UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Facebook, Inc., WhatsApp, Inc.
Petitioners

v.

Uniloc USA, Inc., Uniloc Luxembourg S.A.,
Patent Owner

U.S. Patent No. 8,724,622

TITLE: SYSTEM AND METHOD FOR INSTANT VOIP MESSAGING

DECLARATION OF TAL LAVIAN, PH.D.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS
(continued)

Page

TABLE OF CONTENTS
(continued)

Page

TABLE OF CONTENTS
(continued)

Page

TABLE OF CONTENTS
(continued)

Page

TABLE OF CONTENTS
(continued)

Page

TABLE OF CONTENTS
(continued)

Page

I, Tal Lavian, Ph.D., declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

A. Qualifications and Experience

1. I have more than 25 years of experience in the networking, telecommunications, Internet, and software fields. I received a Ph.D. in Computer Science, specializing in networking and communications, from the University of California at Berkeley in 2006 and obtained a Master's of Science ("M.Sc.") degree in Electrical Engineering from Tel Aviv University, Israel, in 1996. In 1987, I obtained a Bachelor of Science ("B.Sc.") in Mathematics and Computer Science, also from Tel Aviv University.

2. I am employed by the University of California at Berkeley and was appointed as a lecturer and Industry Fellow in the Center of Entrepreneurship and Technology ("CET") as part of UC Berkeley College of Engineering. I have been with the University of California at Berkeley since 2000 where I served as Berkeley Industry Fellow, Lecturer, Visiting Scientist, Ph.D. Candidate, and Nortel's Scientist Liaison. I have taught several classes on wireless devices and smartphones. Some positions and projects were held concurrently, while others were held sequentially.

3. I have more than 25 years of experience as a scientist, educator and technologist, and much of my experience relates to telecommunication, data

communications, and computer networking technologies. For eleven years from 1996 to 2007, I worked for Bay Networks and Nortel Networks. Bay Networks was in the business of making and selling computer network hardware and software. Nortel Networks acquired Bay Networks in 1998, and I continued to work at Nortel after the acquisition. Throughout my tenure at Bay and Nortel, I held positions including Principal Scientist, Principal Architect, Principal Engineer, Senior Software Engineer, and led the development and research involving a number of networking technologies. I led the efforts of Java technologies at Bay Networks and Nortel Networks. In addition, during 1999-2001, I served as the President of the Silicon Valley Java User Group with over 800 active members from many companies in the Silicon Valley.

4. Prior to that, from 1994 to 1995, I worked as a software engineer and team leader for Aptel Communications, designing and developing wireless technologies, mobile wireless devices and network software products.

5. From 1990 to 1993, I worked as a software engineer and team leader at Scitex Ltd., where I developed system and network communications tools (mostly in C and C++).

6. I have extensive experience in communications technologies including wireless technologies, routing and switching architectures and protocols, including

Multi-Protocol Label Switching Networks, Layer 2 and Layer 3 Virtual Private Networks, and Pseudowire technologies. Much of my work for Nortel Networks (mentioned above) involved the research and development of these technologies. For example, I wrote software for Bay Networks and Nortel Networks switches and routers, developed network technologies for the Accelar 8600 family of switches and routers, the OPTera 3500 SONET switches, the OPTera 5000 DWDM family, and the Alteon L4-7 switching product family. I wrote software for Java-based device management, including a software interface for device management and network management in the Accelar routing switch family's network management system. I have also worked on enterprise Wi-Fi solutions, wireless mobility management, and wireless infrastructure.

7. I am named as a co-inventor on more than 100 issued patents and I co-authored more than 25 scientific publications, journal articles, and peer-reviewed papers. Furthermore, I am a member of a number of professional affiliations, including the Association of Computing Machinery ("ACM") and the Institute of Electrical and Electronics Engineers ("IEEE") (senior member). I am also certified under the IEEE WCET (Wireless Communications Engineering Technologies) Program, which was specifically designed by the IEEE Communications Society

(ComSoc) to address the worldwide wireless industry's growing and ever-evolving need for qualified communications professionals.

8. From 2007 to the present, I have served as a Principal Scientist at my company TelecommNet Consulting Inc., where I develop network communication technologies and provide research and consulting in advanced technologies, mainly in computer networking and Internet technologies. In addition, I have served as a Co-Founder and Chief Technology Officer (CTO) of VisuMenu, Inc. from 2010 to the present, where I design and develop architecture of visual IVR technologies for smartphones and wireless mobile devices in the area of network communications.

9. I have worked on wireless and cellular systems using a variety of modulation technologies including time-division multiple-access (TDMA), code-division multiple-access (CDMA), and orthogonal frequency-division multiplexing (OFDM). I have additionally worked on various projects involving the transmission and streaming of digital media content.

10. The above outline of my experience with communications systems is not comprehensive of all of my experience over my years of technical experience. Additional details of my background are set forth in my curriculum vitae, attached as **Exhibit A** to this Declaration, which provides a more complete description of my educational background and work experience.

11. I am being compensated for the time I have spent on this matter at the rate of $400 per hour. My compensation does not depend in any way upon the outcome of this proceeding. I hold no interest in the Petitioners (Facebook, Inc. and WhatsApp Inc.) or the Patent Owner (Uniloc Luxembourg, S.A.) or plaintiff Uniloc USA, Inc.

B. Materials Considered

12. The analysis that I provide in this Declaration is based on my education and experience in the telecommunications and information technology industries, as well as the documents I have considered, including U.S. Patent No. 8,724,622 ("'622" or "'622 patent") **[Ex. 1001/1101]**, which states on its face that it issued from an application filed on July 11, 2012, in turn claiming priority back to an earliest application filed on December 18, 2003. For purposes of this Declaration, I have assumed December 18, 2003 as the effective filing date for the '622 patent. I have cited to the following documents in my analysis below:

Exhibit No.	Title of Document
1001/1101	U.S. Patent No. 8,724,622 to Michael J. Rojas (filed July 11, 2012, issued May 13, 2014)
1003/1103	PCT Patent Application No. PCT/US00/21555 to Herbert Zydney et al. (filed August 7, 2000, published February 15, 2001 as WO 01/11824 A2) ("Zydney") (with line numbers added)
1004/1104	U.S. Patent No. 6,750,881 to Barry Appelman (filed February 24, 1997, issued June 15, 2004) ("Appelman")

Exhibit No.	Title of Document
1005/1105	Excerpts from MARGARET LEVINE YOUNG, INTERNET: THE COMPLETE REFERENCE (McGraw-Hill/Osborne, 2d ed. 2002) ("Young")
1006/1106	N. Borenstein et al., *Request for Comments (RFC) 1521: MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies*, September 1993 ("RFC 1521")
1007/1107	U.S. Patent No. 6,757,365 B1 to Travis A. Bogard (filed October 16, 2000, issued June 29, 2004) ("Bogard")
1008/1108	U.S. Patent No. 6,725,228 to David Morley Clark et al. (filed Oct. 31, 2000, issued April 20, 2004) ("Clark")
1009/1109	Excerpts FROM PAUL S. HETHMON, ILLUSTRATED GUIDE TO HTTP (Manning Publications Co., 1997) ("Hethmon")
1010/1110	Excerpts from CRAIG HUNT, TCP/IP NETWORK ADMINISTRATION (O'Reilly, 2d Ed. 1998) ("Hunt")
1011/1111	HTTP Working Group, Hypertext Transfer Protocol – HTTP/1.1, Nov. 22, 1995 (draft-ietf-http-v11-spec-00.txt)
1012/1112	Excerpts from *Microsoft Computer Dictionary* (3d ed. 1997)
1014/1114	Excerpts from DEBRA LITTLEJOHN SHINDER, COMPUTER NETWORKING ESSENTIALS (Cisco Press, 2002) ("Shinder")
1018/1118	Excerpts from *Microsoft Press Computer Dictionary* (1991) ("Microsoft (1991)")
1019/1119	U.S. Patent No. 6,173,323 to Pratyush Moghe ("Moghe")

II. PERSON OF ORDINARY SKILL IN THE ART

13. I understand that an assessment of claims of the '622 patent should be undertaken from the perspective of a person of ordinary skill in the art as of the earliest claimed priority date, which I understand is December 18, 2003. I have also been advised that to determine the appropriate level of a person having ordinary skill

in the art, the following factors may be considered: (1) the types of problems encountered by those working in the field and prior art solutions thereto; (2) the sophistication of the technology in question, and the rapidity with which innovations occur in the field; (3) the educational level of active workers in the field; and (4) the educational level of the inventor.

14. The '622 patent states that the perceived problem and the purported solution are generally related to the field of Internet telephony (IP telephony). The patent states: "More particularly, the present invention is directed to a system and method for enabling local and global instant VoIP messaging over an IP network, such as the Internet, with PSTN support." ('622, 1:18-22.) The '622 patent purports to describe a "voice messaging system (and method) for delivering instant messages over a packet switched network." (*Id.*, Abstract). The '622 patent purports to depict architectures of Internet and PSTN technologies, global and local IP networks, VoIP switches and gateways, and phone systems. The patent also purports to disclose local and global instant voice messaging servers communicating over an IP Network. In the Summary of the Invention, the applicant states: "The present invention is directed to a system and method for enabling local and global instant VoIP messaging over an IP network, such as the Internet." (*Id.*, 2:57-59.)

15. In my opinion, a person of ordinary skill in the art as of December 2003 would have possessed at least a bachelor's degree in computer science, computer engineering, or electrical engineering with at least two years of experience in development and programming relating to network communication systems (or equivalent degree or experience).

16. My opinions regarding the level of ordinary skill in the art are based on, among other things, my over 25 years of experience in computer science and network communications, my understanding of the basic qualifications that would be relevant to an engineer or scientist tasked with investigating methods and systems in the relevant area, and my familiarity with the backgrounds of colleagues, co-workers, and employees, both past and present.

17. Although my qualifications and experience exceed those of the hypothetical person having ordinary skill in the art defined above, my analysis and opinions regarding the '622 patent have been based on the perspective of a person of ordinary skill in the art as of December 2003.

III. BASIS FOR MY OPINION AND STATEMENT OF LEGAL PRINCIPLES

18. My opinions and views set forth in this declaration are based on my education, training, and experience in the relevant field, as well as the materials I

have reviewed for this matter, and the scientific knowledge regarding the subject matter that existed prior to December 2003.

A. Claim Construction

19. It is my understanding that, when construing claim terms, a claim subject to *inter partes* review receives the "broadest reasonable construction in light of the specification of the patent in which it appears."

B. Anticipation

20. It is my understanding that in order for a patent claim to be valid, the claimed invention must be novel. It is my understanding that if each and every element of a claim is disclosed in a single prior art reference, then the claimed invention is anticipated, and the invention is not patentable according to pre-AIA 35 U.S.C. § 102 effective before March 16, 2013. In order for the invention to be anticipated, each element of the claimed invention must be described or embodied, either expressly or inherently, in the single prior art reference. In order for a reference to inherently disclose a claim limitation, that claim limitation must necessarily be present in the reference.

C. Obviousness

21. Counsel has advised me that obviousness under pre-AIA 35 U.S.C. § 103 effective before March 16, 2013 is the basis for invalidity in the Petitions. Counsel has advised me that a patent claim may be found invalid as obvious if, at

the time when the invention was made, the subject matter of the claim, considered as a whole, would have been obvious to a person having ordinary skill in the field of the technology (the "art") to which the claimed subject matter belongs. I understand that the following factors should be considered in analyzing obviousness: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims; and (3) the level of ordinary skill in the pertinent art. I also understand that certain other factors known as "secondary considerations" such as commercial success, unexpected results, long felt but unsolved need, industry acclaim, simultaneous invention, copying by others, skepticism by experts in the field, and failure of others may be utilized as indicia of nonobviousness. I understand, however, that secondary considerations should be connected, or have a "nexus", with the invention claimed in the patent at issue. I understand that a person of ordinary skill in the art is assumed to have knowledge of all prior art. I understand that one skilled in the art can combine various prior art references based on the teachings of those prior art references, the general knowledge present in the art, or common sense. I understand that a motivation to combine references may be implicit in the prior art, and there is no requirement that there be an actual or explicit teaching to combine two references. Thus, one may take into account the inferences and creative steps that a person of ordinary skill in the art would employ to combine the known

elements in the prior art in the manner claimed by the patent at issue. I understand that one should avoid "hindsight bias" and ex post reasoning in performing an obviousness analysis. But this does not mean that a person of ordinary skill in the art for purposes of the obviousness inquiry does not have recourse to common sense. I understand that when determining whether a patent claim is obvious in light of the prior art, neither the particular motivation for the patent nor the stated purpose of the patentee is controlling. The primary inquiry has to do with the objective reach of the claims, and that if those claims extend to something that is obvious, then the entire patent claim is invalid. I understand one way that a patent can be found obvious is if there existed at the time of the invention a known problem for which there was an obvious solution encompassed by the patent's claims. I understand that a motivation to combine various prior art references to solve a particular problem may come from a variety of sources, including market demand or scientific literature. I understand that a need or problem known in the field at the time of the invention can also provide a reason to combine prior art references and render a patent claim invalid for obviousness. I understand that familiar items may have obvious uses beyond their primary purpose, and that a person of ordinary skill in the art will be able to fit the teachings of multiple prior art references together "like the pieces of a puzzle." I understand that a person of ordinary skill is also a person of

at least ordinary creativity. I understand when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If these finite number of predictable solutions lead to the anticipated success, I understand that the invention is likely the product of ordinary skill and common sense, and not of any sort of innovation. I understand that the fact that a combination was obvious to try might also show that it was obvious, and hence invalid, under the patent laws. I understand that if a patent claims a combination of familiar elements according to known methods, the combination is likely to be obvious when it does no more than yield predictable results. Thus, if a person of ordinary skill in the art can implement a predictable variation, an invention is likely obvious. I understand that combining embodiments disclosed near each other in a prior art reference would not ordinarily require a leap of inventiveness.

