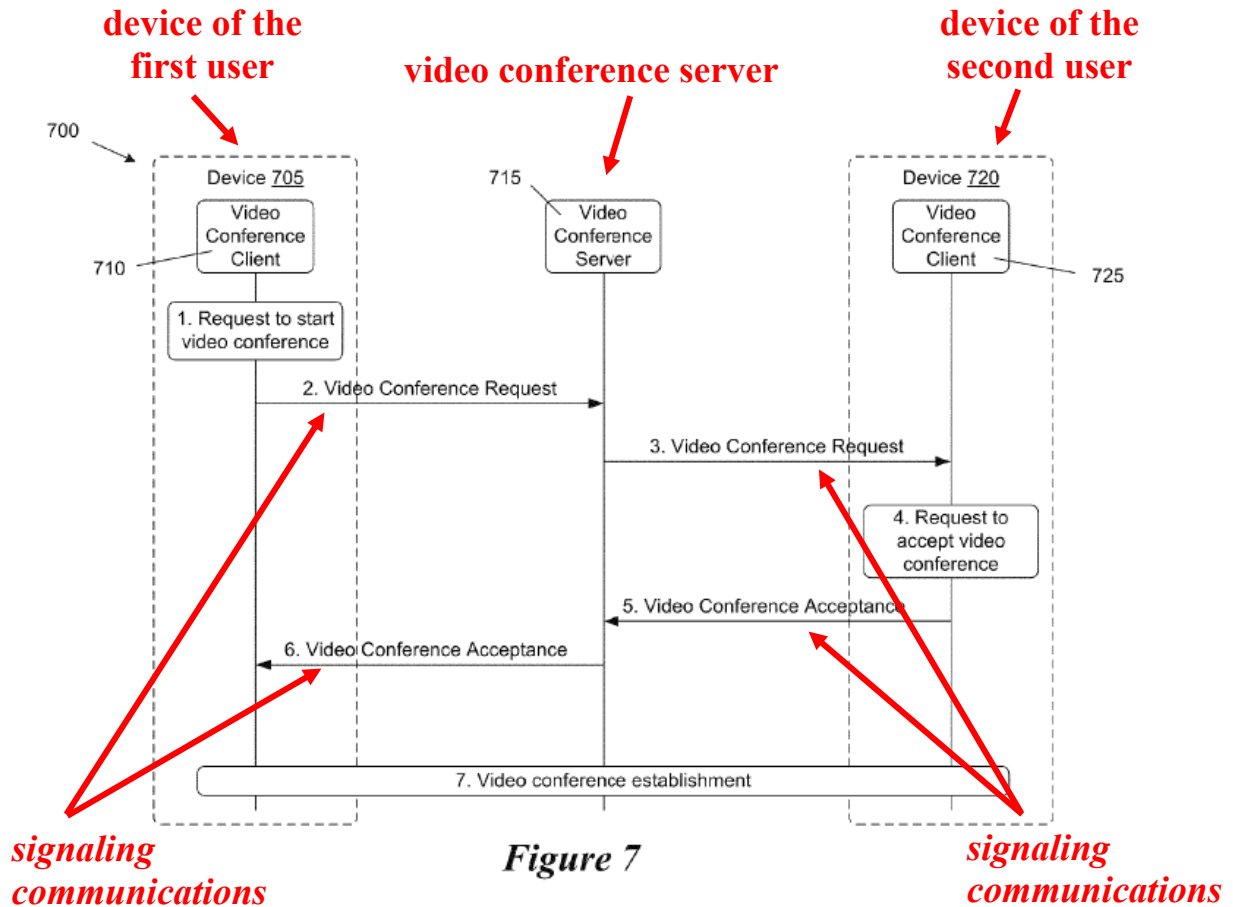


Figure 7

Ex.1005, Fig. 7.

More specifically, as discussed above in [1.2], Abuan teaches that *enabling the establishment of the real-time communication session includes signaling communications between the one or more servers and the function block* because “[e]ach of the video conference clients 710 and 725 of some embodiments communicates with the video conference server 715 over a network[.]” Ex.1005, [0109]. Abuan explains that the networking manager 1314 of the video conference

module 1302 (or network layer 640 of the video conference module 625, *the function block*) “establishes the connections between the dual camera mobile device and the other device of the video conference at the start of the video conference” via the intermediary video conference server 715 (*the one or more servers*). Ex.1005, [0179]; *see also* [0104]; Ex.1003, ¶¶81-82.

Thus, Abuan renders obvious this claim.

**[3.0] The method of claim 2 wherein the signaling communications include session setup, management, and teardown.**

As discussed above for claim 2, Abuan explains that its computing devices include a networking manager 1700 which manages the *signaling communications*. These *signaling communications include session setup, management, and teardown* because, as Abuan elaborates, “the networking manager 1700 **manages** network connections (e.g., **connection establishment, connection monitoring, connection adjustments, connection tear down**, etc.) between a dual camera mobile device on which it operates and a remote device in a video conference.” Ex.1005, [0208]; Ex.1003, ¶¶83-85.

Thus, Abuan renders obvious this claim.

**[4.0] The method of claim 2 wherein the signaling communications use Session Initiation Protocol (SIP) as a signaling protocol.**

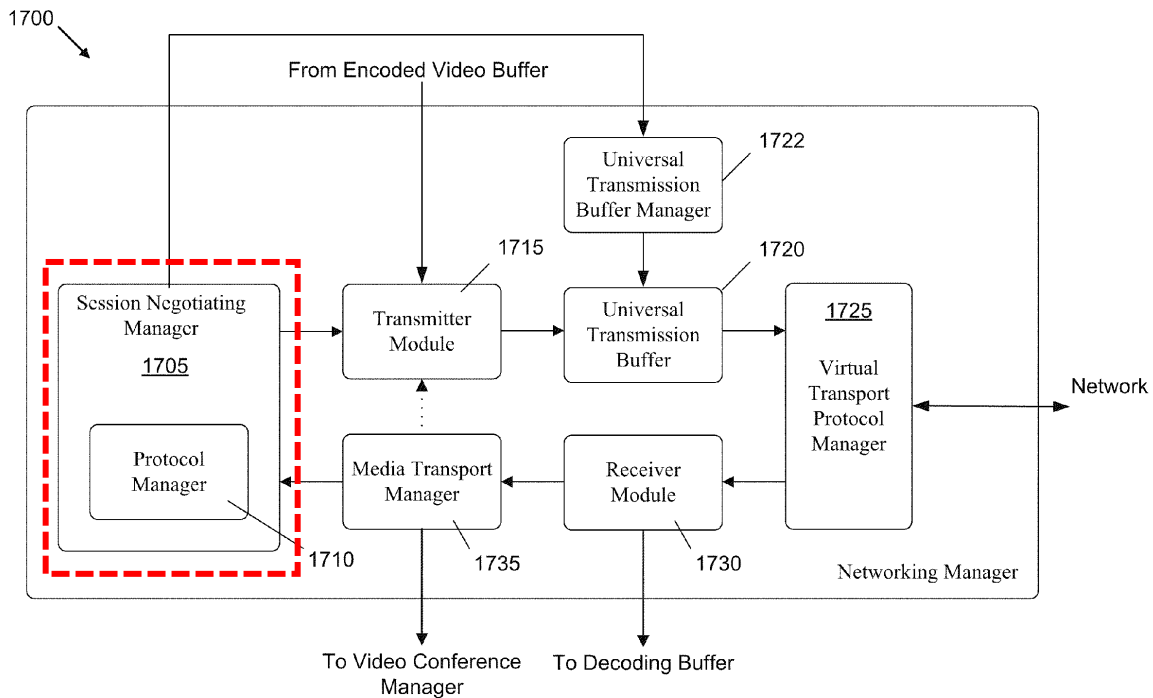
As discussed above for claim 2, Abuan explains that its computing devices include a networking manager 1700 which manages the *signaling communications*.

Abuan further explains that the networking manager 1700 uses *a signaling protocol*, such as *Session Initiation Protocol (SIP)*, for its signaling communications. Ex.1005, Ex.1005, [0211] (“the session negotiating manager 1705 is also responsible for establishing multimedia communication sessions...(e.g., using **a session initiation protocol (SIP)**).”); Ex.1003, ¶¶86-88.

Thus, Abuan renders obvious this claim.

**[5.0] The method of claim 1 wherein the real-time communication session uses Real-time Transport Protocol (RTP) as a data transport protocol.**

As discussed above for claim 2, Abuan explains that its computing devices include a networking manager 1700. As reflected in Figure 17, below, the networking manager 1700 “includes a session negotiating manager 1705, a transmitter module 1715, a universal transmission buffer 1720, a universal transmission buffer manager 1722, a virtual transport protocol (VTP) manager 1725, a receiver module 1730, and a media transport manager 1735.” Ex.1005, [0209]; Ex.1003, ¶¶89-90.



**Figure 17**

**Ex.1005, Fig. 17.**

Abuan’s networking manager 1700 includes a session negotiating manager 1705, which “is responsible for establishing connections between the dual camera mobile device and one or more remote devices participating in the video conference, as well as tearing down these connections after the conference.”

Ex.1005, [0211]. And the session negotiating manager 1705 further “includes a protocol manager 1710” that “ensures that the transmitter module 1715 uses a correct communication protocol to transmit data to a remote device during the video conference and enforces rules of the communication protocol that is used.”

---

Ex.1005, [0210]. Abuan explains that a number of communication protocols are supported by the protocol manager 1710, including “a real-time transport protocol (RTP),” rendering obvious that *the real-time communication session uses Real-time Transport Protocol (RTP) as a data transport protocol*. Ex.1005, [0210]; Ex.1003, ¶¶91-92.

Thus, Abuan renders obvious this claim.

**[6.0] The method of claim 1 wherein enabling the establishment of the real-time communication session includes negotiating signaling and media parameters between the one or more servers and the function block.**

Abuan explains that after a user has accepted a video conference request, video conference establishment includes negotiating the connection. Ex.1005, [0114] (“**the video conference establishment** of some embodiments **includes negotiating a connection** between the device 705 and the device 720, determining a bit rate at which to encode video, and exchanging video between the device 705 and the device 720”); Ex.1003, ¶¶93-94.

More specifically, Abuan explains that, with reference to Figure 17, the networking manager 1700 (part of the *function block*) includes a session negotiating manager 1705, which is “is responsible for establishing connections between the dual camera mobile device and one or more remote devices participating in the video conference, as well as tearing down these connections after the conference.” Ex.1005, [0211]. Abuan’s session negotiating manager 1705 also “includes a



instance, RFC3264 explains an “offer/answer model [] used by protocols like the Session Initiation Protocol (SIP)” to allow two entities to “arrive at a common view of a multimedia session between them,” including codecs to be used. Ex.1041, Abstract. In such implementation, “one participant in the session generates an SDP message that constitutes the offer - the set of media streams and codecs the offerer wishes to use, along with the IP addresses and ports the offerer would like to use to receive the media. The offer is conveyed to the other participant, called the answerer.” Ex.1041, 1-2. “The answerer generates an answer, which is an SDP message that responds to the offer provided by the offerer. The answer has a matching media stream for each stream in the offer, indicating whether the stream is accepted or not, along with the codecs that will be used and the IP addresses and ports that the answerer wants to use to receive media.” Ex.1041, 2; Ex.1003, ¶96.

Accordingly, a POSITA would have recognized that in negotiating a connection between two user devices and video conference server 715 (*one or more servers*) to establish a video conference (*enabling the establishment of the real-time communication session*), Abuan’s session negotiating manager 1705 (part of *the function block*) ensures the correct communication protocol is used (*negotiating signaling... parameters*) and how encoded images are created (*negotiating...media parameters*). Ex.1003, ¶¶97-98.

Thus, Abuan renders obvious this claim.

**[7.0] The method of claim 6 wherein the signaling and media parameters include a bandwidth parameter.**

Abuan explains that the session negotiating manager 1705 (which negotiates *signaling and media parameters*, as discussed above in [6.0]) further provides information including instructions related to a bit rate (*bandwidth parameter*). Ex.1005, [0212] (“the **session negotiating manager 1705** can determine whether too many packets are being sent and **instruct the universal transmission buffer manager 1722 to have the universal transmission buffer 1720 transmit fewer packets (i.e., to adjust the bit rate as described in FIG. 12).**”); *see also* Ex.1005, [0159] (“the set of network condition parameters include one-way latency and a bandwidth estimation bit rate.”), [0166] (“Since the available bandwidth for the video conference can change during the video conference, some embodiments continue to adjust the bit rate based on the set of network condition parameters (i.e., the one-way latency and the bandwidth estimation bit rate) that are received from the remote device”); Ex.1003, ¶¶99-101.

Thus, Abuan renders obvious this claim.

