UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

CISCO SYSTEMS INC.,

Petitioner

_____

IPR2023-00553
U.S. Patent No. 9,210,254

_____

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

TABLE OF CONTENTS

I. INTRODUCTION 4
II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE 7
III. LEVEL OF ORDINARY SKILL IN THE ART 12
IV. RELEVANT LEGAL STANDARDS 13
V. BACKGROUND 14
A. Associating Contact Numbers and Addresses 14
B. Communication Application Login Procedures 17
VI. OVERVIEW OF THE '254 PATENT 18
VII. CLAIM CONSTRUCTION 21
VIII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE 21
IX. THE COMBINATION OF BODART, BANISTER, AND DHARMARAJAN RENDERS OBVIOUS CLAIMS 1, 2, 6-8, AND 12-14 22
A. Summary of Bodart 22
B. Summary of Banister 25
C. Summary of Dharmarajan 26
D. Reasons to Combine Bodart with Banister and Dharmarajan 28
1. Associating a telephone number with an email address 29
2. Associating a telephone number with an IM username 33
3. Logging into an IM service 36
E. Detailed Analysis of Claims 40
1. Claim 1 40
2. Claim 2 70
3. Claim 6 73
4. Claim 7 75
5. Claim 8 76
6. Claim 12 77

7. Claim 13 77
8. Claim 14 80
X. DE CASTRO RENDERS OBVIOUS CLAIMS 1, 2, 6-8, AND 12-14 81
A. Summary of de Castro 81
B. Detailed Analysis of Claims 85
1. Claim 1 85
2. Claim 2 113
3. Claim 6 117
4. Claim 7 118
5. Claim 8 120
6. Claim 12 120
7. Claim 13 121
8. Claim 14 123
XI. CONCLUSION 123

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

I. INTRODUCTION

1. I am making this declaration at the request of Cisco Systems, Inc. in the matter of the *Inter Partes* Review of U.S. Patent No. 9,210,254 ("the '254 patent") to Hunt et al.

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

3. I have been asked to provide my opinions regarding whether the subject matter of claims 1, 2, 6-8, and 12-14 ("the Challenged Claims") of the '254 patent would have been obvious to a person having ordinary skill in the art ("POSITA") at the time of the alleged invention, in light of the prior art. It is my opinion that the Challenged Claims would have been obvious to a POSITA.

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

a. the '254 patent, Ex.1001.

b. the prosecution history of the '254 patent ("'254 File History"), Ex. 1002.

c. U.S. Pat. Pub. No. 2007/0043687 to Bodart et al. ("Bodart"), Ex.1005.

d. U.S. Pat. No. 6,621,892 to Banister et al. ("Banister"), Ex.1006.

e. U.S. Pat. No. 7,089,585 to Dharmarajan ("Dharmarajan"), Ex.1007.

f. U.S. Pat. Pub. No. 2012/0016932 to de Castro et al. ("de Castro"), Ex.1008.

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

a. Ben Langfeld, Jose de Castro, "XEP-0327: Rayo," XMPP Standards Foundation (archived August 18, 2013) ("the Rayo Specification"), Ex.1009.

b. U.S. Pat. No. 6,298,128 to Ramey et al. ("Ramey"), Ex.1010.

c. John Letzing "Facebook Raises Hackles on Email," Wall Street Journal Online, June 26, 2012 (archived June 30, 2012), ("Letzing") Ex.1011.

d. Shelley O'Hara, "Easy Microsoft® Office Outlook® 2003," (Que Publishing, September 2003) ("O'Hara"), Ex.1012.

e. U.S. Pat. No. 8,130,751 to Berkley et al. ("Berkley"), Ex.1013.

f. U.S. Pat. No. 7,929,519 to Goldman et al. ("Goldman"), Ex.1014.

g. U.S. Pat. Pub. No. 2007/0143231 to Pegaz-Paquet et al. ("Pegaz-Paquet"), Ex.1015.

h. U.S. Pat. No. 1,276,825 to Swope ("Swope"), Ex.1016.

i. U.S. Pat. No. 2,308,778 to Prince ("Prince"), Ex.1017.

j. U.S. Patent No. 8,090,787 to Polis et al. ("Polis"), Ex.1022.

6. I have also considered the following documents as demonstrating the prior art status of the non-patent literature documents listed above, including the Rayo Specification, Letzing, and O'Hara:

a. Affidavit of Authenticity of "XEP-0327: Rayo," confirming the prior art status of the Rayo Specification, Ex.1018.

b. Amazon.com listing for "Easy Microsoft® Office Outlook® 2003" (archived February 1, 2008) ("Amazon"), confirming the prior art status of O'Hara, Ex.1019.

c. Affidavit of Authenticity of for Letzing and Amazon, confirming the prior art status of Letzing and O'Hara, Ex.1020.

d. Declaration of Jose de Castro under 37 C.F.R. § 1.68, confirming the prior art status of the Rayo Specification, Ex.1021.

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

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

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

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

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

11. I first entered telecommunications in 1987 when I worked as a summer intern at AT&T Bell Laboratories as part of a five-year dual degree program at MIT. I continued to work at AT&T Bell Laboratories as part of this MIT program. While at MIT, I was a teaching assistant ("TA") in the Electrical

Engineering and Computer Science Department's core Computer Architectures course. I first was a TA as a senior for a role typically reserved for graduate students. I later became head TA. The course covered various topics in computer architectures. As a TA, I helped write homework assignments, lab assignments, and exams. I also taught in the recitation sections.

12. Later, as part of my doctoral research at MIT from 1991-1998, I was a research assistant in the Telemedia Network Systems ("TNS") group at the Laboratory for Computer Science. The TNS group built a high-speed gigabit network and created applications that ran over the network. Example applications included ones for remote video capture, processing, and display of video on computer terminals. In addition to working on the design of core network components, designing and building the high-speed links, and designing and writing the device drivers for the interface cards, I also set up the group's web server.

13. I also helped to build the web pages that initiated the above-mentioned video sessions via a web interface. Vice President Al Gore visited our group in 1996 and received a demonstration of—and remotely drove—a radio-controlled toy car with a wireless video camera mounted on it that was built by our group. This toy car device received commands transmitted over a network from a remote computer, and video data from the toy car was transmitted wirelessly then

over a computer network back to the user controller. On occasion, we allowed users visiting our web site to drive the toy car from their remote computer while they watched the video on their computer. The video stream was encoded by TNS-designed hardware, streamed over the TNS-designed network, and displayed using TNS-designed software.

14. I defended and submitted my Ph.D. thesis, titled "Designing Networks for Tomorrow's Traffic," in January 1998. As part of my thesis research, I analyzed local area and wide area flows to show a more efficient method for routing packets in a network, based on traffic patterns at the time.

15. From 1997 to 1999, I was a Senior Scientist and Engineer at NBX Corporation, a start-up that made business telephone systems for streaming packetized audio over data networks instead of using traditional telephone lines. NBX was later acquired by 3Com Corporation and the phone system is still used today by numerous businesses.

16. As part of my work at NBX, I designed the core audio reconstruction algorithms for the telephones, as well as the packet transmission algorithms. I also designed and validated the core packet transport protocol used by the phone system. The protocol was used for all signaling in the phone system, including for the setup of conference calls.

17. The NBX system also featured a computer interface for initiating phone calls, which could also initiate conference calls. The NBX system also supported the Telephony Application Programming Interface ("TAPI") that allowed other computer programs to integrate with our system telephony features. We obtained U.S. Patent No. 6,697,963, entitled "Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data," as part of this work.

18. From 1999-2004, I was employed by Empirix or its predecessor company, Teradyne. Empirix was a leader in test tools for telecommunications protocols and systems, providing functional testing tools as well as load testing tools. From 2000-2001, I conceived and built a test platform for testing Voice-over-IP (VoIP). The first application on this new test platform was a cloud emulator for simulating the effects of transmitting VoIP over a busy network.

19. In 2006, as part of my role at BBN Technologies, I helped found PodZinger Inc., now known as RAMP Inc. PodZinger utilized BBN's speech recognition algorithms to search through the spoken words in audio and video segments. While I was Vice President of Operations and Technology, PodZinger followed its initial prototype with a full streaming audio and video search solution. I also created a social networking web site, which BBN sold to a venture-funded

startup company. In the process of creating the web site, I designed and specified the authentication and authorization protocols.

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

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

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

III. LEVEL OF ORDINARY SKILL IN THE ART

21. I understand there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including (1) the levels of education and experience of persons working in the field at the time of the invention; (2) the sophistication of the technology; (3) the types of problems encountered in the field; and (4) the prior art solutions to those problems.

22. A Person of Ordinary Skill in The Art ("POSITA") in October 2013 would have had a working knowledge of the telecommunications and networking art that is pertinent to the '254 patent. A POSITA would have had a bachelor's degree in electrical engineering, computer science, computer engineering, or an

equivalent, and two years of relevant experience in telecommunications and networking. Lack of professional experience can be remedied by additional education, and vice versa.

23. For purposes of this Declaration, in general, and unless otherwise noted, my statements and opinions, such as those regarding my own experience and what a POSITA would have understood or known generally (and specifically related to the references I consulted herein), reflect the knowledge that existed in the relevant field as of the priority date of the '254 patent.

IV. RELEVANT LEGAL STANDARDS

24. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the '254 patent, I am relying on certain basic legal principles that Cisco's counsel has explained to me.

25. I understand that prior art to the '254 patent includes patents and printed publications in the relevant art that predate the priority date of the '254 patent. For purposes of this Declaration, I am applying October 9, 2013, as the priority date of the '254 patent.

26. I have been informed by Cisco's counsel that a claimed invention is unpatentable under 35 U.S.C. § 103 if the differences between the claimed invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA. I have also been

informed by Cisco's counsel that the obviousness analysis considers factual inquiries, including the level of ordinary skill in the art, the scope and content of the prior art, and the differences between the prior art and the claimed subject matter.

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

V. BACKGROUND

A. Associating Contact Numbers and Addresses

28. Storing telephone numbers and other contact addresses, such as email addresses and instant messaging (IM) usernames, is a well-known concept in

telecommunications applications. Communications applications are typically installed on a user's computer to provide the user access to one or more communications services, such as email or IM. One example of a communications application is an email application which may include an email client for accessing an email server. A user's email address, credentials, the server and protocol to be used are configured within the application to customize the access to email for a particular user.

29. Communications applications often provide a database or digital rolodex of contact information that can be managed and modified by the user. *See, e.g.*, O'Hara, pp. 112-118 (describing how contact information is stored using Microsoft Outlook 2003). This allows the user of the application to input and store the usernames, addresses, telephone numbers, website URLs, physical addresses, and any other addressing mechanisms for each individual in their personal or professional network. The addresses are typically stored as a database and in such a way that each address or contact number is associated with its corresponding individual. In the database, the user may store their own contact numbers and addresses as well as those of their associates.

30. Databases that are used to store contact information are commonly structured as relational databases. A relational database is typically embodied in a set of tables where each row in each table is associated with a record, such as for

an individual, designated by their name or some other unique identifier. A computer language such as Structured Query Language (SQL) may be used for accessing a relational database. However, the communications software, such as an email client, provides a more user-friendly visual interface for the user to search or browse the stored contact information in the database. The columns or entries in a row in a table may include the contact numbers and addresses, including email addresses, IM addresses, or physical addresses associated with a particular user or individual. A contact database would typically indicate a communication application or platform for each type of address in the database. For example, the each entry in an IM account table would indicate IM addresses and the associated IM platform, such as Microsoft Network (MSN) Messenger, since users may have multiple IM accounts on different platforms and different usernames on each. The structure of the relational database facilitates simple associations of its entries to an individual, such as their telephone number and associated email address. *See, e.g.*, Berkley, 6:55 – 7:3 (describing active user registry (AUR) databases); Goldman, 11:8-17; (AUR database stores telephone numbers, fax numbers, mobile telephone numbers, e-mail addresses, etc.); Pegaz-Paquet, [0116] (a reverse address match database for locating an address associated with an originating telephone number).

B. Communication Application Login Procedures

31. Access to communications services such as email or IM typically involves gaining authenticated or secure access to a remote server over the internet. Software installed on the user's computer typically provides a user-friendly interface to login to the remote server by providing security credentials, such as a username and password. In some cases, such as with email applications, the user also provides server settings or configurations, such as the server protocol, and the name or address of the server providing the service. *See* O'Hara, pp. 104-105; Polis, 5:27-44. For email service, the server protocol may be post office protocol (POP), such as POP3, internet message access protocol (IMAP), or Microsoft Exchange, for example. *See* O'Hara, pp. 104-105; Polis, 5:27-44. For IM, the server protocol may be Extensible Messaging and Presence Protocol (XMPP). The user's computer is commonly referred to as the client computer, and the software that accesses the remote server is called a software client. If the security credentials are accepted by the remote server, the server may provide an acknowledgement to the client computer to allow the client computer to access the user's messages and data stored on the server.

32. The acknowledgement may be provided with a digital token, or by establishing an authenticated session. *See, e.g.*, Dharmarajan, 5:46-54. A security token is an encrypted file that has a secret key. An authenticated session also

involves the server providing a file to the client computer. The session file specifies a time period or timing parameters for the authenticated access.

33. All of the features and functionality described in this section were well known in the early 2010s, and a POSITA would have been familiar with them.

VI. OVERVIEW OF THE '254 PATENT

34. The '254 patent "relates generally to integrating communications services, and in particular to managing services and applications using telephone numbers as common subscriber identifiers." '254 patent, 1:15-19. The '254 patent describes provisioning telecommunications services from service providers with different ecosystems using a unified software platform. '254 patent, Abstract, 3:55-58, 11:37-42. The '254 patent explains that a variety of different applications may be added on request:

> [C]lient 110 may request to add services and applications to the existing telephone service provided by provider 120. Any of various service and applications may be requested, including video conferencing, text messaging, interactive texting, hosted private branch exchange (PBX) seats, voice mail systems, and other services and applications.

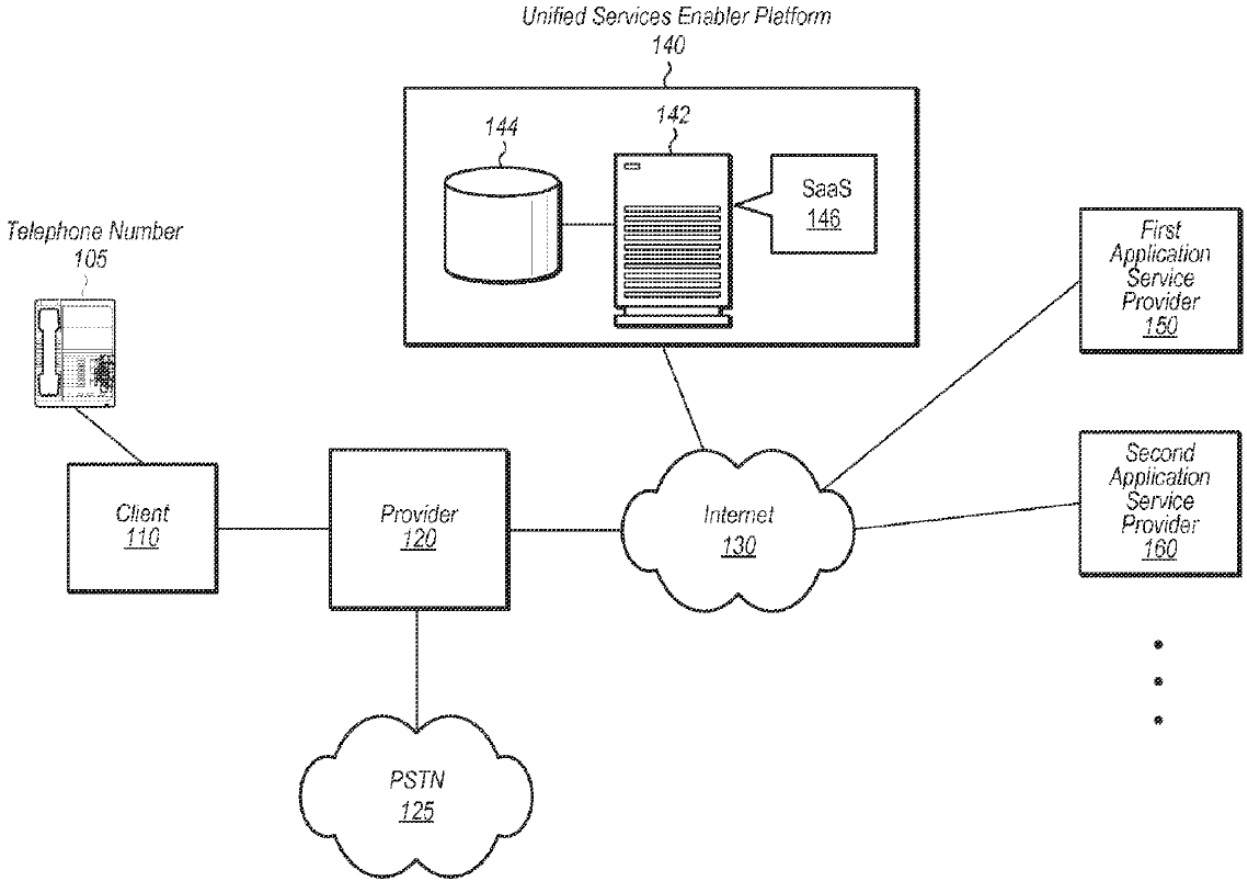
'254 patent, 3:53-58.

35. The '254 patent explains that for "large numbers of customers … the telephone number remains the universal identifier (ID) for communicating with the outside world." '254 patent, 1:20-26. "However, many services and applications

are supplied by third-party providers in different ecosystems. These third-party ecosystems often use **different types of IDs** and have different rules and policies for managing and activating their services and applications." '254 patent, 1:29-33. The '254 patent offers allegedly "new techniques for integrating advanced services and applications with legacy telephone service." '254 patent, 1:34-36.

36. To integrate third-party applications with telephone service, the '254 patent proposes "using the telephone number as the common subscriber ID." '254 patent, 1:40-42. The '254 patent further explains that "the telephone number may be used as the central, unifying ID to associate services and applications that are utilized by the corresponding user." '254 patent, 1:46-48. When a customer adds a service, the customer's "telephone number may be translated to a second ID that is utilized as a login ID or user name within the ecosystem of the second [service] provider." '254 patent, 8:47-49. This allegedly provides "a unified services enabler platform" that "may allow for the association of internet protocol (IP) enabled services with telephone numbers tied to legacy telephone services." '254 patent, 1:43-46. The customer's telephone number is "associated with the relevant interworking data which links all of the IDs and third-party applications and services associated with the telephone number." '254 patent, 12:53-58.

37. Fig.1 below illustrates the unified services enabler platform 140 connected by internet 130 to client users 110, telephone service provider 120, and application service providers 150, 160. '254 patent, Fig.1, 3:35-37, 4:14-28.



'254 patent, Fig.1.

38. As explained in more detail below, these concepts were well known as of the priority date of the '254 patent. In particular, it was well-known in the prior art to integrate and provision services from different ecosystems for a single user. It was also well-known in the prior art to associate a user's telephone number with

a user ID corresponding to a different service provider to provision a service. *See, e.g.*, Banister, 2:1-5; de Castro, [0139]-[0141].

VII. CLAIM CONSTRUCTION

39. It is my understanding that in order to properly evaluate the '254 patent, the terms of the claims must first be interpreted. It is my understanding that for the purposes of this *inter partes* review, the claims are to be construed under the so-called *Phillips* standard, under which claim terms are given their ordinary and customary meaning as would have been understood by a POSITA in light of the specification and prosecution history, unless the inventor has set forth a special meaning for a term. I have also been informed that claim terms only need to be construed to the extent necessary to resolve the obviousness inquiry. I have reviewed the entirety of the '254 patent, as well as its prosecution history. It is my understanding that no terms in the Challenged Claims require specific construction.

VIII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

40. The discussion in this Declaration provides a detailed analysis of how the asserted prior art references teach each limitation of the Challenged Claims.