1. Motivation to Combine

22. I have been advised by counsel that obviousness may be shown by demonstrating that it would have been obvious to modify what is taught in a single piece of prior art to create the patented invention. Obviousness may also be shown by demonstrating that it would have been obvious to combine the teachings of more than one item of prior art. I have been advised by counsel that a claimed invention

may be obvious if some teaching, suggestion, or motivation exists that would have led a person of ordinary skill in the art to combine the invalidating references. Counsel has also advised me that this suggestion or motivation may come from the knowledge of a person having ordinary skill in the art, or from sources such as explicit statements in the prior art. Alternatively, any need or problem known in the field at the time and addressed by the patent may provide a reason for combining elements of the prior art. Counsel has advised me that when there is a design need or market pressure, and there are a finite number of predictable solutions, a person of ordinary skill may be motivated to apply common sense and his skill to combine the known options in order to solve the problem. The following are examples of approaches and rationales that may be considered in determining whether a piece of prior art could have been combined with other prior art or with other information within the knowledge of a person having ordinary skill in the art:

(1) Some teaching, motivation, or suggestion in the prior art that would have led a person of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;

(2) Known work in one field of endeavor may prompt variations of it for use in the same field or a different field based on design incentives or other market

forces if the variations would have been predictable to a person of ordinary skill in the art;

(3) Combining prior art elements according to known methods to yield predictable results;

(4) Applying a known technique to a known device, method, or product ready for improvement to yield predictable results;

(5) Applying a technique or approach that would have been "obvious to try" (choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success);

(6) Simple substitution of one known element for another to obtain predictable results; or

(7) Use of a known technique to improve similar products, devices, or methods in the same way.

IV. RELEVANT TECHNOLOGY BACKGROUND

23. The '622 patent, entitled "System and method for instant VoIP messaging," purports to disclose and claim a system and method for delivering instant voice messages over a packet-switched network. ('622, Abstract.) In this section, I provide a brief background discussion on technologies pertinent to the '622 patent prior to December 2003.

A. The Internet and TCP/IP Protocol Suite

24. **The Internet** is the global packet-switched network based on a protocol suite known as Transmission Control Protocol/Internet Protocol (TCP/IP). The Internet originated in the late 1960s as a Department of Defense project known as ARPANET and, by the 1980s, was in use by a large number of universities and organizations. As the Internet advanced in size and speed over the years, a vast amount of research and development was invested to develop technologies and standards for enabling voice communications over IP networks (VoIP). These significant investments in research and development yielded approved standards and large scale implementations based on these standards prior to the year 2003. Some of these key standards are discussed in the following sections.

25. The Internet is based on a globally unique address space based on the Internet Protocol (IP)[1] and is able to support communications using the TCP/IP suite or its subsequent extensions/follow-ons. In addition, the Internet provides, uses or makes accessible, either publicly or privately, high level services layered on the communications infrastructure. The TCP/IP protocol suite includes many different standard protocols including IP, TCP, UDP, VoIP, RTP, FTP, BGP, SMTP, DHCP,

[1] *See* IETF Network Working Group RFC 791 (Sept. 1981), RFC 1726 (Dec. 1994).

HTTP, and others. Internet standards are typically published in the form of documents known as "Requests for Comments" (RFCs), which are today maintained by the Internet Engineering Task Force (IETF).

B. Voice over IP (VoIP)

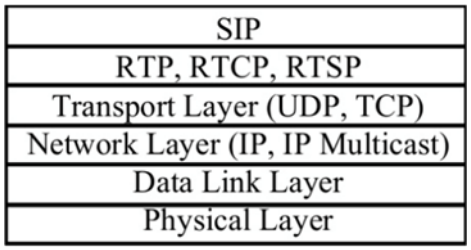
26. **Voice over IP (VoIP)** is a family of standard technologies which allows IP networks to be used for voice applications. VoIP generally involves the transmission of voice "data packets" from a device at one IP address over the Internet to a device at another IP address. The ability to transmit voice data packets from one IP address to another over the Internet is one of the background technologies relevant to the '622 patent and the claims at issue, which recite communication over a "packet-switched network."

27. The technologies that enabled VoIP and implementation of applications based on these technologies were available long before the '622 patent's filing date. For example, an early public domain VoIP application called NetFone (Speak Freely) was released in 1991 by Autodesk. A commercial internet VoIP application was released by VocalTec in February of 1995.[2]

[2] *See* William M. Bulkeley, *Hello World! Audible chats On the Internet*, WALL STREET JOURNAL, Feb. 10, 1995.

28. The real-time transport protocol (RTP) is an Internet protocol for the transfer of real-time data including voice and video. Version 1.0 of RTP was published in the early 1990s, and it was approved as a standard with the publication of RFC 1889 in January 1996.

29. RTP runs on top of an IP transport (depicted in the figure below).



Some relevant points of the protocol design are quoted from the standard: [3]

> This document defines RTP, consisting of two closely-linked parts:
> - The ***real-time transport protocol*** **(RTP)**, to carry data that has real-time properties.
>
> - the ***RTP control protocol*** **(RTCP)**, to monitor the quality of service and to convey information about the participants in an on-going session. The latter aspect of RTCP may be sufficient for "loosely controlled" sessions, i.e., where there is no explicit membership control and set-up, but it is not necessarily intended to support all of an application's control communication requirements. This functionality may be fully or partially subsumed by a separate session control protocol, which is beyond the scope of this document.

[3] All emphasis in quoted text in this Declaration has been added, unless otherwise noted.

Source: RFC 1889, § 1 (available at https://www.ietf.org/rfc/rfc1889.txt).

Definitions

RTP payload: The data transported by RTP in a packet, for example audio samples or compressed video data. The payload format and interpretation are beyond the scope of this document.

RTP packet: A data packet consisting of the fixed RTP header, a possibly empty list of contributing sources (see below), and the payload data. Some underlying protocols may require an encapsulation of the RTP packet to be defined. Typically one packet of the underlying protocol contains a single RTP packet, but several RTP packets may be contained if permitted by the encapsulation method (see Section 10).

RTCP packet: A control packet consisting of a fixed header part similar to that of RTP data packets, followed by structured elements that vary depending upon the RTCP packet type. The formats are defined in Section 6. Typically, multiple RTCP packets are sent together as a compound RTCP packet in a single packet of the underlying protocol; this is enabled by the length field in the fixed header of each RTCP packet.

Port: The "abstraction that transport protocols use to distinguish among multiple destinations within a given host computer. TCP/IP protocols identify ports using small positive integers." [3] The transport selectors (TSEL) used by the OSI transport layer are equivalent to ports. RTP depends upon the lower-layer protocol to provide some mechanism such as ports to multiplex the RTP and RTCP packets of a session.

Transport address: The combination of a network address and port that identifies transport-level endpoint, for example an IP address and a UDP port. Packets are transmitted from a source transport address to a destination transport address.

Id., RFC 1889, § 3 ("Definitions").

30. The ’622 patent acknowledges the use of VoIP in the prior art. For example, the patent explains that voice messaging was known in voice over internet protocol (“VoIP”) systems. (’622, 2:22.) According to the patent, in a VoIP system, a user would access a terminal device, such as a VoIP phone or a personal computer running VoIP client software, to connect with other such VoIP devices over the Internet. (*Id.*, 1:35-45.) The user would use a microphone connected to the terminal device to record messages and speakers or headphones to listen to messages. (*Id.*, 1:45-48.) Those messages would be transmitted over the Internet in packets. (*Id.*, 1:37-43.)

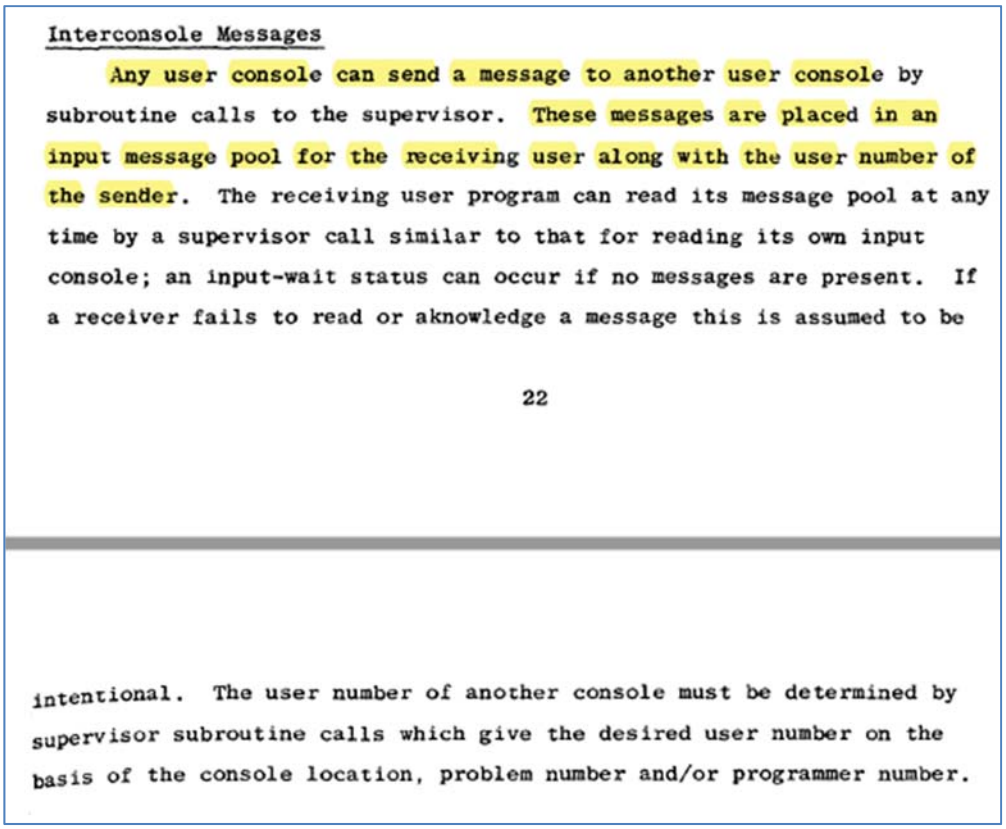
C. Instant messaging (IM)

31. Instant messaging (IM) was also well known before the ’622 patent’s priority date. An IM solution generally includes software with a user interface that allows users to exchange information with other users, including text, voice data, and/or files. The user software typically allows a user to select one or more recipients from lists of registered users which are displayed in a window. IM clients typically communicate through a server which either forwards messages directly to recipients or stores them if the recipients are not currently available to receive messages.

32. Different clients may vary in terms of what types of information they can send, how they indicate availability, how they can group users, and whether and how they secure the communications. However, the most popular clients available before the '622 patent filing date, provided the various functions proposed by the '622 patent. I start with a brief history of IM solutions.

33. An early example of an instant messaging solution dates back to the 1960s, as is shown by the following except of an instruction manual for "Interconsole messages" from an MIT programming manual for the "compatible time sharing system" which was published in 1963[4] (highlight added):

[4] *The Compatible Time Sharing System*, The MIT Press, 1963.



Interconsole Messages

Any user console can send a message to another user console by subroutine calls to the supervisor. These messages are placed in an input message pool for the receiving user along with the user number of the sender. The receiving user program can read its message pool at any time by a supervisor call similar to that for reading its own input console; an input-wait status can occur if no messages are present. If a receiver fails to read or aknowledge a message this is assumed to be

22

intentional. The user number of another console must be determined by supervisor subroutine calls which give the desired user number on the basis of the console location, problem number and/or programmer number.

34. MIT and Digital Equipment later developed the "Zephyr Notification Service" in the 1980s. The service used Unix to locate and send messages to users:

> Zephyr is a notice transport and delivery system under development at Project Athena. Zephyr is for use by network-based services and applications with a need for immediate, reliable and rapid communication with their clients. Zephyr meets the high throughput, high fan-out communications requirements of large-scale workstation environments. It is designed as a suite of 'layered services' based on a reliable, authenticated notice protocol. Multiple, redundant Zephyr servers provide basic routing, queueing, and dispatching services to clients that communicate via the Zephyr Client Library. More advanced

communication services are built upon this base.[5]

35. CompuServe's CB Simulator, released in 1980 to simulate citizens band radio through text-based messages and user handles, is considered by some to be the one of the first commercial services dedicated to online chat.

36. In 1982, Commodore International released the Commodore 64 computer. An Internet service known as Quantum Link (also known as Q-Link) was designed for use with Commodore computers. Q-Link, which later became known in the 1990s as America Online (AOL), allowed users to send text-based messages to another user via modem. The receiving user had the option of responding to or ignoring the messages.

37. One of the most popular IM applications was ICQ (I Seek You). The following is an excerpt from a press release: "Launched in November 1996, ICQ's instant communication and chat technology informs users when family, friends and business colleagues are online and enables them to exchange messages in real-time to help its users build their own communities. ICQ also gives its users the ability to

[5] DellaFera, C. Anthony, et al., *The Zephyr Notification Service*, USENIX, Winter 1988, at Abstract.

play games and exchange files and URLs."[6] AOL purchased ICQ in June 1998. AOL also had its own IM product called AOL Instant Messenger (AIM).

38. In the years before 2003, online instant messaging grew to include hundreds of millions of registered users. As explained by **Young [Ex. 1005/1105]**, published in 2002, ICQ and AIM each had more than 100 million registered users at the time. (Young at 331, 336.)

1. IETF in RFC 2778 – "A Model for Presence and Instant Messaging"

39. With the proliferation of IM systems, the IETF identified a need to generalize a model of IM services and protocols which would enable different IM systems to communicate. Instant messaging was addressed by the IETF in RFCs 2778 and 2779. An IM model was proposed by the IETF in RFC 2778 published in February 2000. RFC 2778 "defines the various entities involved, defines terminology, and outlines the services provided by the system." RFC 2778, Abstract. The motivation for defining the model is described: "A presence and instant messaging system allows users to subscribe to each other and be notified of

[6] From http://www.timewarner.com/newsroom/press-releases/1998/06/08/america-online-inc-acquires-mirabilis-ltd-and-its-icq-instant.

changes in state, and for users to send each other short instant messages. To facilitate development of a suite of protocols to provide this service, we believe that it is valuable to first develop a model for the system. The model consists of the various entities involved, descriptions of the basic functions they provide, and most importantly, definition of a vocabulary which can be used to facilitate discussion." *Id*., Introduction.