**[8.0] The method of claim 6 wherein the signaling and media parameters include a codec parameter.**

Abuan explains that the session negotiating manager 1705 (which negotiates *signaling and media parameters*, as discussed above in [6.0]) further provides

information including how encoded images are created and transmitted. Ex.1005, [0213] (“**The manner in which the encoded images are created and sent to the transmitter module 1715 can be based on instructions or data received from the media transport manager 1735 and/or the session negotiating manager 1705.**”). Abuan’s information about how to create encoded images is a “*codec parameter.*” And as discussed in [6.0], it was well-known for such parameter negotiation to be performed “using [] session initiation protocol (SIP)” as contemplated by Abuan. Ex.1005, [0211]; *see* Ex.1041, 1-2 (describing the negotiation of codecs that will be used via SIP messages); Ex.1003, ¶¶102-105.

Thus, Abuan renders obvious this claim.

**[11.0] The method of claim 1 further comprising sending a notification from the one or more servers to the function block.**

Abuan teaches that as part of the establishment of real-time communication session between the users, a video conference request sent to the video conference server. Ex.1005, [0111] (“**the video conference client 710 sends (at operation 2) a video conference request, which indicates the device 720 as the recipient based on input from the user, to the video conference server 715**”); Ex.1003, ¶¶106-108. This is reflected in operation 2 of Figure 7, below:

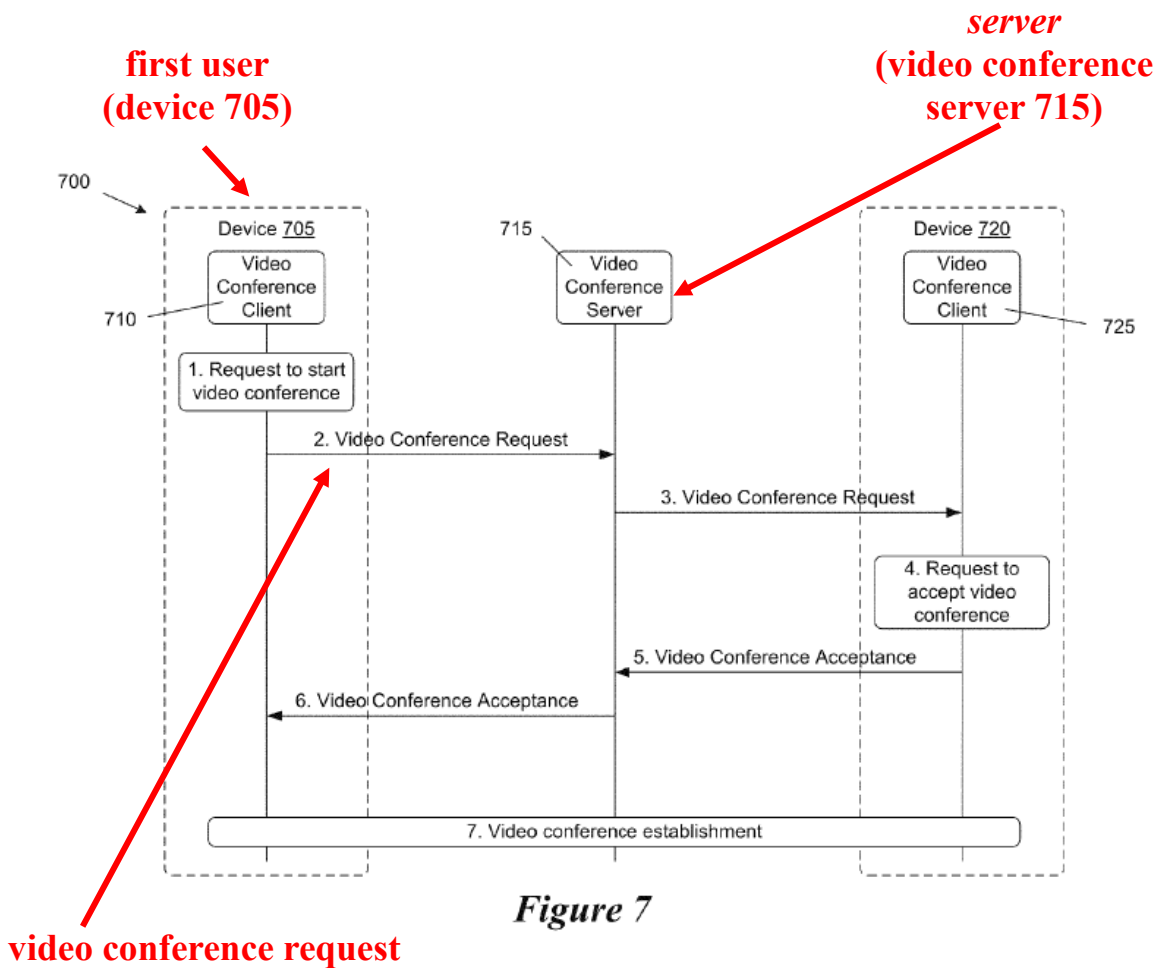


Figure 7

Ex.1005, Fig. 7.

Abuan explains that the video conference request is then sent (*sending a notification*) from the video conference server to the video conference client of a second device (*from the one or more servers to the function block*), where user interface is displayed to indicate that a request to start a video call by the first user. Ex.1005, [0108] (“FIG. 7 conceptually illustrates an example video conference request messaging sequence 700 of some embodiments...**one device (i.e., the**

device 705) requests a video conference and another device (i.e., the device 720) responds to such request.”); Ex.1003, ¶¶109-110. This is reflected in operation 4 of Figure 7, below:

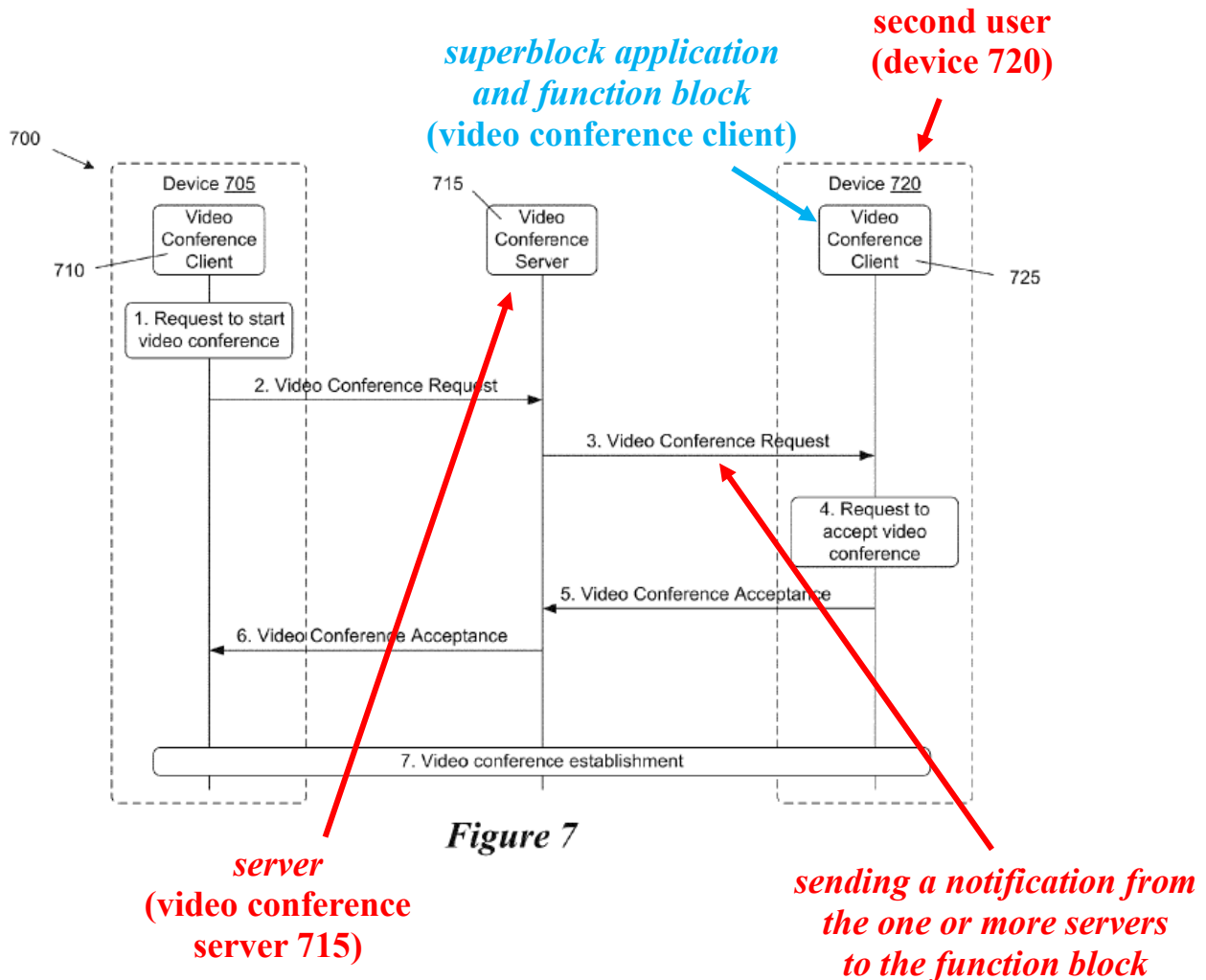
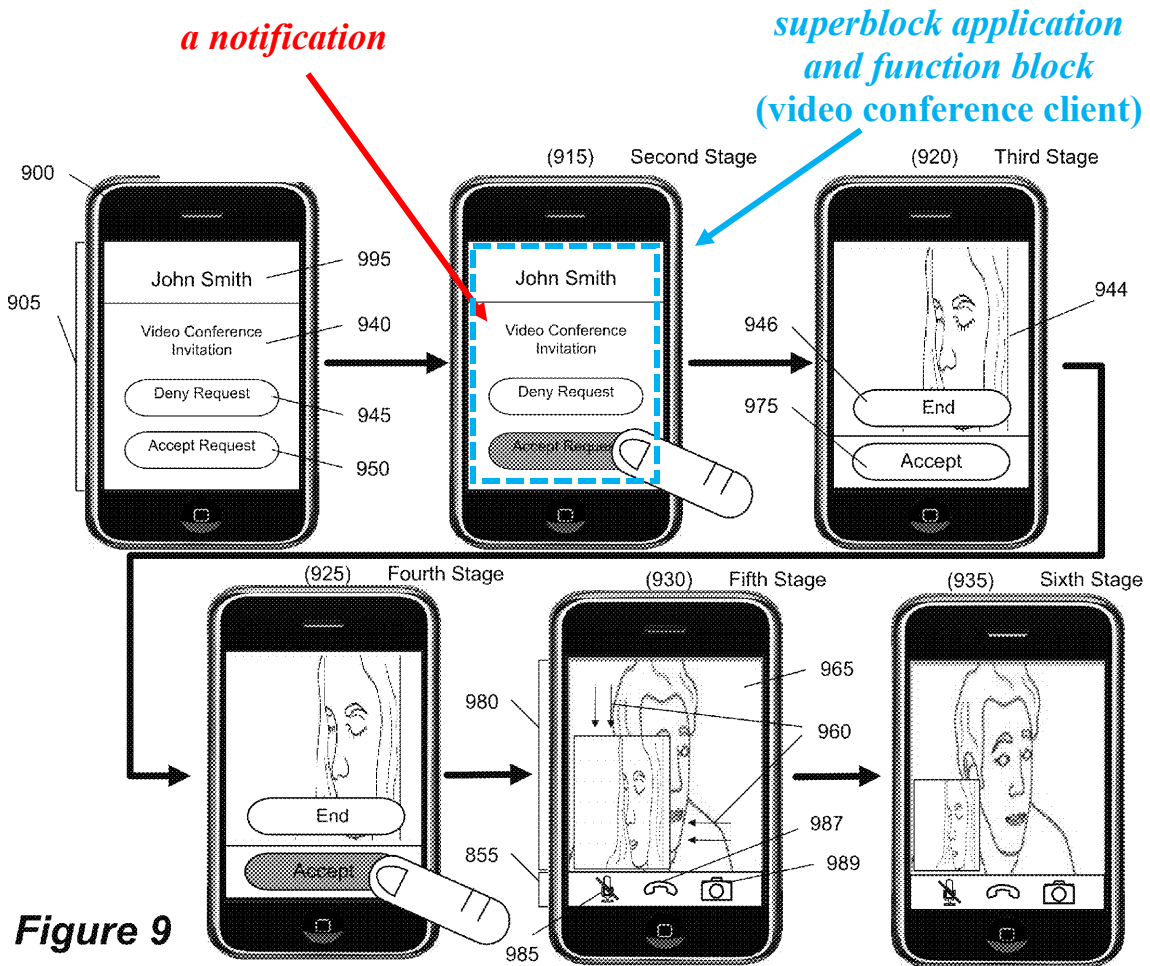


Figure 7

Ex.1005, Fig. 7.