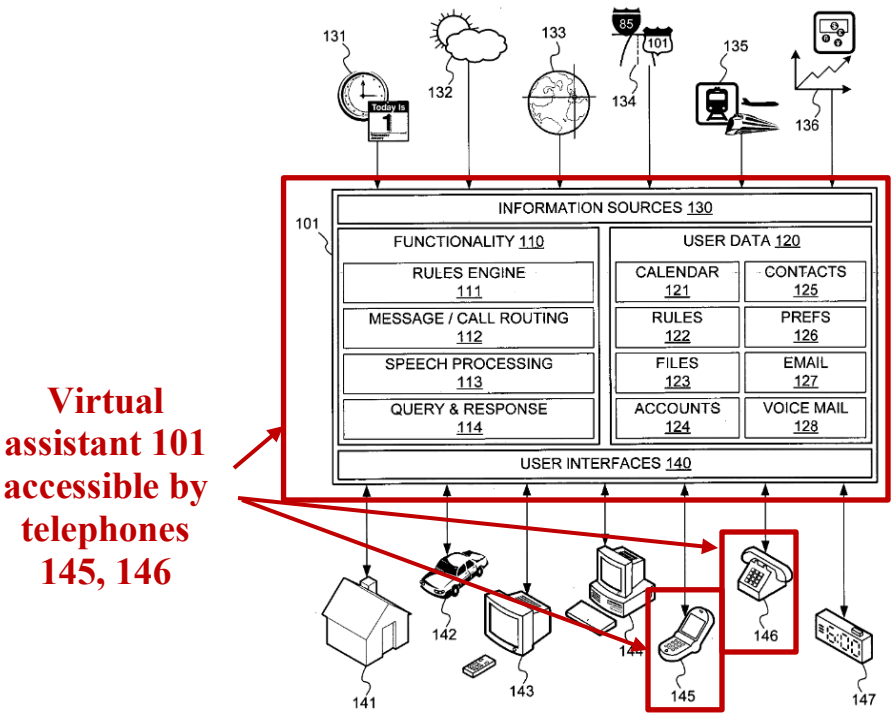
41. As part of my analysis, I have considered, and discuss in detail, the scope and content of the prior art and any differences between the alleged invention and the prior art.

42. It is my opinion that the alleged invention recited in the Challenged Claims would have been obvious in view of the teachings of the asserted prior art and the knowledge of a POSITA.

IX. THE COMBINATION OF BODART, BANISTER, AND DHARMARAJAN RENDERS OBVIOUS CLAIMS THE CHALLENGED CLAIMS

A. Summary of Bodart

43. Bodart relates to "personal productivity systems" that provide for "the integration of personal and public information to enable more productive individuals by automating routine tasks." Bodart, [0001]. Like the '254 patent, Bodart's systems provide users access to internet-based communications services such as voicemail, text messaging, and voice attendants. *See* '254 patent, 4:14-15, 24-28, 11:37-40; Bodart, Abstract, [0022], [0025], [0026], [0044], [0064], [0072]. Bodart teaches a virtual assistant having "access to a user's contacts, calendar, and location," as well as "information about the weather, traffic, and mass transit...." Bodart, Abstract. Access to this information "augment [the virtual assistant's] ability to make decisions on the user's behalf." Bodart, [0020]. The virtual assistant includes "one or more user interfaces 140 which allow a user to interact with virtual assistant" including "cell phone 145" and "conventional telephone 146." Bodart, [0030]. A schematic representation of Bodart's system is illustrated in Fig.1, reproduced in annotated form below:



Bodart, Fig. 1 (annotated)

44. Bodart explains that "a remote user can call into his or her virtual assistant and provide voice commands." Bodart, [0026]. Further, "[v]irtual assistant 101 may include the ability to receive and direct calls and messages directed to user 201." Bodart, [0044]. The functionalities that the virtual assistant 101 provides to users include routing incoming communications (Bodart, [0025], [0044]), accessing information sources (Bodart, [0029], [0041]), setting calendar appointments (Bodart, [0027], [0045]), setting custom alarms (Bodart, [0038], [0045]), retrieving contact information (Bodart, [0027], [0045]), and online banking (Bodart, [0022], [0042]).

45. For electronic communications, Bodart explains that "incoming messages including instant messages and/or SMS messages, and even email messages, may [] be routed using message routing 214." Bodart, [0044]. Bodart explains that software applications may be used or incorporated with the virtual assistant to provide user interfaces or otherwise facilitate some of its functionalities, such as email and instant messaging (IM). For example for IM services, Bodart explains that "a user may be able to send messages using an instant message client such as America Online (AOL) Instant Messenger, Microsoft Network (MSN) Messenger, and so forth." Bodart, [0034]. For email services, contact information, and calendar appointments, a "personal information manager (PIM)" such as Microsoft Outlook "may form the basis of user data to be used by virtual assistant." Bodart, [0040]. The PIM data may be accessed through a single application such as Outlook, or may "be managed by individual applications (e.g., iCal and Mail on an Apple Computer)." Bodart, [0040].

46. Some of the functionalities contemplated by Bodart for the virtual assistant include "having a user's email read to her by phone" (Bodart, [0002]) notifying a user by IM about an incoming phone call (Bodart, [0060]), and routing or forwarding incoming calls to the user based on the user's location (Bodart, [0059]), the identity of the caller (Bodart, [0057]), or the user's calendar information and scheduled appointments ([0060]).

47. Bodart is relevant to the problems described by the '254 patent, that is, to integrate and provision communications services from different providers. *See e.g.*, '254 patent, 11:33-42; Bodart, [0005], [0030]. Analogous to the '254 patent, Bodart describes a system "for receiving data from a plurality of information providers" and making the information sources and functionalities accessible by telephone. Bodart, [0020], [0030]. Both Bodart and the '254 patent provide users access to internet-based communications services such as voicemail, text messaging, and voice attendants. '254 patent, 4:14-15, 42-25, 11:37-40; Bodart, Abstract, [0022], [0025], [0026], [0044], [0064], [0072].

B. Summary of Banister

48. Banister "relates to the fields of computer systems and communications." Banister, 1:8-9; *see* '254 patent, 1:15-19, 6:42-44. Banister teaches a method for "converting an electronic mail message to audio for delivery via a telephone call." Banister, 1:8-10. Banister's method includes "retrieving a telephone number associated with an intended recipient of the message, ... establishing a telephonic connection with the recipient and playing an audio form of the message text." Banister, 1:51-57.

49. Banister explains how a mail server maps an email address to the user's telephone number:

> the mail server may maintain **a list or database of addresses with associated telephone numbers**, in which case a message sent to such

> an address will be converted to audio and forwarded to the specified number.

Banister, 2:1-5.

50. In one example, Banister explains that "an address specified by an originator of a message may be converted to a telephone number or used to look-up a telephone number when the message is received at electronic mail server 104." Banister, 4:14-17. Banister also explains that the mail server may apply a set of rules "to determine if the prefix [of the email address to which the message is directed] includes or is mapped to a telephone number to which an audio form of the message can or should be delivered." Banister, 7:9-11.

51. Banister is from the same field of endeavor as the '254 patent, namely integrating communications services. '254 patent, 1:16-17. Banister is also relevant to the problems described by the '254 patent, that is, to associate an electronic ID with a telephone number to provision a communication service to the user. *See e.g.*, '254 patent, 11:33-42; Banister, 2:1-5, 4:14-17. Like the '254 patent, Banister describes mapping an email address to a telephone number. *See* '254 patent, 7:51-55, 9:55-58; Banister, Abstract, 2:1-5.

C. Summary of Dharmarajan

52. Dharmarajan relates to "a method and system for authorizing a client computer to access a server computer system," such as an instant messaging computer system. Dharmarajan, 1:7-11. Dharmarajan describes providing "instant

messaging capabilities to a user" with an "instant messaging client application program 12 communicat[ing] through the Internet 8 with an instant message server application 4." Dharmarajan, 4:45-52.

53. Dharmarajan also explains, with respect to Fig.1 (reproduced below), a technique with which a client computer gains access to instant messaging service:

> According to an actual embodiment of the present invention, the client computer 10 executes an instant messaging client application program 12. The instant messaging client application program 12 communicates through the Internet 8 with an instant messaging server application 4 executing on the instant messaging server computer 2. Together, the instant messaging client application program 12 and the instant messaging server application program 4 provide instant messaging capabilities to a user of the client computer 10.

Dharmarajan, 4:43-52

54. Dharmarajan explains that "when a user launches the instant messaging client, the user may be required to provide a username and password." Dharmarajan, 1:33-35. Dharmarajan also describes a secure login procedure facilitated by an instant messaging client application program:

> The secure login is typically initiated by a user of the instant messaging client application program. The login procedure may include the transmission of a username and password from the client computer to the instant messaging server computer in a secure fashion.

Dharmarajan, 9:47-51.

55. Dharmarajan's disclosure of secure login procedures are pertinent to the '254 patent. For example, Dharmarajan provides details for allowing a user to request access to third-party communication services associated with a user ID. '254 patent, 13:53-64; Dharamarajan, 2:19-23. Similar to the disclosure of Dharmarajan, the embodiments of the '254 patent describe using "a login ID or user name" to activate services. '254 patent, 8:44-49, 8:65-9:5. Further, both the '254 patent and Dharmarajan describe authenticating customers for communication services using an email address. *See* '254 patent, 13:44-66; Dharmarajan, 5:39-49.

D. Reasons to Combine Bodart with Banister and Dharmarajan

56. A POSITA would have combined Bodart with Banister and Dharmarajan because each of the references describes providing users access to telephone-based and internet-based communications services such as IM and email. *See* Bodart, Abstract, [0025], [0064], [0072]; Banister, 1:8-11; Dharmarajan, 4:43-55. A POSITA would have been motivated to combine the references for at least the reasons discussed below and would have had a reasonable expectation of success in the combination. One possible combination of Bodart and Banister, as described below, would configure the virtual assistant with a table or set of tables in the user data module. This table or set of tables would associate the user's telephone number with the user's electronic addresses. The

combination of all three references—Bodard, Banister, and Dharmarajan—would, as also described below in more detail, further configure the virtual assistant to login to an IM service or email service using login credential, such as a username and/or password, stored in the user data module. A POSITA would have had the necessary knowledge to implement this combination and would have had a reasonable expectation of success. A POSITA would also have understood that other implementations of the combination of Bodard, Banister, and Dharmarajan are possible and that the specific details of such implementation would be left to a POSITA's design choice.

1. Associating a telephone number with an email address

57. Bodart describes various functionalities that involve a combination of telephone-based communications service (e.g., telephone calls) and other electronic communications services (*e.g.*, IM, email), such as "having a user's email read to her by phone." Bodart, [0002]. Bodart describes that the virtual assistant "**intercept[s]** incoming telephone calls, instant messages, and other real time **communication requests**." Bodart, [0025]. Bodart explains that the virtual assistant can access the user's emails based on a spoken request from the user:

> Using a keyboard **or speech processing**, a user may query virtual assistant with regard to various aspects of user data 120 and/or information sources 130.... Similar queries may search user files 123, **emails** 127 ... and so forth.

Bodart, [0027].

58. Moreover, Bodart explains that the spoken commands can be provided by calling into the virtual assistant:

> Using this functionality, a remote user can **call into** his or her virtual assistant and **provide voice commands**. These can be interpreted into commands which control the functioning of virtual assistant 101.

Bodart, [0026].

59. Thus, Bodart contemplates various examples of giving the user access to her emails and email functionalities via telephone calls. However, Bodart otherwise provides limited implementation details about how the virtual assistant "intercepts" and reads emails to the user by telephone, for example.

60. Banister provides details on how an electronic mail server converts an email into audio and delivers the audio to the email recipient by telephone. *See* Banister, Abstract, 3:35-50, 4:14-17. Banister explains the email recipient's "address specified by an originator of a message may be converted to a telephone number or used to look-up [the recipient's] telephone number when the message is received at electronic mail server." Banister, 4:14-17. Banister explains how a mail server maps an email address to the user's telephone number:

> the mail server may maintain **a list or database of addresses with associated telephone numbers**, in which case a message sent to such an address will be converted to audio and forwarded to the specified number.

Banister, 2:1-5.

61. In one example, Banister explains that "an address specified by an originator of a message may be converted to a telephone number or **used to look-up a telephone number** when the message is received at electronic mail server 104." Banister, 4:14-17.

62. A POSITA would have found it obvious to use the teachings of Banister to implement Bodart's email-reading functionality. Banister simply provides implementation details that Bodart expects. Bodart, [0002]; Banister, Abstract 3:35-50; For instance, Banister explains that delivering email via telephone involves looking up a telephone number from an email address. Banister, 2:1-5. Banister also describes a mail server determining whether "the prefix [of a received email message] includes or **is mapped to a telephone number** to which an audio form of the message can or should be delivered." Banister, 7:8-10. Bodart describes storing, in a user data module, "user accounts, including email passwords, website passwords, and so forth." Bodart, [0022]. Bodart also teaches storing "contact email addresses and phone numbers." Bodart, [0021]. In view of Banister's disclosure of maintaining a list or database of email addresses with associated telephone numbers and looking up a telephone number from an email address, it would have been obvious to a POSITA for Bodart's user data module to

maintain a list or database indicating an association between the user's email address and telephone number. Banister, 2:1-5, 4:14-17.

63. The combination of Bodart and Banister would be simply combining prior art elements according to known methods to yield predictable results. The known methods would involve programming the virtual assistant to create a table or set of tables in the user data module where the entries in a row include email addresses and credentials associated with a telephone number, and the specific email server and protocol to be used for each email address. The software change would also include relating the email address entry to a particular user and their telephone number. *See* Section V.A. The result would be a database comprising a table or set of tables associating email addresses with telephone numbers. The results of the combination would have been predictable because both Bodart and Banister describe reading emails to a user by phone, and because Banister merely provides implementation details that Bodart expects. *See* Bodart, [0002]; Banister, Abstract.

64. Furthermore, storing account information including electronic addresses was well-known in the art, and adding relationships between items in a relational database was also well-known in the art. *See* Section V.A; Berkley, 6:55 – 7:3; Goldman, 11:8-17; Pegaz-Paquet, [0116]. Therefore, the proposed combination would have been predictable and a POSITA would have a reasonable

expectation of success, because the combination of Bodart's email functionality with Banister's implementation details would have been well-within the skill level of a POSITA. For example, a POSITA would have been capable of implementing software changes to enable Bodart's virtual assistant to maintain a database including a telephone number with an associated email address in the user data module. *See* Banister, 2:1-5, 59-64; Berkley, 6:55 – 7:3; Goldman, 11:8-17; Pegaz-Paquet, [0116]. This confirms the predictability and reasonable expectation of success of the combination.

2. Associating a telephone number with an IM username

65. Bodart describes various functionalities that involve IM service. Bodart, [0025], [0059], [0060]. For instance, Bodart discloses that the virtual assistant "intercept[s] incoming telephone calls, **instant messages**, and other real time communication requests." Bodart, [0025]. Further, "**instant messages** ... may [] be routed using message routing 214." Bodart, [0044]. In one example, Bodart explains that, in response to an incoming call, the "virtual assistant 101 may put contact 215 on hold and attempt to reach user 201 via ... **instant message**." Bodart, [0060].

66. Thus, Bodart contemplates giving the user access to functionalities that involve a combination of telephone service (receiving an incoming phone call) and IM service. Bodart, [0060]. However, Bodart otherwise provides limited

implementation details about how the virtual assistant reaches the user by IM in response to the incoming phone call.

67. Banister describes techniques for looking up telephone numbers associated with electronic addresses. Banister, 3:35-50. Bodart teaches that "[an] electronic mail server is configured to receive a mail message and parse an address to retrieve a telephone number included in or associated with said address." Banister, Abstract. To facilitate this telephone number look-up:

> "the mail server may maintain a list or **database of addresses with associated telephone numbers**, in which case a message sent to such an address will be converted to audio and forwarded to the specified number."

Banister, 2:1-5.

68. Further, "the prefix [of an email address] may be used to look up a telephone number to which the message is to be sent." Banister, 6:57-59. Banister also describes applying "a set of rules ... to determine if the prefix [of an email address] includes or is **mapped to a telephone number** to which an audio form of the message can or should be delivered." Banister, 7:8-10.

69. Thus, Banister provides implementation details for associating a recipient's phone number with an electronic address, and looking up the phone number based on the electronic address indicated in the message. Banister, 2:1-5, 7:8-10. These are implementation details which Bodart expects. For instance,

Bodart describes storing, in a user data module, "**user accounts**, including email passwords, website passwords, and so forth." Bodart, [0022]. Bodart also describes storing "contact email addresses and phone numbers." Bodart, [0021].

70. Although Banister relates specifically to email messages and email addresses, a POSITA would have found it obvious to apply Banister's teachings of storing and mapping electronic addresses to the other types of electronic communications contemplated by Bodart, including IMs. This would facilitate the virtual assistant's functionalities in which an incoming phone call triggers the virtual assistant to send an IM. Thus, in view of Banister's disclosure of storing and mapping electronic addresses to associated telephone numbers, it would have been obvious to a POSITA for Bodart's user data module to store an association between the user's IM username and telephone number.

71. Using the implementation details provided by Banister with Bodart's virtual assistant would involve simple computer programming changes that are well-within the skill set of a POSITA. The computer programming changes would involve creating a stored association between at least one IM username and a corresponding telephone number in Bodart's user data module. *See* Bodart, [0022]. The software change would involve creating a table or set of tables in the user data module where the entries in a row include a username to be associated with a telephone number. The table or set of tables would also indicate the specific

platform or server to be used for each username (e.g., IM application username, email address). *See* Section V.A. This software change would have been predictable given that both Bodart and Banister describe storing contact addresses and telephone numbers. *See* Bodart, [0002]; Banister, Abstract. Furthermore, Bodart specifically describes storing, in the user data module 120 "contact information 125, the equivalent of a software rolodex, where users can store contact email addresses and phone numbers, along with other information." Bodart, [0021]. Similar to Bodart's disclosure of storing contact information for the user's contacts, it would have been obvious to a POSITA to store the user's own contact information, including the user's telephone number, email address, IM username, and any other addresses for the user.

3. Logging into an IM service

72. Bodart describes various IM functionalities of the virtual assistant, including a "message & call routing" functionality that "enable[s] virtual assistant 101 to intercept incoming telephone calls, **instant messages**, and other real time communication requests." Bodart, [0025]. Bodart's virtual assistant can provide automated control over IMs and other incoming communications:

> The virtual assistant also has a rules engine enabling a user to create **rules for handling** incoming calls and **instant messages**, rerouting calls based on their caller identification.

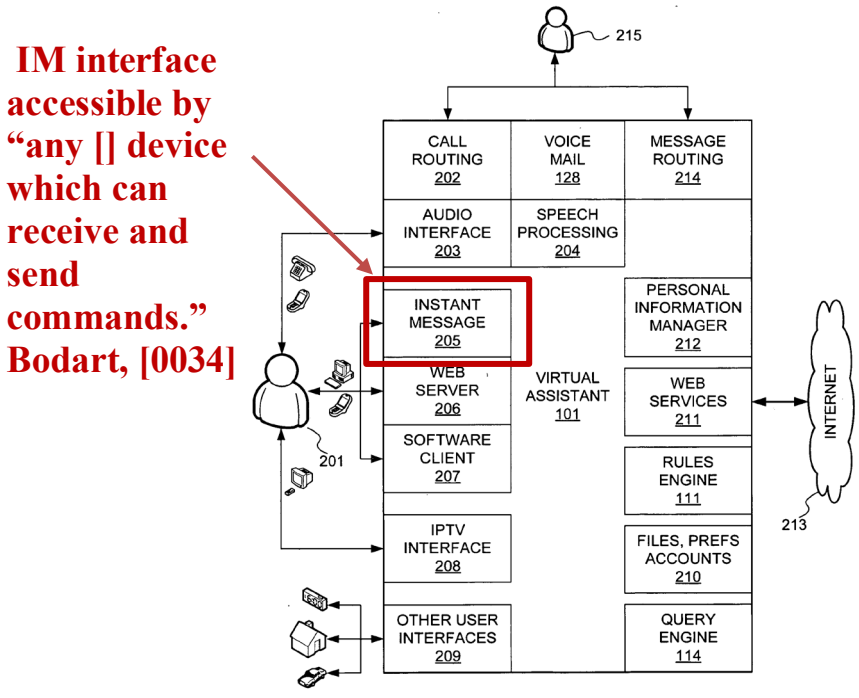
Bodart, Abstract.