2. IETF RFC 2779 "Instant Messaging / Presence Protocol Requirements"

40. Using the RFC 2778 model, a basic architecture for instant messaging was proposed by the IETF RFC 2779 published in February 2000: "Applications of presence and instant messaging currently use independent, non-standard and non-interoperable protocols developed by various vendors. The goal of the Instant Messaging and Presence Protocol (IMPP) Working Group is to define a standard protocol so that independently developed applications of instant messaging and/or presence can interoperate across the Internet. This document defines a minimal set of requirements that IMPP must meet." RFC 2779, Abstract.

3. Prior Art Instant Messaging ("IM") Systems

41. The '622 patent explains that known instant messaging ("IM") systems generally included client devices, IM software installed on those client devices, and IM servers. ('622, 2:34-38.) IM systems communicated over a packet-switched

network, such as the Internet. ('622, 1:37-38, 2:34-38.) The IM server maintained a list of users that were currently "online" and able to receive messages and presented this list to the users via the instant messaging software. ('622, 2:38-41.) A user could select one or more recipients and send them a message. ('622, 2:42-44.) The IM server would transmit the message to the recipients and the message would be displayed to the recipients by the IM software. ('622, 2:44-46.)

42. Other elements of voice messaging systems were also well-known by December 2003. For example, **Bogard**, U.S. Patent No. 6,757,365, explains that visual interfaces, called **buddy lists**, could be used to identify available users and start voice messaging sessions. (Bogard, Ex. 1007/1107, 1:25-48.) Bogard describes the "**buddy list**" from AOL Instant Messenger's ("AIM") client software as "allow[ing] a user of AIM to see **which buddies** (other user**s** of interest to our particular user, e.g. friend**s**, co-worker**s**, family member**s**) **are signed on**." (*Id.*, 1:27-31.) Figure 1 of Bogard, reproduced below, shows an AIM client identifying both "**available**" buddies, i.e., Buddy1, Buddy2, Buddy3, and Buddy4, and **"offline" buddies**. (*Id.*, Fig. 1.)

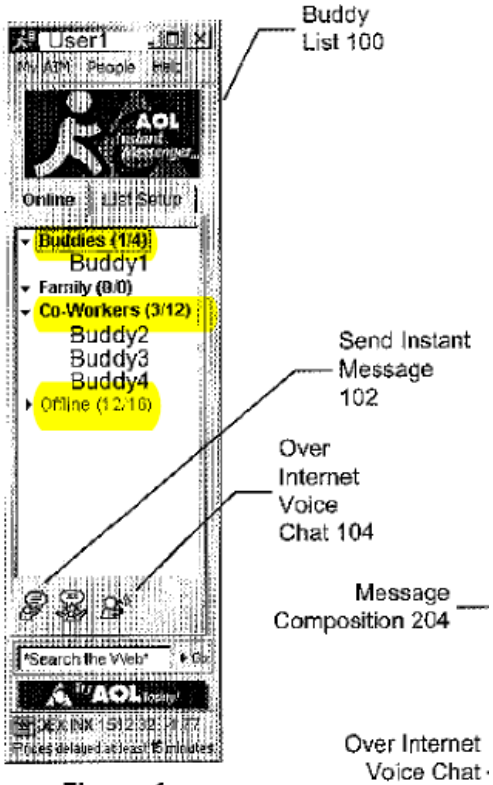


Figure 1
(Prior Art)

(*Id.*, Fig. 1.) Bogard also describes a method by which AIM users could start a voice communication session over the Internet through the buddy list by clicking a "Voice Chat" button. (Bogard, 1:40-48; *see also id.*, Fig. 1 (identifying the "Over Internet Voice Chat 104" button).)

43. **Young** likewise describes the AIM and ICQ instant message systems that were used by hundreds of millions of users prior to 2003, including their buddy list interfaces showing users' online and offline statuses, as shown in the figures reproduced below.

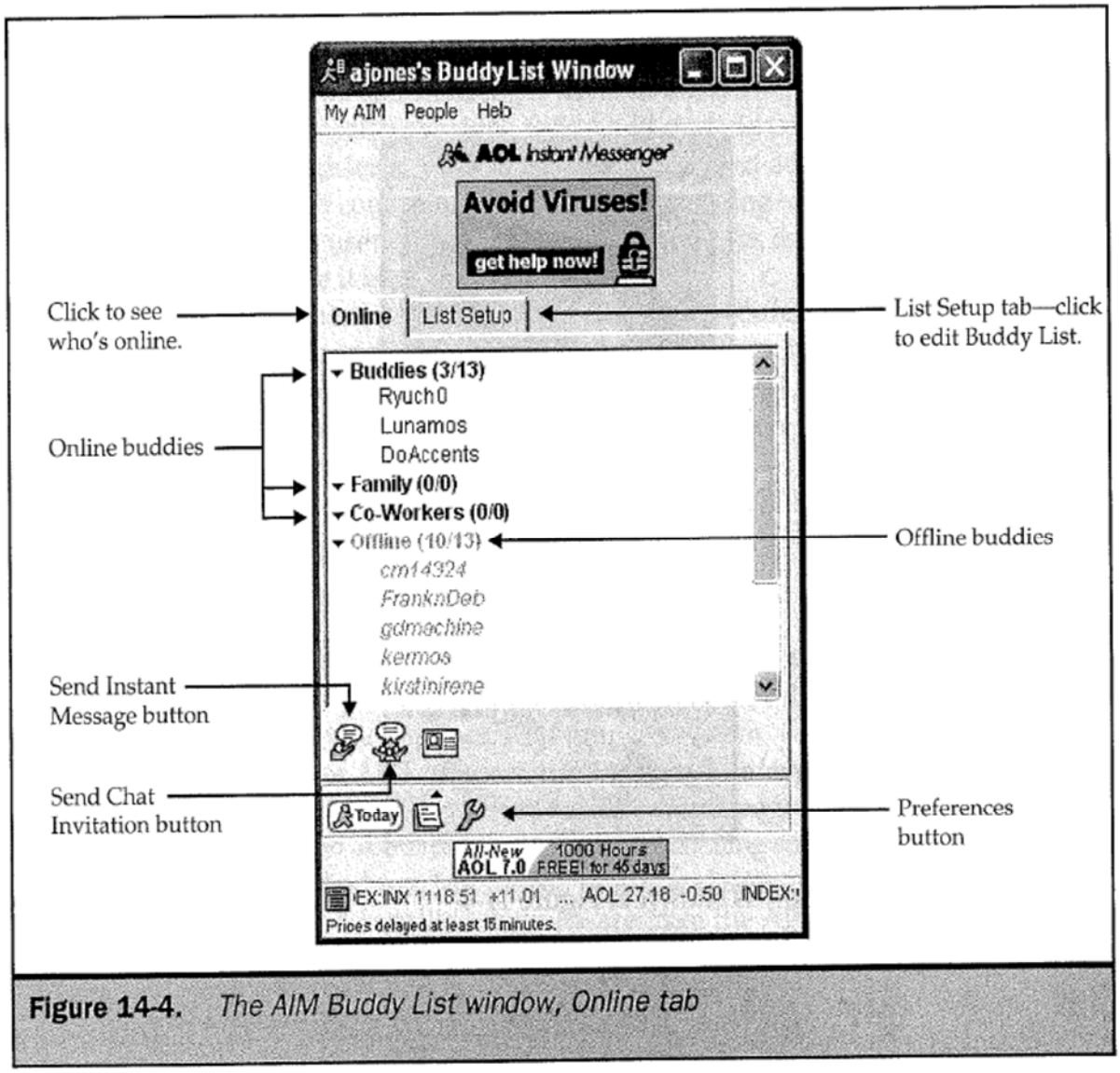


Figure 14-4. *The AIM Buddy List window, Online tab*

(Young at 338, Fig. 14-4.)

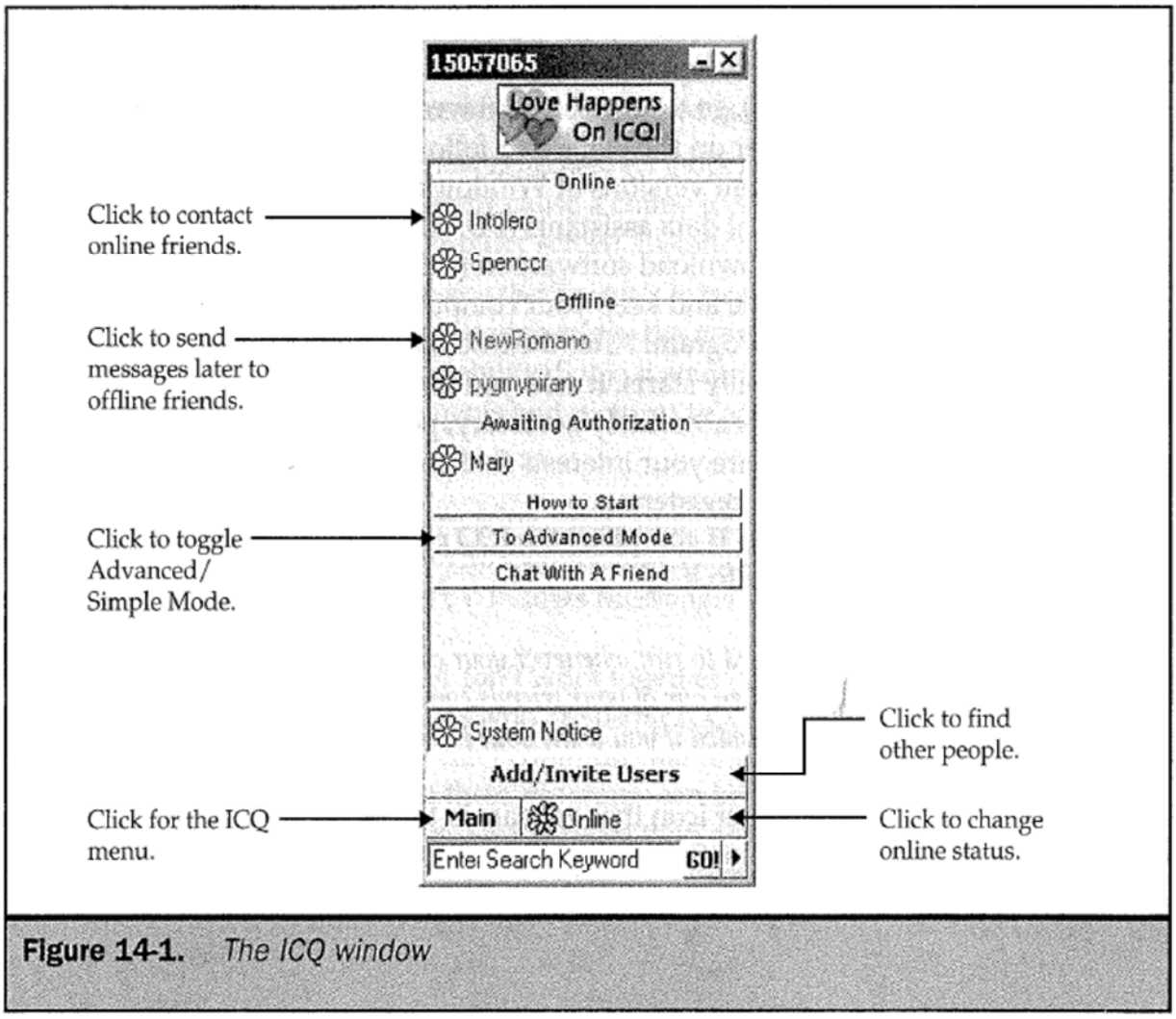


Figure 14-1. *The ICQ window*

(*Id.* at 332, Fig. 14-1.)

V. THE '622 PATENT

A. The Specification

44. The '622 patent purports to describe a system and method for delivering instant voice messages over a packet-switched network. ('622, Abstract.) In this system, a client such as a VoIP telephone or PC computer "enabled for IP telephony"

is connected to a server and instant voice message ("IVM") recipients through a network(s). (*Id.*, 1:43-50, 2:60-3:4, 6:65-7:2.)

45. In one embodiment, when a user chooses to send an IVM, the IVM client displays a "list of one or more IVM recipients." (*Id.*, 7:65-8:4.) This recipient list is provided and stored by an IVM server. (*Id.*) Once recipients are selected, the user records a message, such as by using a microphone to record a digitized audio file. (*Id.*, 8:7-11.)

46. The patent states that one or more files may be attached to the instant voice message, such as by using a conventional "drag-and-drop" technique. (*Id.*, 12:26-38, 13:33-38.)

47. Once the voice message is generated, the client transmits the voice message to the server for delivery to one or more recipients. (*Id.*, 8:21-26.) After receiving the instant voice message, the server transmits the voice message to the one or more recipients. (*Id.*, 8:26-29.) Recipients that are "available" (currently connected to the IVM server) will receive the instant voice message. (*Id.*, 8:32-34.) If a recipient is unavailable (offline), the server temporarily saves the voice message and transmits it once the recipient becomes available. (*Id.*, 8:34-39.) The recipient is notified of the new voice message and can play the audio file aloud. (*Id.*, 8:29-

32.) If the voice message had attachments, the recipient can also access the attached files. (*Id.*, 13:3-10.)

B. The Claims of the '622 Patent

48. This Declaration addresses claims 3-8, 10-35, 38, and 39. Claims 3, 24, 27, and 38 are independent claims; claims 4-8 and 10-23 directly or indirectly depend from claim 3; claims 25-26 depend from claim 24; claims 28-35 directly or indirectly depend from claim 27; and claim 39 depends from claim 38. I address the claims in my detailed analysis in **Part VI** below.