Abuan explains that the video conference client operating on device 720 (i.e., including the function block) receives a video conference request, the user

interface displays *a notification*. Ex.1005, [0112] (“When the video conference client 725 of some embodiments receives the video conference request, a user interface is displayed on the device 720 to indicate to the user of the device 720 that the user of the device 705 sent a request to start a video conference”); Ex.1003, ¶111. This is further illustrated in Figure 9, below, as a user interface displays a notification with items including “a name field 995, a message field 940, and two selectable UI items 945 and 950.” Ex.1005, [0130]. “Upon seeing the ‘Video Conference Invitation’ notation displayed in the message field 940, the invite recipient may deny or accept the request by selecting the Deny Request option 945 or Accept Request option 950 in the UI, respectively.” Ex.1005, [0131]. “The selectable UI item 975 is an Accept button 975 that the user may select to start video conferencing.” Ex.1005, [0134]; Ex.1003, ¶¶112-113.



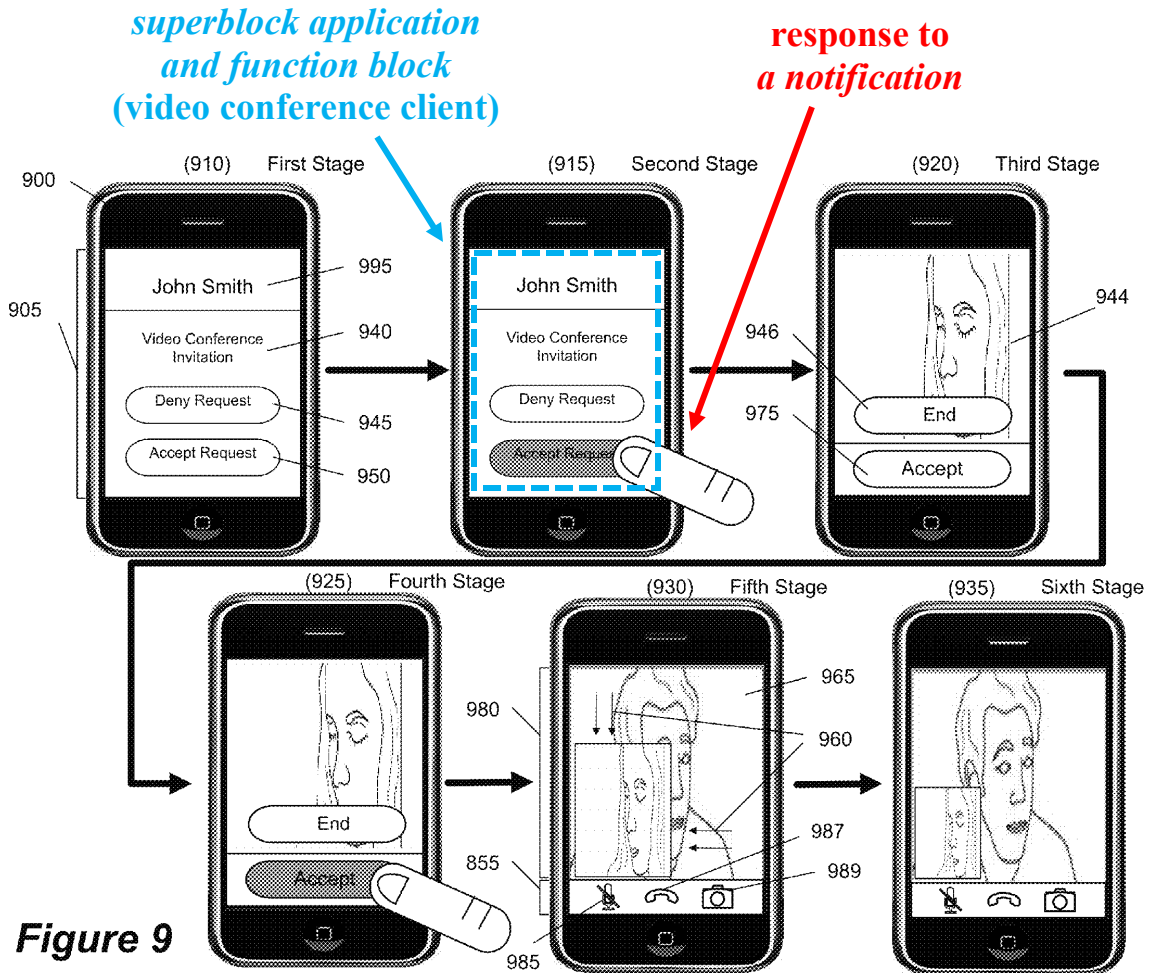
Ex.1005, Fig. 9.

Thus, Abuan discloses or renders obvious this claim.

**[12.0] The method of claim 11 wherein the notification requires a response and the establishment of the real-time communication session is the response to the notification.**

As discussed at [11.0], Abuan explains that a user may respond to a video conference request by accepting or denying the request shown at step 915 (“the

*notification*”). Ex.1005, Fig. 9. “Upon seeing the ‘Video Conference Invitation’ notation displayed in the message field 940, the invite recipient may deny or accept the request by selecting the Deny Request option 945 or Accept Request option 950 in the UI, respectively.” Ex.1005, [0130]. Since the user interface at step 915 offers only two mutually exclusive options, it would have been obvious *the notification requires a response*. Ex.1003, ¶115. When “Accept Request” is selected, a video conference is established “between the device 705 and the device 720” (such that *the establishment of the real-time communication session is the response to the notification*). Ex.1005, [0111]-[0112]. Figure 9, below, provides an example of where “Accept Request” is selected. Ex.1003, ¶¶114-116.



Ex.1005, Fig. 9.

Thus, Abuan discloses or renders obvious this claim.

**[13.0] The method of claim 11 wherein the notification is a request to initiate the real-time communication session, and an API call indicates whether the request has been granted or denied by the superblock application.**

Abuan illustrates, with respect to Figure 7, below, the establishment of a video conference, where a video conference request sent to the video conference

server (step 2), which is then sent from the video conference server to the video conference client of a second device (step 3), where a user interface displays a notification to indicate that a request to start a video call has been made by the first user (*the notification is a request to initiate the real-time communication session*).

Ex.1003, ¶¶117-119.

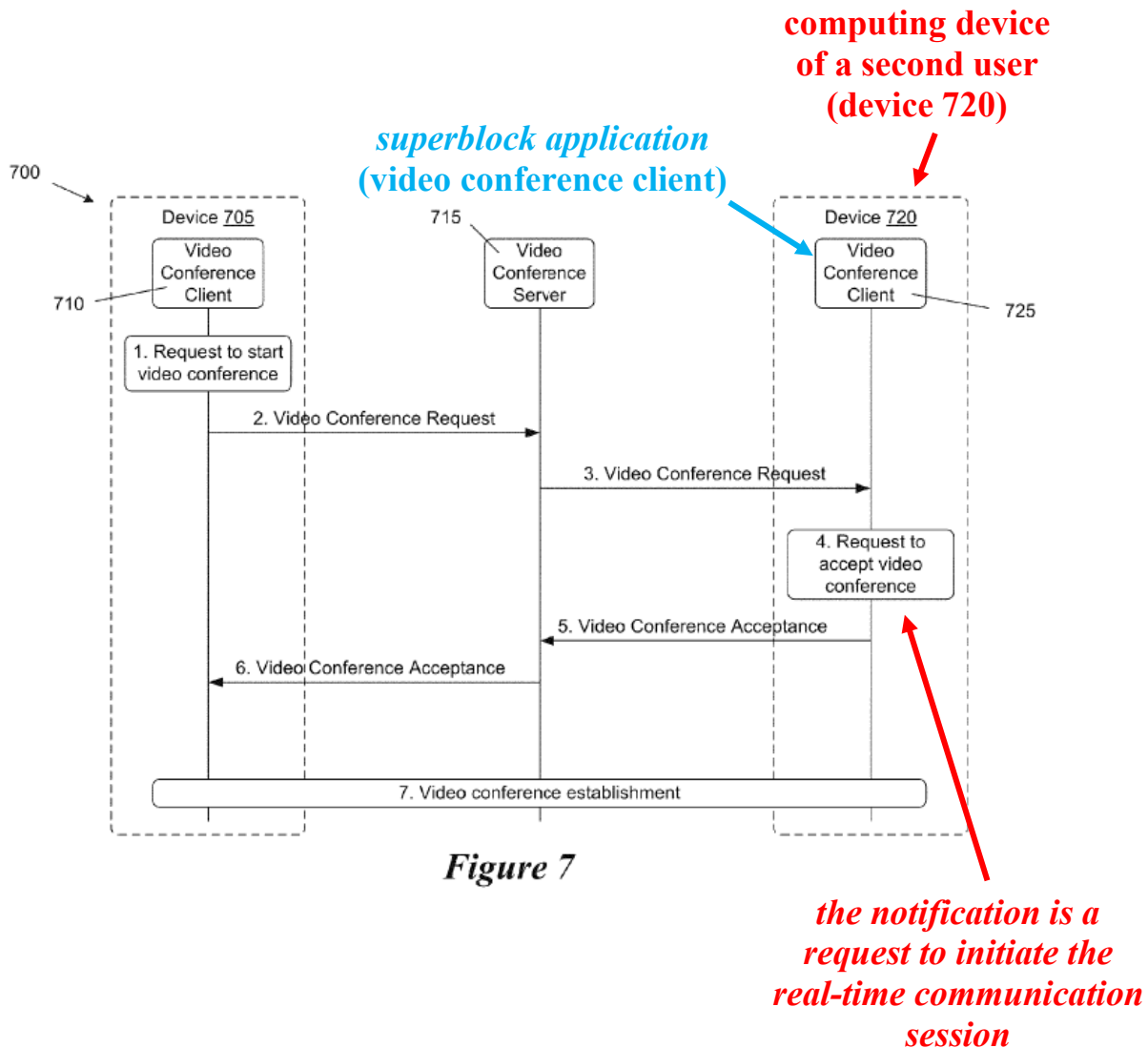
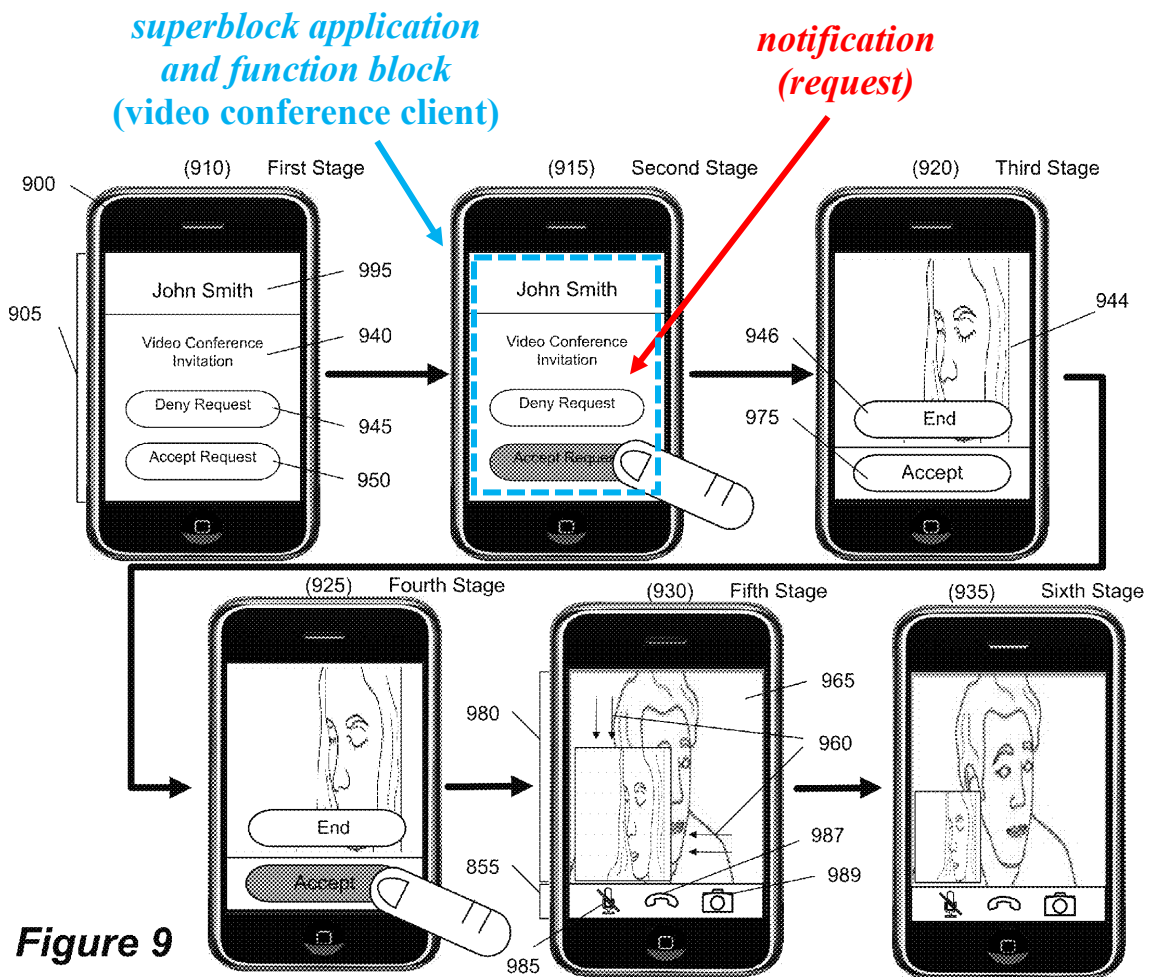


Figure 7

Ex.1005, Fig. 7.