73. Bodart teaches that IM access and control may be facilitated by an IM client application providing an IM software interface:

> A visual interface may include those provided by **instant messenger 205**, web server 206, software client 207, or any other electronic interface. User 201 may interact using a computer, cell phone, or any other device which can receive and send commands. For instant message interface 205, a user may be able to send messages using **an instant message client such as** America Online (AOL) Instant Messenger, **Microsoft Network (MSN) Messenger**, and so forth.

Bodart, [0034].

74. The instant messenger application used by the virtual assistant is illustrated in Fig.2 of Bodart, reproduced in annotated form below:



Bodart, Fig.2 (annotated)

75. However, Bodart otherwise provides limited details about how the virtual assistant gains access to IM services through the IM interface.

76. Dharmarajan provides IM client login procedures which Bodart expects. For instant, Dharmarajan describes "secure login mechanism[s] to gain access to [an] instant messaging server application program." Dharmarajan, 5:4-7. Dharmarajan discloses that "when a user launches the instant messaging client, the user may be required to provide a username and password." Dharmarajan, 1:33-35. Dharmarajan explains that its methods and techniques "may be embodied in an instant messaging server application program and **an instant messaging client application program**, such as those utilized to provide **the MSN Messenger service**." Dharmarajan, 3:55-59.

77. A POSITA implementing the IM functionality and applications of Bodart's virtual assistant would have found it obvious to consider other references like Dharmarajan, which describes implementation details for IM service login procedures that Bodart expects. For instance, Bodart teaches storing user accounts, including "email passwords [and] website passwords." Bodart, [0022]. Bodart further explains that the virtual assistant may include or employ "an instant message client such as ... Microsoft Network (MSN) Messenger." [0034]. Dharmarajan teaches that a user provides a username and password when launching an IM client. Dharmarajan, 1:33-35. It would have been obvious to a

POSITA to store an IM username and password in Bodart's user data module for logging into the IM service. Bodart, [0034]; Dharmarajan, 2:22-26. This would allow the virtual assistant to gain authenticated access to the IM service without prompting the user to provide the IM login details every time the IM service is used.

78. The combination of Bodart and Dharmarajan would be simply combining prior art elements according to known methods to yield predictable results. In that regard, the combination would include configuring Bodart's IM-enabled virtual with an IM username and password according to Dharmarajan's implementation details for logging into an IM service. This combination would be facilitated by well-known computer programming methods that would have been within the skill set of a POSITA. The result of the combination would be the virtual assistant configured to login to the IM service with the provided IM username and password. This result would have been predictable to a POSITA because the combination simply uses known login techniques as explained by Dharmarajan to enable the virtual assistant's IM services. Further, a POSITA implementing the combination would have had a reasonable expectation of success. As explained above, Bodart already describes storing and using login credentials to access online services. Bodart, [0022], [0042]. Further, both Bodart and Dharmarajan describe using the same software for IM service, such as MSN

Messenger. *See* Bodart, [0034]; Dharmarajan, 4:52-55. Combining Bodart's IM functionality with Dharmarajan's implementation for logging into an IM service would involve software changes easily accomplished by a POSITA. The software changes would include programming the virtual assistant to login to the IM server computer with an IM username and password retrieved from the user data module.

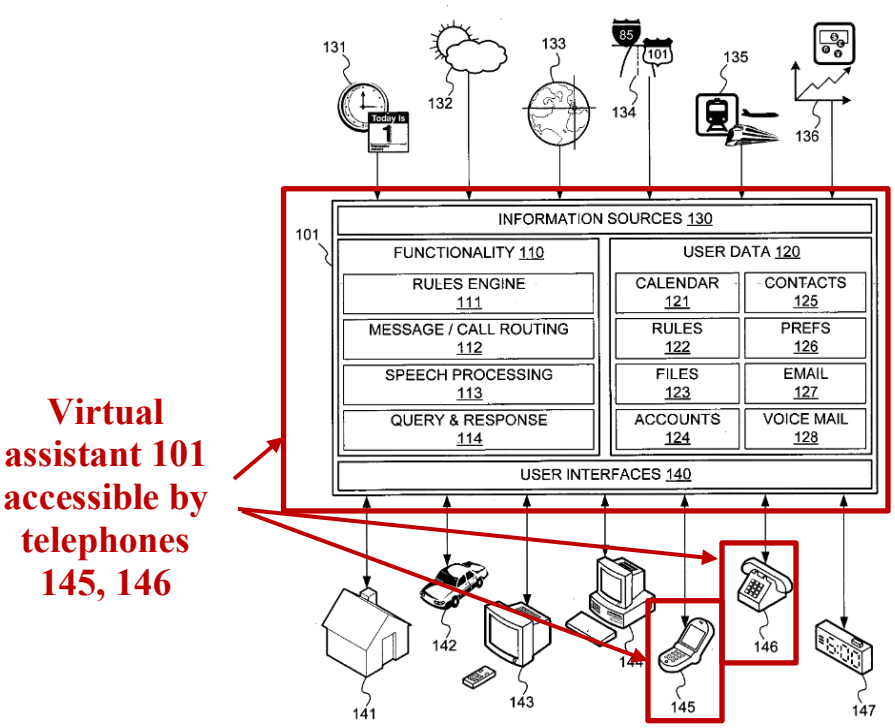
E. Detailed Analysis of Claims

1. Claim 1

a. [1.0] *A telephone system, comprising:*

79. To the extent that the preamble is limiting, Bodart discloses it.

80. Bodart teaches "[s]ystems and methods for assisting a user with a variety of tasks." Bodart, Abstract. For example, Bodart teaches a virtual assistant having "the ability to access user data 120, combined with technical functionality 110, to assist the user and make the most accurate decisions possible." Bodart, [0020]. Fig.1 of Bodart, reproduced in annotated form below, illustrates "one possible set of components of [the] virtual assistant." Bodart, [0020].



Bodart, Fig.1 (annotated)

81. Fig.1 of Bodart shows that the virtual assistant 101 is accessible through a variety of interfaces, including "conventional telephone 146," and "cell phone 145." Bodart, [0030]. For example, using speech processing functionality 113, "a remote user can **call into** his or her virtual assistant and provide voice commands." Bodart, [0026]. Further, Bodart discloses that the virtual assistant 101 includes "Message / Call Routing 112" functionality which "may enable virtual assistant 101 to intercept **incoming telephone calls**." Bodart, [0025]. Bodart explains that the virtual assistant 101 may have "the ability to initiate **telephone calls** and fax calls." Bodart, [0027].

82. The virtual assistant, which is accessible by telephone and has functionality for receiving and placing telephone calls, is a "*telephone system*."

83. Therefore, Bodart teaches "*[a] telephone system*," as recited in the preamble.

b. [1.1] *a database; and*

84. The prior art renders this limitation obvious.

85. Bodart discloses that a "**user data module** may store data about a specific user." Bodart, [0005]. The virtual assistant combines the user data and technical functionality "to assist the user and make the most accurate decisions possible." Bodart, [0020]. Bodart explains that the user data "may include contact information 125 ... where users can store contact email addresses and phone numbers, along with other information." Bodart, [0021]. Bodart further discloses looking up "phone numbers of contacts in the user's **database**." Bodart, [0057].

86. It would have been obvious that Bodart's user data module would include "*a database*." Databases were well-known and common components of computer systems as a way to store information, such as user data, software, user preferences, and configuration information. *See, e.g.*, Berkley, 6:55 – 7:3 (describing active user registry (AUR) databases); Goldman, 11:8-17; (AUR database stores telephone numbers, fax numbers, mobile telephone numbers, e-mail addresses, etc.); Pegaz-Paquet, [0116] (a reverse address match database for locating an address

associated with an originating telephone number). For example, Banister discloses storing "addresses with associated telephone numbers" in a "**database**." Banister, 2:1-5. Dharmarajan also contemplates a server computer having a "**database** containing [] information." Dharmarajan, 2:59-61. A POSITA would recognize that a database was a common storage technique for storing account information and user data, for example. *See* Bodart, [0029]. Further, implementing a database in Bodart's virtual assistant to store user data would have been well within the skillset of a POSITA.

87. Therefore, the prior art renders obvious "*a database*."

c. [1.2] *a server coupled to the database,*

88. The prior art renders this limitation obvious.

89. **First**, Bodart renders obvious that the virtual assistant includes "*a server*." For instance, Bodart contemplates that the virtual assistant uses servers to access information:

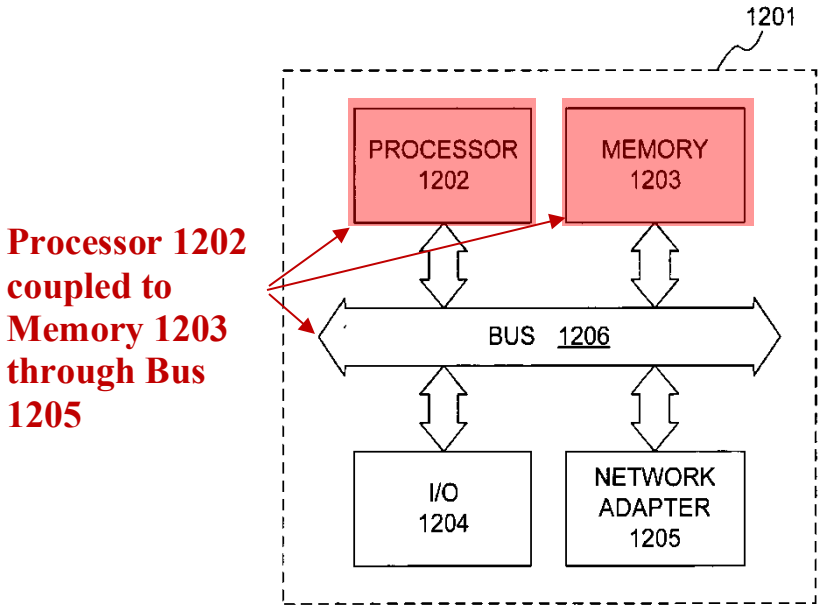
> Virtual assistant 101 may also access user files, preferences and account information 210 stored either locally on the same machine, or on **remote servers** and devices.

Bodart, [0042].

90. Further, Bodart explains that the information sources 130 accessed by the virtual assistant "may reside on **computer servers** located within the same

entity housing virtual assistant 101, or they may reside on remote networks, interconnected for example, via the Internet." Bodart, [0028]. Bodart also discloses using a "web server 206," which "may supply an interface from virtual assistant 101 to any computer, cell phone, or other electronic device...." Bodart, [0035].

91. Bodart also discloses that the virtual assistant may be implemented in the "operating environment" shown in Fig. 12, reproduced in annotated form below.



Bodart, Fig.12 (annotated)

92. Bodart explains that the virtual assistant 101 may be implemented using the operating environment of Fig.12:

> This operating environment may be found within any of the devices described above, and use of virtual assistant 101 may only be limited by the speed and size of both the associated processor and memory.

Bodart, [0075].

93. Referring to Fig.12, Bodart explains that the device 1201 in the operating environment:

> may be any sort of hardware minimally containing the components shown [in Fig. 12] ... This includes desktop computers, laptop computers, **servers**, ... and so forth. Bodart, [0075].

94. Thus, Bodart renders obvious that the virtual assistant may be implemented or otherwise facilitated by "*a server*." Accordingly, Bodart renders obvious that Bodart's virtual assistant "*compris[es] ... a server*."

95. **Second**, to the extent Bodart does not explicitly teach that the server is "*coupled to the database*," Bodart, Banister, and Dharmarajan render this limitation obvious. As discussed in Section IX.E.1.b above, Bodart, Banister, and Dharmarajan render obvious that the virtual assistant comprises "*a database*." For instance, the database may store the user data. It would have been obvious to a POSITA that the virtual assistant, which comprises and utilizes a server, would be coupled to a database storing the user data. For instance, it would have been obvious that the server would utilize the user data stored in the database to make decisions for the user 201. Bodart, [0029].

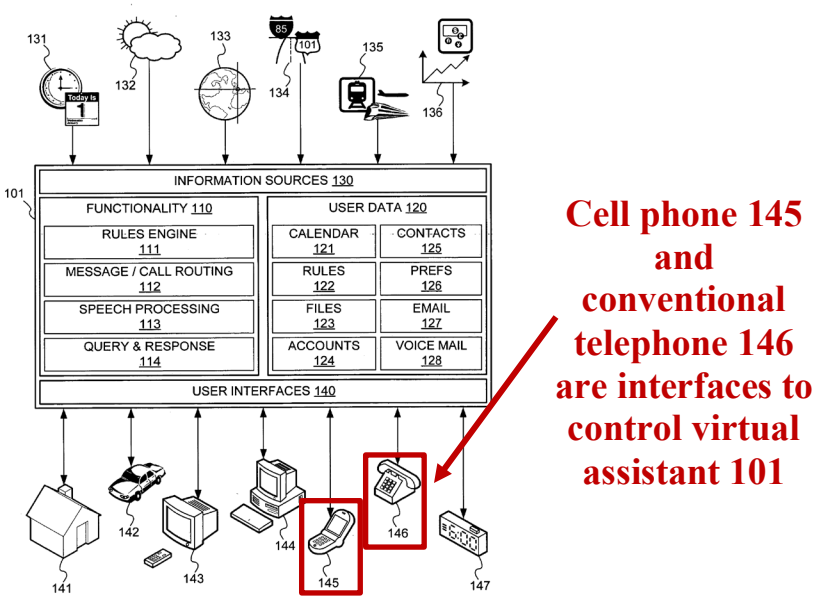
96. Therefore, the prior art renders obvious "*a server coupled to the database*."

d. [1.3] *wherein the server is utilized by a first provider to provide a first service to a first user, wherein the first service is telephone service, and wherein the telephone service is associated with a first telephone number.*

97. The prior art renders this limitation obvious.

98. Bodart teaches that the virtual assistant is able to receive and initiate phone calls. *See* Bodart, Abstract, [0025], [0040]. For instance, Bodart discloses that the "virtual assistant also has a rules engine enabling a user to create rules for **handling incoming calls** and ... rerouting calls based on their caller identification." Bodart, Abstract. Further, "Message & call routing 112 may enable virtual assistant 101 to intercept **incoming telephone calls**." Bodart [0025]. Further still, the "[v]irtual assistant 101 may include the ability to **receive and direct calls** and messages directed to **user 201**." Bodart, [0044].

99. Additionally, Bodart teaches that the virtual assistant may have "the ability to **initiate** telephone calls." Bodart, [0027]. Bodart discloses that the virtual assistant is accessible through a conventional telephone 146 and a cell phone 145.



Bodart, Fig. 1 (annotated)

100. By providing the ability to receive, route, and initiate calls, the virtual assistant is "*utilized*" to provide "*telephone service to the first user*."

101. Further, it would have been obvious to a POSITA that the telephone service would be provided by a telephone service provider ("*first provider*"). Fig.1 above shows a conventional telephone 146. It was well-known that telephone companies or carriers provided telephone service for use by conventional, landline telephones. *See*, e.g., Swope, p. 2, col. 2, ll. 80-84 ("...without cutting or in any way interfering with **the telephone company's** installation."); Prince, p. 1, col. 2, ll. 5-6 ("...**the telephone company** uses only a few of the numbers from 1 to 0"); Pegaz-Paquet, [0106] (example local and long-distance telephone carriers include Verizon, AT&T, and Sprint). Thus, it would have been obvious that the virtual

assistant 101, which includes "*a server*," is "*utilized by a first provider to provide a first service to a first user, wherein the first service is telephone service*."

102. Third, it would have been obvious that the "*telephone service is associated with a first telephone number*." The "*first telephone number*" may be the user's telephone number, or the telephone number the user calls to access the virtual assistant. Regarding the user's telephone number, Bodart discloses that "a remote **user can call** into his or her virtual assistant." Bodart, [0026]. Bodart also dicloses that the virtual assistant can route calls addressed to the user:

> Virtual assistant 101 may include the ability to receive and direct calls and messages directed to user 201.... By routing calls and messages, user 201 does not have to screen her calls. Rather, virtual assistant 101 is able to recognize an identity of the sender (via caller ID or some other identification method) and decide whether or not to route the call to voice mail 128 or some other system.

Bodart, [0044].

103. This disclosure renders obvious that the remote user has as telephone number from which to make the call. Further, Bodart discloses that "[v]irtual assistant 101 may include the ability to **receive and direct calls** and messages directed to **user 201**." This renders obvious the user having a telephone number. Bodart, [0044]; Thus, the telephone service used to access the virtual assistant is

"*associated with*" a telephone number for the user, the virtual assistant, or both. Bodart, [0026], [0044].

104. Therefore, the prior art renders obvious "*wherein the server is utilized by a first provider to provide a first service to a first user, wherein the first service is telephone service, and wherein the telephone service is associated with a first telephone number*."

e. [1.4] *wherein the server is configured to: receive a request to provision a second service for the first user,*

105. The prior art renders this limitation obvious.

106. Bodart teaches that the virtual assistant facilitates a variety of services that can be provided to the user, including email service (Bodart, [0002], [0022], [0027]), instant messaging service (Bodart, Abstract, [0025], [0034]), geographic location services (Bodart, [0029]), file storage (Bodart, [0022], [0043]), online banking (Bodart, [0022], [0042]), and others. This Declaration will refer to instant messaging service and email service for limitations involving the "*second service*."

107. Instant messaging example: Bodart teaches that "[t]he virtual assistant also has a rules engine enabling a user to create rules for **handling** incoming calls and **instant messages**." Bodart, Abstract. Further, the virtual assistant's message and call routing functionality "may enable virtual assistant 101 to intercept incoming ... instant messages." Bodart, [0025]. *See also*, Bodart, [0044] ("incoming messages including instant messages ... may similarly be routed using

message routing 214."). Referring to one specific example involving an incoming telephone call, Bodart discloses "virtual assistant 101 may put contact 215 on hold and attempt to reach user 201 via phone or instant message." Bodart, [0060].

108. Bodart teaches that a third-party instant message client may be used as an interface for sending instant messages:

> For instant message interface 205, a user may be able to send messages using an instant message client such as America Online (AOL) Instant Messenger, Microsoft Network (MSN) Messenger, and so forth.

Bodart, [0034].

109. The virtual assistant's functionality to "send," "intercept," and/or "route" instant messages teaches "*a second service*." The instant message service may be facilitated by the virtual assistant using an instant message client, for example. Bodart, [0034].

110. It would have been obvious that Bodart's virtual assistant would "*receive a request to provision*" the instant message service. For instance, Dharmarajan teaches that "when a user launches [an] instant messaging client, the user may be required to **provide a username and password**." Dharmarajan, 1:33-35. Bodart discloses storing "information for ... **user accounts**, including email passwords, website passwords, and so forth." Bodart, [0022]. It would have been obvious to a POSITA that the Bodart virtual assistant's user data module would receive, from the user, a selection of an IM platform to use (e.g., MSN Messenger)

and a corresponding IM username and password ("*request to provision [the] second service*") as explained in Dharmarajan. Bodart, [0034]; Dharmarajan, 5:24-38; In this way, the virtual assistant would use the username and password to login to the instant messaging server. Bodart, [0022]. This would allow the virtual assistant to access the IM service to perform IM-related functionalities such as routing incoming IMs or notifying the user of any newly received IM messages. Bodart, [0044].

111. Thus, configuring Bodart's virtual assistant by providing the username and password for accessing the instant messenger service renders obvious "*wherein the server is configured to: receive a request to provision [the] second service for the first user.*" Dharmarajan, 1:33-35.

112. Email example: Bodart teaches that the virtual assistant includes "Query and response 114 abilities" through which a user may submit a query to "search user files 123, **emails 127**, calendar 121 appointments, and so forth." Bodart, [0027]. Bodart teaches storing email passwords in an Accounts 124 module for accessing email service:

> Accounts 124 may include personal financial account information for the user, possibly including bank login information. Accounts 124 may also include information for other types of user accounts, including **email passwords**, website passwords, and so forth.

Bodart, [0022].