C. Claim Construction

1. "an instant voice messaging application"

49. It is my opinion that the broadest reasonable interpretation of "an instant voice messaging application" in the context of the claims of the '622 patent is "**hardware and/or software used for instant voice messaging**."

50. Claim 13 recites "[t]he system according to claim 3, wherein each of the instant voice message client systems comprises an instant voice messaging application generating an instant voice message and transmitting the instant voice message over the packet-switched network to the messaging system."

51. Claim 27 similarly recites: "an instant voice messaging application installed on the client device, wherein the instant voice messaging application includes a client platform system for generating an instant voice message and a

messaging system for transmitting the instant voice message over the packet-switched network via the network interface."

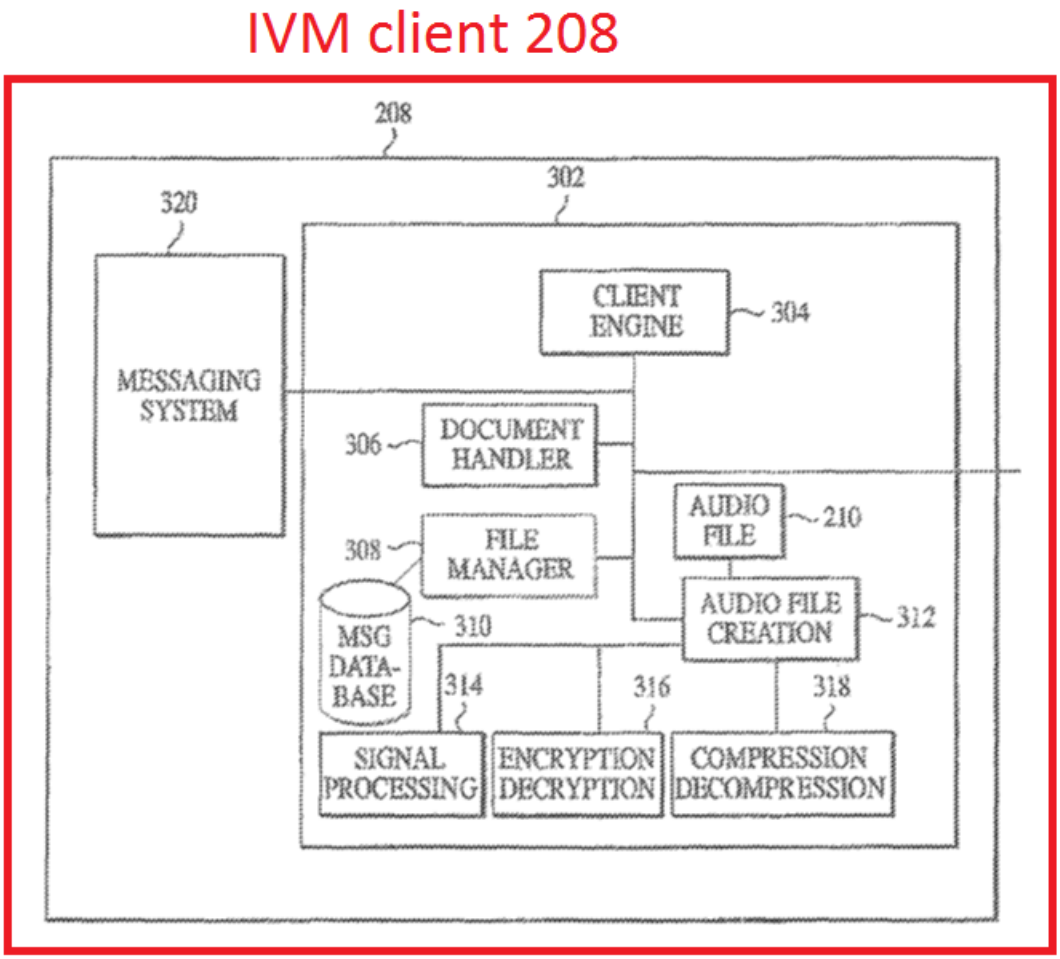
52. Claim 38 similarly recites: "an instant voice messaging application installed on the client device, wherein the instant voice messaging application includes a client platform system for generating an instant voice message and a messaging system for transmitting the instant voice message over the packet-switched network via the network interface."

53. The written description of the '622 patent does not use the word "application" to refer to any aspect of the alleged invention.[7] A person of ordinary skill in the art typically uses the term "application" to refer to computer software for performing a particular function. (Ex. **1012/1112**, Microsoft Computer Dictionary (1997), at p.27 (defining "application" as "[a] program designed to assist in the performance of a specific task, such as word processing, accounting, or inventory management.").) In this case, however, the written description of the '622 patent

[7] The only instances of the word "application" in the entire written description appear in the "Cross-Reference to Related Application" section that lists various patent applications in the same family as the '622 patent. ('622, 1:4-14.)

suggests that the term should not be limited to just software under its broadest reasonable construction.

54. The written description also does not identify any particular software program capable of performing all of the functions associated with the "instant voice messaging application" recited in the claims. To the contrary, it describes these functions as being performed by an instant voice messaging client, IVM client **208**, which is a "general-purpose programmable computer." ('622, 12:11-14.) The IVM client **208** contains various boxes labeled with functions including client platform **302**, which contains boxes labeled client engine **304**, document handler **306**, file manager **308**, audio file creation **312**, signal processing **314**, encryption/decryption **316**, and compression/decompression **316**. (*Id.*, 12:17-21.) The IVM client **208** also contains a box labeled messaging system **320**. (*Id.*, 12:6-11.) Figure 3, an excerpt of which is reproduced below with red annotations added, shows these various boxes inside IVM client **208**.

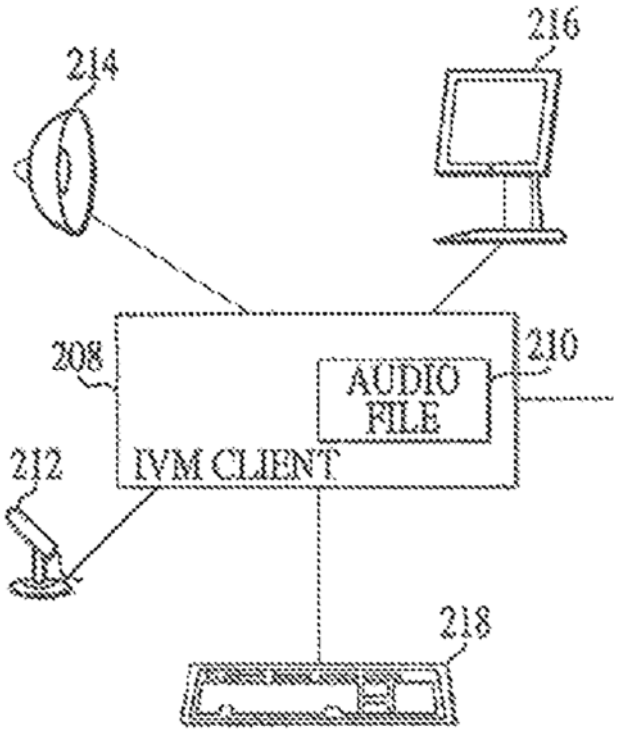


(*Id.*, Fig. 3 (annotated).)

55. I note that the claims recite that the "instant voice message application" includes a "client platform system" and a "messaging system." But as shown in Figure 3 above, the written description shows both the client platform **302** and messaging system **320** inside IVM client **208**, and does not identify a specific software program that contains those two components. Therefore, a person of ordinary skill in the art would have understood that, under the broadest reasonable

construction, the claimed "instant voice messaging application" would not be limited to just a software program, but could encompass multiple different software programs and/or hardware components.

56. My conclusion is further supported by claim 21, which recites that the claimed instant voice messaging application "displays a list of one or more potential recipients for the instant voice message." The written description does not state that any of the boxes inside IVM client **208**, or any software, provides the claimed display capability. The written description instead states that a "display device **216** is connected to the IVM client **208** to display instant voice messages recorded and/or received by a user of the IVM client **208**." (*Id.*, 7:15-17; *see also id.*, 7:19-22 ("It is noted that the microphone **212**, audio device **214**, display device **216** and input device **218** may form integral parts of the IVM client **208**.").) The written description also states that the IVM client **208**, which is a general-purpose programmable computer, can display a list of recipients on display **216**: "The IVM client **208** displays a list of one or more IVM recipients on its display **216**…." (*Id.*, 7:65-66.) Figure 2, an excerpt of which is reproduced below, shows IVM client **208** connected to display device **216**.



(*Id.*, Fig. 2.)

57. Therefore, a person of ordinary skill in the art would have understood that under the broadest reasonable construction, the claimed "instant voice messaging application" is not limited to software and could have included hardware such as a general purpose computer and display device **216**. Accordingly, a person of ordinary skill in the art would have understood that the broadest reasonable interpretation of "instant voice messaging application" in the context of the claims of the '622 patent is "**hardware and/or software used for instant voice messaging**."

2. "client platform system"

58. As noted in my discussion of the claimed "instant voice messaging application," one of the components the application must include is a "client platform system." Claim 27, for example, states that the instant voice messaging application includes "a client platform system for generating an instant voice message." ('622, 26:24-25.) In my opinion, the broadest reasonable construction of "**client platform system**" is "**hardware and/or software on a client for generating an instant voice message**."

59. The written description does not use the term "client platform system" but does describe a "client platform **302**" whose purpose is "generating an instant voice message." (*Id.*, 12:7-8). The written description further states that the client platform **302** "comprises a client engine **304**, which controls other components" such as the document handler, file manager, and encryption/decryption. (*Id.*, 12:17-21.)

60. The written description does not provide any description of what the composition or content of "client engine **304**" actually is, such as whether it is implemented in hardware and/or software. The written description instead describes client engine **304** functionally as being involved in functions including (1) communicating with the server and (2) performing operations required to generate an instant voice message. (*Id.*, 12:24-25, 13:15-28.) Figure 3 of the '622 patent also

shows client engine **304** as a similarly nondescript box within client platform **302**. (*Id.*, Fig. 3.)

61. Nevertheless, as I explained above, the claimed "instant voice messaging application" is composed of hardware and/or software under its broadest reasonable construction. Because the claimed "client platform system" is part of the "instant messaging application" in the challenged claims, the "client platform system" under its broadest reasonable construction should similarly be defined as hardware and/or software. In my opinion, therefore, "client platform system" should be defined under its broadest reasonable construction as "**hardware and/or software on a client for generating an instant voice message**."

62. I am informed that in pending litigation, the Patent Owner has proposed to construe "**a client platform system**" to mean "the system of the client engine which controls other components used to generate an instant voice message." This definition has various flaws and, in my opinion, is not consistent with the broadest reasonable interpretation. To begin with, the definition gets the relationship between the "client engine" and the "client platform" backwards by reciting that the "client platform system" is a part of the "client engine." But the written description makes clear that client engine **304** is actually a part of the client platform **302**, not the other way around. (*Id.*, Fig. 3, 12:17-18 ("The client platform **302** comprises a client

engine **304**, which controls other components…").) Second, the claims themselves do not recite a "client engine," and the recitation of a "client engine" does not appear to add anything meaningful to the Patent Owner's proposed construction. Third, the written description of the preferred embodiment and the claim language do not expressly define that a "client platform system" as a system that "controls other components used to generate an instant voice message" and, under the broadest reasonable interpretation, do not require that the term would be limited to this definition for all embodiments of the claimed invention. Nevertheless, my analysis of the claims below will show that the prior art discloses the claimed "client platform system" even under the Patent Owner's proposed construction.

3. "communication platform system"

63. I am informed and understand that the Patent Owner in the pending litigation has proposed to construe "communication platform system" in the context of claims 3 and 24 of the '622 patent to mean "the system of the server which relays communications and/or tracks client connection information." I have been asked to apply this construction for my analysis.

64. I note that the '622 patent's written description does not contain the term "communication platform system." The written description contains the following statements:

> The IVM server **202** comprises a server communication platform **402**, a messaging system **436** and a database **414**, thereby enabling instant voice messaging according to the present invention. The server communication platform **402** comprises a server engine **404**, client manager **406**, station manager **408**, gateway manager **410**, database manager **412** that accesses database **414**, supplemental servers **416** (including particular server subsystems **418**-**424**), as well as a control layer **426** (including non-proprietary server subsystems **428**, **430** and proprietary server subsystems **432**, **434**).

('622, 13:46-55.)