This request is further illustrated in Figure 9, below:



**Ex.1005, Fig. 9.**

Further, as discussed above for [1.3], Abuan teaches that API calls are used to allow a client program to access services provided by libraries, such that *an API call indicates whether the request has been granted or denied by the superblock application*. More specifically, Abuan explains that the video conference client application sends instructions from a user of the application to the video

conference module 1302, including instructions to start a video conference.

Ex.1005, [0172] (“client application 1365 of some embodiments sends instructions to the video conference module 1302 such as instructions to start a conference” and “routes instructions from a user of the dual camera mobile device to the video conference module 1302,” referencing Fig. 13), [0187] (“the instructions are received from the client application 1365 or are received from a user through a user interface displayed on the dual camera mobile device and forwarded to the video conference manager 1304 by the client application 1365,” referencing Fig. 15); *see* Ex.1005, [0186] (“The operation of the video conference manager 1304 of some embodiments will now be described by reference to FIG. 15.”); Ex.1003, ¶120.

Accordingly, because the client application 1365 (*superblock application*) communicates with the video conference module (*function block*) via API calls, including instructions from a user such as instructions to start a video conference, Abuan renders obvious *an API call indicates whether the request has been granted or denied by the superblock application*. Ex.1003, ¶121.

**[14.0] The method of claim 11 wherein the notification is a presence notification.**

As discussed above in [11.0], Abuan teaches that as part of the establishment of real-time communication session between the users, a video conference request sent to the video conference server, which is sent in turn to the device of the user

receiving the request, as reflected in Figure 7, below. Ex.1003, ¶¶122-123.

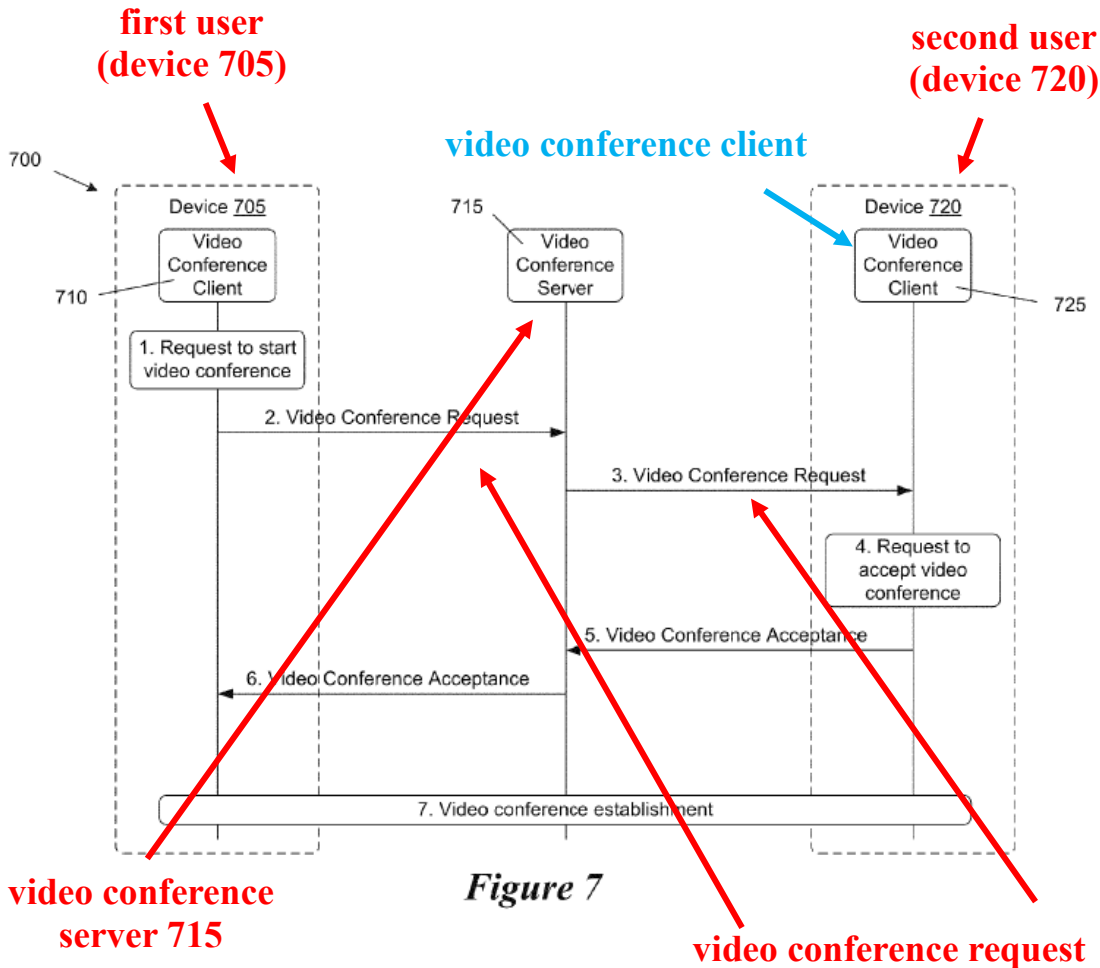


Figure 7

Ex.1005, Fig. 7.

Abuan explains that information regarding the incoming call, including the identity of the caller and options to accept or deny the video conference request, are displayed to the user when a video conference is requested. Ex.1005, [0130] (“The first stage 910 illustrates **the UI 905 when the invite recipient receives an invitation to a video conference from the invite requestor... The name field 995 displays the name of a person who is requesting a video conference... The**

message field 940 displays an invite from the invite requestor to the invite recipient.”); Ex.1003, ¶124.

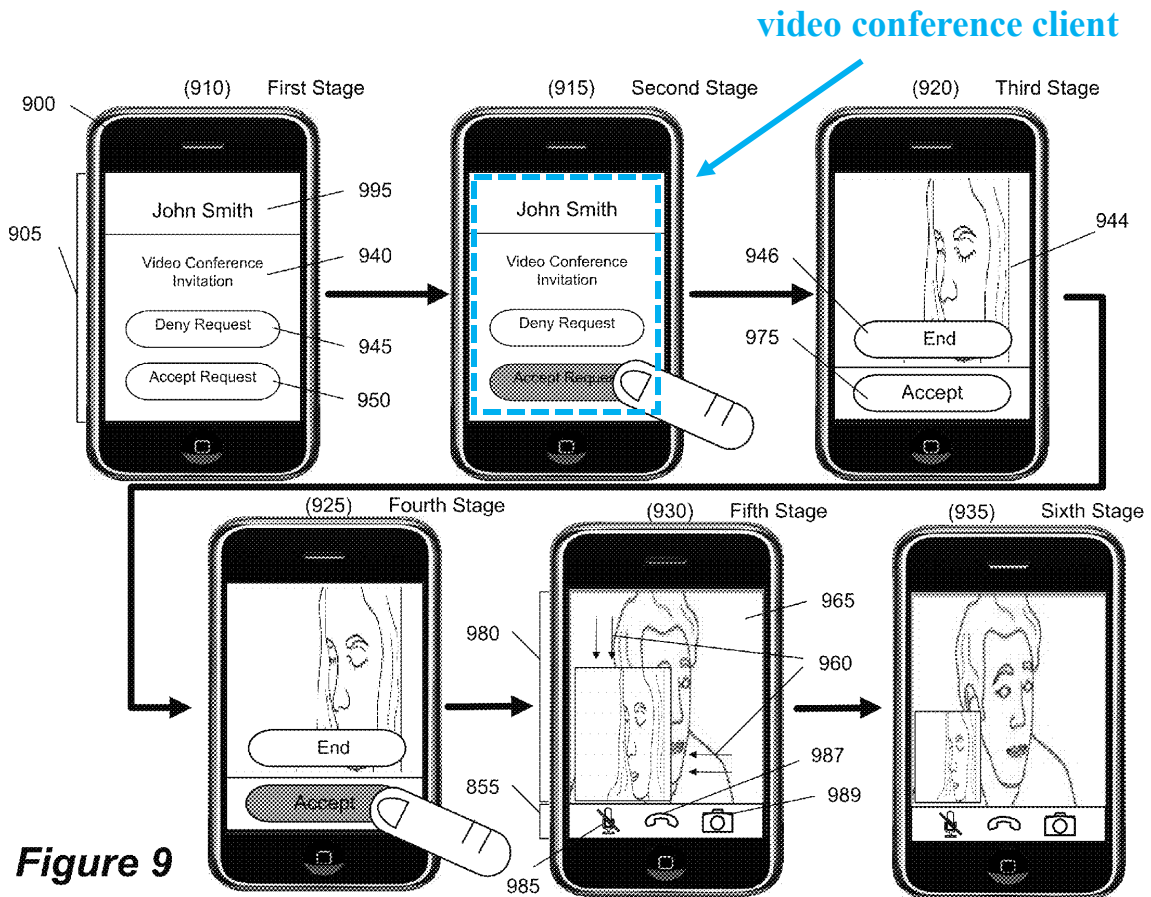


Figure 9

Ex.1005, Fig. 9.

Abuan’s video conference invitation represents the *presence* of the user sending the request. For example, by sending the request, the user demonstrates that they are both online and awaiting a response to the request. See Ex.1005, [0111]. By communicating that a video conference is being requested and

providing the identity of the user sending the video conference invitation, Abuan renders obvious that the notification transmitted from a first user, to the server, to the second user *is a presence notification*. Ex.1003, ¶¶125-126.

Thus, Abuan renders obvious this claim.

**[16.0] The method of claim 1 further comprising receiving, by the one or more servers, a notification from the function block.**

As discussed above in [11.0], Abuan teaches that as part of the establishment of real-time communication session between the users, a video conference request sent to the video conference server. Ex.1005, [0111] (“After the video conference client 710 receives the request, **the video conference client 710 sends (at operation 2) a video conference request**, which indicates the device 720 as the recipient based on input from the user, **to the video conference server 715.**”); Ex.1003, ¶¶127-129. This is reflected in operation 2 of Figure 7, below:

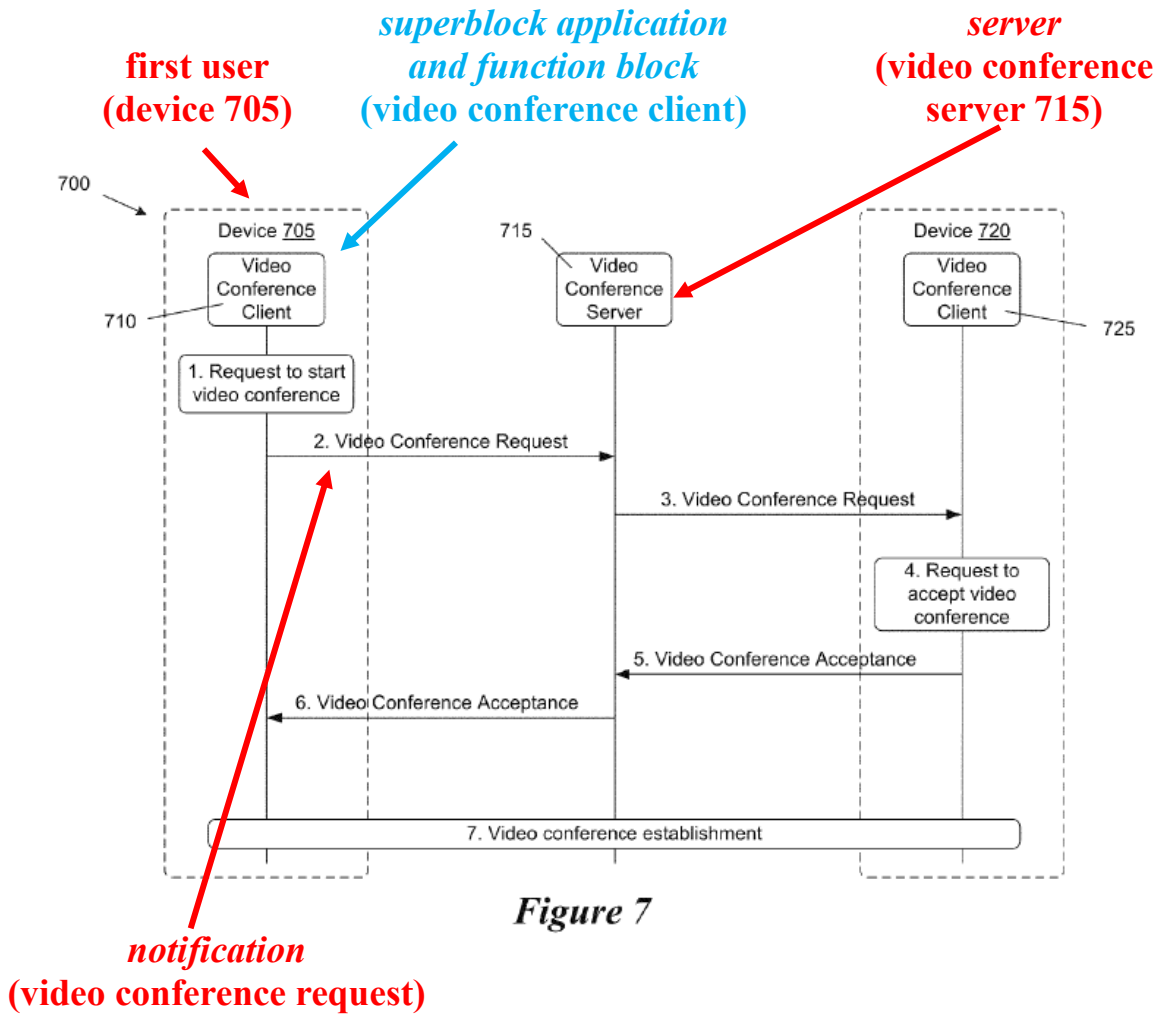


Figure 7

Ex.1005, Fig. 7.