113. Bodart further explains that the virtual assistant may rely on a personal information manager (PIM) such as "**Microsoft Outlook**" to access user data. Bodart, [0040]. "The user data to which a PIM may provide access includes ... **email messages**." Bodart, [0040]. Bodart describes the virtual assistant providing a variety of email functionalities, including reading email messages to the user via telephone (Bodart, [0002]), routing incoming email messages (Bodart, [0044]), alerting the user by email of an incoming telephone call (Bodart, [0062]), and sending a user's document to a recipient's email (Bodart, [0068]).

114. In light of Bodart's teachings regarding email access and storing email passwords and other user accounts information, it would have been obvious that the virtual assistant would be configured to "*receive a request to provision a second service for the first user*." For example, configuring a computer with login credentials to access internet-based services like IM or email ("*second service*") was a common technique for gaining authorized access to those services. It was well-known to allow users to install software applications to access email (e.g., Microsoft Outlook) or IM services (e.g., MSN Messenger). *See* Bodart, [0044]; Dharmarajan, 3:55-59; Ramey, 2:52-54. Users would configure the software applications by providing their login credentials (e.g., username and password), and the server settings (e.g., server protocol and name). Dharmarajan, 1:33-35. Polis, 5:27-44; O'Hara, pp. 104-105; Section V.B. By entering the login

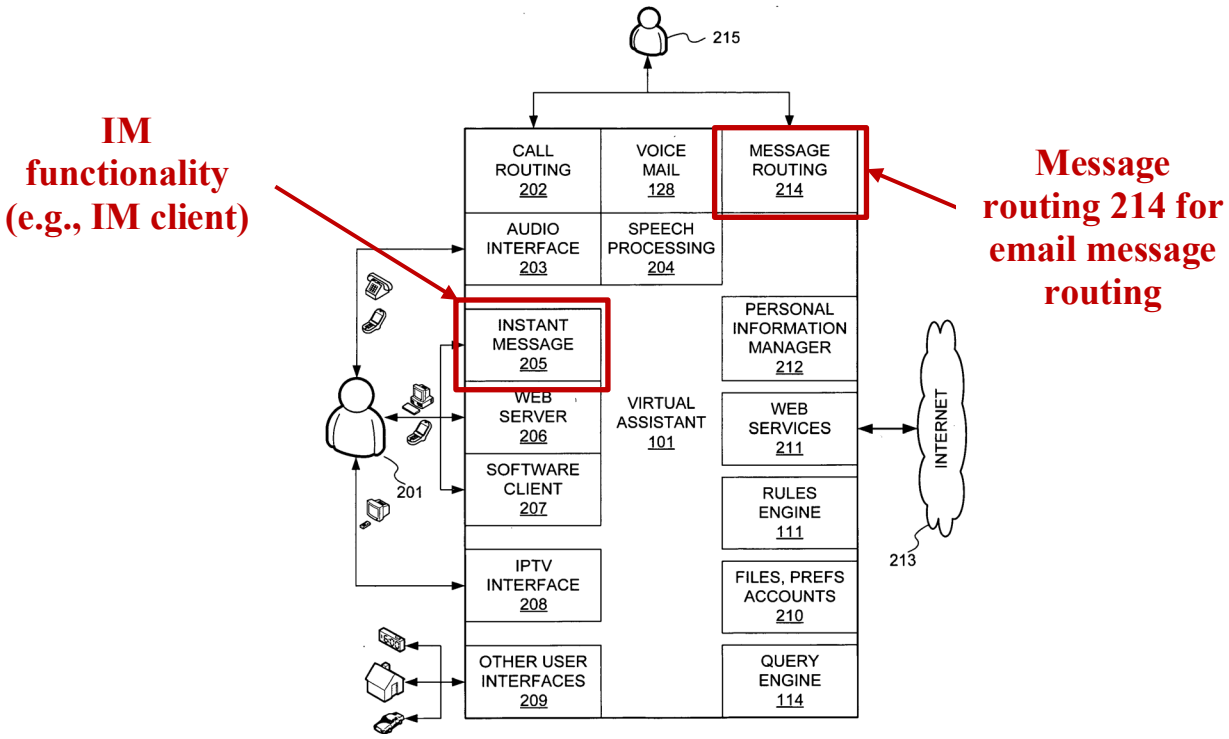
credentials into the software application, the user would request, or cause the software application to request, access to the service facilitated by the software. The software application may include or implement a software client that can request access to a server providing the service. The software client may communicate with a server client to authorize the user to and activate the service (e.g., email, IM). Thus, it would have been obvious for the virtual assistant to receive the email server and login credentials from the user ("*request to provision a second service for the first user*") so that the virtual assistant can perform email-related functions such as reading email to the user by phone. Bodart, [0002].

115. Therefore, the prior art renders obvious "*wherein the server is configured to: receive a request to provision a second service for the first user*."

f. [1.5] *wherein the second service is provided by a second provider, and wherein the second provider is different than the first provider;*

116. The prior art renders this limitation obvious.

117. Instant messaging example: Fig.2 of Bodart shows that the virtual assistant includes IM functionality 205. Bodart, [0034].



Bodart, Fig. 2 (annotated)

118. Referring to the IM functionality 205, Bodart explains that "a user may be able to send messages using an instant message client such as America Online (AOL) Instant Messenger, Microsoft Network (MSN) Messenger, and so forth." Bodart, [0034]. AOL and Microsoft are example of instant message service providers. Thus, Bodart teaches "*a second provider*" (such as AOL or Microsoft) providing IM service ("*second service*"). Bodart, [0034].

119. Further, a POSITA would have found it obvious that AOL or Microsoft would be different from the telephone service provider ("*first provider*"). For example, a POSITA would have known that common telephone service providers were AT&T, Verizon, and T-Mobile. *See, e.g.*, Pegaz-Paquet,

[0106] (discussing Verizon, AT&T, and Sprint as example local and long-distance telephone carriers). By contrast, the core business of IM providers such as AOL and Microsoft was typically not telephone service. Thus, it would have been obvious that the IM service provider for the virtual assistant would be different from the telephone service provider. Therefore, Bodart renders obvious that the IM service is "*provided by a second provider, wherein the second provider is different than the first provider*."

120. Email example: Referring to Fig.2, Bodart teaches that the virtual assistant includes message routing functionality 214. Bodart explains that "email messages [may be] routed using message routing 214." Bodart [0044]. In one specific use case for email message functionality, Bodart explains that the virtual assistant may intercept a phone call to the user, and "may attempt to send a text or email message to user 201 to determine the user's availability to take a particular call." Bodart, [0064].

121. To facilitate access to email services, Bodart teaches using PIMs such as "Microsoft Outlook." Bodart, [0040]. Bodart further explains:

> As opposed to one single application, each of these types of PIM data may alternatively be managed by individual applications (e.g., iCal, Address Book, and Mail on an Apple Computer).

Bodart, [0040].

122. It would have been obvious to a POSITA that the server system associated with Outlook or Apple's Mail application would provide email service. Alternatively, a POSITA would understand that an email application such as Outlook or Apple's Mail application would provide access to other well-known email services such as Gmail or Hotmail. Bodart, [0036], [0040]; Dharmarajan, 5:28-31; Letzing, pp. 1-2. For instance, a POSITA would have understood that Outlook and Apple Mail are email clients that can be configured with login credentials to a login to a variety of email providers, such as Gmail, Hotmail, or a private company-operated email system. It would also have been obvious to a POSITA that an email service provider such as Gmail would be "*different than*" the telephone service provider ("*the first provider*"). Bodart, [0040].

g. [1.6] *responsive to receiving the request: create a mapping from the first telephone number to a second identifier (ID), wherein the second ID is utilized by the second provider to identify the first user;*

123. Bodart, Banister, and Dharmarajan render obvious this limitation.

124. Instant messaging example: As discussed at Section IX.E.1.e, Bodart and Dharmarajan render obvious a user providing an IM username and password into the user data module. *See* Section IX.E.1.e. The IM username (*see* Dharmarajan, 1:33-35) teaches "*the second ID*." Further, the combination of Bodart and Dharmarajan renders obvious a "*second ID [] utilized by the second provider to identify the first user*." For instance, a POSITA would have understood

that an IM service provider such as AOL or Microsoft (Bodart, [0025]) uses the IM username to identify the first user in order to deliver IMs addressed to the IM username.

125. Bodart describes functionalities of the virtual assistant that combine telephone service associated with a telephone number, and a second communication service associated with a different electronic ID or address. In one example, Bodart describes the virtual assistant intercepting incoming phone calls and determining a disposition for the call:

> If contact 215 attempts to telephone user 201, the call may be intercepted by virtual assistant 101. Virtual assistant 101 attempts to recognize contact 215 using caller ID (or some other system) and comparing the incoming number with the phone numbers of contacts in the user's database.

Bodart, [0057].

126. Based on the user's calendar information, the virtual assistant can send an IM to the user notifying them of the incoming call:

> User 201 may be able to prioritize and/or categorize the appointments on his calendar, and based on the current time and date, virtual assistant 101 may decide whether to forward a call based on the priority of the recognized contact and on the priority of the current appointment (e.g., sleeping is not to be interrupted, weekly status meeting may be interrupted by client calls). **Alternatively, virtual assistant 101 may**

> **put contact 215 on hold and attempt to reach user 201 via** phone or **instant message**.

Bodart, [0060].

127. Thus, Bodart describes the functionality of the server sending an IM to a user receiving a phone call. Bodart also teaches storing "email passwords, website passwords, and so forth" in the user data module. Bodart, [0022]. However, Bodart otherwise provides limited details on how the virtual assistant IMs the user to notify them of the incoming phone call. Banister and Dharmarajan provide implementation details for providing access to electronic communications by phone. Banister, Abstract; Dharmarajan, 1:33-38. For instance, Banister teaches a method "for converting a textual portion of an electronic mail message into audio and delivering the audio through a phone call to a recipient." Banister, Abstract. Banister explains that "an **address** specified by an originator of a message may be converted to a telephone number or used to **look-up a telephone number** when the message is received at electronic mail server 104." Banister, 4:14-17. To facilitate the look-up, a "mail server may maintain a list or **database of addresses with associated telephone numbers**." Banister, 2:1-5. Thus, Banister teaches "*a mapping from the first telephone number to*" an email address.

128. While Banister involves email addresses, Dharmarajan teaches that logging into an IM server involves providing an IM username. Dharmarajan, 1:33-35. Similar to Banister's teachings of mapping a telephone number to an email

address in a database, it would have been obvious to a POSITA for Bodart's virtual assistant to "c*reate a mapping from the first telephone number*" to an IM username ("*second ID*"). For instance, Bodart teaches that the user data module may store user account information "including email passwords, website passwords, and so forth." Bodart, [0022]. Bodart also teaches that "users can store contact email addresses and phone numbers" in the user data module. Bodart, [0021]; *see also* [0057] ("Virtual assistant 101 attempts to recognize contact 215 ... comparing the incoming number with the **phone numbers** of contacts **in the user's database**.")

129. It would have been obvious that the user's IM username would be stored in the user data module with the telephone number of the user or the telephone number of the virtual assistant. For instance, as explained above, it would have been obvious for the virtual assistant to configure the user data module with the IM username and password ("*request to provision a second service*") to allow the virtual assistant to access to the IM service. The virtual assistant would "*create*" the mapping in the user data module when the user configures the virtual assistant with the IM login credentials ("*responsive to receiving the request*"). This would allow the virtual assistant to IM the user in response to an incoming phone call, as explained above. Bodart, [0060].

130. Email example: As explained above, Banister's teaching of maintaining "a list or **database of addresses with associated telephone**

numbers" teaches "*a mapping from the first telephone number to*" the user's IM username. In the email example, the user's email address is "*the second ID*." A POSITA would understand that the email address, similar to the IM username, would be used by an email service provider (e.g., Gmail, Hotmail) "*to identify the first user*."

131. Similar to the IM example, Bodart and Banister render obvious "*create a mapping from the first telephone number to*" the user's email address. For instance, Bodart describes a virtual assistant functionality for "having a user's email read to her by phone." Bodart, [0002]. Banister provides implementation details for this technique. Banister explains that "an address specified by an originator of a message may be converted to a telephone number or used to look-up a telephone number when the message is received at electronic mail server 104." Banister, 4:14-17. Further, the "mail server may maintain a list or database of addresses with associated telephone numbers." Banister, 2:1-5. Thus, Banister teaches using "*a mapping from the first telephone number to*" the email address to facilitate Bodart's email reading functionality.

132. It would have been obvious that the user's email address would be stored in the user data module with the telephone number of the user or the telephone number of the virtual assistant ("*the first telephone number*"). For instance, it would have been obvious for the virtual assistant to configure the user

data module with the email address and password ("*request to provision a second service*") to allow the virtual assistant to access to the email service. Bodart, [0022]. In one example, Bodart describes using an email client such as Microsoft Outlook to access the email service. Bodart, [0044]. O'Hara teaches that a well-known feature of Microsoft Outlook is storing contact information, including email addresses and associated telephone numbers for a variety of individuals. O'Hara, pp. 116-117. *See also*, Ramey, Fig.2; Berkley, 6:55 – 7:3; Goldman, 11:8-17; Pegaz-Paquet, [0116].

133. It would have been obvious to similarly configure the virtual assistant with the user's email address and telephone number to facilitate the virtual assistant calling the user to read her emails over the phone. Bodart, [0002]. The virtual assistant would "*create*" the mapping in the user data module when the user configures the virtual assistant with the email login credentials ("*responsive to receiving the request*"). *See* Section IX.E.1.e.

134. Thus, the prior art renders obvious "*responsive to receiving the request: create a mapping from the first telephone number to a second identifier (ID), wherein the second ID is utilized by the second provider to identify the first user*."

h. [1.7] *[responsive to receiving the request:] activate the second service with the second provider utilizing the second ID as a user ID for the first user; and*

135. The prior art renders this limitation obvious.

136. Instant messaging example: Dharmarajan's disclosure of the IM username (*see* Dharmarajan, 1:33-35) teaches the "*second ID*." *See* Section IX.E.1.g. Dharmarajan explains that a user may provide an IM username and password "to gain access to the server computer that provides the instant messaging service." Dharmarajan, 1:33-38.

137. It would have been obvious to a POSITA that Bodart's virtual assistant would login to an IM service when the user configures the virtual assistant with the IM username and password ("*responsive to receiving the request*"). Logging into the IM service is "*activat[ing] the second service*." For example, the virtual assistant would login to the IM service to confirm that the IM username and password are correct and to confirm that the connection to the IM server can be established. Further, Dharmarajan teaches logging into an IM server computer to initiate an authenticated session of a specified length. Dharmarajan, Abstract. In this way, the client computer (e.g., Bodart's virtual assistant) may repeatedly access the IM server computer during the limited duration without having to provide the IM username and password each time, and be notified upon receipt of an IM message.

138. Thus, logging into the IM service responsive to the user providing the IM username and password would be an obvious part of the virtual assistant using IM-related functionalities, including "handling incoming... instant messages" and "send[ing] messages using an instant message client." Bodart, Abstract, [0034]; Dharmarajan, 9:57-61. It would also have been obvious that the IM service provider (e.g., Microsoft, AOL) would utilize the IM username as "*a user ID*" for the user because the IM username identifies the user. Bodart, [0034]; Dharmarajan 1:33-35; *see* Section IX.E.1.g.

139. Email example: As explained above at Section IX.E.1.g, the stored email address in the account information is the "*second ID*." It would have been obvious to a POSITA that Bodart's virtual assistant would login to an email service ("*activate the second service*") when the user configures the virtual assistant with the email address and password ("*responsive to receiving the request*"). For instance, it would have been obvious to a POSITA that the virtual assistant would login to the email service to verify that the email address and password are correct, or to confirm that a connection with the email server can be established. Further, a POSITA would have found it obvious to incorporate Dharmarajan's teachings related to initiating an authenticated session with a server computer with Bodart's virtual assistant. Dharmarajan, Abstract, 2:32-38. In this way, the virtual assistant would be able to gain repeated access to the email service without having to

provide the email address and password each time. Dharmarajan, 1:58-62. Establishing the authenticated session would "*activate*" the email service ("*the second service*").

140. It would also have been obvious to a POSITA that the email service provider would utilize an email address as "*a user ID*" for the user because that is how email services commonly worked. Bodart, [0040]. For instance, a user typically provides their email address, along with their password and server settings, to login to the email server. O'Hara pp. 104-105; Polis, 5:27-44. It would have been obvious for the virtual assistant to login to the email server when the user provides their email address and password into the user data module ("*responsive to receiving the request*"). For example, logging in to the email server would allow the virtual assistant to confirm that the access credentials (email address and password) were provided correctly.

141. Thus, the prior art renders obvious "[*responsive to receiving the reques*t:] *activate the second service with the second provider utilizing the second ID as a user ID for the first user*."

i. [1.8] [*responsive to receiving the request:*] *associate the second service with the first telephone number, wherein the first telephone number is associated with routing and interworking data necessary to utilize the second service.*

142. The prior art renders obvious this limitation.

143. Instant messaging example: Bodart discloses various IM functionalities of the virtual assistant, including the ability to send IMs (Bodart, [0034]) and to route incoming IMs (Bodart, [0044]). In one example, Bodart describes, in response to an incoming phone call, "virtual assistant may **put [the call] on hold** and attempt to reach user 201 via... **instant message**." Bodart, [0060]. Thus, Bodart teaches using IM service ("*the second service*") to reach the user when the virtual assistant intercepts a call to a telephone number associated with the user or virtual assistant ("*the first telephone number*"). Bodart, [0060]. The virtual assistant "*associate[s]*" IM service with "*the first telephone number*," because the virtual assistant uses the IM service in response to incoming phone calls to the user's telephone number or the virtual assistant's telephone number. In this regard, Bodart describes storing rules in the user data module to execute the functionalities of the virtual assistant:

> User data 120 may include a **set of rules** for use with rules engine 111, setting forth **the logic used by virtual assistant to aid the user**.

Bodart, [0021].

144. Bodart further explains:

> A rules engine may be compared with a software compiler which translates user input (i.e. computer source code) into a series of instructions which a computer processor can understand (i.e. computer object code). Rules engine 111 takes **user-created rules, which may**

> **be represented as a series of textual if-then-else statements**, and translates them into a series of actions.

Bodart, [0024].

145. Bodart describes using rules to automate the virtual assistant's functionalities including call handling (Bodart, [0057]) and adjusting alarms (Bodart, [0052]). It would have been obvious to a POSITA to store a rule or set of rules in the user data module to automate the functionality that causes the virtual assistant to send an IM to the user in response to an incoming call. Bodart, [0060]. Thus, an association between the IM service ("*the second service*") and "*the first telephone number*" would be embodied in the rule or set of rules stored in the user data module, where "the first telephone number" corresponds to the telephone number of the user, the telephone number of the virtual assistant, or both. *See* Section IX.E.1.d. Executing rules for IM-related functionalities would involve the IM username and password provided by the user. Therefore, the virtual assistant using the rule embodying the association between the IM service and the telephone number would be "*responsive to receiving the request*." *See* Section IX.E.1.e (user providing IM username and password is a "*request to provision [the] second service*"). Accordingly, Bodart renders obvious *"[responsive to receiving the request:] associate the second service with the first telephone number*."

146. Bodart teaches that "a user may be able to send messages using an instant message client." Bodart, [0034]. A POSITA would have understood that an

IM client accesses an IM service using the name or address of the IM server computer. For instance, Dharmarajan explains that an "instant messaging client" performs a login "to gain access to **the server computer** that provides the instant messaging service." Dharmarajan, 1:35-38. The name or address of the IM server computer is "*routing and interworking data necessary to utilize*" the IM service.

147. It would have been obvious to a POSITA that Bodart's user data module would store the name or address of the IM server computer for gaining access to the IM service. Bodart, [0034], [0060]. As explained above at Section IX.E.1.g, it also would have been obvious to store the telephone number of the user, the telephone number of the virtual assistant, or both in the user data module. Thus, the name or address of the IM server computer ("*routing and interworking data*") would be stored in the user data module with the telephone number of the virtual assistant or user, and therefore would be associated with "*the first telephone number*." Bodart, [0022].