65. The "server communication platform" is shown labeled as item **402** in Figure 4, reproduced below.

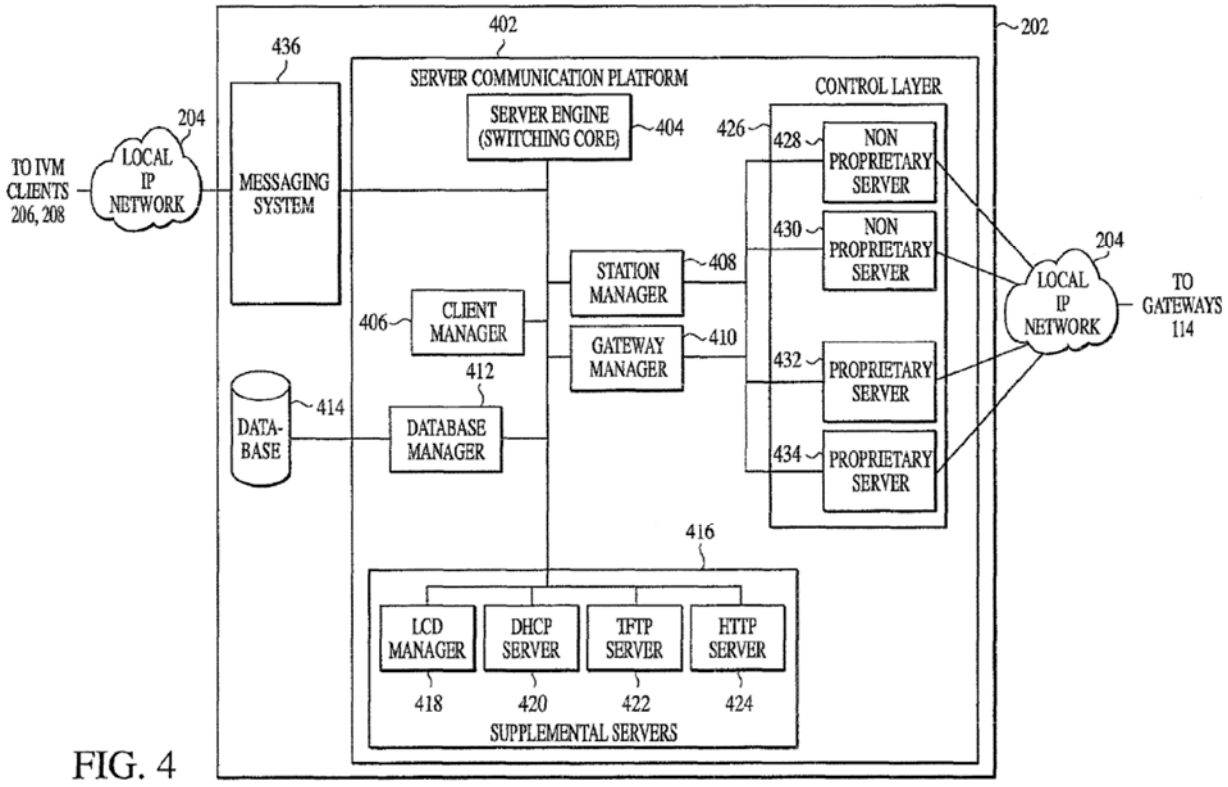


FIG. 4

(*Id*., Fig. 4.)

4. "connection object messages"

66. I am informed and understand that the parties in litigation have agreed that "**connection object messages**" in the context of claim 24 of the '622 patent should be construed to mean "**messages containing data representing the state of the connection and code (one or more methods) for establishing and maintaining the logical connections between an instant voice messaging server and instant voice messaging clients**." I have been asked to apply this construction as the broadest reasonable construction for purposes of my analysis.

67. At the outset, in my opinion, the agreed-upon construction is consistent with the meaning of "connection object messages" to a person of ordinary skill in the art who has reviewed the '622 patent. The wording of the agreed-upon construction comes from the specification of the '622 patent, which provides the following description of connection objects:

> Connection objects maintain the logical connections between the IVM server **202** and IVM clients **206**, **208** connected to the IVM server **202**. More specifically, a connection object comprises data representing the state of the connection and code (one or more methods) for establishing and maintaining the logical connections between the IVM server **202** and the IVM clients **206**, **208** within the IVM system **200** of FIG. **2**. The connection object can contain both data and/or commands, including information that describes the socket, the size of the data to be transferred, and the priority of the transfer (e.g., high, normal, low, unknown). On start up the local IVM server **202** generates and maintains a list for each IVM client **206**, **208**. The local IVM server **202** then waits to receive connection objects from the IVM clients **206**, **208** that are stored in the respective lists, decodes the received connection objects to obtain specific requests, and then services the specific requests from the IVM clients **206**, **208**.

('622, 14:47-63.)

68. In my opinion, the claims of the '622 patent do not require that the claimed "connection object messages" contain ***executable*** code, such as an ".EXE"

software program. I note that the above-quoted passage from the '622 specification states that a connection object can comprise "**code (one or more methods)**," which does not on its face mandate the inclusion of machine-executable computer program code. The passage goes on to explain that the IVM server **202** receives and "decodes" the connection objects, but does not say anything about execution of any "code" contained in them. In my opinion, therefore, neither the patent nor the agreed-upon construction, applying the broadest reasonable interpretation, requires that the connection object message include executable code.

VI. APPLICATION OF THE PRIOR ART TO THE CLAIMS

69. I have reviewed and analyzed the prior art references and materials listed in **Part I.B** above. In my opinion, the claims of the '622 patent are obvious based on the following prior art. The independent claims are **bolded**.

References	Claims
Zydney and Shinder	**3**, 6-8, 10, 11, 13, 18-21, 23, **27**, 32-35, **38**
Zydney, Shinder, and Appelman	22, 39
Zydney, Shinder, and Clark	14-17, 28-31
Zydney, Shinder, and Hethmon	4, 5, **24**-26
Zydney, Shinder, and Microsoft (1991) and Moghe	12

70. In my opinion, as fully explained below, claims **3**, 6-8, 10-13, 18-21, 23, **27**, 32-35, and **38** are obvious based on Zydney and Shinder. Claims 22 and 39 are obvious based on Zydney in view of Shinder and Appelman. Claims 14-17, and 28-31 are obvious based on Zydney in view of Shinder and Clark. Claims 4, 5, and 24-26 are obvious based on Zydney in view of Shinder and Hethmon. Claim 12 is obvious based on Zydney in view of Shinder, in further view of Microsoft (1991) and Moghe. I understand that each of these references qualifies as prior art to the '622 patent because it was filed or published before December 18, 2003, the earliest claimed filing date for the '622 patent.

A. Brief Description and Summary of the Prior Art

1. Brief Summary of Zydney [Ex. 1003/1103]

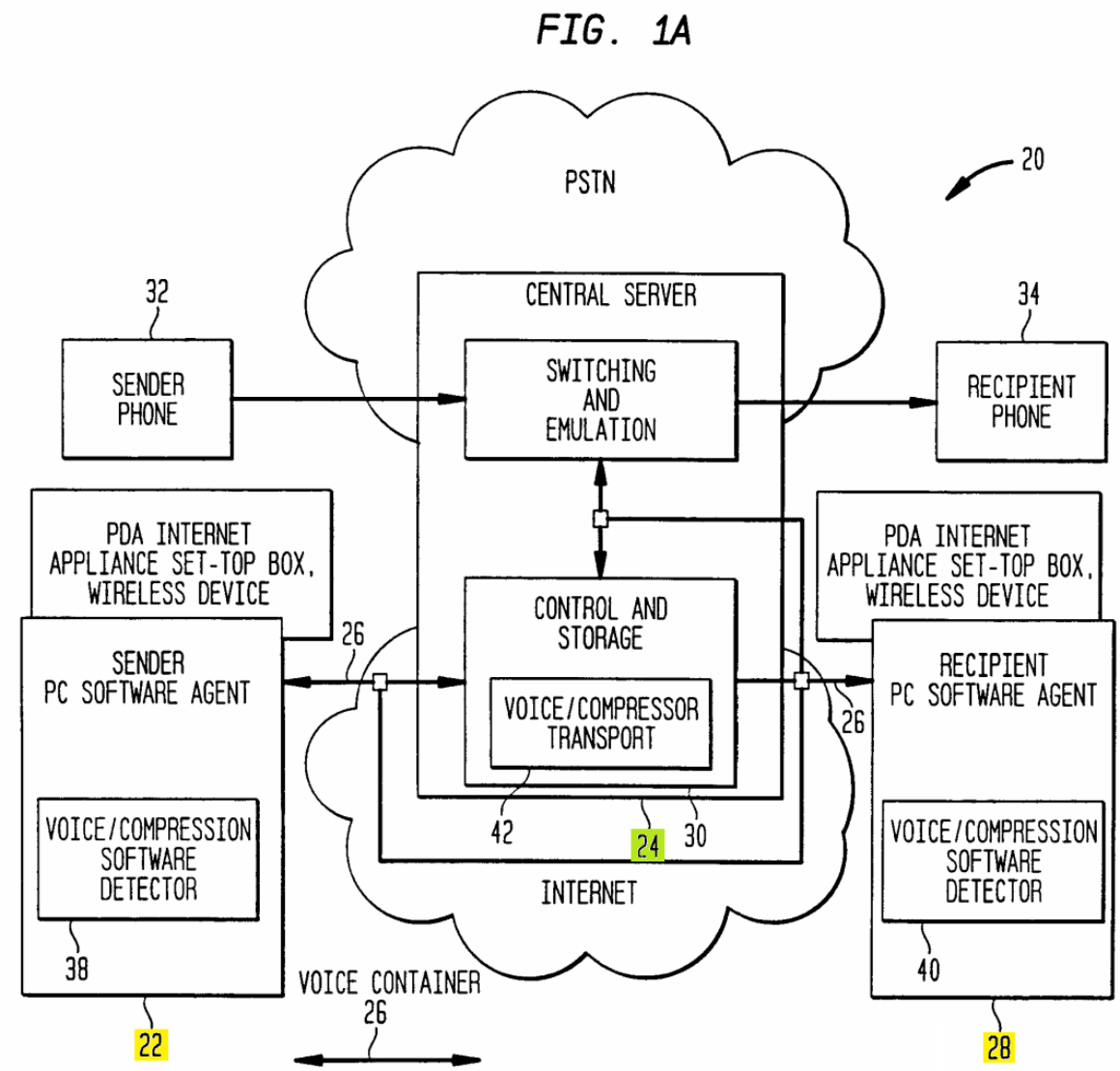
71. **Zydney** describes a system for voice communication that enables a client to interact with a server in order to "send, receive and store messages using voice containers." (Zydney, 2:2-3.)[8] The system transmits the voice containers

[8] I am informed that the Zydney reference does not contain line numbers. Accordingly, Exhibit 1003/1103 contains a copy of Zydney in which line numbers were added to the left of each page (beginning on page 1), to facilitate precise citation to the passages of the reference that I discuss in this Declaration.

"instantaneously or stored for later delivery," depending on whether or not the recipient is currently online. (*Id.*, 1:19-22, 15:8-21.) This Declaration relies on Zydney as the primary reference that discloses the claims' limitations.

72. Zydney describes that its method and system "is particularly well suited for use in connecting Internet users." (*Id.*, 5:3-4.) The system is generally shown in Figure 1A, reproduced below.

FIG. 1A



(*Id.*, Fig. 1A.)[9]

73. Three key components of the system include the "SENDER PC SOFTWARE AGENT" shown on the left (22), the "RECIPIENT PC SOFTWARE AGENT" shown on the right (28), and the "CENTRAL SERVER" shown in the middle (24) of Figure 1A. (*Id.*, 10:19-11:1.) Zydney explains that the sender and recipient software agents may work on any suitable client device such as "a personal computer, wireless handheld computer such a personal data assistant (PDA), digital telephone, or beeper." (*Id.*, 11:14-20.) Central server (24) facilitates instant voice messaging between the sender and the recipient. (*Id.*, 10:20-11:1.) The sender, recipient, and central server communicate with each other using a communications network, as shown with the bottom cloud labeled "INTERNET" in Figure 1A. (*Id.*, Fig. 1A; *see also id.*, 5:4-5, 5:15-18, 10:11-14, 14:2-5.)[10]

[9] Unless otherwise indicated, all underlining or boldface type in quotations and all highlighting in figures in this Declaration has been added for emphasis.

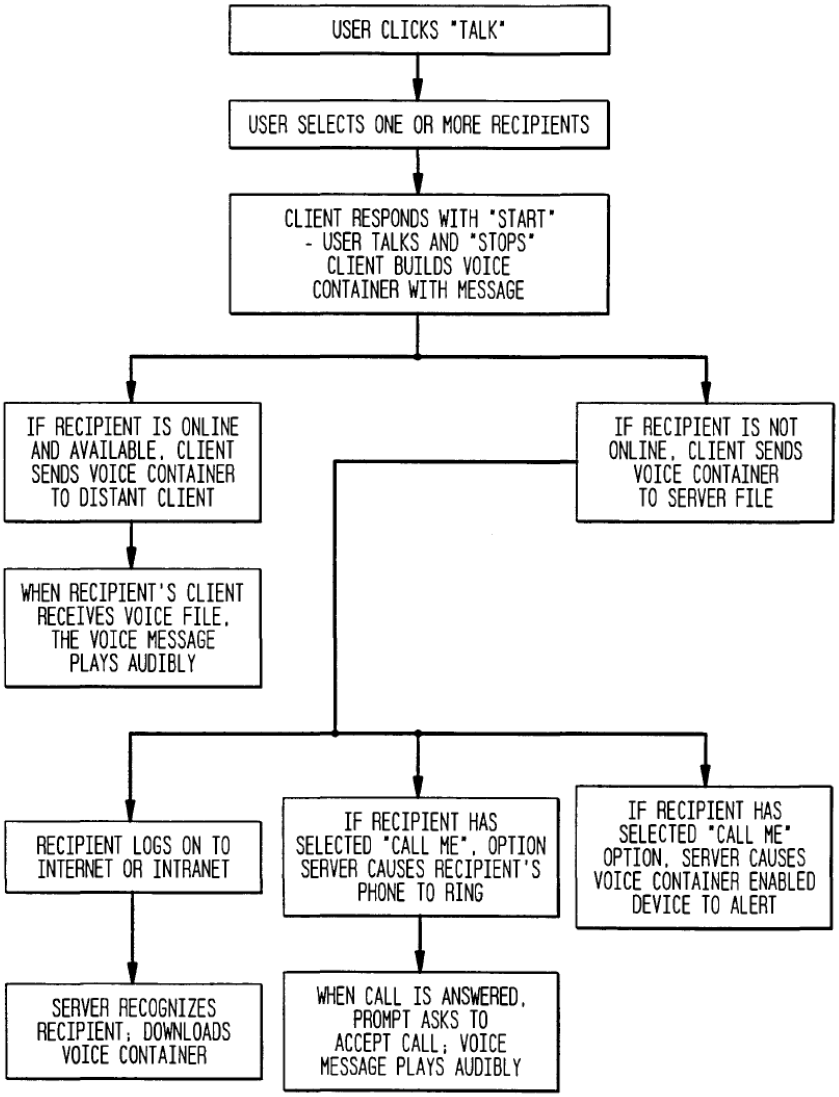
[10] Figure 1A also depicts an alternative embodiment in which a sender and recipient can communicate using phones (32, 34) connected over the Public Switched Telephone Network (PSTN). My Declaration will focus the Internet-connected embodiment described in the text.