As discussed in further detail above in [11.0], this *notification* is forwarded from the video conference server to device 720 in step 3, where the second user accepts or denies the request for a video conference. Ex.1005, [0112]. Accordingly, the video conference request sent in step 2 of Figure 7, above, is a *notification from the function block that is receiv[ed] by the one or more servers*, i.e., video conference server 715. Ex.1003, Ex.1003, ¶¶130-131.

Thus, Abuan discloses or renders obvious this claim.

**[17.0] The method of claim 1 wherein the real-time communication session is for instant messaging.**

Abuan describes how establishing a videoconference session includes establishing “a control communication channel [] for messaging between the local mobile device and a remote device.” Ex.1005, [0218]. This control communication channel is used for “sending and receiving requests, notifications, and acknowledgements,” plus “remote control instruction messages.” Ex.1005, [0218]. A POSITA would have recognized that such messages would be exchanged substantially instantly because they relate to the establishment, maintenance, and control of the real-time videoconference communication session. As confirmation, Abuan explains that the control communication channel uses “protocols like a **real-time** transport control protocol (RTCP).” Ex.1005, [0218]. Abuan’s discussion of establishing a real-time messaging channel for controlling videoconference session renders obvious that “*the real-time communication session is for instant messaging.*” Ex.1003, ¶132-133; *see also* Ex.1001, 17:34-35 (“instant messaging (IM) control module”); Material Fact #4.

Further, Abuan explains that its invention contemplates various types of *real-time communication session[s]*, including ones *for instant messaging* over an instant messaging application. Ex.1005, [0171] (“The client application 1365 may be an application that uses the video conferencing functions of the video

conference module 1302, such as...an instant messaging application.”); *see also* Ex.1005, [0270] (“**additional services can be provisioned**...such as data services like...**instant messaging**”); Ex.1003, ¶¶134-135.

Thus, Abuan renders obvious this claim.

**[18.0] The method of claim 1 wherein the real-time communication session is a voice call.**

Abuan explains that its invention contemplates various types of *real-time communication session[s]*, including a *voice call* such as over a voice-over-IP application. Ex.1005, [0171] (“The client application 1365 may be an application that uses the video conferencing functions of the video conference module 1302, such as...a voice-over-IP (VoIP) application (e.g., Skype)”); Material Fact #3. Further, as discussed above in claims 6-7, Abuan explains that video conference module 1302 (part of the *function block*) initializes modules for *voice call[s]*, including audio processing manager 1415, microphone 1425, speaker 1430, and performs “encoding [of] audio data captured by the microphone 1425 and decoding [of] audio data stored in the buffer 1410” Ex.1005, [0189]; Ex.1003, ¶¶136-139.

Thus, Abuan renders obvious this claim.

**[19.0] The method of claim 1 wherein the real-time communication session is a video call.**

Abuan explains that its invention contemplates various types of *real-time communication session[s]*, including a *video call* over a video conferencing

application. Ex.1005, [0171] (“The client application 1365 may be an application that uses the video conferencing functions of the video conference module 1302, such as a video conferencing application”); Ex.1003, ¶¶140-142.

Thus, Abuan renders obvious this claim.

**[20.0] The method of claim 1 wherein the superblock application is further adapted to operate with iOS as an operating system for the computing device.**

Abuan’s teachings support that frameworks provided by Apple, which are intended to be used on the iOS operating system, can be implemented as a media exchange module. Ex.1005, [0044] (“One example of such a media exchange module 310 is the Core Media framework provided by Apple Inc.”); Ex.1042, 3:28-33 (describing “a mobile device executing the iOS™ operating system” including a “media processing module” (framework) that “provides a low-level programming interface for managing and playing audiovisual media” such as “Core Media framework”); *see also* Ex.1043, 1; Ex.1048 (Wayback Machine capture showing “Core Media Framework Reference” from Apple developer documentation for “iOS application”); Ex.1003, ¶¶143-144. As explained in [1.1], Abuan’s video conferencing and processing module 1300 (including the client application 1365 and the various modules) is *the superblock application*, which includes media exchange module 1320. Ex.1005, [0181] (“is the same as the media exchange module 310 shown in FIG. 3, with more detail provided”). Apple’s Core Media framework is part of “a rich set of [APIs] and tools to create innovative

---

applications for iPhone and iPod® touch.” Ex.1043, 1. Using Core Media framework—provided by Apple for applications running on iOS—as the media exchange module for Abuan’s video conferencing application 1365 means the application is *adapted to operate with iOS as an operating system*, as the framework provides interfaces for “iOS application[s].” Ex.1048; Ex.1003, ¶¶144-145.

Thus, Abuan renders obvious this claim.

**[22.0] The method of claim 1 wherein the computing device is a cellular phone.**

Abuan explains that its invention contemplates a *computing device*, such as user device 705 and device 720 discussed above, including implementations where the computing device is a mobile phone (i.e., *cellular phone*). Ex.1005, [0032] (“Some embodiments of the invention provide a mobile device with two cameras that can take pictures and videos. Examples of mobile devices include mobile phones[.]”); Ex.1003, ¶¶146-148.

Thus, Abuan discloses or at least renders obvious *the computing device is a cellular phone*.

**[23.0] The method of claim 1 wherein the computing device is a smart phone.**

Abuan explains that its invention contemplates a *computing device*, such as user device 705 and device 720 discussed above, including implementations where the computing device is a *smart phone*. Ex.1005, [0032] (“Some embodiments of

the invention provide a mobile device with two cameras that can take pictures and videos. Examples of mobile devices include...smartphones[.]”). Ex.1003, ¶¶149-151.

Thus, Abuan discloses or at least renders obvious *the computing device is a smart phone*.

**[24.0] The method of claim 1 wherein the computing device is a tablet.**

Abuan explains that its invention contemplates a *computing device*, such as user device 705 and device 720 discussed above, including implementations where the computing device is a *tablet*. Ex.1005, [0032] (“Some embodiments of the invention provide a mobile device with two cameras that can take pictures and videos. Examples of mobile devices include...tablet personal computers[.]”). Ex.1003, ¶¶152-154.

Thus, Abuan discloses or at least renders obvious *the computing device is a tablet*.

**[25.0] The method of claim 1 wherein the computing device is a personal digital assistant.**

Abuan explains that its invention contemplates a *computing device*, such as user device 705 and device 720 discussed above, including implementations where the computing device is a *personal digital assistant*. Ex.1005, [0032] (“Some embodiments of the invention provide a mobile device with two cameras that can take pictures and videos. Examples of mobile devices include...personal digital

---

assistants (PDAs)[.]”). Ex.1003, ¶¶155-157.

Thus, Abuan discloses or at least renders obvious *the computing device is a personal digital assistant*.

**[26.0] The method of claim 1 wherein the computing device is a laptop computer.**

Abuan explains that its invention contemplates a *computing device*, such as user device 705 and device 720 discussed above, including implementations where the computing device is a *laptop computer*. Ex.1005, [0032] (“Some embodiments of the invention provide a mobile device with two cameras that can take pictures and videos. Examples of mobile devices include...laptops[.]”). Ex.1003, ¶¶158-160.

Thus, Abuan discloses or at least renders obvious *the computing device is a laptop computer*.

**[27.0] The method of claim 1 wherein the computing device is a desktop computer.**

While Abuan explains that its invention contemplates devices such as user device 705 and device 720 discussed above, Abuan further explains that its “embodiments are used in cases involving a video conference between a dual camera mobile device and another device, such as a single camera mobile device, a **computer**, a phone with video conference capability, etc.” Ex.1005, [0279].

Accordingly, a POSITA would have recognized that Abuan’s teachings would be

---

applicable to a *desktop computer*. Ex.1003, ¶¶161-163.

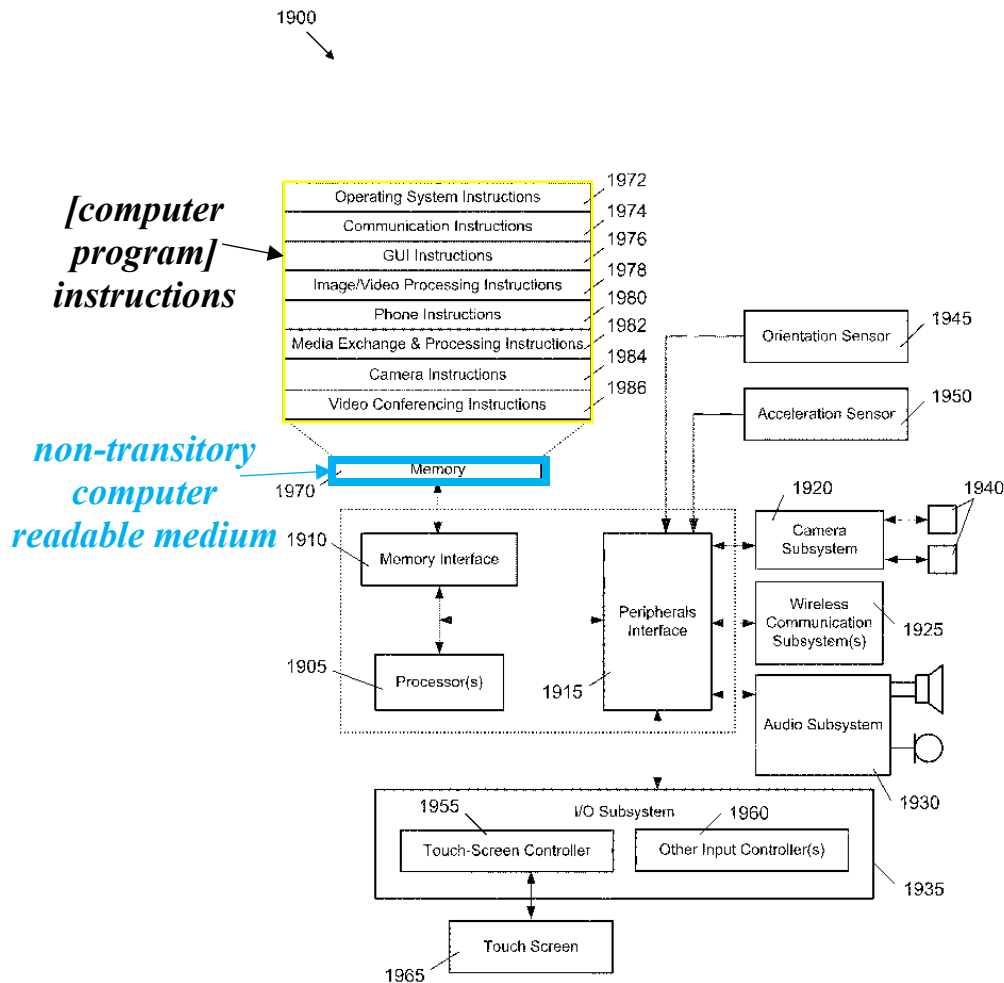
Thus, Abuan renders obvious *the computing device is a desktop computer*.

### 3. Independent Claim 55

**[55.0] A non-transitory computer readable medium embodying a computer program for providing a real-time communication session over the internet for a superblock application intended for use on a computing device, the computer program comprising instructions that include**

As discussed above in [1.0], Abuan discloses or at least renders obvious *providing a real-time communication session over the internet for a superblock application intended for use on a computing device*. Further, Abuan’s methods “are implemented as software processes.” Ex.1005, [0235]. The software comprises program code embodied as computer program instructions residing in read-only memory. Ex.1005, [0234] (“In this specification, the term ‘software’ is meant to include firmware **residing in read-only memory** or applications stored in magnetic storage which can be read into memory for processing by a processor.”); Ex.1003, ¶¶164-165.

Further, Abuan explains that the software (*computer program comprising instructions*) is stored in “a machine readable medium,” and “in a form readable by...a computer.” Abuan, [0248]; Ex.1003, ¶166.