148. Email example: Bodart teaches that the virtual assistant can search the user's emails (Bodart, [0022]) and route incoming emails (Bodart, [0044]). Bodart contemplates "having a user's email read to her by phone." Bodart, [0002]. In another example, Bodart discloses that "virtual assistant 101 may attempt **to send** a text or **email message** to user 201 to determine the user's availability to take a particular **call**." Bodart, [0064]. Each of these examples involves an integration of

telephone service *vis-a-vis* "*the first telephone number*" with email service ("*the second service*") *vis-a-vis* the user's email address.

149. For instance, Bodart teaches using rules to automate the virtual assistant's functionalities. Bodart, [0021], [0024], [0052], [0057]. It would have been obvious to a POSITA to store a rule or set of rules in the user data module to automate the functionality that causes the virtual assistant to read emails to the user by phone. Bodart, [0002]. Thus, an association between the email service ("*the second service*") and "*the first telephone number*" would be embodied in the rule or set of rules stored in the user data module, where "*the first telephone number*" is the telephone number of the user, the telephone number of the virtual assistant, or both. The settings and credentials involved with accessing the email service to execute email-based rules would include the server settings (e.g., server protocol and server name), and the email address and password provided by the user. *See* Section V.B. Therefore, the virtual assistant using the rule embodying the association between the email service and the telephone number would be "*responsive to receiving the request*." *See* Section IX.E.1.e (user providing email address and password is a "*request to provision [the] second service*"). Accordingly, Bodart renders obvious *"[responsive to receiving the request:] associate the second service with the first telephone number*."

150. Similar to IM service, a POSITA would have understood that accessing an email service would involve using an email client to access an email server. *See* O'Hara, pp. 104-105. The email client would access the email server using the name or address of the email server and the server protocol ("*routing and interworking data necessary to utilize the second service*"). *See* O'Hara, pp. 104-105 (adding email account involves selecting type of email server and inputting server name, username, and password); Polis, 5:27-44 (connecting to email service typically involves inputting name and protocol of server, such as POP3 server). It would have been obvious to a POSITA that Bodart's user data module would store the name or address of the email server computer to allow the virtual assistant to access the email service on behalf of the user. Bodart, [0022], [0044], [0072]; As explained above at Section IX.E.1.g, it also would have been obvious that the user data module would store the telephone number of the user or the virtual assistant. Thus, it would have been obvious that the user data module would store the name or address of the email server and the telephone number of the user or virtual assistant, such that the name or address of the email server would be "*associated with the first telephone number*." Bodart, [0022].

151. Thus, the prior art renders obvious "*associate the second service with the first telephone number, wherein the first telephone number is associated with routing and interworking data necessary to utilize the second service*."

2. Claim 2

a. [2.0] *The telephone system as recited in claim 1, wherein the server is further configured to associate a plurality of services with the first telephone number,*

152. Bodart, Banister, and Dharmarajan render obvious *"[t]he telephone system as recited in claim 1*." *See* analysis at Section IX.E.1.a-i.

153. Bodart explains that the virtual assistant is accessible by telephone interfaces, such as a cell phone and a conventional telephone:

> "one or more user interfaces 140 which allow a user to interact with virtual assistant 101. These may include ... **cell phone 145** ... [and] a **conventional telephone 146**."

Bodart, [0030].

154. Further, "a user may control virtual assistant 101 using an audio interface 203 using a conventional telephone." Bodart, [0033]. To control the virtual assistant by phone, Bodart explains that the virtual assistant includes "[s]peech processing 113 functionality [which] can enable virtual assistant 101 to comprehend spoken language commands. Bodart, [0026]. "Using this functionality, a remote user can call into his or her virtual assistant and provide voice commands." Bodart, [0026].

155. As also explained above at Section IX.E.1.i, the prior art renders obvious associating both IM services and email services with the user's telephone number or the virtual assistant's telephone number ("*first telephone number*"). It

would have been obvious to a POSITA that the virtual assistant would associate multiple IM accounts and multiple email accounts for the user ("*plurality of services*"). For example, the virtual assistant may facilitate access to a work email address and a personal email address, where each can be accessed by phone.

156. Further, Bodart's teaches that the virtual assistant is capable of accessing a wide variety of other services in addition to email and IM. For instance, Bodart describes functionalities that incorporate services from different service providers, including "search[ing] for the text string among all documents" (Bodart, [0066]), sending a document "by facsimile" (Bodart, [0068]), "controlling functions within a house" (Bodart, [0074]), and "checking of voicemail, the monitoring of a calendar, the viewing of email, and access to additional user data." Bodart, [0072]. To perform such tasks, Bodart discloses that the "virtual assistant 101 may receive information from **various sources 130** using one or more computer networks." Bodart, [0028]. The virtual assistant 101 also accesses information from third-party applications such as "**Google Maps**" and "**iCal**." Bodart, [0040], [0046], [0047].

157. Because these other services can be accessed by telephoning the virtual assistant, it would have been obvious that the virtual assistant "*associate[s*]" these services "*with the first telephone number*." *See* Bodart, [0030]; Section IX.E.1.g. For instance, it would have been obvious that Bodart's

virtual assistant would be configured to provide the user access to email, IM, or calendar information based on the user's telephone number, the virtual assistant's telephone number, or both. Section IX.E.1.a-b. Thus, Bodart renders obvious that the server providing the virtual assistant is "*configured to associate a plurality of services with the first telephone number*."

b. [2.1] *wherein the plurality of services are provided by a plurality of providers different than the first provider, and wherein the plurality of services are utilized by the first user.*

158. The prior art renders obvious this limitation.

159. **First**, Bodart's virtual assistant "is able to retrieve data from a variety of information sources 130, in part to augment its ability to make decisions on the user's behalf." Bodart, [0020]. For example, "**information sources 130** ... may reside on remote networks, interconnected for example, via the Internet." Bodart, [0028]. The information sources 130 "may include date & time information 131, weather information 132, user geographic location information 133, road traffic information 134, mass transit information 135, and finance information 136." Bodart, [0029]. Further, Bodart explains that other third party services may be used to obtain information, including "Google Maps," "Yahoo! Maps," "Mapquest," "iCal," and "the Weather Channel," Bodart, [0040], [0046], [0047]. Thus, Bodart teaches that the information sources 130 are "*provided by a plurality of providers*."

160. Further, it would have been obvious that the providers of the information sources 130 (e.g., Google Maps, iCal) would be *"different than*" the telephone service provider ("*first provider*"). For example, a POSITA would have known that common telephone service providers were AT&T, Verizon, and T-Mobile. *See, e.g.*, Pegaz-Paquet, [0106]. By contrast, the core business of weather, calendar, and mapping information providers such as Google Maps and the Weather Channel was typically not telephone service. Thus, it would have been obvious that the IM service provider for the virtual assistant would be different from the telephone service provider.

161. **Second**, Bodart teaches that "a user may **query** virtual assistant with regard to various aspects of user data 120 and/or **information sources 130**." Bodart, [0027]. For instance, the user may access the virtual assistant by calling into the virtual assistant, or accessing the virtual assistant on a computer. Bodart, [0027], [0034]. Thus, Bodart teaches that the information sources 130 "*are utilized by the first user*" via the virtual assistant 101.

162. Therefore, the prior art renders obvious "*wherein the plurality of services are provided by a plurality of providers different than the first provider, and wherein the plurality of services are utilized by the first user.*"

3. Claim 6

a. [6.0] *The telephone system as recited in claim 1, wherein the second service is a conferencing*

application, voice attendant application, or text messaging application.

163. The prior art discloses this feature.

164. As explained above, both the IM service and email service teach "*the second service*." Section IX.E.1.e. Each of the IM service and email service is a "*text messaging application*."

165. In another example, Bodart teaches that "[i]nteractive voice response (IVR) technologies may enable speech processing 113 abilities for virtual assistant 101." Bodart, [0026]. Bodart explains that IVR makes the virtual assistant's functionality accessible via audio interfaces (e.g., a telephone) so that the virtual assistant can "process spoken commands and translate them into text or another format which virtual assistant commands." Bodart, [0033]. For instance, Bodart describes using IVR to "find a document" by its title. Bodart, [0066]. Further, Bodart describes the virtual assistant using "speech processing" "search user files 123, emails 127, calendar 121 appointments, and so forth." Bodart, [0027]. The IVR or speech processing functionality for issuing commands to the virtual assistant is a "*voice attendant application*."

166. Thus, the prior art discloses "*wherein the second service is a conferencing application, voice attendant application, or text messaging application.*"

4. Claim 7

a. [7.0] *A method comprising:*

167. To the extent the preamble is limiting, the prior art discloses this limitation.

168. Bodart teaches "[s]ystems and **methods** for assisting a user with a variety of tasks." Bodart, Abstract. In one example illustrated in Fig.6, Bodart discloses "a **method** of routing an incoming communication." Bodart [0061]. Thus, Bodart discloses the preamble. Figs. 4 and 8 of Bodart also illustrate "method[s]" that involve some of the functionalities of the virtual assistant described above with respect to claims 1, 2, and 6. Bodart, [0009], [0013]. The combination of Bodart, Banister, and Dharmarajan discloses a method "*comprising*" the recited elements.

b. [7.1] *providing a first service to a first user by a first provider, wherein the first service is telephone service, and wherein the first user has a first telephone number.*

169. Bodart renders obvious this limitation; *see* [1.2] and [1.3] above.

c. [7.2] *receiving a request to provision a second service for the first user, wherein the second service is provided by a second provider, and wherein the second provider is different than the first provider.*

170. Bodart in view of Dharmarajan render obvious this limitation; *see* [1.4] and [1.5] above.

d. [7.3] *responsive to receiving the request: creating a mapping from the first telephone number to a second*

identifier (ID), wherein the second ID is utilized by the second provider to identify the first user.

171. Bodart in view of Banister and Dharmarajan render obvious this limitation; *see* [1.6] above.

e. [7.4] *activating the second service with the second provider utilizing the second ID as a user ID for the first user.*

172. Bodart in view of Banister and Dharmarajan render obvious this limitation; *see* [1.7] above.

f. [7.5] *associating the second service with the first telephone number, wherein the first telephone number is associated with routing and interworking data necessary to utilize the second service.*

173. Bodart in view of Banister and Dharmarajan render obvious this limitation; *see* [1.8] above.

5. Claim 8

a. [8.0] The method as recited in claim 7, further comprising associating a plurality of services with the first telephone number,

174. Bodart in view of Banister renders obvious this limitation; *see* [2.0] above.

b. [8.1] wherein the plurality of services are provided by a plurality of providers different than the first provider, and wherein the plurality of services are utilized by the first user.

175. Bodart renders obvious this limitation; *see* [2.1] above.

6. Claim 12

a. [12.0] The method as recited in claim 7, wherein the second service is a conferencing application, voice attendant application, or text messaging application.

176. Bodart teaches this limitation; *see* [6.0] above.

7. Claim 13

a. [13.0] A non-transitory computer readable storage medium comprising program instructions, wherein when executed the program instructions are operable to:

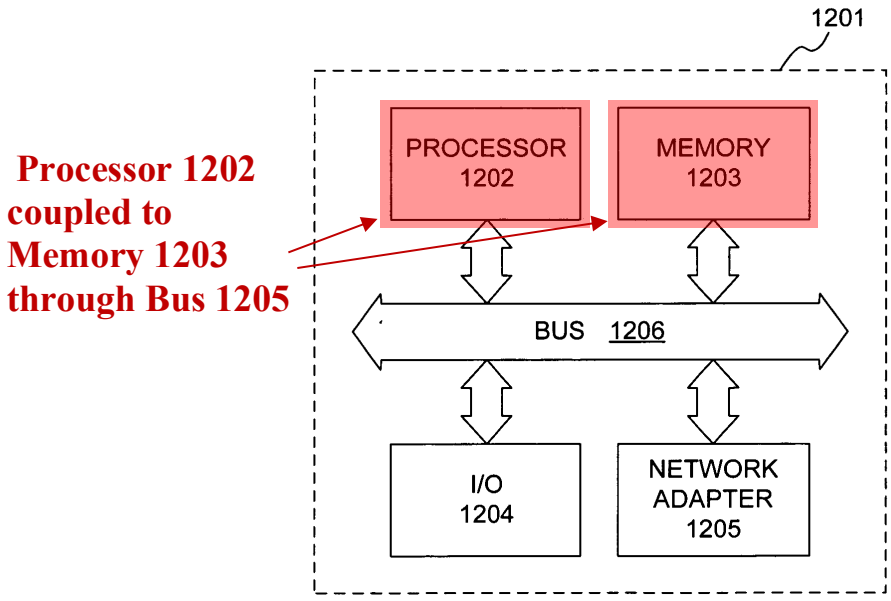
177. Bodart describes that:

> The term virtual assistant 101 is used to describe **software** designed to simplify, streamline and enhance the often repetitive activities which occupy the daily lives of many people.

Bodart, [0020].

178. As explained above, a POSITA would have been generally familiar with computer software. Section IX.D. A POSITA would have understood that computer software was typically implemented as executable program instructions stored on a non-transitory computer readable medium. Bodart itself contemplates the use of volatile and non-volatile memory to access user data. Bodart, [0023]. Thus, to the extent the preamble is limiting, Bodart's discussion of virtual assistant software renders obvious "*non-transitory computer readable storage medium comprising program instructions*."

179. Fig.12 of Bodart (below) further illustrates an "operating environment in which one or more aspects of the invention may be implemented." Bodart, [0075]. The device 1201 of Fig.12 "may be any sort of hardware minimally containing the components shown [in Fig.12]," including a memory 1203 in communication with a processor 1202. Bodart, Fig.12.



Bodart, Fig. 12 (annotated).

180. Bodart further teaches that the virtual assistant 101 can include a rules engine that "may be compared with a software compiler which translates user input (i.e. computer source code) into **a series of instructions which a computer processor can understand** (i.e. computer object code)." Bodart, [0024]. Further, Bodart explains that:

> Virtual assistant 101 may also access user files, preferences and account information 210 stored either locally on **the same machine**, or on **remote servers and devices**.

Bodart, [0042].

181. Bodart explains that the information sources and interfaces associated with the virtual assistant may be remotely connected to the virtual assistant via a network. Bodart, [0028], [0030], [0074].

182. Bodart's further teachings of processor 1202, memory 1203, compiling computer-understandable instructions, and remote connection to information sources and communication interfaces further render obvious the preamble. Bodart, [0075].

b. [13.1] provide a first service to a first user by a first provider, wherein the first service is telephone service, and wherein the first user has a first telephone number.

183. Bodart renders obvious this limitation; *see* [1.2] and [1.3] above.

c. [13.2] receive a request to provision a second service for the first user, wherein the second service is provided by a second provider, and wherein the second provider is different than the first provider.

184. Bodart in view of Dharmarajan render obvious this limitation; *see* [1.4] and [1.5] above.

d. [13.3] responsive to receiving the request: create a mapping from the first telephone number to a second

identifier (ID), wherein the second ID is utilized by the second provider to identify the first user.

185. Bodart in view of Banister and Dharmarajan render obvious this limitation; *see* [1.6] above.

e. [13.4] activate the second service with the second provider utilizing the second ID as a user ID for the first user; and

186. Bodart in view of Banister and Dharmarajan render obvious this limitation; *see* [1.7] above.

f. [13.5] associate the second service with the first telephone number, wherein the first telephone number is associated with routing and interworking data necessary to utilize the second service.

187. Bodart in view of Banister and Dharmarajan render obvious this limitation; *see* [1.8] above.

8. Claim 14

a. [14.0] The non-transitory computer readable storage medium as recited in claim 13, wherein the program instructions are further operable to associate a plurality of services with the first telephone number,

188. Bodart in view of Banister render obvious this limitation; *see* [2.0] above.

b. [14.1] wherein the plurality of services are provided by a plurality of providers different than the first provider, and wherein the plurality of services are utilized by the first user.

189. Bodart renders obvious this limitation; *see* [2.1] above.

X. DE CASTRO RENDERS OBVIOUS CLAIMS 1, 2, 6-8, AND 12-14

A. Summary of de Castro

190. De Castro relates to "a system and method for providing a telephony-enabled service via a message-based API interface." De Castro, [0002]. De Castro explains that because many third party developers use a variety of preferred programming languages, "it is difficult for a hosting facility to provide compatible browsers for each of the possible programming languages and protocols." De Castro, [0035]. Further:

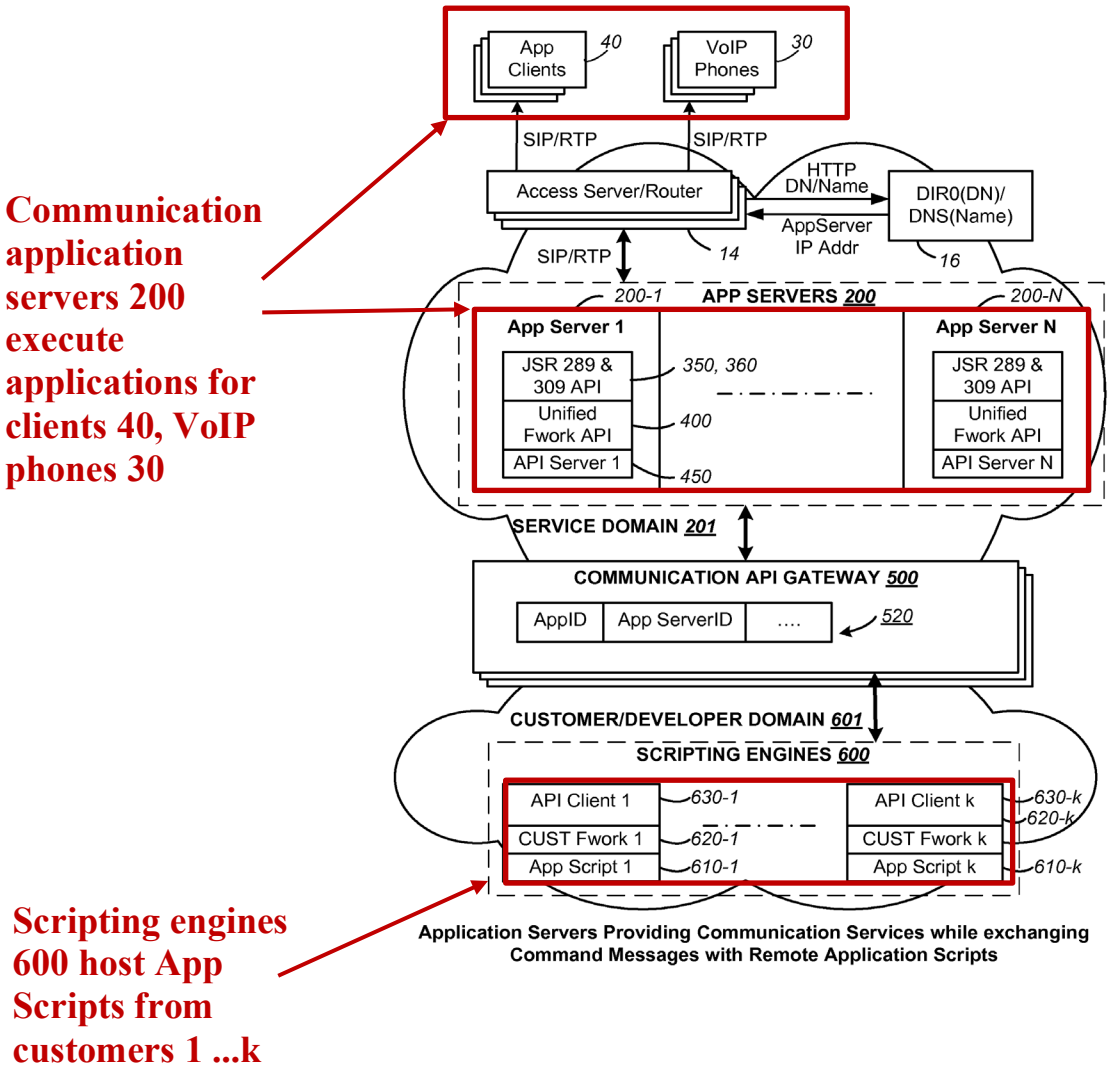
> different scripts running in the same virtual machine may contend with each other, resulting in poor performance, memory leaks and, worst still, object collisions. Also, having to support a wide set of possible scripts make resource provisioning and budgeting in the communication platform difficult and indefinite.

De Castro, [0036].