74. Sending a voice instant message from a sender to a recipient in Zydney is straightforward. A message sender (originator) "selects one or more intended recipients from a list of names that have been previously entered into the software agent." (*Id.*, 14:17-19.) The sender also "digitally records messages for one or more recipients using a microphone-equipped device and the software agent. The software agent compresses the voice and stores the file temporarily on the PC if the voice will be delivered as an entire message." (*Id.*, 16:1-4; *see also id.*, 20:11-14, 21:11-16 (describing "the recording of one or more voice packet messages on a personal computer" as "voice files [that] can be played and recorded using voice container enabled devices.").) The voice message is placed into a "**voice container**," which can be transmitted to the destination. (*Id.*, 10:20-11:3.)

75. Zydney describes at least two modes in which voice messages can be transmitted. First, an instant voice message to an online recipient can be transmitted using "**pack and send**" mode. "A pack and send mode of operation is one in which the message is first acquired, compressed and then stored in a voice container 26 which is then sent to its destination(s)." (*Id.*, 11:1-3.) Second, Zydney discloses an "**intercom**" mode in which participants communicate in a real-time call or conversation. (*Id.*, 15:8-10.)

76. An exemplary process of transmitting an instant message is illustrated in Figure 4 of Zydney, reproduced below.

FIG. 4



(*Id.*, Fig. 4.)

77. As reflected in Figure 4, Zydney describes that, in order to send a voice instant message, a user selects one or more recipients from a list. (*Id.*, 14:17-19.)

Once the delivery mode is determined, the sender records a message using a microphone. (*Id.*, 16:1-3.) In the "pack and send" mode, as noted, the software agent compresses and stores the voice message file, which Zydney refers to as a "voice container," on the client device. (*Id.*, 16:3-4, 12:1-8, 10:20-11:3.) The sender also can include "multimedia attachments" with the voice message, as illustrated for example in Figure 6. (*Id.*, 19:2-8, 22:17-20, Fig. 6.)

78. The software agent then transmits the voice container (and any attachments) to either the central server for delivery to the recipient or, alternatively, directly to the recipient. (*Id.*, 12:1, 12:20-23, 16:7-10.)

79. If the recipient is **online**, it receives the voice container immediately. (*Id.*, 1:21-22 ("routed to the appropriate recipients instantaneously.").) If the recipient is **offline**, the server stores the voice container until the recipient is available, as shown in Figure 4. (*Id.*, 13:12-15, 14:9-11, Fig. 4 ("if recipient is not online, client sends voice container to server file").) More specifically, the central server will forward the stored voice container to the recipient once they log in. (*Id.*, claim 1, 14:14-16, Fig. 4 ("recipient logs on to internet or intranet," "server recognizes recipient, downloads voice container"), 16:10-12 ("If the intended recipient has a compatible active software agent on line after log on, the central server downloads the voice recording almost immediately to the recipient.").)

80. Upon receipt of a voice container, the recipient's software agent unpacks the voice container and any attachments and presents them to the recipient. (*Id.*, Fig. 18, 35:20-22.) The software agent can then audibly play the voice message to the recipient through the speakers or headset attached to the device. (*Id.*, 13:19-22, 14:14-16, 16:10-14.)

2. Brief Summary of Shinder [Ex. 1014/1114]

81. Shinder, entitled "Computer Networking Essentials," was published by Cisco Systems, a well-known supplier of networking equipment. The book was written to "help[] you understand the fundamentals of computer networking concepts and implementation and introduce[] you to the client and server operating systems that run on networked PCs." (Shinder, Introduction, p.xxii.)

82. I have cited Shinder primarily in connection with limitations in the independent claims reciting a "**network interface**" that facilitates communication with a network. Zydney, the primary reference cited in my Declaration, does not specifically describe the computing hardware used to connect the client or server devices to a network. But Shinder confirms that there was nothing inventive or non-obvious about providing a "**network interface**" in the manner recited in the claims. For example, one well-known example of a network interface was known as a network interface controller (NIC), which was widely available. (Shinder, p.195

("The most basic piece of hardware required to network computers is the NIC, also called a network adapter or network card.").) Shinder further teaches:

> **Some sort of network interface is always required to communicate over a network**. . . . The NIC is the basic hardware component of network communications. It translates the parallel signal produced by the computer into the serial format that is sent over the network cable. The 1s and 0s of binary communications are turned into electrical impulses, pulses of light, radio waves, or whatever signaling scheme is used by the network media.

(*Id*., pp.195-196.) This passage confirms that a networked system, such as the instant messaging system of Zydney, cannot even function without a network interface for connecting the computing devices to the network. Shinder thus confirms the claimed "**network interface**" would have been apparent and obvious, and provides no meaningful distinction over Zydney.

3. Brief Summary of Appelman [Ex. 1004/1104]

83. **Appelman**, entitled "User Definable On-Line Co-User Lists," describes an instant messaging system that keeps track of the logon status of users in an on-line system. (Appelman, Abstract.) I cite Appelman with respect to claims 22 and 39 for its teachings regarding displaying an indication as to whether a potential recipient is available to receive a message.

84. Appelman describes a technique for allowing a user to create a list of users called a "**Buddy List**," which records the names of selected other co-users with whom the user may wish to communicate. (*Id.*, 1:53-59, Fig. 3.) The buddy list also keeps track of whether the other co-users are currently logged onto the system. "When a user logs on to a system, the user's set of buddy lists is presented to the buddy list system. The buddy list system attempts to match co-users currently logged into the system with the entries on the user's buddy list. Any matches are displayed to the user. As co-users logon and logoff, a user's buddy list is updated to reflect these changes." (*Id.*, 1:64-2:2; *see also id.*, 2:51-3:6.)

85. Figure 2a of Appelman, reproduced below, shows an example Buddy List table **32** that records the screen name of each "buddy" user as well as the connectivity status of each user (whether the user is logged "in" or logged "out").

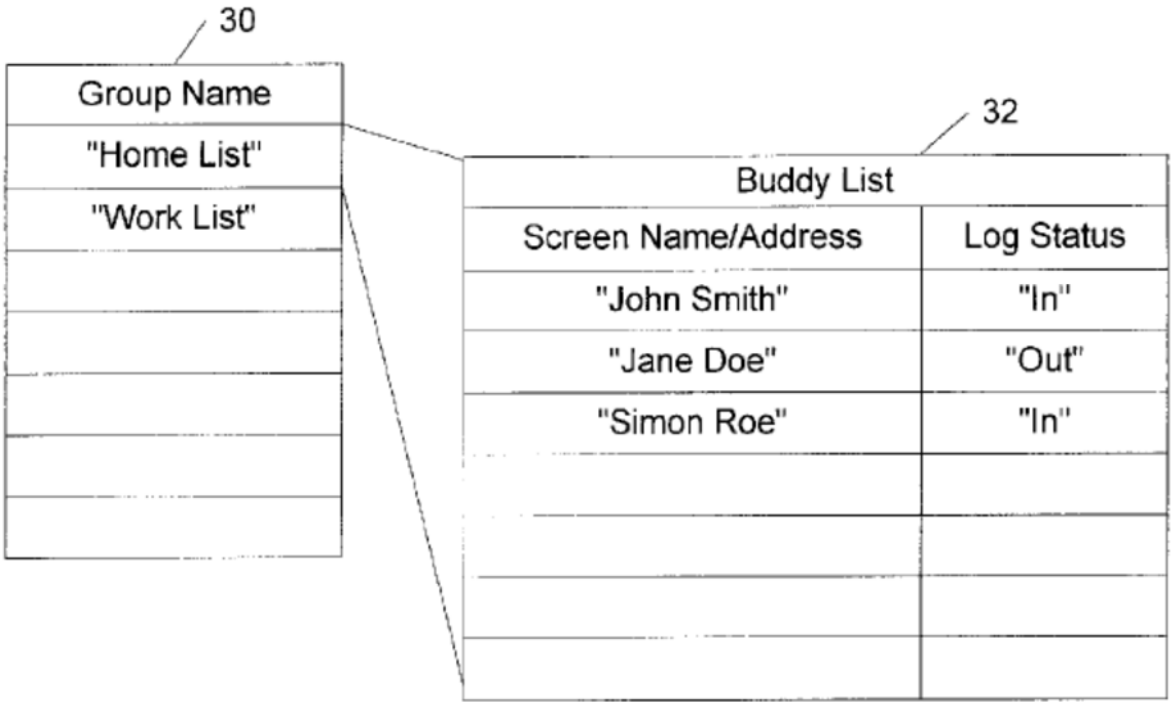


FIG. 2a

(Appelman, Fig. 2a.) Appelman explains that each user can create different buddy lists. (*Id.*, 3:61-64.) In the example above, the user has created two buddy lists ("Home List" and "Work List"). The buddy list called "Home List" contains the name/address and logon status for three users: "John Smith," "Jane Doe" and "Simon Roe." (*Id.*, Fig. 2a, 3:41-47.) For each user, the table indicates either "**IN**" to indicate that the user is currently logged into the system, or "**OUT**" to indicate that the user is not logged in. (*Id.*, 3:43-47, 4:4-7.)

86. When a user logs in, the system presents a user interface to the user that displays the buddy list, including the name and connectivity status of each co-user on the list. (*Id.*, 2:66-3:8, 4:28-36.) Figure 3 below shows an exemplary user interface for the "Home List" buddy list shown in Figure 2a above:

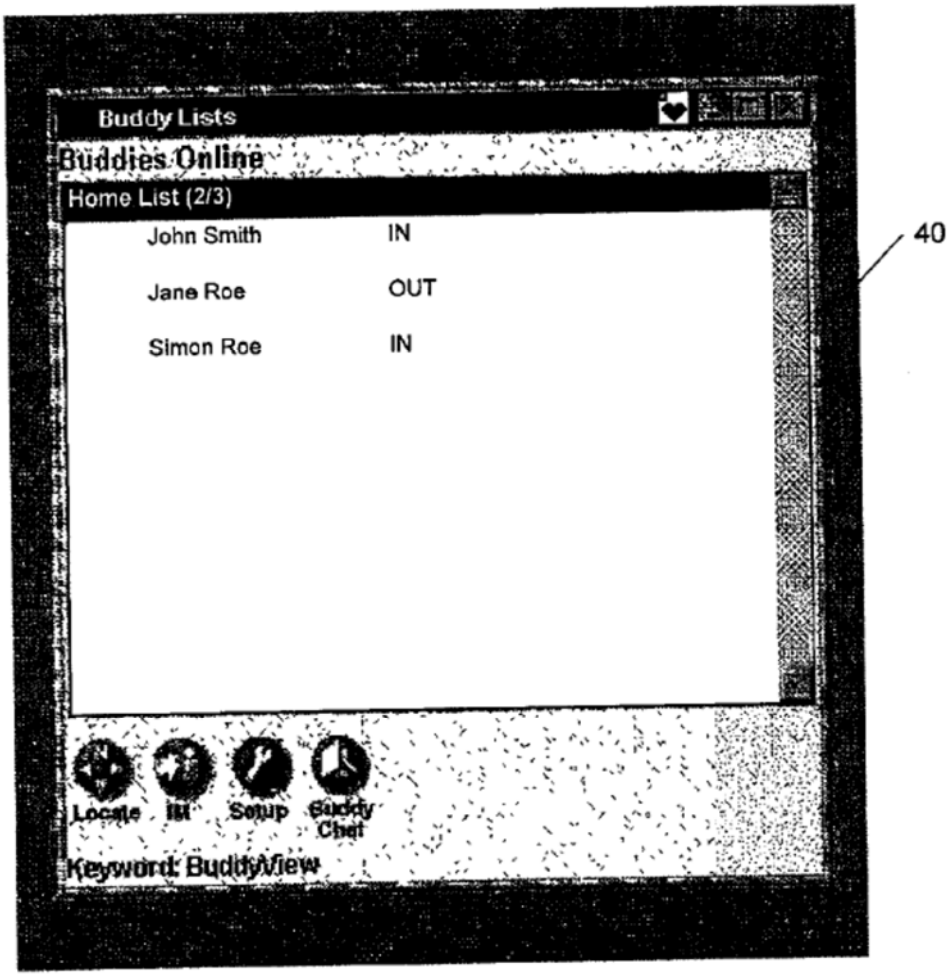


FIG. 3

(*Id.*, Fig. 3; *id.*, 4:29-32 ("In the preferred embodiment, when the user first logs into the system the Buddy List window **40** [in Figure 3] opens, informing the user which

of the user's buddy list members are currently online.").) The example above shows that users "John Smith" and "Simon Roe" are "IN," but "Jane Roe" is "OUT," directly corresponding to the Buddy List table **32** in Figure 2a. (*Id.*, Figs. 3, 2a.) "A user can also keep tabs on each list by checking out the numbers posted in parenthesis next to the buddy list names. This number tells the user how many people on that list are logged in [sic; or] out of the total number of screen names on the buddy list. In the illustrated example, 2/3 means that two of the three people on the 'Home List' are currently online." (*Id.*, 4:23-29.)