**Ex.1005, Fig. 19 (annotated).**

Accordingly, Abuan’s “memory” storing computer program instructions for a video conference client application is a *non-transitory computer readable medium embodying a computer program... the computer program comprising instructions that perform the functionality described above in the analysis of claim 1. Ex.1003,* ¶167.

**[55.1] a function block for use in adding additional functionality to a third party superblock application that has its own functionality and display window,**

See [1.1]. Ex.1003, ¶¶168, 57-64.

**[55.2] wherein the function block is configured to be compiled into the superblock application and is configured to add the additional functionality to provide the real-time communication session using one or more servers connected over the internet, and**

See [1.2]. Ex.1003, ¶¶169, 65-69.

**[55.3] wherein the function block is configured to interact with the superblock application through one or more application programming interface (API) calls, and**

See [1.3]. Ex.1003, ¶¶170, 70-74.

**[55.4] wherein the function block is further configured to enable establishment of the real-time communication session between the one or more servers and the function block compiled into the superblock application so that the function block can provide the real-time communication session to the superblock application.**

See [1.4]. Ex.1003, ¶¶171, 75-77.

## **C. Ground 2**

### **1. Eisenberg**

Eisenberg describes a “system and method for initiating and supporting network video conferences[.]” Ex.1019, Abstract. Like Abuan, Eisenberg contemplates that “one or more instant messaging enabled clients can initiate a video conference.” Ex.1019, Abstract. Further, an intermediary server, “e.g., a video conferencing server, supports video conferences between video conference participants, a video conference being initiated between video conference participants in response to an instant message transmitted between the at least two

client nodes.” Ex.1019, Abstract. Eisenberg recognizes that “[i]nstant messaging (IM) is becoming an increasingly popular utility on networks such as the Internet.” Ex.1019, [0003]. Eisenberg details that within such services, a “presence” feature is provided “which permits IM clients to know the status of other clients on the network serviced by the messenger service.” Ex.1019, [0003]. This allows users to communicate their availability or otherwise block incoming messages when unavailable. Ex.1019, [0003]; Ex.1003, ¶¶172-173.

Eisenberg is analogous art to the ’116 patent because it is in the same field of endeavor as the ’116 patent, namely, the “manner in which functionality is accessed in certain environments, such as mobile device environments.” Ex.1001, 1:58-60; *see* Ex.1019, [0005]-[0006] (describing “the convenience of instant messaging with video conferencing capability to allow a user to initiate a video conference using IM” on “client nodes” or other types of “video conferencing system[s]”). Eisenberg and the ’116 patent also both generally relate to implementing functionality via software. *See* Ex.1001, Abstract (“providing additional functionality to existing software”); Ex.1005, [0027] (describing a device including “an IM client software module 16, a video conference initiation software module 18 and a video conferencing client software module 28.”), [0023] (“each IM client 12 includes a video conference module 18 which interfaces with both the IM client module 16 and the CTM server 20 to initiate a video

conference”). Eisenberg and the ’116 patent both describe at length features for facilitating videoconferencing. Ex.1001, Abstract, 4:43-51 (purpose of a function block is to provide functions including “audio, video,...conferencing, meetings, and/or other functions”); Ex.1019, [0005] (“a video conference is initiated between video conference participants”), [0023] (“each IM client 12 includes a video conference module 18 which interfaces with both the IM client module 16 and the CTM server 20 to initiate a video conference”). Ex.1003, ¶174.

## **2. Reasons to Combine**

A POSITA would have been motivated to include a presence indicator, as taught by Eisenberg, in Abuan’s video conference client because doing so would have been the combination of prior art elements (Abuan’s video conference client with Eisenberg’s presence features) according to known methods to yield predictable results (providing information regarding the availability of a user before placing a call to them using Abuan’s video conference client). Ex.1003, ¶175. Eisenberg explicitly teaches a presence feature for instant message clients, and more broadly contemplates a “system and method for initiating and supporting network video conferences.” Ex.1019, Abstract. And Abuan similarly teaches an instant message client, such as its video conference client, that is able to establish video conferences over a network between two devices. Ex.1005, [0109]; Ex.1003, ¶175. A POSITA would have had a reasonable expectation of success in making

such a combination because Eisenberg is directed to video conferencing, and compliments Abuan's teachings of establishing a video conference between two devices with additional information related to the users interacting with Abuan's video conference client. Ex.1003, ¶175.

### 3. Dependent claim 15

**[15.0] The method of claim 14 wherein the function block is further configured to update a presence indicator within the superblock application in response to receiving the presence notification.**

As discussed above for [1.0], Abuan discloses “an application that uses the video conferencing functions of the video conference module 1302, such as a video conferencing application, a voice-over-IP (VoIP) application (e.g., Skype), or **an instant messaging application.**” Ex.1005, [0171]. Eisenberg provides further implementation details regarding an instant messaging application, such as contemplated by Abuan. Ex.1003, ¶¶176-177. More specifically, Eisenberg explains that “[i]nstant messaging (IM) is becoming an increasingly popular utility on networks such as the Internet.” Ex.1019, [0003]. For such applications, Eisenberg notes that it is typical to include a *presence indicator* regarding whether a client is online or offline:

The messenger service also typically provides a “presence” feature which permits IM clients to know the status of other clients on the network serviced by the messenger service. The status information for a client

includes whether the client is online and available to receive instant messages from other clients and whether the client presently at his workstation, which can be determined by the level of activity at the client's workstation. The presence feature also allows clients to declare themselves unavailable to receive instant messages so that incoming messages can be blocked.

Ex.1019, [0003].

To that end, Eisenberg's teaching of providing a presence feature is advantageous for a video conference client, such as the one contemplated by Abuan, because Eisenberg explains that it allows users to communicate their availability or otherwise block incoming messages when unavailable. Ex.1019, [0003]; Ex.1003, ¶¶177-178. And as discussed above in Section XII.C.2, a POSITA would have been motivated to combine Eisenberg's teaching of a presence feature in Abuan's video conference client, and would have had a reasonable expectation of success in doing so because Eisenberg specifically contemplates such features in the context of video conferencing. Ex.1003, ¶¶178-179.

Thus, Abuan in view of Eisenberg renders obvious this claim.

**D. Ground 3**

**1. Beilis**

Beilis is related to "the field of mobile communications," and

---

telecommunications applications running “on a mobile computing platform.”

Ex.1046, [0003]. In an example, Beilis discloses an “android device,” that “run[s] a mobile platform such as the well-known android platform,” and is running a “telecommunications video application.” Ex.1046, [0024]. Beilis further explains that an application may command the telecommunications video application “to launch a video call.” Ex.1046, [0027]; Ex.1003, ¶180.

Beilis is in the same field of endeavor as the ’116 patent, namely, the “manner in which functionality is accessed in certain environments, such as mobile device environments.” Ex.1001, 1:58-60; *see* Ex.1046, [0003]-[0005] (describing the ways mobile applications run in different computing platforms, such as Android). Beilis and the ’116 patent also both generally relate to implementing functionality via software. *See* Ex.1001, Abstract (“providing additional functionality to existing software”); Ex.1046, [0011], [0013], [0024] (describing mobile device running software and software applications, such as telecommunications video applications). Beilis and the ’116 patent both describe features for facilitating videoconferencing. Ex.1001, Abstract, 4:43-51 (purpose of a function block is to provide functions including “audio, video,...conferencing, meetings, and/or other functions”); Ex.1046, [0027]-[0028], [0039] (describing running a mobile application to “launch,” “render,” “transfer,” and “resize” a video call); Ex.1003, ¶181.

Beilis is reasonably pertinent to a problem allegedly addressed by the '116 patent. The '116 patent describes purported problems associated with using multiple applications on a mobile device, potentially requiring a user to “switch[] back and forth between the video window of [a] call and [a] superblock application.” Ex.1001, 4:22-26. The '116 patent’s proposed solution is to provide functionality—such as communication functions—to a superblock application. Ex.1001, 4:43-61. Beilis similarly describes the complexities of having multiple applications running in parallel on a mobile device, where “a user must select the application and engage in manual task performance to replace the original display with new data from the other application.” Ex.1046, [0008]. Beilis integrates the capability to perform function calls while continuously occupying the full screen so that the user does not need to switch back and forth between applications. Ex.1046, [0029]; Ex.1003, ¶182.

## **2. Reasons to Combine**

A POSITA would have been motivated to configure Abuan’s stand-alone video conferencing application available on the Android platform, as taught by Beilis, because doing so would have been a predictable, routine combination of prior art element using known software development techniques to yield predictable results (Abuan’s video conferencing application configured to run on the well-established Android platform). Ex.1003, ¶183. A POSITA would have

---

appreciated that configuring Abuan’s application to run on the Android platform is an implementation detail that would not change the underlying videoconferencing functionalities disclosed by Abuan. Ex.1003, ¶183. A POSITA would have had reasonable expectation of success in the combination because Beilis shows that video communications applications were available on, and operable within, the Android platform. Ex.1003, ¶183.

### 3. Dependent claim 21

**[21.0] The method of claim 1 wherein the superblock application is further adapted to operate with Android as an operating system for the computing device.**

Abuan alone or in view of Beilis renders obvious this claim.

As discussed above, Abuan contemplates that its video conference client is “an application that may use the video conferencing functions” such as Skype. Ex.1005, [0105], [0171]. It was well-known that video conferencing applications, including Skype itself, were also available on the Android operating system. Ex.1044. And while Abuan describes frameworks provided by Apple Inc. as examples of its modules, Abuan does not limit itself to iOS as the only operating system it contemplates. Ex.1005, [0044]; *see also* Ex.1007, [0048]; Ex.1003, ¶¶184-185.

Beilis provides additional implementation details for operating mobile applications on different devices and platforms, including “android device[s]”

running the “android platform.” Ex.1046, [0016]; Ex.1003, ¶186. Beilis explains that an android device may run a “telecommunications video application” on the android platform (“*Android as an operating system*”), which application is used to “launch” and “render” a video call. Ex.1046, [0024], [0027], [0029]. Ex.1003, ¶186. Thus, Beilis’s telecommunications application is “*adapted to operate with Android as an operating system for the computing device.*”

A POSITA would have been motivated to *adapt* Abuan’s video conference client (“superblock application”) to run on the “Android operating system,” as taught by Beilis, because doing so would have been a predictable, routine engineering choice aimed at making the same mobile video-conferencing functionality available to the large and well-established Android ecosystem, without changing the underlying conferencing concepts disclosed by Abuan. Ex.1003, ¶187. Beilis shows that video communications applications were available on, and operable within, the Android platform, and that launching and conducting video calls through an Android application was a known and achievable implementation detail. Ex.1003, ¶¶187-188.

## **E. Ground 4**

### **1. Guzman**

Guzman describes a video-conferencing application that facilitates account setup, authentication, contact management, and call initiation for video

---

conferencing sessions. Ex.1007, Abstract, [0002]–[0006], [0033], [0037], [0049]–[0051]. Guzman’s application and its interface allows a user to sign in to a service (for instance, via a remote server) using login credentials to gain access to the user’s contacts, and thereby gain the ability to initiate video conference calls to those contacts. Ex.1007, [0044]–[0046], [0060]–[0063], [0074]–[0075], [0058], [0070]–[0071]; Ex.1003, ¶¶189-190.

Guzman’s video conferencing application includes an “auto-login” option that allows the application to automatically provide stored login credentials to the authenticating system without the user having to input those credentials for every video conferencing session. Ex.1007, [0056]–[0059]; Ex.1003, ¶191.

## **2. Reasons to Combine**

A person of ordinary skill in the art would have seen clear, practical reasons to combine Abuan’s videoconferencing framework with Guzman’s login and auto-login mechanisms. Ex.1003, ¶192.

Abuan recognizes that a video call can only proceed with the recipient’s permission, as reflected in its discussion of converting a phone call into a video conference “with the permission of the other party.” Abuan, [0117]. Abuan also shows, in Figure 7, that these video sessions depend on an intermediary device—a videoconference server—to set up, manage, and carry the session. In practice, that server is a shared, finite resource. A skilled engineer would thus appreciate that the

server's participation is not just functional but also conditional, in the sense that the server must decide whether to allow or deny additional sessions based on access control and capacity. Ex.1003, ¶193.

Without some gatekeeping, the system would be exposed to misuse and overload. A system that allows any client to start a session at any time without authentication increases the risk that the server will be overwhelmed, whether by intentional attacks or simply by volume spikes and abusive patterns. This is a textbook problem in networked services: denial-of-service conditions occur when requests flood a server, consuming bandwidth or compute such that legitimate users are deprived of service. Material Fact #1. Engineers routinely mitigate these risks by requiring authenticated access and by tying service initiation to verified user identity. Ex.1003, ¶194.

Guzman addresses precisely this issue by a login process through which a user supplies a username and password to access a service such as videoconferencing. Ex.1007, [0056]. Guzman further proposes an auto-login procedure to more conveniently allow a user who has already authenticated successfully to bypass repeated logins on subsequent uses. Ex.1007, [0056]. That approach is a well-known way to preserve access control while keeping the user experience efficient. Ex.1003, ¶195.