191. Accordingly, de Castro proposes a "unified framework for call control and media control" to address these issues. De Castro, [0038].

192. The architecture of de Castro includes "a plurality of communication application servers 200" that "allow[] application clients 40 ... to access the application hosted in the communication application servers." De Castro, [0115]. Further, "a plurality of scripting engines 600" are used to "host and process the customers' applications scripts 610." De Castro, [0119]. VoIP phones 30,

application clients 40 (i.e., users), application servers 200, and scripting engines 600 are illustrated in Fig.14 below.



De Castro, Fig.14 (annotated)

193. An example messaging API protocol used by the communication application servers is the "Rayo™ protocol." De Castro, [0138]-[0139]. De Castro

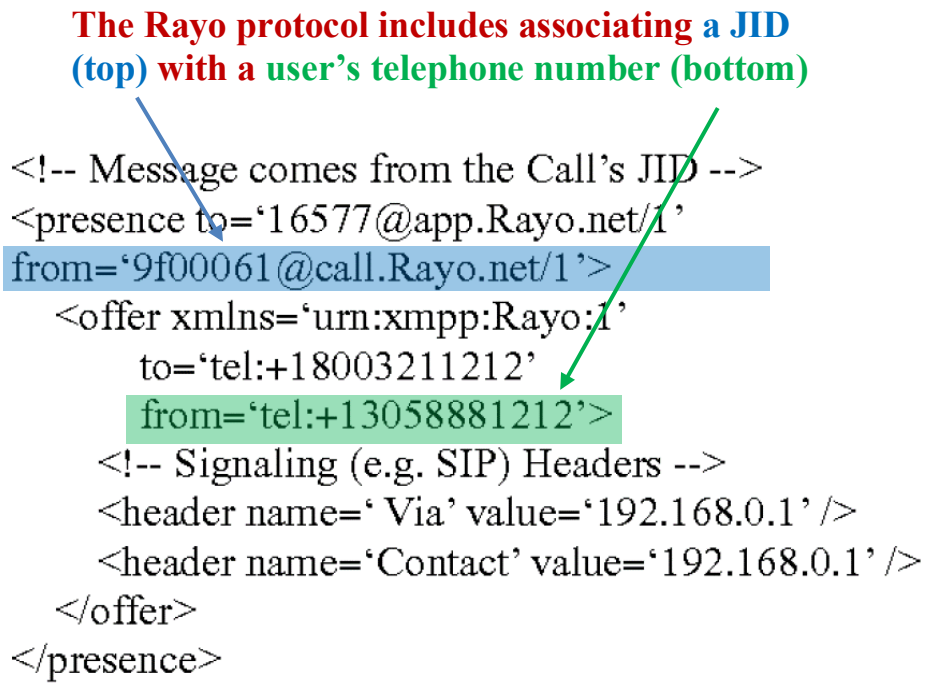
teaches that the API server 450 can execute an application via the Rayo protocol to control a call using a JID:

> The following are examples of **the commands** supported by the unified framework API 400 through the API server 450 and a specification of the messaging API protocol (hereinafter referred to as the "Rayo™" protocol).
>
> **Calls**
>
> The Rayo protocol primarily deals with calls. Inbound calls originate from the PSTN or via SIP and are offered to Rayo clients[1] via XMPP using a Jabber Identifier (JID). **Each call is in turn represented by its own unique JID allowing a two way conversation between the Rayo client and the server that's handling the call signaling and media**.

De Castro, [0138]-[0139].

194. The "JID allow[s] a two way conversation between the Rayo client and the server that's handling the call signaling and media." De Castro, [0139]. In the excerpt below, de Castro shows how the Rayo server notifies a Rayo client of an incoming call with a message that indicates both the user's telephone number ("from='tel:+13058881212'") and a JID ("from= '9f00061@call.Rayo.net/1'") for the call. De Castro, [0141].

[1] In Fig.14 and the corresponding disclosure (e.g., [0115], [0116]), de Castro refers to the end users of the App scripts as "clients," such as "App Clients 40." These "App Clients 40" should not be confused with the "Rayo clients" described in [0141] of de Castro, for example. The "*Rayo* clients" correspond to the customers/developers whose app scripts are hosted by the App servers 200. The "Rayo server" corresponds to an App server 200.



De Castro, [0141] (annotated)

195. De Castro is relevant art to the '254 patent. For example, de Castro describes a "unified framework" that is used to provide application clients access to a variety of communication applications from a variety of developers (also referred to as "customers"). De Castro, Abstract, [0039]. Similarly, the '254 patent provides a "unified services enabler platform" to access "a variety of services and applications from multiple different ecosystems." '254 patent, Abstract. De Castro proposes a defined "messaging API protocol" to "enable[] an application script to pass commands remotely to the communication application server to operate it." De Castro, [0109], Abstract. Similarly, the '254 patent proposes allowing service providers to "integrate their portal into the API [of the unified services enabler

platform].” ’254 patent, 4:60-62. De Castro is also reasonably pertinent to the particular problem of the ’254 patent. For example, de Castro describes how service providers using electronic IDs such as the Jabber ID can provide communications services to clients by telephone (e.g., ’254 patent, Abstract, 11:33-42; De Castro, [0138]-[0141]).

B. Detailed Analysis of Claims

1. Claim 1

a. [1.0] *A telephone system, comprising:*

196. To the extent that the preamble is limiting, de Castro discloses it.

197. De Castro teaches a “network architecture” that includes a “**service domain 201** and a customer/developer domain 601,” and an “**API gateway 500**.” De Castro, [0114] and [0120], Fig.14. De Castro explains that the service domain 201 provides “telephony and communication services”:

> “In the **service domain 201** is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 **provide telephony and communication services** to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers.”

De Castro, [0115].

198. De Castro further explains that the API gateway 500 is a messaging broker for the service domain 201:

> "one or more communication API gateway 500 is deployed on the network to serve as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201."

De Castro, [0120].

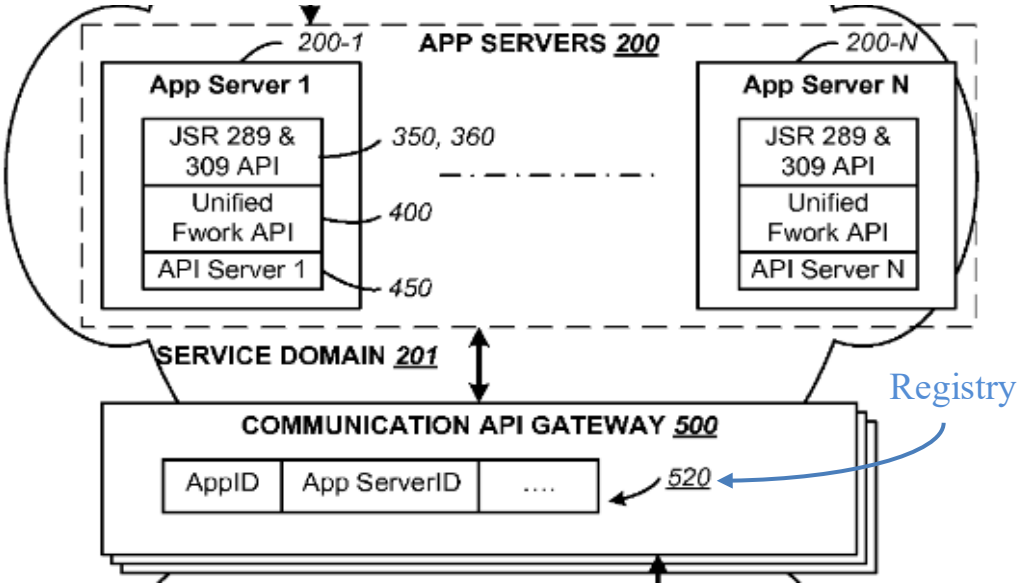
199. The service domain 201 having a plurality of communication application servers 200 that provide telephony and communication services, as well as the API gateway 500 that brokers messages to the service domain 201 from the customer/developer domain 601 are a "*telephone system*."

200. Therefore, de Castro discloses "*[a] telephone system*," as recited in the preamble.

b. [1.1] *a database; and*

201. De Castro renders obvious this limitation.

202. De Castro, Fig.14, shows that the API Gateway 500 includes a Registry 520. A Registry 520 includes a database.



De Castro, Fig. 14 (partial annotated)

203. For example, de Castro further explains that the registry 520:

> "registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630."

De Castro, [0130].

204. Furthermore, de Castro explains that the data in the registry 520 is accessed:

> FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630

> (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by **referencing the information in the registry 520**, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601.

De Castro, [0130].

205. A POSITA would understand that the registry 520 renders obvious "*a database*," because registering applications and logging properties and statuses of the application servers 200 and API clients 630, and accessing the information, describes the function and capability of a database.

206. Alternatively, it would have been obvious to implement registry 520 as a "*database*." Registries were commonly implemented as databases for storing information like that stored by the registry 520. Berkley, Abstract, 6:55 – 7:3; Goldman, 11:8-12.

207. Therefore, de Castro renders obvious "*a database*."

c. [1.2] *a server coupled to the database,*

208. De Castro renders this limitation obvious.

209. **First**, de Castro teaches that the service domain 201 comprises a plurality of communication application servers 200:

> "In the service domain 201 is a plurality of **communication application servers 200**, **such as App Server 1** to App Server N. As in FIG. 13, the communication application servers 200 provide

> telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers."

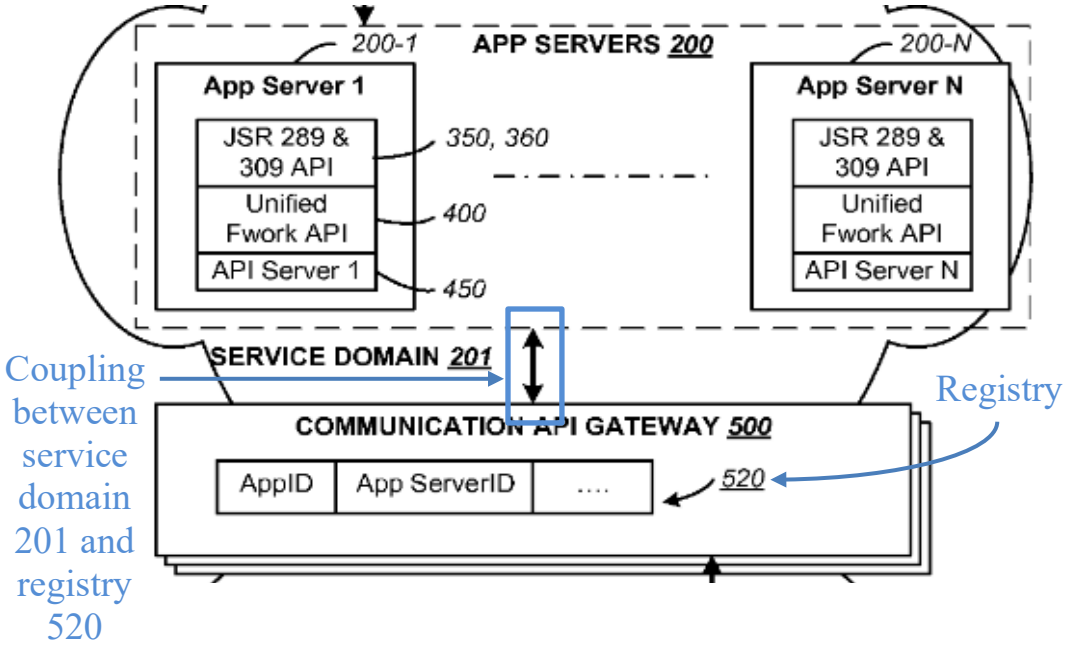
De Castro, [0115].

210. Any of the communication application servers 200, such as App Server 1, teaches the "*server*."

211. **Second**, De Castro teaches that the registry 520 of the API Gateway 500 is coupled to the communication application servers 200 because de Castro teaches that the API Gateway 500 brokers messages with the communication application servers 200:

> "one or more communication API gateway 500 is deployed on the network to serve as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201."

De Castro, [0120]



De Castro, Fig. 14 (partial annotated)

212. The communication application servers are coupled to the registry 520, such that the registry 520 can register the communication application servers 200. For example, de Castro teaches that the registry 520:

> "**registers the communication application servers 200** in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status **associated with each communication application server 200** and the properties and status associated with each API clients 630."

De Castro, [0130].

213. Therefore, de Castro teaches that App Server 1 200-1 ("*server*") is in communication with (double arrows) Registry 520. De Castro, Fig.14.

214. Thus, de Castro discloses "*a server coupled to the database*."

d. [1.3] *wherein the server is utilized by a first provider to provide a first service to a first user, wherein the first service is telephone service, and wherein the telephone service is associated with a first telephone number*

215. De Castro renders this limitation obvious.

216. First, de Castro teaches that at least one of the communication application servers 200 "*is utilized by a first provider to provide a first service to a first user*." de Castro teaches that a customer 1 ("*first provider*") uses a scripting engine to host and process an app script: **"a customer 1** has the scripting engines 600 host and process an application script such as App Script 1 610-1." de Castro, [0119]. To provide application clients 40 with an application service, "API client 1 [associated with customer 1] 630-1 [] interprets App Script 1 and renders it in terms of the commands of the finite set defined at the API server 450 of the communication application server 200." Ex.1008, [0119]. The customers described by de Castro provide App scripts embodying communication applications, and are therefore "*provider[s]*."

217. De Castro further teaches that clients 40 access the communication applications hosted on communication application servers 200 ("*server is utilized by a first provider to provide a first service to a first user*"):

> "the communication application servers 200 provide telephony and communication services to applications and allows **application clients 40** such as, for example, VoIP phones to access **the applications hosted in the communication application servers**."

De Castro, [0115].

218. De Castro explains, with respect to the embodiment of Fig.1, that an App Script defines a communication application: "The application script 210 defines **the communication application** 300 and directs how a call is to be handled." De Castro, [0011]. By analogy, the App Script 610-1 "defines" customer 1's ("*first provider['s]*") communication application "and directs how a call is to be handled." Thus, the communication application embodied by App Script 610-1 is the "*first service*."

219. Fig.14 shows that App Scripts 610, including the App Script 610-1 embodying the first service, are accessible by the clients 40 through the communication application servers 200:

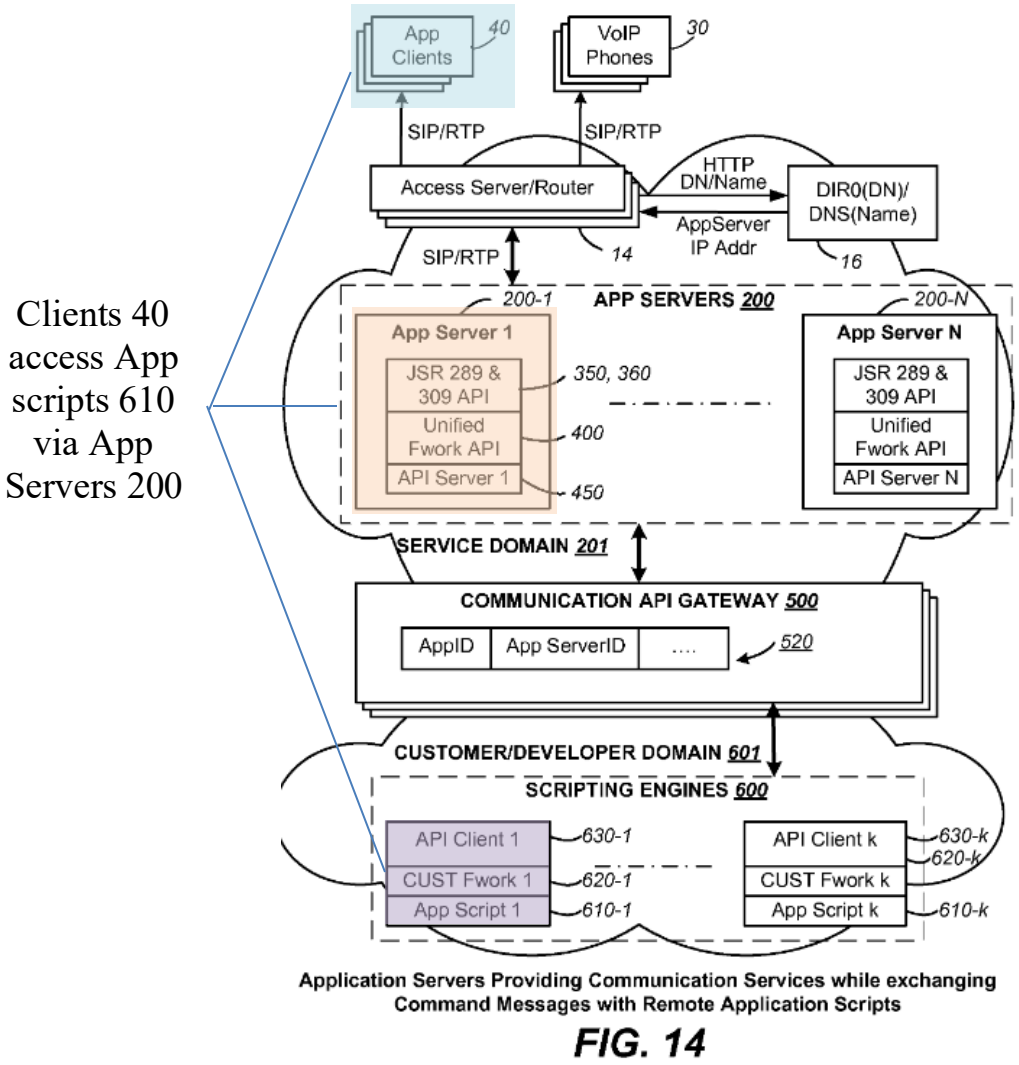


FIG. 14

De Castro, Fig. 14 (annotated)

220. Any of the clients 40 may be the "*first user*." Further, it would have been obvious to a POSITA any of the communication application servers 200 would provide any of the clients 40 access to any communication application hosted by the communication application servers 200, including the communication application embodied in App Script 610-1 ("*first service*").

221. **Second**, de Castro teaches "*wherein the first service is telephone service*." de Castro explains that "application clients such as VoIP phones 30, is [sic] able to access a specific communication application by calling a dialed number or a URL." de Castro, [0116].

222. De Castro further explains in the "Summary and Objects of the Invention" that the communication applications provide "*telephony and communication services*": "the communication application server is among a group of similar communication application servers on the network to provide **telephony and communication** services to a plurality of customers with application scripts hosted on scripting engines." De Castro, [0039]. The Abstract of de Castro explains that the described unified framework "conform[s] to a telephony object model." de Castro, Abstract.

223. A POSITA would therefore understand that the communication applications hosted by the communication application servers 200, including the communication application ("*the first service*") embodied in App Script 610-1, is "*telephone service*."

224. **Third**, de Castro teaches that "*the telephone service is associated with a first telephone number*."

225. For example, de Castro explains that a communication application can be accessed by calling a number:

> [A]pplication clients such as VoIP phones 30, is [sic] able to access a specific communication application by calling a dialed number or a URL. The call is routed by an access server/router 14 into the IP network 10 to one of the communication application servers 200, such as App server 1.

De Castro, [0116].

226. The telephone number of the client seeking to access a communication application by dialing a number is the "*first telephone number*." de Castro provides exemplary communication application actions and components for incoming calls from the user's telephone number:

> Incoming Calls
>
> <!-- Message comes from the Call's JID -->
> <presence to='16577@app.Rayo.net/1'
> from='9f00061@call.Rayo.net/1'>
> <offer xmlns='urn:xmpp:Rayo:1'
> to='tel:+18003211212'
> from='tel:+13058881212'>
> <!-- Signaling (e.g. SIP) Headers -->
> <header name=' Via' value='192.168.0.1' />
> <header name='Contact' value='192.168.0.1' />
> </offer>
> </presence>.

De Castro, [0141].

227. Thus, de Castro teaches that the commands of the communication application explicitly include the client's telephone number. Accordingly, de Castro teaches that the communication application ("*the first service*") the client is accessing via telephone is "*associated with*" the client's telephone number ("*a first telephone number*").