87. The "**IN**" and "**OUT**" statuses shown in Figure 2a and Figure 3 correspond, respectively, to the claimed "**indicia for each of the one or more potential recipients indicating whether the potential recipient is currently available to receive an instant voice message**," as recited in claims 22 and 39 of the '622 patent. This is because users who are logged into the system (who are "**IN**") are available for instant message communications. (Appelman, 6:1-5 ("Instant Messages [] Once a co-user is displayed on a user's buddy list, indicating that the co-user is currently logged into the network system, the preferred embodiment of the invention enables a simple way of communicating with that co-user.").) For example, the bottom of Figure 3 shows a button called "**IM**" that allows the user to send an instant message to another user. (*Id.*, 6:1-16 (describing Instant Message

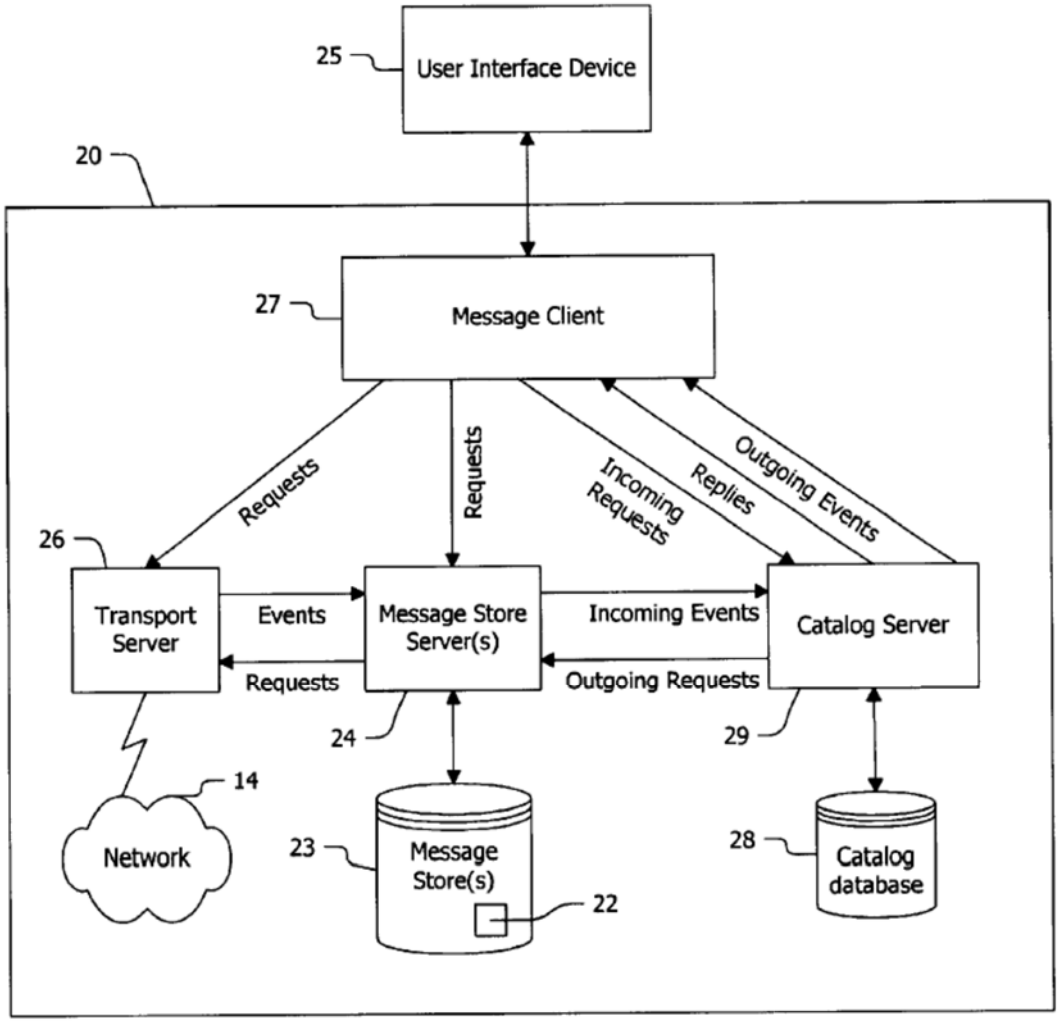
functionality); *id;*, 6:13-16 ("The Instant Message function can also be activated by highlighting a co-user name in the buddy list and clicking on the IM button in the Buddy List window **40** (FIG. 3).").)

4. Brief Summary of Clark [Ex. 1008/1108]

88. **Clark**, entitled "System for Managing and Organizing Stored Electronic Messages," describes a system that "catalogs and retrieves electronic messages saved in a message store." (Clark, Abstract.) I cite Clark for its teachings relating to a "**message database**" and "**file manager system**" and related limitations as recited in claims 14-17 and 28-31.

89. Clark discloses a system that can "automatically organize stored electronic messages, such as e-mail messages, instant messages, voice messages and fax messages." (*Id.*, 4:9-12.) Clark's system works with a client-server messaging system that operates over a computer network, as shown in Figure 1A of Clark. "FIG. 1A shows a very simple computer network **14** which connects a server computer **12** and two user computers **16** and **18**." (*Id*., 7:67-8:2; Fig. 1A.) The messaging system could be an email system or "could be applied equally well to other types of messages or to messaging in a mixed environment handling different types of messages." (*Id*., 8:7-10.)

90. Each user computer may operate "messaging client software," and the disclosed message database system can "advantageously be integrated with messaging client software . . . to facilitate the organization of electronic messages." (*Id*., 4:35-38.) In particular, the user's computer **18** contains a message store **23** that discloses a **message database**. Figure 2 of Clark, reproduced below, shows an example computer system **20**, which may be user computer **18** operating in a client/server network as shown in Figure 1A. (*Id*., 4:61-64, 9:7-22.)



(*Id*., Fig. 2.)

91. As shown in Figure 2, "[a] collection of electronic messages **22** is stored in one or more message stores **23**. Each message store **23** comprises a memory, file or database structure that provides temporary or permanent storage for the contained messages **22**." (*Id*., 9:11-15.) A "message store server **24** manages the messages **22** in message store **23**," including receiving requests for messages from other parts of the system and providing the messages. (*Id*., 9:15-20.)

92. Figure 4A, reproduced below, provides another illustration of an embodiment where a user's computer **18** contains the message client **27**, message store server(s) **24**, and message store **23**. (*Id*., 10:27-33, Fig. 4A.)

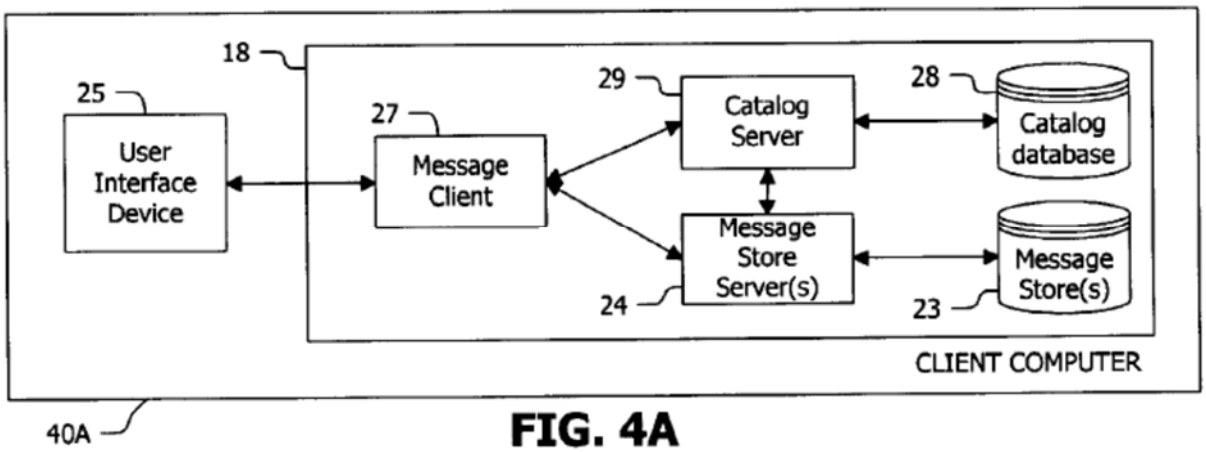


FIG. 4A

(*Id*., Fig. 4A.)

93. Clark describes that the message database "can be applied to organizing any sort of electronic messages which are to be temporarily or permanently stored,"

including "instant messages," "voice mail messages," and "any other present or future types of electronic messages," which may also include attachments of various types. (*Id*., 8:31-44.) In the disclosed embodiment, the message store **23** stores both outgoing (sent) and incoming (received) messages, including messages that are currently "unsent." (*Id*., 17:12-22.) The messages may be organized by fields such as MessageDateTime (date and time the message was sent or received), DisplayNames (the sender's name for received messages, or recipient names for sent or unsent outgoing messages), and Subject. (*Id*.)

5. Brief Summary of Hethmon [Ex. 1009/1109]

94. **Hethmon**, entitled "Illustrated Guide to HTTP," is a book published in 1997 that describes the HyperText Transfer Protocol version 1.1 (HTTP/1.1). I cite Hethmon in connection with the "**action field**" limitations of claims 4-5 and the "**connection object messages**" limitations of claims 24-26.

95. HTTP is a well-known protocol used to send and receive messages between clients and servers on the Internet. As Zydney explains, HTTP "is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods (commands). A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being

transferred clients and servers." (Zydney, 7:21-8:3.) HTTP was the subject of published standards, including an Internet Engineering Task Force (IETF) document cited by and incorporated-by-reference in Zydney. (*Id.*, 8:3-6.) Although Zydney discloses the use of HTTP, it does not disclose the details of how HTTP operates, presumably because those details were already well-known as noted.

96. I cite Hethmon to show that the "action field" and "connection object messages" limitations in claims 4-5 and 24-26, respectively, are nothing more than features that were built-in to HTTP/1.1 as described in Hethmon.

97. HTTP is commonly referred to as a "request-response" protocol, which refers to a communications protocol in which a client application sends a "request message" to a server, which responds by sending back a "response message." (Hethmon, p.10.) This basic interaction is illustrated in the figure below from Hethmon:

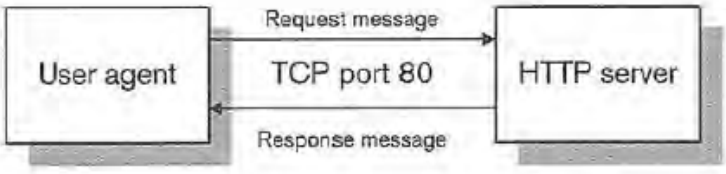


Figure 2.2 Basic client to server HTTP operation

(Hethmon, Figure 2.2 at p.11.)

98. For purposes of my Declaration, a key aspect of HTTP is the content of the HTTP "request message." Hethmon explains that each HTTP request message contains a "Request-Line," that contains information about the request. One of the fields of the "Request-Line" is known as a "Method," which as I will explain in my detailed analysis of claim 4, discloses the claimed "**action field**" of that claim. Hethmon explains that the Method is a keyword "to indicate the type [of] action the request is asking the server to execute." (*Id*., p.55.) Furthermore, in HTTP/1.1, the Request can specify any one of a predetermined set of seven different methods, namely OPTIONS, GET, HEAD, POST, PUT, DELETE, and TRACE. (*Id*., p.55-61 ("With HTTP/1.1, there are seven basic methods . . .").) The Request-Line therefore discloses **an action field identifying one of a predetermined set of permitted actions requested by the user**.

99. Another aspect of HTTP that is pertinent to my declaration is the way in which HTTP requests satisfy the "**connection object messages**" of claims 24-26. Hethmon discloses that in HTTP, the request sent to the server can result in establishing and maintaining a connection. (Hethmon, pp.10-11.) I discuss these features in more detail in my discussion of claims 24-26.

100. As I will show below, the claimed "action field" and "connection object messages" describe nothing more than built-in features of HTTP/1.1 that were well-

known to persons of ordinary skill in the art, and thus, present no non-obvious distinction over Zydney.

6. Brief Summary of *Microsoft Press Computer Dictionary* (1991) [Ex. 1018/1118] and Moghe [Ex. 1019/1119]

101. Dependent claim 12 recites the step of "updat[ing] the connection information for each of the instant voice message client systems" by "periodically transmitting a connection status request" to a client. My Declaration cites to the *Microsoft Press Computer Dictionary* (1991) and Moghe for their description of a well-known technique known as "polling" that discloses and renders obvious the claimed connection status request.

102. Zydney discloses a central server that, among other things, keeps track of the connection status of clients in the instant messaging system. Zydney explains that the central server "will track and maintain the status of all software agents," which "is frequently conveyed to the software agent by the central server." (Zydney, 14:8-9, 14:20-22.) But Zydney does not describe in detail **how** the central server continuously maintains updated status information about those software agents. Zydney explains that software agents on the client devices notify the central server of their status upon logging on (*id.*, 14:2-4), but Zydney does not describe a mechanism by which the central server obtains updated status information from those clients. I accordingly cite **Microsoft (1991)** and **Moghe** for the

straightforward proposition that it would have been obvious to adapt the system of Zydney to obtain that status by periodically requesting connection status from each client using a well-known technique known as "polling."

103. The *Microsoft Press Computer Dictionary* (1991) states that it is "designed to be a comprehensive and authoritative source of definitions for computer-related terms and acronyms." (Microsoft (1991), Ex. 1018/1118, p.vii.) It provides the following definition of "polling" or "autopolling":

> **Autopolling** Also called polling. The process of periodically determining the status of each device in a set so that the active program can process events generated through each device. The process can be used to determine the status of a range of events such as whether a key or a mouse button was pressed or whether new data is available at a serial port. Autopolling can be compared with event-driven processing, in which a low-level routine in the operating system alerts a program or routine to an event occurring in a device with an interrupt or message, rather than requiring the program to check each device in turn.

(Microsoft (1991), pp. 26-27 (boldface in original).)

104. As the definition above confirms, polling is often contrasted with "event-driven" systems in which the second system detects a status change and automatically notifies the first system of the change. In an "event-driven" system, therefore, there is no need for continuous "polling" for status changes because

affected systems themselves automatically issue notifications upon detection of status changes. Polling and event-driven processing were, and still are, two well-known and alternative ways of obtaining status from all manner of devices, including devices in a network. Microsoft (1991) notes, for example, that polling of "network nodes" may be performed. (*Id*., p.272 (defining "polling cycle" as "[t]he time and sequence required for a program to poll each of its devices or network nodes. *See also* autopolling.") (italics in original).)