A skilled practitioner would find it natural and straightforward to incorporate Guzman's login and optional auto-login into Abuan's clients and server. Abuan already relies on a centralized server to broker and maintain sessions, and the addition of a credential check at the server boundary is a standard software pattern. The combination does not require novel algorithms or unconventional architectures; it calls for routine programming work within ordinary skill to integrate authentication workflows, maintain session state, and, where desired, persist tokens or credentials to enable auto-login. The expected outcome is predictable: restricting access to videoconference services to authenticated users, thereby preventing unauthorized use and reducing the risk of server overload from anonymous or automated traffic. Ex.1003, ¶196.

Put differently, applying Guzman's login technique to Abuan's videoconferencing system is the application of a known access-control mechanism to another known server-mediated communication service that would have benefited from, and expected, an authorization mechanism. The resulting benefits—controlling who can initiate or join a session, improving system stability under load, and aligning with standard security practices—are entirely foreseeable. The same logic holds for both the client and server components in Abuan: implementing login at the client side to capture credentials and at the server side to verify and authorize access is the ordinary way similar systems are improved—

namely, by limiting functionality to legitimate, logged-in users in order to protect shared server resources while maintaining a smooth user experience via auto-login.

Ex.1003, ¶197.

### 3. Dependent Claims 9-10

**[9.0] The method of claim 1 further comprising providing an authorization key for use by the superblock application, wherein the authorization key unlocks the function block for use by the superblock application.**

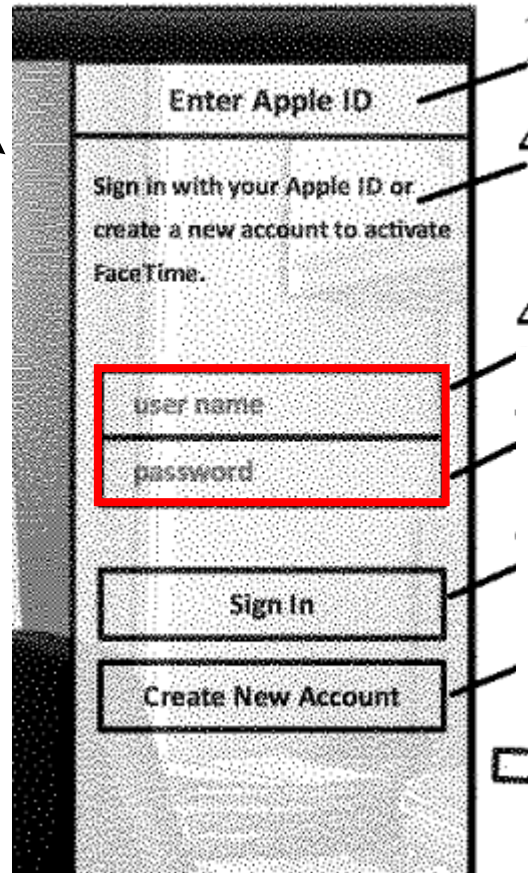
Abuan's Figure 7 assumes that the video conference client 710 is authorized to use the video conference module 1302 to communicate with the video conference server 715 and the user of device 705 is already logged into the video conference server 715. It would have been obvious to a POSITA, however, for the video conference server to require users to login to use the server's videoconferencing capabilities, because requiring a user to login was nearly universally employed by servers at the time. Material Fact #2. Consistent with that understanding and with Abuan's general architecture, Guzman describes a videoconferencing application for providing videoconference communications between two or more clients. Ex.1007, Abstract, [0034]. Ex.1003, ¶¶198-199.

Guzman discloses a "video conferencing application [] for conducting a video conference between [a] first electronic device and a second electronic device." Ex.1007, Abstract. Like Abuan's video conference module 1302 that provides video conferencing functions to the client application 1365 on a mobile

device, Guzman's video conferencing application "enables a user of [an electronic] device to engage in a video conference with a remote user of a second electronic device that also includes a video conferencing application." Ex.1005, [0171]; Ex.1007, [0034], [0221] (describing application as a "client"); Ex.1003, ¶200.

Guzman describes a user inputting a "username" and "password" (*authorization credentials*) into an "overlay display area" of a "video conferencing application." Ex.1007, [0050]. In one example, Guzman describes a log-in procedure wherein the application receives, as a user input from UI (part of the *superblock application*) a username and password. Ex.1007, [0074], Fig. 3. With respect to Fig. 4 below, Guzman shows the overlay display area UI with two "entry fields" for "user name" and "password." Ex.1003, ¶201.

*“superblock application”*



**UI fields for application  
to receive  
“authorization  
credentials”**

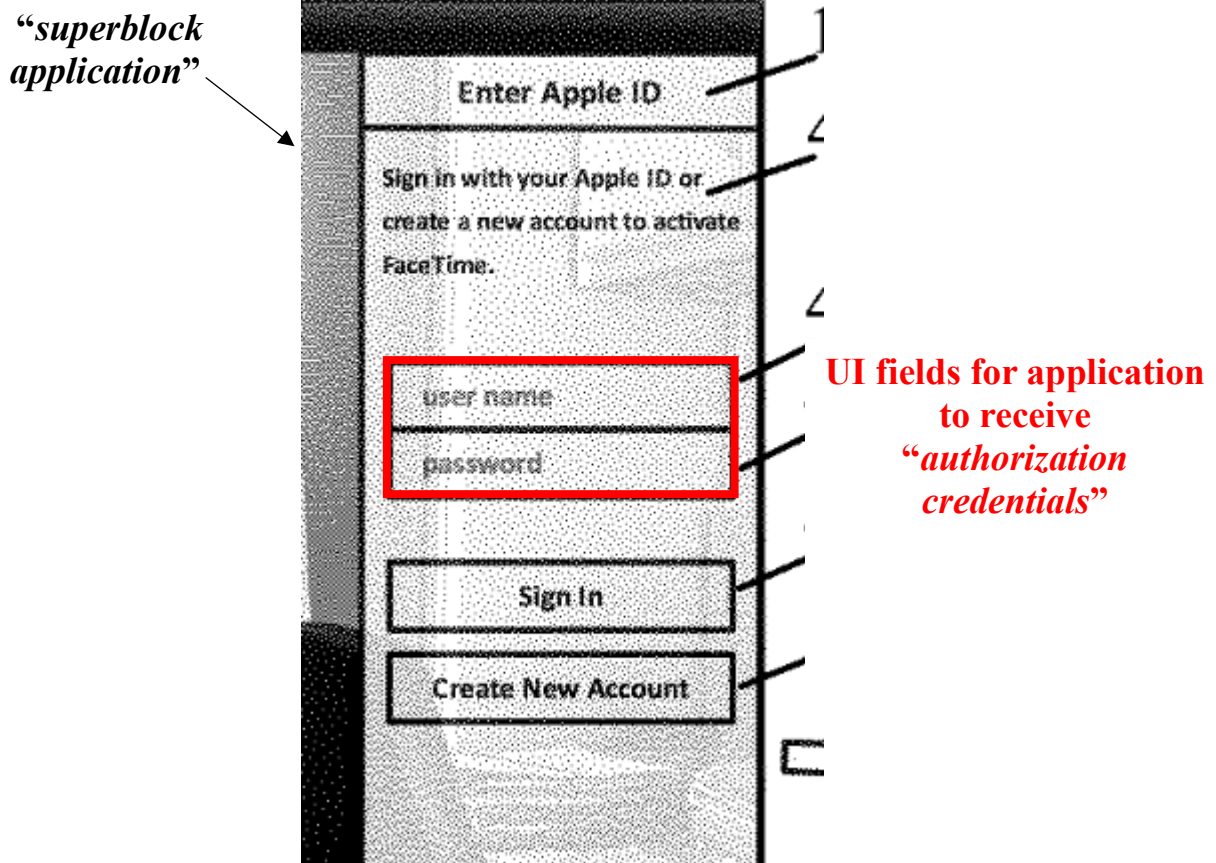
**Ex.1007, Fig. 4 (partial annotated)**

“[W]hen the user selects the “Sign In” item 465, the video conferencing application verifies whether the login information entered by the user is valid.” Ex.1007, [0077]. Guzman describes different ways to verify and authenticate the user’s credentials. First, the credentials can be verified by the video conferencing application itself. Ex.1007, [0057]. Second, the credentials can be verified “by accessing a remote server that stores login information.” Ex.1007, [0057]; Ex.1003, ¶202.

Thus, Guzman’s disclosure of verifying login credentials by the video conferencing application itself discloses “*providing an authorization key for use by the superblock application.*” Further, the user interface interaction module 2205 (*function block*) receives the authorization credentials from the UI of the video conferencing application and facilitates login. When the credentials have been verified, the user’s contacts are loaded and made available so that the user can initiate a video conference (*the authorization key unlocks the function block for use by the superblock application*). Ex.1007, [0058], Fig. 4; Ex.1003, ¶203.

**[10.0] The method of claim 1 further comprising providing authentication credentials for use by the superblock application, wherein the authentication credentials enable the function block to use the one or more servers for the real-time communication session.**

As discussed above in [9.0] and reflected in Figure 4, below, Guzman describes a user inputting a “username” and “password” (*authorization credentials*) into an “overlay display area” of a “video conferencing application.” Ex.1007, [0050]; Ex.1003, ¶¶204-205.



Ex.1007, Fig. 4 (partial annotated)

“[W]hen the user selects the “Sign In” item 465, the video conferencing application verifies whether the login information entered by the user is valid.” Ex.1007, [0077]. Guzman describes different ways to verify and authenticate the user’s credentials. First, the credentials can be verified by the video conferencing application itself. Ex.1007, [0057]. Second, the credentials can be verified “by accessing a remote server that stores login information.” Ex.1007, [0057]; Ex.1003, ¶206.

For instance, Guzman’s video conferencing application 2200 includes “an account setup and verification module 2210” (a “*function block*”) that facilitates a user logging into the videoconferencing server. Ex.1007, [0222], [0230]. In one example, a “user inputs an account name and password which is **passed by the user interface interaction module 2205 to the account setup and verification module 2210.**” Ex.1007, [0230]; Ex.1003, ¶207.

Thus, Guzman’s disclosure of verifying login credentials by accessing a remote server discloses “*providing authorization credentials for use by the superblock application.*” Further, the user interface interaction module 2205 (*function block*) receives the authorization credentials from the UI of the video conferencing application and facilitate login with the videoconferencing server (*the authentication credentials enable the function block to use the one or more servers for the real-time communication session*). When the credentials have been verified, the user’s contacts are loaded and made available so that the user can initiate a video conference. Ex.1007, [0058], Fig. 4; Ex.1003, ¶208.

#### **4. Independent Claim 28 and Dependent Claims 29-41, 43-47, and 49-54**

Independent claim 28 is substantially similar to independent claim 1, with limitations [28.0] through [28.3] being word-for-word identical to limitations [1.0] through [1.3], respectively. Limitation [28.4] is substantially similar to claim [10.0] and is addressed below. Dependent claim 29 is substantially similar to

limitation [1.4], except it does not recite the language “*compiled into the superblock application*” found in limitation [1.4]. Dependent claims 30-32 are substantially similar to claims 2-4, respectively, claim 36 is substantially similar to claim 5, claims 33-35 and 37 are substantially similar to claims 6-9, claims 38-41 are substantially similar to claims 11-14, claims 43-47 are substantially similar to claims 16-20, and claims 49-54 are substantially similar to claims 22-27.

Accordingly, claims 28-41, 43-47, and 49-54 obvious for the same reasons as discussed above. *See* Ex.1047 (comparing claims 1-27 with claims 28-54 in redline); Ex.1003, ¶209.

**[28.4] providing authentication credentials for use by the superblock application, wherein the authentication credentials enable the function block to use the one or more servers to establish the real-time communication session for the superblock application.**

Limitation [28.4] is substantially similar to claim 10. As discussed above at [10.0], Abuan and Guzman render obvious “*providing authentication credentials for use by the superblock application, wherein the authentication credentials enable the function block to use the one or more servers.*” Ex.1003, ¶214.

Limitation [28.4] further recites “*to establish the real-time communication session for the superblock application.*” As discussed for [1.4], Abuan explains, with reference to Figure 7 above below, that a “video conference server 715 forwards (at operation 3) [a] video conference request [from device 705] to the video conference client 725 of the device 720” to establish a video conference

between device 705 and device 720 (*establish the real-time communication session*). Ex.1005, [0110]-[0111]. This communication between the video conference client 725—operating on device 720—and the video conference server 715, is performed by the video conference module 1302 on that device, such that *the function block [] use[s] the one or more servers* via communications over the network. More specifically, Abuan discloses that “[e]ach of the video conference clients 710 and 725...communicates with the video conference server 715 over a network (e.g., a cellular network, a local area network, a wireless network, a network of networks, the Internet etc.) through a network interface such as the network interface 650.” Ex.1005, [0109]. Abuan explains that the networking manager 1314 of the video conference module 1302 (or network layer 640 of the video conference module 625) “establishes the connections between the dual camera mobile device and the other device of the video conference at the start of the video conference.” Ex.1005, [0179]; *see also* [0104]; Ex.1005, [0156] (“A more detailed version of this video conference module [625] will be described below by reference to FIG. 13.”), [0171] (“the client application 1365 is the same as the video conference client 645 of FIG. 6.”). In this way, the video conferencing module 1302 (*the function block*) establishes the video conference session (*establish the real-time communication session*) between the video conference

client of device 705 (*the superbloc application*) and device 720 using video conference server 715 (*use the one or more servers*). Ex.1003, ¶215.