228. Thus, de Castro renders obvious "*wherein the server is utilized by a first provider to provide a first service to a first user, wherein the first service is telephone service, and wherein the telephone number is associated with a first telephone number.*"

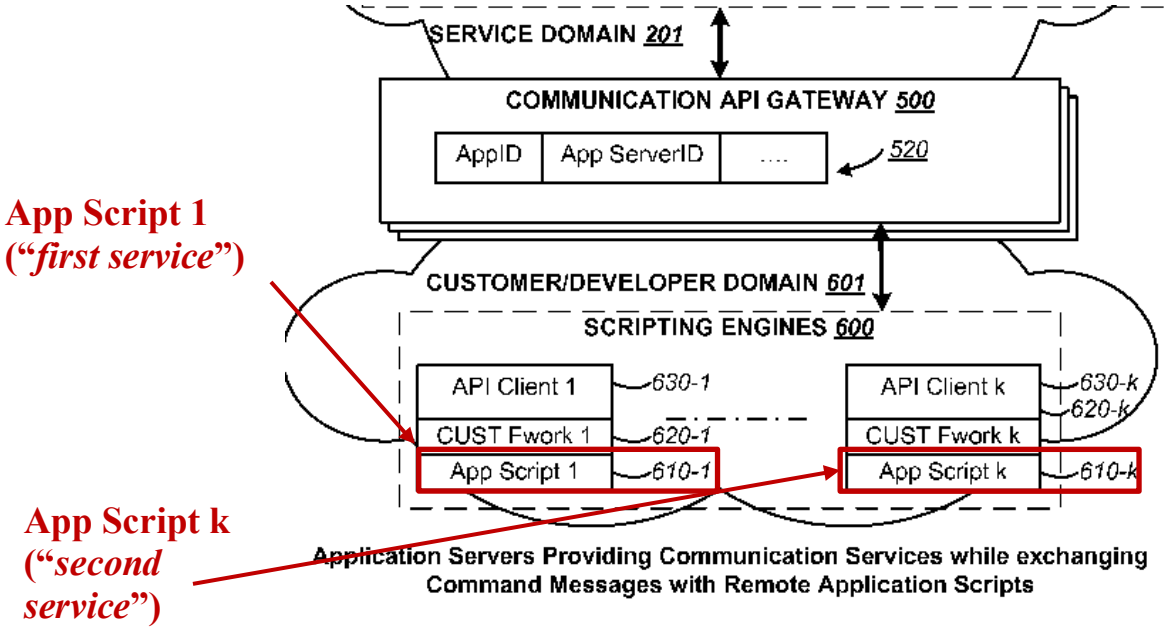
e. [1.4] *wherein the server is configured to: receive a request to provision a second service for the first use*r,

229. Referring to Fig.14, de Castro teaches that customers/developers host their App Scripts 610 ("*service[s]*") in the scripting engines 600:

> in the customer/developer domain 601 is a plurality of scripting engines 600 for the customers to host and process the customers' **application scripts 610**. For example, a customer 1 has the scripting engines 600 host and process an application script such as App Script 1 610-1.

De Castro, [0119], Fig.14.

230. For instance, Fig.14 of de Castro reproduced below shows "App Script 1" ("*first service*") and "App Script k" ("*second service*") hosted in respective scripting engines. De Castro, Fig.14, [0119].



De Castro, Fig.14 (partial annotated)

231. Second, de Castro teaches one of the application servers 200 ("*server*") receiving, from an application client ("*first user*") a call to access to one of the communication applications ("*request to provision a second service*"):

> application clients such as VoIP phones 30, [are] able to **access a specific communication application by calling a dialed number** or an URL. The call is routed by **an access server/router 14** into the IP network 10 **to one of the communication application servers 200**, such as App server 1. The routing is performed by **looking up** a directory service DIR0 16 with the IP address/dialed number in order to return **the address of the communication application server** executing the specific communication application.

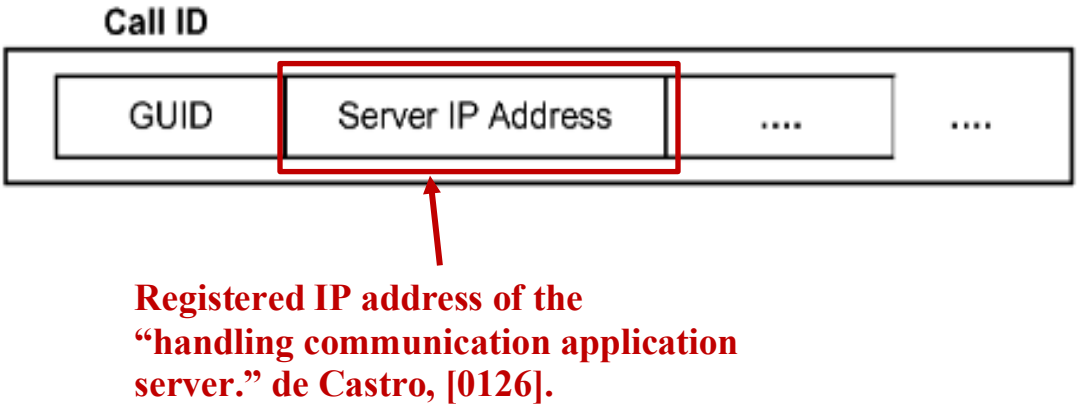
De Castro, [0116].

232. De Castro teaches that an incoming call prompts the app server 200 to register a call ID for the call and start a call actor:

> When a call 1 is received into the communication application server 200, the call manager register [sic] the Call ID into a call ID registry 262 and start a call actor 1 270-1, and similarly for call 2, . . . , and call i.

De Castro, [0124].

233. As illustrated in Fig.16 of de Castro, reproduced below, the call ID "includes a GUID (Globally Unique Identifier) and **a server IP address**. In this way, the call and **the handling communication application server are readily identified**." De Castro, [0126].



De Castro, Fig.16 (annotated)

234. Thus, de Castro teaches, in response to an incoming call from the client 40, that the access server router 14 identifies an IP address of a server to execute a communication application for the call. De Castro, [0116], [0124]. Then,

the App server 200 registers the IP address and a GUID for the call to provision an App Script for the client 40 dialing in. De Castro, [0126]. Therefore, a POSITA would have understood that the incoming call that is registered with the GUID and IP server address is "*a request to provision a [] service for the first user*." De Castro, [0116]. It would have been obvious that one of the app servers 200 would receive a routed call from a client to access App Script k (610-k) ("*a second service*") as shown in Fig.14, for example. Thus, a POSITA would have found it obvious that one of the app servers 200 "*receive[s] a request to provision a second service*." De Castro, [0116].

235. Thus, de Castro teaches that the app servers 200 receive a routed call from a client 40 to access a communication application, such as the communication application embodied in app script 610-k, which renders obvious "*wherein the server is configured to: receive a request to provision a second service for the first user*."

f. [1.5] *wherein the second service is provided by a second provider, and wherein the second provider is different than the first provider*;

236. As explained above at Section X.B.1.c, de Castro teaches that "**customer 1**" ("*a first provider*") uses "the scripting engines 600 [to] host and process an application script such as App Script 1 610-1" ("*first service*"). Fig.14 of de Castro further shows another application script "**App Script k**" ("*second*

service") being hosted by a scripting engine. *See* Section X.B.1.e; De Castro, [0119].

237. It would have been obvious to a POSITA that the App Script k would be "*provided by a second provider*" that is "*different than the first provider*." For instance, de Castro teaches that its telephone system architecture provides "a messaging broker between the **plurality of customers with application scripts** and the group of communication application servers." De Castro, [0039]. De Castro explains that the customer/developer domain can be used by multiple customers/developers to host app scripts:

> [I]n the customer/developer domain 601 is a plurality of scripting engines 600 for the **customers** to host and process the **customers'** application scripts 610. For example, a customer 1 has the scripting engines 600 host and process an application script such as App Script 1 610-1.

De Castro, [0119].

238. Thus, de Castro teaches that the app scripts 610 are provided by a plurality of customers/developers, one of which is "customer 1" associated with App Script 1. [0119]. Further, de Castro teaches that each scripting engine 600 uses a "**customer-specific** framework 620," which "supplies a set of **customer-specific** libraries." De Castro, [0119]. De Castro further explains that, in its system, a "**customer using the communication services can code the application script**

in a preferred programming language using a custom framework providing a set of **customer-specific libraries**." De Castro, [0041].

239. Thus, de Castro teaches a plurality of customers/developers hosting their app scripts 610 based on their preferred programming languages and customer-specific libraries. It would have been obvious to a POSITA that the customer/developer hosting App Script k would be using a different customer-specific framework than the customer/developer hosing App Script 1. Accordingly, it would have been obvious to a POSITA that the App Script k "*is provided by a second provider, and wherein the second provider is different than the first provider*."

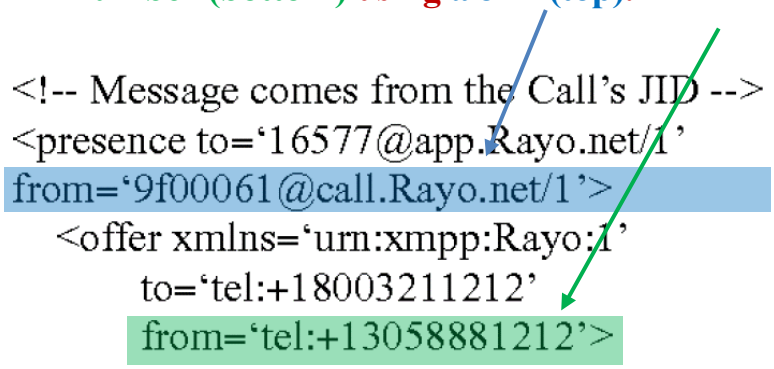
g. [1.6] *responsive to receiving the request: create a mapping from the first telephone number to a second identifier (ID), wherein the second ID is utilized by the second provider to identify the first user*;

240. As explained above at Section X.B.1.e, de Castro teaches that clients 40 can access communication applications by dialing a phone number associated with an app script 610. De Castro, [0116]. The communication application server 200 receiving a routed call renders obvious "*receive a request to provision a second service*." Further, de Castro describes commands of a communication application responding to "[i]ncoming calls" ("*responsive to receiving the request*"). De Castro, [0141]. For instance, de Castro explains that "inbound calls

... are offered to Rayo clients via XMPP using a Jabber Identifier (JID)." De Castro, [0139].

241. De Castro describes offering an inbound call from a telephone number ("*the first telephone number*") to a Rayo client using a "from" JID ("*second ID*"):

Rayo server offers a call from a user's telephone number (bottom) using a JID (top).



De Castro, [0141] (annotated)

242. Once the call has been offered by the Rayo server, de Castro shows that the Rayo client uses the JID in its command messages to identify and control the call. De Castro, [0141]. For example, the portion of de Castro, reproduced below shows a command message from a Rayo client invoking an "Accept" command in response to the Rayo server offering the call:

The Rayo client can now control the call by using one of the following commands.

```
<!-- Accept (e.g. SIP 180/Ringing). Only applies to
incoming calls. -->
<iq type='set' to='9f00061@call.Rayo.net/1'
from='16577@app.Rayo.net/1'>
   <accept xmlns='urn:xmpp:Rayo:1'>
      <!-- Sample Headers (optional) -->
      <header name="x-skill" value="agent" />
      <header name="x-customer-id". value="8877" />
   </accept>
</iq>
```

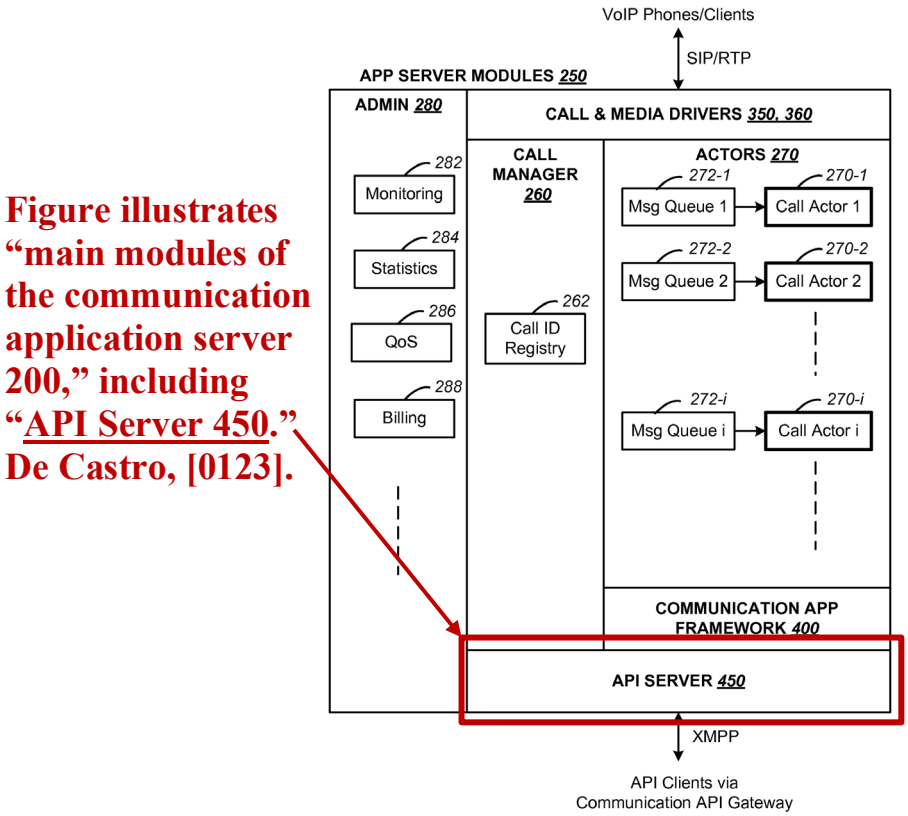
JID now used as "to" address by Rayo client to accept incoming call

De Castro, [0141] (annotated)

243. In the message shown above, the JID previously indicated by the Rayo server as the "from" JID has been switched to the "to" JID. In other words, the Rayo client now uses the JID previously indicated by the Rayo server as corresponding to the user's telephone number to identify the call. Thus, a POSITA would have understood that the Rayo server associating the end user's telephone number with a JID to offer the call to a Rayo client teaches to "*create a mapping from the first telephone number to a second identifier (ID)*."

244. De Castro teaches that the messages and commands described above (e.g., Accept, Answer) are handled through the API server 450 of a communication application server 200: "The following are examples of the commands supported by the unified framework API 400 through the **API server 450**" De Castro, [0138].

245. Fig. 15 of de Castro (reproduced below) shows that the API server 450 is part of the communication application server 200:



De Castro, Fig.15 (annotated), [0123]

246. Thus, de Castro teaches that the communication application server 200 ("*the server*") is configured to "*create [the] mapping from the first telephone number to [the] second identifier (ID)*." Additionally, because de Castro teaches the Rayo server indicates the JID in response to an incoming call (de Castro, [0139], [0141]), de Castro renders obvious that the mapping is created "*responsive to receiving the request*." *See* Section X.B.1.e).

247. Further, it would have been obvious to a POSITA that the JID "*is utilized by the second provider to identify the first user*." De Castro teaches, in the context of the Rayo protocol, that "[e]ach call is ... represented by its own unique JID." De Castro, [0139]. A POSITA would have understood that the JID representing the call is "*utilized ... to identify*" the end user (e.g., client 40). De Castro explains that "The JID follows a specific format. In XMPP the JID is constructed as <node>@<domain>/<resource>." De Castro, [0140]. Further, XEP-0327 (the "Rayo Specification") describes that, in a JID addressing scheme, all of the actors (e.g., client, call) "are represented by XMPP entities with a JID of their own." Rayo Specification, [5.2].

Actor	JID format	Example JID
Server	[service domain]	shakespeare.lit
Client	any JID	juliet@capulet.lit/balcony
Call	<call ID>@[<call sub-domain>.]<service domain>	f88eh2@call.shakespeare.lit
Mixer	<mixer name>@[<mixer sub-domain>.]<service domain>	conf1@mixer.shakespeare.lit
Call Component	<call ID>@[<call sub-domain>.]<service domain>/<component ID>	f88eh2@call.shakespeare.lit/8f83jf
Mixer Component	<mixer name>@[<mixer sub-domain>.]<service domain>/<component ID>	conf1@mixer.shakespeare.lit/932eu
Server Component	<service domain>/<component ID>	shakespeare.lit/f3fg4

Rayo client can use "any JID."

JIDs for client, call, mixer, and component actors have construction: <node>@<domain>/<resource>

Rayo Specification, Table 1 (annotated), [5.2]

248. As shown in Table 1 above, the JIDs for all of the actors (client, call, mixer) include the same construction described by de Castro ("<node>@<domain>/<resource>"). De Castro, [0140]; Rayo Specification, [5.2]. De Castro shows that an app server offers a call from a telephone number ("tel:+13058881212") using a JID ("9f00061@call.Rayo.net/1"). This JID also has the construction "<node>@<domain>/<resource>," where "9f00061" is the node, "call.Rayo.net" is the domain, and "1" is the resource."

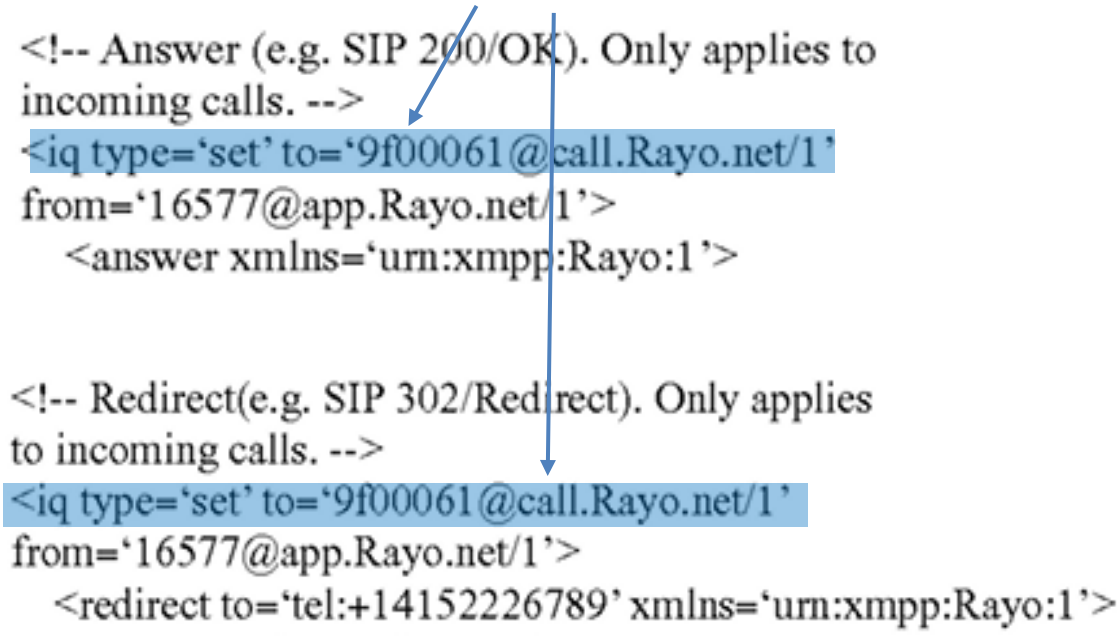
Rayo server offers a call from a user's telephone number (bottom) using a JID (top).

```
<!-- Message comes from the Call's JID -->
<presence to='16577@app.Rayo.net/1'
from='9f00061@call.Rayo.net/1'>
   <offer xmlns='urn:xmpp:Rayo:1'
         to='tel:+18003211212'
         from='tel:+13058881212'>
```

De Castro, [0141] (annotated)

249. Once the app server has offered the call to the Rayo client, the Rayo client controls the call with commands (e.g., Answer, Redirect) using the call's JID ("9f00061@call.Rayo.net/1"):

Rayo client issues commands for the call using JID only.



De Castro, [0141] (annotated)

250. The "to" JID in the command messages (e.g., Answer, Redirect) from the Rayo client are thus used to direct the commands to the appropriate call maintained by the app server 200. Accordingly, the "to" JID in these command messages addresses the commands to the user on the call. A POSITA would have thus understood that the JID of de Castro "*identif[ies] the first user*."