## **F. Ground 5**

### **1. Dependent Claim 42**

Claim 42 depends, indirectly, from claim 28, which is obvious over Abuan and Guzman as discussed above in Ground 4. Claim 42 further recites the same limitation recited in claim 15, shown to be obvious over Abuan above in Ground 1. Thus, claim 42 would have been obvious for the same reasons discussed above for claims 28 and 15. *See* Ex.1047 (comparing claims 1-27 with claims 28-54 in redline); Ex.1003, ¶240.

## **G. Ground 6**

### **1. Dependent Claim 48**

Claim 48 depends from claim 28, which is obvious over Abuan and Guzman as discussed above in Ground 4. Claim 48 further recites the same limitation recited in claim 21, shown to be obvious over Abuan above in Ground 1. Thus, claim 48 would have been obvious for the same reasons discussed above for claims 28 and 21. *See* Ex.1047 (comparing claims 1-27 with claims 28-54 in redline); Ex.1003, ¶242.

## **XIII. Conclusion**

Petitioner has established a reasonable likelihood that the Challenged Claims

are unpatentable.

Respectfully submitted,

Dated: January 13, 2026  
HAYNES AND BOONE, LLP  
2801 N. Harwood St., Suite 2300  
Dallas, Texas 75201  
Customer No. 27683

/Theodore M. Foster/  
Theodore M. Foster  
Lead Counsel for Petitioner  
Registration No. 57,456

#### **XIV. Mandatory notices**

##### **A. Real party-in-interest**

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner certifies that the real party-in-interest is Cisco Systems, Inc.

##### **B. Related matters**

Pursuant to 37 C.F.R. § 42.8(b)(2), to the best knowledge of the Petitioner, the '116 Patent is or was involved in the following cases:

<b>Case Heading</b>	<b>Number</b>	<b>Court</b>	<b>Filed</b>
<i>Damaka, Inc. v Cisco Systems, Inc.</i>	2-25-cv-00593	EDTX	May 30, 2025
<i>Damaka, Inc. v. The Cigna Group</i>	2:25-cv-00594	EDTX	May 30, 2025

##### **C. Lead and back-up counsel and service information**

###### Lead Counsel

Theodore M. Foster  
HAYNES AND BOONE, LLP  
2801 Harwood St. Suite 2300  
Dallas, TX 75201

Phone: (303) 382-6205  
Fax: (214) 200-0853  
ipr.theo.foster@haynesboone.com  
USPTO Reg. No. 57,456

###### Back-up Counsel

Calmann J. Clements  
HAYNES AND BOONE, LLP  
2801 Harwood St. Suite 2300  
Dallas, TX 75201

Phone: (972) 739-8638  
Fax: (214) 200-0853  
calmann.clements.ipr@haynesboone.com  
USPTO Reg. No. 66,910

David L. McCombs  
HAYNES AND BOONE, LLP  
2801 Harwood St. Suite 2300  
Dallas, TX 75201

Phone: (214) 651-5533  
Fax: (214) 200-0853  
david.mccombs.ipr@haynesboone.com  
USPTO Reg. No. 32,271

**Back-up Counsel**

Brett Bostrom  
HAYNES AND BOONE, LLP  
2801 N. Harwood St., Suite 2300  
Dallas, TX 75201

Phone: (720) 484-3715  
Fax: (214) 200-0853  
brett.bostrom.ipr@hayneboone.com  
USPTO Reg. No. 73,009

Dagim Tilahun  
HAYNES AND BOONE, LLP  
2801 Harwood St. Suite 2300  
Dallas, TX 75201

Phone: (202) 654-4582  
Fax: (214) 200-0853  
dagim.tilahun.ipr@haynesboone.com  
USPTO Reg. No. 75,722

Matthew Beck  
HAYNES AND BOONE, LLP  
2801 Harwood St. Suite 2300  
Dallas, TX 75201

Phone: (202) 654-4573  
Fax: (214) 200-0853  
matthew.beck.ipr@haynesboone.com  
USPTO Reg. No. 79,791

Please address all correspondence in this proceeding to lead and back-up counsel. Petitioner consents to service in this proceeding by email at the addresses above on condition that Patent Owner likewise consents to email service.

**XV. Claims Appendix**

- [1.0] A method for providing a real-time communication session over the internet for a superblock application intended for use on a computing device, the method comprising:
- [1.1] providing a function block for use in adding additional functionality to a third party superblock application that has its own functionality and display window,
- [1.2] wherein the function block is configured to be compiled into the superblock application and is configured to add the additional functionality to provide the real-time communication session using one or more servers connected over the internet, and
- [1.3] wherein the function block is configured to interact with the superblock application through one or more application programming interface (API) calls; and
- [1.4] enabling establishment of the real-time communication session between the one or more servers and the function block compiled into the superblock application so that the function block can provide the real-time communication session to the superblock application.
- [2.0] The method of claim 1 wherein enabling the establishment of the real-time communication session includes signaling communications between the

one or more servers and the function block.

[3.0] The method of claim 2 wherein the signaling communications include session setup, management, and teardown.

[4.0] The method of claim 2 wherein the signaling communications use Session Initiation Protocol (SIP) as a signaling protocol.

[5.0] The method of claim 1 wherein the real-time communication session uses Real-time Transport Protocol (RTP) as a data transport protocol.

[6.0] The method of claim 1 wherein enabling the establishment of the real-time communication session includes negotiating signaling and media parameters between the one or more servers and the function block.

[7.0] The method of claim 6 wherein the signaling and media parameters include a bandwidth parameter.

[8.0] The method of claim 6 wherein the signaling and media parameters include a codec parameter.

[9.0] The method of claim 1 further comprising providing an authorization key for use by the superblock application, wherein the authorization key unlocks the function block for use by the superblock application.

[10.0] The method of claim 1 further comprising providing authentication credentials for use by the superblock application, wherein the authentication credentials enable the function block to use the one or more

servers for the real-time communication session.

[11.0] The method of claim 1 further comprising sending a notification from the one or more servers to the function block.

[12.0] The method of claim 11 wherein the notification requires a response and the establishment of the real-time communication session is the response to the notification.

[13.0] The method of claim 11 wherein the notification is a request to initiate the real-time communication session, and an API call indicates whether the request has been granted or denied by the superblock application.

[14.0] The method of claim 11 wherein the notification is a presence notification.

[15.0] The method of claim 14 wherein the function block is further configured to update a presence indicator within the superblock application in response to receiving the presence notification.

[16.0] The method of claim 1 further comprising receiving, by the one or more servers, a notification from the function block.

[17.0] The method of claim 1 wherein the real-time communication session is for instant messaging.

[18.0] The method of claim 1 wherein the real-time communication session is a voice call.

[19.0] The method of claim 1 wherein the real-time communication session is a

video call.

[20.0] The method of claim 1 wherein the superblock application is further adapted to operate with iOS as an operating system for the computing device.

[21.0] The method of claim 1 wherein the superblock application is further adapted to operate with Android as an operating system for the computing device.

[22.0] The method of claim 1 wherein the computing device is a cellular phone.

[23.0] The method of claim 1 wherein the computing device is a smart phone.

[24.0] The method of claim 1 wherein the computing device is a tablet.

[25.0] The method of claim 1 wherein the computing device is a personal digital assistant.

[26.0] The method of claim 1 wherein the computing device is a laptop computer.

[27.0] The method of claim 1 wherein the computing device is a desktop computer.

[28.0] A method for providing a real-time communication session over the internet for a superblock application intended for use on a computing device, the method comprising:

[28.1] providing a function block for use in adding additional functionality to a third party superblock application that has its own functionality and display window,

[28.2] wherein the function block is configured to be compiled into the superblock application and is configured to add the additional functionality to provide

the real-time communication session using one or more servers connected over the internet, and

[28.3] wherein the function block is configured to interact with the superblock application through one or more application programming interface (API) calls; and providing authentication credentials for use by the superblock application,

[28.4] wherein the authentication credentials enable the function block to use the one or more servers to establish the real-time communication session for the superblock application.

[29.0] The method of claim 28 further comprising enabling the establishment of the real-time communication session between the one or more servers and the function block so that the function block can provide the real-time communication session to the superblock application.

[30.0] The method of claim 29 wherein enabling the establishment of the real-time communication session includes signaling communications between the one or more servers and the function block.

[31.0] The method of claim 30 wherein the signaling communications include session setup, management, and teardown.

[32.0] The method of claim 30 wherein the signaling communications use Session Initiation Protocol (SIP) as a signaling protocol.

- [33.0] The method of claim 29 wherein enabling the establishment of the real-time communication session includes negotiating signaling and media parameters between the one or more servers and the function block.
- [34.0] The method of claim 33 wherein the signaling and media parameters include a bandwidth parameter.
- [35.0] The method of claim 33 wherein the signaling and media parameters include a codec parameter.
- [36.0] The method of claim 28 wherein the real-time communication session uses Real-time Transport Protocol (RTP) as a data transport protocol.
- [37.0] The method of claim 28 further comprising providing an authorization key for use by the superblock application, wherein the authorization key unlocks the function block for use by the superblock application.
- [38.0] The method of claim 28 further comprising sending a notification from the one or more servers to the function block.
- [39.0] The method of claim 38 wherein the notification requires a response and the establishment of the real-time communication session is the response to the notification.
- [40.0] The method of claim 38 wherein the notification is a request to initiate the real-time communication session, and an API call indicates whether the request has been granted or denied by the superblock application.

[41.0] The method of claim 38 wherein the notification is a presence notification.

[42.0] The method of claim 41 wherein the function block is further configured to update a presence indicator within the superblock application in response to receiving the presence notification.

[43.0] The method of claim 28 further comprising receiving, by the one or more servers, a notification from the function block.

[44.0] The method of claim 28 wherein the real-time communication session is for instant messaging.

[45.0] The method of claim 28 wherein the real-time communication session is a voice call.

[46.0] The method of claim 28 wherein the real-time communication session is a video call.

[47.0] The method of claim 28 wherein the superblock application is further adapted to operate with iOS as an operating system for the computing device.

[48.0] The method of claim 28 wherein the superblock application is further adapted to operate with Android as an operating system for the computing device.

[49.0] The method of claim 28 wherein the computing device is a cellular phone.

[50.0] The method of claim 28 wherein the computing device is a smart phone.

- [51.0] The method of claim 28 wherein the computing device is a tablet.
- [52.0] The method of claim 28 wherein the computing device is a personal digital assistant.
- [53.0] The method of claim 28 wherein the computing device is a laptop computer.
- [54.0] The method of claim 28 wherein the computing device is a desktop computer.
- [55.0] A non-transitory computer readable medium embodying a computer program for providing a real-time communication session over the internet for a superblock application intended for use on a computing device, the computer program comprising instructions that include
- [55.1] a function block for use in adding additional functionality to a third party superblock application that has its own functionality and display window,
- [55.2] wherein the function block is configured to be compiled into the superblock application and is configured to add the additional functionality to provide the real-time communication session using one or more servers connected over the internet, and
- [55.3] wherein the function block is configured to interact with the superblock application through one or more application programming interface (API) calls, and
- [55.4] wherein the function block is further configured to enable establishment of

the real-time communication session between the one or more servers and the function block compiled into the superblock application so that the function block can provide the real-time communication session to the superblock application.

**CERTIFICATE OF WORD COUNT**

Pursuant to 37 C.F.R. § 42.24(d), Petitioner hereby certifies, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 12,684. Pursuant to 37 C.F.R. § 42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under § 42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: January 13, 2026

/Theodore M. Foster/  
Theodore M. Foster  
Lead Counsel for Petitioner  
Registration No. 57,456

**CERTIFICATE OF SERVICE**

Pursuant to 37 C.F.R. § 42.6(e), the undersigned certifies that a true and correct copy of the foregoing was served on the Patent Owner as detailed below:

<i>Date of service</i>	January 13, 2026
<i>Manner of service</i>	FedEx
<i>Documents served</i>	<b>PETITION FOR <i>INTER PARTES</i> REVIEW UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104; PETITIONER'S POWER OF ATTORNEY; PETITIONER'S EXHIBIT LIST AND EXHIBITS Ex.1001-1048</b>
<i>Persons Served</i>	Munck Wilson Mandala L.L.P 2000 McKinney Ave Ste 1900 Dallas, TX 75201

/Theodore M. Foster/  
Theodore M. Foster  
Lead Counsel for Petitioner  
Registration No. 57,456