251. Thus, de Dastro teaches using JIDs to control calls from any of the app clients 40, where the calls are maintained by the app server 200. It would have been obvious that the scripting engine 600-k would utilize a JID to control a call for "*the first user*" calling into the telephone number associated with "*the second*

service." Thus, it would have been obvious for the customer ("*second provider*") associated with scripting engine 600-k to utilize the JID "*to identify*" the client 40 ("*first user*").

h. [1.7] [*responsive to receiving the request:*] *activate the second service with the second provider utilizing the second ID as a user ID for the first user*; *and*

252. De Castro describes the process by which the App Scripts of the customers/developers, such as App Script k ("*the second service*"), are executed for clients 40 calling a number associated with the App Script. *See* De Castro, Fig.14, [0118]-[0124]. In response to incoming calls, the communication application server 200 registers Call IDs of the incoming calls "into a call ID registry 262 and start[s] a call actor 1 270-1." De Castro, [0124]. Once a call is received and a call actor has been started, de Castro explains that the command messages are queued and applied to calls in order:

> As command messages from the API clients 630 (see FIG.14) are received by the application server 200, they are identified with a call actor and **queued in an associated message queue 272, which are then applied to the call in order**.

De Castro, [0124].

253. From the perspective of the customers/developers, De Castro also explains how the command messages (via API clients 630) are rendered from their respective App Scripts and transmitted to control the calls:

> [A] customer 1 has the scripting engines 600 host and process an application script such as App Script 1 610-1... The processing is performed by an API client 630 such as API client 1 630-1 which **interprets App Script 1 and renders it in terms of the commands of the finite set defined at the API server 450** of the communication application server 200. The API client 630 then packages the commands as messages in accordance with the messaging API protocol to be sent to a connected communication application server 200 such as App server 1 200-1.

De Castro, [0119].

254. De Castro explains that rendering the commands based on the finite set defined at the API server allows "customers of the communication application server on the network to execute their applications as a series of these commands." [0118]. Next, de Castro explains that the rendered commands can be parsed, queued, and executed by the app server 200, as mentioned above:

> [w]hen **a command** carried in a message is received at the communication application server 200, the API server 450 will parse out the command in accordance with the messaging API protocol and **allow the command to be queued with the associated call (actor) to be executed**.

De Castro, [0122].

255. A POSITA would have understood that registering a call ID of an incoming call and starting an associated call actor for an App Script 610 would include "*activat[ing]*" the App Script 610. In this regard, registering the call ID

and starting the call actor allow command messages from the customers to be queued, sent and executed to control the call, thus "*activat[ing]*" the commands and components of the App Script 610 for the client 40 on the call. De Castro, [0118], [0122].

256. In light of de Castro teachings of registering calls and queueing resulting commands of an App Script to be executed, it would have been obvious to a POSITA that the communication application server 200 would execute App Script 610-k ("*the second service*") in response to an incoming call from a client 40, which renders obvious "[*responsive to the request:*] *activate the second service*."

257. Further, it would have been obvious to a POSITA that the app server 200 would "*activate*" App Script k "*utilizing the second ID as a user ID for the first user*." For instance, de Castro teaches that a Rayo client ("*second provider*") uses the JID to direct commands from the communication application to the user in the call. De Castro, [0141]; Section X.B.1.g. As explained above, de Castro renders obvious that the "*second ID is utilized by the second provider to identify the first user*." *See* Section X.B.1.g. It would have been obvious to a POSITA that the customer associated with scripting engine 600-k ("*second provider*") would similarly utilize the JID ("*second ID*") to direct commands from the App Script

610-k to the client 40 during a call session. De Castro, Fig.14, [0119]. This renders obvious "*utilizing the second ID as a user ID for the first user.*"

i. [1.8] [*responsive to receiving the request:*] *associate the second service with the first telephone number, wherein the first telephone number is associated with routing and interworking data necessary to utilize the second service*.

258. De Castro teaches that the API server 450 can execute an application via the Rayo protocol to control a call using a JID:

> The following are examples of the commands supported by the unified framework API 400 through the API server 450 and a specification of the messaging API protocol (hereinafter referred to as the "Rayo™" protocol).
>
> **Calls**
>
> The Rayo protocol primarily deals with calls. Inbound calls originate from the PSTN or via SIP and are offered to Rayo clients via XMPP using a Jabber Identifier (JID). **Each call is in turn represented by its own unique JID allowing a two way conversation between the Rayo client and the server that's handling the call signaling and media**.

De Castro, [0138]-[0139].

259. As discussed at Section X.B.1.g, de Castro teaches that the JID is mapped to the telephone number ("*the first telephone number*") of the app client 40 accessing the App Script associated with the number the client is calling. The app server 200, via the access server/router 14, maintains the call for which the App Script is executed. De Castro, [0116]. A POSITA would therefore understand that

the app server 200 acts as an intermediary between the customers/developers hosting their App Scripts in the Customer/Developer domain 601, and the telephone call from app client 40 routed through the Access Server/Router 14. *See* de Castro, Fig.14. Thus, the app server 200 "*associate[s]*" the App Script with both the telephone number of the app client 40 and the JID used by the scripting engine 600 to send commands for controlling the call.

260. Because the app server 200 executes commands from the App Script for a phone call from the client 40's telephone number, the app server 200 "*associate[s]*" the App Script "*with the first telephone number*." Thus, it would have been obvious to a POSITA that the app server 200 would "*associate*" the App Script 610-k ("*the second service*") with the client 40's telephone phone number ("*the first telephone number*").

261. Further, it would have been obvious to a POSITA that "*the first telephone number is associated with routing and interworking data necessary to utilize the second service*." De Castro teaches that "the call manager register [sic] the Call ID into a call ID registry 262 and start[s] a call actor 1 270-1, and similarly for call 2, . . . , and call i." De Castro, [0124]. De Castro also teaches that "[i]nbound calls originate from the PSTN [public switched telephone network] or via SIP [session initiation protocol] and are offered to Rayo clients via XMPP using a Jabber Identifier (JID)." De Castro, [0139]; *see also* [0003] (providing

background description of the PSTN) and [0018] (providing background description of SIP). A POSITA would understand that a phone call from the PSTN is associated with the client 40's telephone phone number, while the command messages received by the app server 200 from the scripting engines 600 use the JID. It would have been obvious that the app servers 200 would store an association between the client 40's phone number and the JID in the call ID registry 262. In this way, the app servers 200 can identify the correct phone call corresponding to the JID indicated in the command messages from the developers.

262. Thus, de Castro renders obvious associating a telephone number and JID in a call ID registry to identify the correct call for which to execute commands of the App Scripts. The stored association would be "*routing and interworking data necessary to utilize the second service*." This renders obvious to "*associate the second service with the first telephone number, wherein the first telephone number is associated with routing and interworking data necessary to utilize the second service*."

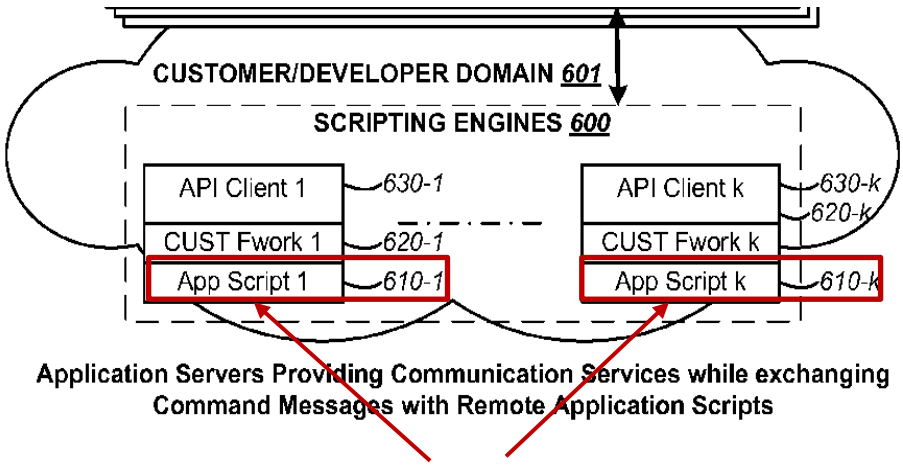
2. Claim 2

a. [2.0] *The telephone system as recited in claim 1, wherein the server is further configured to associate a plurality of services with the first telephone number*,

263. It would have been obvious to a POSITA in view of de Castro that the communication application server 200 is "*configured to associate a plurality of*

services with the first telephone number." As explained above at Section X.B.1.i, de Castro renders obvious that the communication application server 200 "*associate[s]*" the App Script 610-k ("*the second service*") with the telephone number of the client 40 ("*the first telephone numbe*r"). De Castro, [0141].

264. De Castro teaches that "in the customer/developer domain 601 is a plurality of scripting engines 600 for the customers to host and process the customers' **application scripts 610**." De Castro, [0119]. Thus, de Castro teaches that a plurality of application scripts 610 ("*plurality of services*"), including the App Scripts 610-1 ("*the first service*") and 610-k ("*the second service*") are hosted in the scripting engines 600. De Castro, [0119].



De Castro, Fig.14 (partial annotated)

265. It would have been obvious to a POSITA that the communication application server 200 would be "*configured to associate*" the App Scripts 610

with the client's 40 telephone number. For example, as explained above at Section X.B.1.i, de Castro renders obvious that the server is configured to "*associate the second service with the first telephone number*." Like the App Script 610-1, a client 40 may call various numbers to access any of the App Scripts 610, such as App Script 610-k. De Castro, [0116]. Accordingly, it would have been obvious that the communication application server 200 associates two or more of the App Scripts (e.g., 610-1 ... 610-k) with the client 40's telephone number. For example, the same client 40 may, at different times, call different phone numbers that each invoke a different App Script. It would have been obvious that the app server 200 would similarly associate each of these invoked App Scripts with the client 40's telephone number when they are executed. In this way, the app server 200 would be able to allow any of the clients 40 to access any of the App Scripts hosted by the scripting engines 600. Thus, it would have been obvious to a POSITA that the communication application server 200 would "*associate a plurality of services with the first telephone number*."

b. [2.1] *wherein the plurality of services are provided by a plurality of providers different than the first provider, and wherein the plurality of services are utilized by the first user*.

266. As explained above at Section X.B.2.a, de Castro teaches that "a plurality of customers" have their respective App Scripts ("*plurality of services*") hosted on the scripting engines 600. De Castro, [0110]. De Castro further describes

a "**plurality of customers** with applications scripts in the customer/developer domain 601." De Castro, [0129]. Thus, a POSITA would have understood that the App Scripts ("*plurality of services*") are "*provided by*" the plurality of customers ("*plurality of providers*").

267. Because there are multiple customers, it would have been obvious to a POSITA that at least one of the customers would be different from first customer ("*the first provider*") associated with the App Script 610-1. For instance, de Castro explains that the messaging protocol used by the app server 200 allows customers to "code the application script in a preferred programming language." De Castro, [0041]. "In this way, application scripting is decoupled from the operation of the communication application server, which only needs to focus on providing basic communication services." De Castro, [0041]; *see also* [0113]. Accordingly, it would be obvious to a POSITA that the app server 200 would associate multiple App Scripts coded in different programming languages with the telephone number of the same client 40 so that the client 40 can access the services provided from each of the App Scripts.

268. Further, it would have been obvious to a POSITA in view of de Castro that the plurality of communication applications would be "*utilized by the first user*." De Castro teaches:

> "the communication application servers 200 ... allows [sic] application **clients** 40 ... to access the **applications** hosted in the communication application servers."

De Castro, [0115].

269. As explained in Section X.B.2.b above, it would have been obvious that the same client 40 would call different telephone numbers corresponding to different App Scripts. Thus, it would have been obvious that a plurality of the communication applications, including those embodied in App Scripts 610-1 to 610-k, would be "*utilized by the first user*."

3. Claim 6

a. [6.0] *The telephone system as recited in claim 1, wherein the second service is a conferencing application, voice attendant application, or text messaging application.*

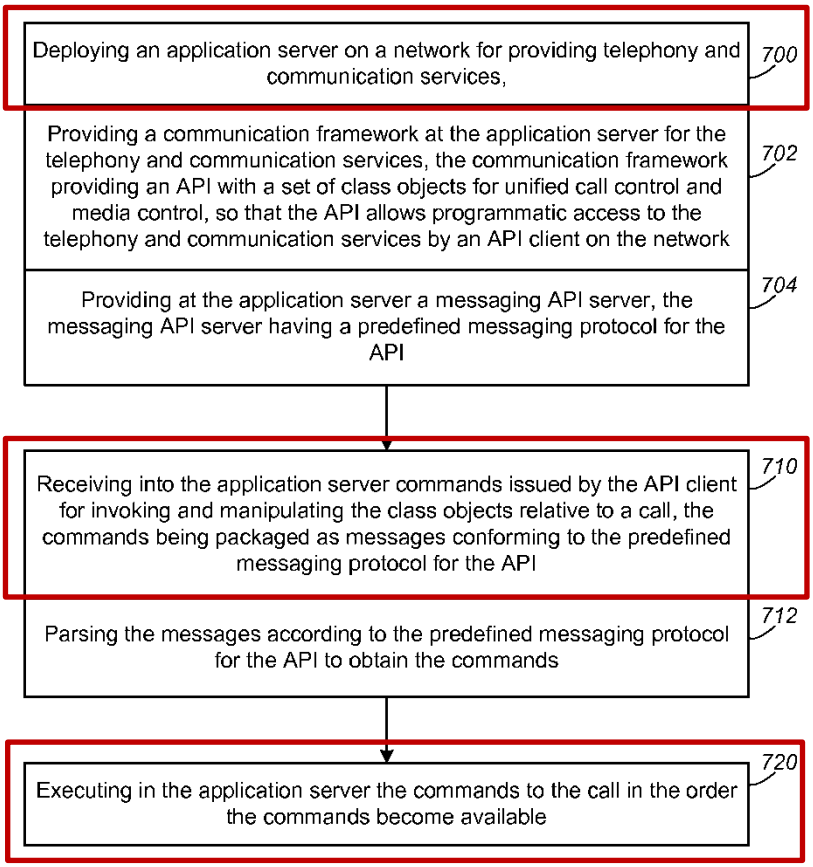
270. De Castro renders obvious that "*the second service is a conferencing application*" or a "*voice attendant application*." For example, de Castro teaches that the applications supported by the unified communication framework includes interactive voice response (IVR) and "Conferencing" applications. It would have been obvious that the App Script 610-k would include a "*conferencing application*." De Castro, [0075].

271. De Castro further describes the IVR concept and other "voice-enabled self-help applications," which further renders obvious that "*the second service*" is a "*voice attendant application*." *See* De Castro, [0104], [0008]-[0010].

4. Claim 7

a. [7.0] *A method comprising:*

272. De Castro relates to a "system and **method** for providing a telephony-enabled service via a message-based API interface." De Castro, [0002]. Fig.20 of de Castro illustrates "a **method** of operating the communication application server." De Castro, [0131]. The method shown in Fig.20, reproduced below, includes similar aspects of the network architecture of Fig.14 described above in Section IX.E.1.c-i, including providing a "communication framework" allowing "programmatic access to the telephony and communication services by an API client on the network," receiving "commands issued by the API client ... packaged as messages confirming to the predefined messaging protocol," and "executing in the application server the commands to the call." De Castro, Fig.20.



De Castro, Fig.20 (annotated)

273. Thus, de Castro teaches the preamble.

b. **[7.1]** ***providing a first service to a first user by a first provider, wherein the first service is telephone service, and wherein the first user has a first telephone number;***

274. De Castro renders obvious this limitation; *see* [1.2] and [1.3] above.

c. **[7.2]** ***receiving a request to provision a second service for the first user, wherein the second service is provided by a second provider, and wherein the second provider is different than the first provider;***

275. De Castro renders obvious this limitation; *see* [1.4] and [1.5] above.

d. [7.3] ***responsive to receiving the request: creating a mapping from the first telephone number to a second identifier (ID), wherein the second ID is utilized by the second provider to identify the first user;***

276. De Castro renders obvious this limitation; *see* [1.6] above.

e. [7.4] ***activating the second service with the second provider utilizing the second ID as a user ID for the first user; and***

277. De Castro renders obvious this limitation; *see* [1.7] above.

f. [7.5] ***associating the second service with the first telephone number, wherein the first telephone number is associated with routing and interworking data necessary to utilize the second service.***

278. De Castro renders obvious this limitation; *see* [1.8] above.

5. Claim 8

a. [8.0] ***The method as recited in claim 7, further comprising associating a plurality of services with the first telephone number,***

279. De Castro renders obvious this limitation; *see* [2.0] above.

b. [8.1] ***wherein the plurality of services are provided by a plurality of providers different than the first provider, and wherein the plurality of services are utilized by the first user.***

280. De Castro renders obvious this limitation; *see* [2.1] above.

6. Claim 12

a. [12.0] ***The method as recited in claim 7, wherein the second service is a conferencing application, voice attendant application, or text messaging application.***

281. De Castro renders obvious this limitation; *see* [6.0] above.

7. Claim 13

a. [13.0] *A non-transitory computer readable storage medium comprising program instructions, wherein when executed the program instructions are operable to:*

282. De Castro renders obvious a "*non-transitory, computer readable storage medium comprising program instructions*." De Castro teaches that "the communication application server 200 shown in Fig.14 supports a Java virtual machine that **maintains a number of software modules** 250...." De Castro, [0123]. Further, de Castro explains:

> Each communication application server 200 comprises a virtual machine for **executing codes** with access to the unified communication API of the **unified communication framework** 400 as described in connection with FIG. 4 above. The unified communication framework 400 is a further abstraction of a basic call control API and media control API.

De Castro, [0117].

283. With respect to software frameworks, de Castro describes that:

> A **software framework**, in computer programming, is an abstraction in which common code providing generic functionality can be selectively specialized by user code providing specific functionality. Frameworks are a special case of software libraries in that they are reusable abstractions of code wrapped in a well-defined API....

De Castro, [0067].

284. It would have been obvious to a POSITA for de Castro's software modules and software framework to be embodied in a "*non-transitory computer readable storage medium comprising program instructions*" because that is how computer software was commonly stored. *See, e.g.*, Bodart, [0075] (describing a computer system with non-volatile memory storing executable instructions); Banister 2:65 – 3:1 (describing computer-executable instructions stored on a computer-readable medium); Goldman, 8:58-63 (describing computer program stored on a computer-readable medium); Pegaz-Paquet, [0047] (operating system software stored in system memory).

285. Thus, it would have been obvious to a POSITA that the app server 200 of de Castro would include a "*non-transitory computer readable storage medium comprising program instructions*" to operate the software modules 250 of the app server 200 and to perform the functions of the server as described above in Section X.B.1.a-i.

b. [13.1] *provide a first service to a first user by a first provider, wherein the first service is telephone service, and wherein the first user has a first telephone number;*

286. De Castro renders obvious this limitation; *see* [1.2] and [1.3] above.

c. [13.2] *receive a request to provision a second service for the first user, wherein the second service is provided by a second provider, and wherein the second provider is different than the first provider;*

287. De Castro renders obvious this limitation; *see* [1.4] and [1.5] above.

d. [13.3] ***responsive to receiving the request: create a mapping from the first telephone number to a second identifier (ID), wherein the second ID is utilized by the second provider to identify the first user;***

288. De Castro renders obvious this limitation; *see* [1.6] above.

e. [13.4] ***activate the second service with the second provider utilizing the second ID as a user ID for the first user; and***

289. De Castro renders obvious this limitation; *see* [1.7] above.

f. [13.5] ***associate the second service with the first telephone number, wherein the first telephone number is associated with routing and interworking data necessary to utilize the second service.***

290. De Castro renders obvious this limitation; *see* [1.8] above.

8. Claim 14

a. [14.0] ***The non-transitory computer readable storage medium as recited in claim 13, wherein the program instructions are further operable to associate a plurality of services with the first telephone number,***

291. De Castro renders obvious this limitation; *see* [2.0] above.

b. [14.1] ***wherein the plurality of services are provided by a plurality of providers different than the first provider, and wherein the plurality of services are utilized by the first user.***

292. De Castro renders obvious this limitation; *see* [2.1] above.

XI. CONCLUSION

293. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true,

and that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code.

Dated: 2/1/23

Henry Houh

Henry Houh, Ph.D.