105. In the context of computer networking, "polling" often involves a first system sending a status request to a second system at some specific interval in order to check on the status of the second system. This is confirmed in **Moghe**, which confirms that polling is applicable in the context of a networked computer system. The Background section of Moghe explains that "polling" provides a means for requesting the status of other devices or resources on a network:

> Typically one host on the network is assigned the task of network manager ("NM") **10**, running appropriate software, while the remaining hosts and resources are identified as agents. The manager **10** will periodically request information from the agents using one of a variety of protocols, e.g. Simple Network Manager Protocol ("SNMP") at the application layer, or Packet Internet Groper ("PING") at the IP layer, and expect a response from each agent using the same protocol. This process is referred to as "polling."

(Moghe, 1:14-22.) Moghe further explains that "[e]fficient polling is becoming increasingly important with new bandwidth-intensive applications such as conferencing and web-push applications." (*Id.*, 1:23-24.)

106. Moghe goes on to describe an enhanced polling technique in which the rate or frequency of periodic polling requests may be varied based on, among other things, network congestion. (*Id.*, 2:3-7.) I have cited Moghe for the more general proposition, confirmed above, that polling was a well-known technique for obtaining status information from network-connected devices.

107. In my opinion, the recitation of a "connection status request" in claim 12 recites nothing more than an implementation of known polling techniques that would have been obvious to a person of ordinary skill in the art.

B. Zydney and Shinder Render Obvious Claims 3, 6-8, 10, 11, 13, 18-21, 23, 27, 32-35, and 38.

1. Independent Claim 3

108. I have reproduced independent claim 3 below using bracketed notations (e.g. "**[a]**," "**[b]**," etc.):

3. A system comprising:

[a] a network interface connected to a packet-switched network;

[b] a messaging system communicating with a plurality of instant voice message client systems via the network interface; and

[c] a communication platform system maintaining connection information for each of the plurality of instant voice message client systems indicating whether there is a current connection to each of the plurality of instant voice message client systems,

[d] wherein the messaging system receives an instant voice message from one of the plurality of instant voice message client systems, and

[e] wherein the instant voice message includes an object field including a digitized audio file.

('622, Claim 3.) Zydney and Shinder disclose and render obvious claim 3.

(a) Preamble of claim 3: "A system comprising:"

109. To the extent the preamble is limiting, **Zydney [Ex. 1003/1103]** discloses "[a] system" comprising the features discussed below.

(b) "a network interface connected to a packet-switched network" (Claim 3[a])

110. I will address this limitation in two parts, the first addressing the "network interface" limitation, the second addressing the requirement that the interface be "connected to a packet-switched network."

(i) "a network interface"

111. The written description of the '622 patent does not provide much detail regarding the claimed "**network interface**" or much guidance on the meaning of this term. With respect to the IVM server **202**, the specification merely states: "The

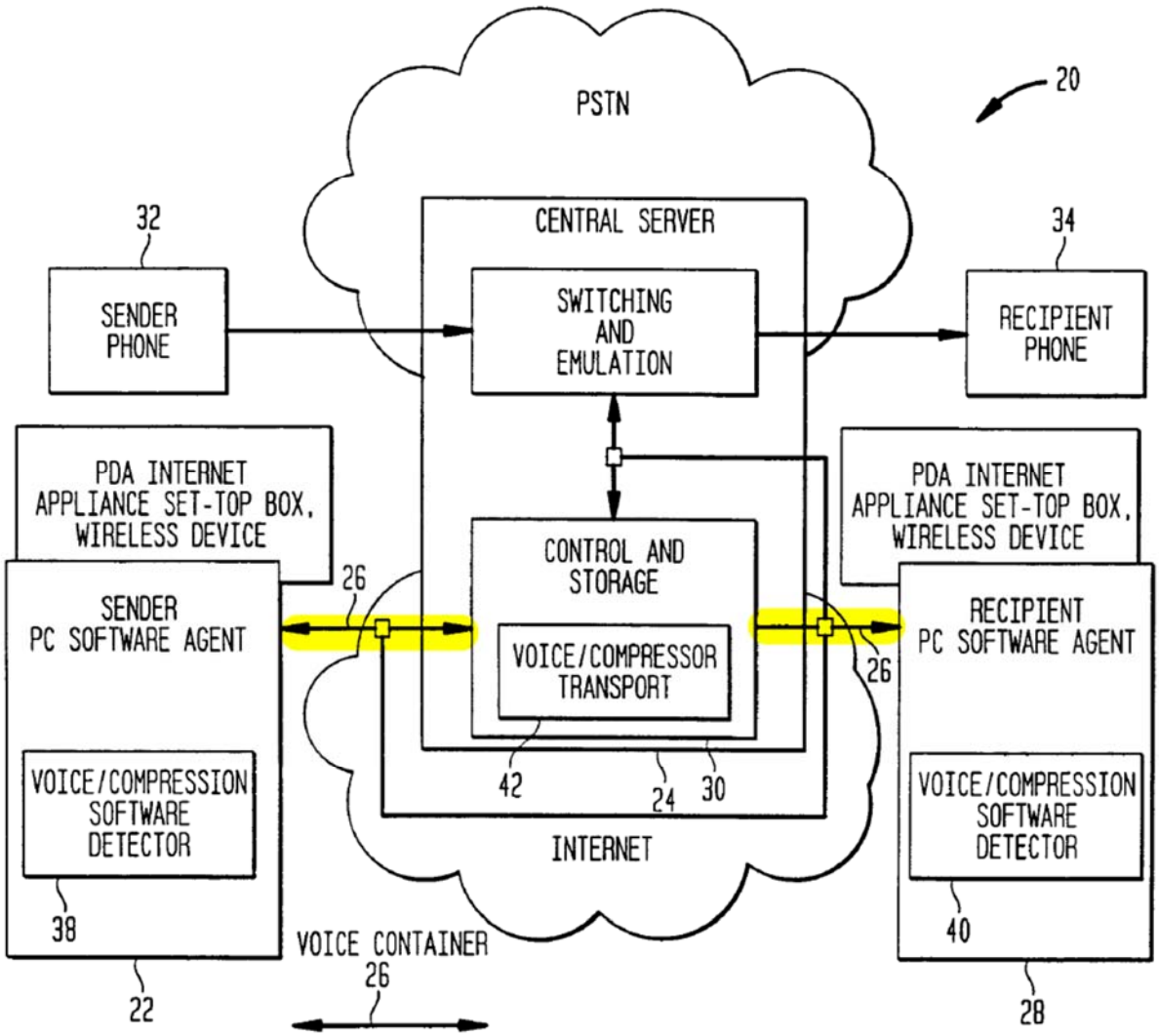
IVM server **202** is a general-purpose programmable computer equipped with a network interface, such as an Ethernet card, to provide connectivity to a network **204**." ('622, 13:41-44.) As I will explain in more detail below, the exemplary "Ethernet card" mentioned in the '622 patent was a well-known piece of computing hardware that provided an interface by which a computer could communicate on a network. The specification thus indicates that the term "**network interface**" should be used under its ordinary meaning to refer to computing hardware that provides connectivity to a computer network. I am informed that in the pending litigation, the Patent Owner has proposed to construe "**network interface**" as a "component that provides connectivity to the network." This definition appears to be consistent with my own broadest reasonable construction, and as such, my analysis below would not change even if Patent Owner's proposal were adopted.

112. Zydney, alone and in combination with Shinder, discloses and renders obvious the claimed "**network interface**." As I explained in the brief summary of Zydney in **Part VI.A.1** above, Zydney describes a system where clients create and transmit instant voice messages in the form of "voice containers." (Zydney, 12:1-8, 10:20-11:3, Fig. 1A.) The client systems include software agents that can transmit the voice containers over the Internet to a **central server**, which can then deliver the

voice container to one or more recipients over the Internet or store it for later delivery. (*Id.*, 13:1-6, 13:12-18, 14:6-13, Figs. 4, 8.)

113. Zydney discloses that the voice containers are transmitted over the Internet to the central server in data packets using well-known TCP/IP protocols. (*Id.*, 23:11-12 ("[t]he voice container will be sent using standard TCP/IP transport"), Fig. 2 (identifying "Transport Processes (TCP/IP, UDP, PSTN, Others)").) This is shown in Figure 1A, which depicts "transmission line 26" connecting the client systems to the central server through the Internet, as highlighted below. (*Id.*, 10:21-23, Fig. 1A (line 26).)

FIG. 1A



(*Id.*, Fig. 1A.)

114. Although Zydney does not describe the specific hardware used by the central server to connect to the Internet, it would have been obvious to a person of ordinary skill in the art that the central server in Zydney included a "network interface" as claimed. This is because the central server would have needed such an

interface in order to connect to the Internet and communicate information over the Internet using TCP/IP, as shown in Figure 1A. In my opinion, the presence of a "network interface" would have been obvious based on the disclosures of Zydney alone, when combined with the knowledge of a person of ordinary skill in the art.

115. Nevertheless, in the event the Patent Owner were to argue that the disclosures of Zydney alone were insufficient, the obviousness of the claimed "network interface" is further confirmed by **Shinder** [Ex. 1014/1114]. Shinder describes network interface controllers (NICs),[11] which were well-known in the prior art as noted previously. (Shinder, pp.195-196.) Shinder explains: "The most basic piece of hardware required to network computers is the NIC, also called a network adapter or network card." (*Id*., p.195.) Shinder further teaches:

> **Some sort of network interface is always required to communicate over a network**. . . . The NIC is the basic hardware component of network communications. It translates the parallel signal produced by the computer into the serial format that is sent over the network cable. The 1s and 0s of binary communications are turned into electrical impulses, pulses of light, radio waves, or whatever signaling scheme is

[11] Persons of ordinary skill in the art use the acronym "NIC" to refer to "network interface controller," and alternatively, "network interface card." For purposes of my analysis, the difference is immaterial.

used by the network media.

(*Id*., pp.195-196.) Shinder also explains that the network interface (such as a network interface card) controls all incoming and outgoing data traffic to and from a networked computer: "Along with preparing the data to go onto the network media, the NIC is responsible for controlling the flow of data between computers and media and for receiving incoming data." (*Id*., p.196.) Shinder thus discloses the claimed "**network interface**."

116. ***Rationale and Motivation to Combine***: It would have been obvious to a person of ordinary skill in the art to combine Zydney with Shinder, with no change in their respective functions, predictably resulting in the voice instant messaging system of Zydney in which the central server contained a "network interface," such as a network interface card (NIC), to connect the server to the Internet. Shinder itself provides a motivation to combine by explaining that "[s]ome sort of network interface is **always** required to communicate over a network." (Shinder, pp.195-196.) To the extent this requirement was not already known to a person of ordinary skill in the art (see my discussion below), Shinder would have clearly motivated a person of ordinary skill in the art to provide a network interface to connect the central server of Zydney to the Internet. Shinder would have confirmed, for a person of ordinary skill in the art, that a "network interface" is not only desirable, but essential

to allow the central server of Zydney to perform its communications functions within the instant messaging system.

117. Shinder further explains that various types of network interfaces were well-known and could have been selected as a matter of implementation choice based on considerations such as data transfer speed, network architecture, media type, and available bus type. (Shinder, pp.196-197.) A person of ordinary skill in the art would have thus found it obvious that the central server would be coupled to a network interface for communicating over the Internet, and the choice of which particular network interface to use would have been well within the grasp of a person of ordinary skill in the art. (*See id*.)

118. Finally, as I alluded to above, providing a network interface was within the basic knowledge of persons of ordinary skill in the art. The '622 patent itself explains that the network interface can include "an Ethernet card, to provide connectivity to a network **204**." ('622, 13:41-44.) The term "card" would have been understood by a person of ordinary skill in the art to refer to hardware logic contained, for example, on a board (often called a "card") that plugs into the computer's expansion slots. An Ethernet card is an example of a NIC (as described above), and such cards were commonly installed in computers, including computers used as servers. These cards typically contained physical sockets that allowed the

user to plug-in an Ethernet cable that supplied a connection to the network,[12] and ubiquitous on computers that were connected to a network. A person of ordinary skill in the art would thus have fully understood the need for a network interface in implementing the voice instant messaging system of Zydney. Zydney in view of Shinder therefore renders obvious the claimed "**network interface**."

(ii) "…connected to a packet-switched network"

119. Zydney and Shinder also disclose and render obvious that the network interface is connected to "**a packet-switched network**," as claimed. (Zydney, 10:21-23, Figs. 1, 1A.) The term "**packet-switched network**," as recited in the claim, generally refers to a type of communications network in which information is transferred through a series of data units called "packets." Such a network is disclosed and obvious in view of Zydney.

120. As I noted previously, Zydney discloses that the voice containers are transmitted over the Internet. It was well-known to persons of ordinary skill in the art that the Internet was a packet-switched network. The '622 patent itself confirms

[12] Ethernet network interfaces remain prevalent on computers today, but many computer motherboards today have built-in Ethernet support and thus do not require a separate "card" to provide Ethernet network connectivity.

as much. ('622, 1:37-40 ("a VoIP terminal device is connected to a packet-switched network (e.g., Internet)"), 1:51-53 ("FIG. 1 is an illustrative example of a prior art IP telephony system 100 [which] comprises a packet-switched IP network 102, such as the Internet").) Shinder also confirms that the Internet is a packet-switched network. (Shinder, p.170 ("An example of a packet-switched network is the Internet.").) This is also confirmed in Zydney, which describes transmission of data packets over the Internet using well-known TCP/IP and UDP protocols. (Zydney, 23:11-12 ("[t]he voice container will be sent using standard TCP/IP transport"), Fig. 2 (identifying "Transport Processes (TCP/IP, UDP, PSTN, Others)").)

121. It would also have been obvious to a person of ordinary skill in the art that the Internet as disclosed in Zydney would have been a packet-switched network, as claimed. The Internet was the most well-known and most widely used packet-switched network as of December 2003. Zydney itself explains that its system "relates to the field of packet communications, and more particularly to voice packet communication systems." (Zydney, 1:4-5.) As Zydney explains, "Transaction Control Protocol/Internet Protocol (TCP/IP) is the communications standard between hosts on the Internet. TCP/IP defines the basic format of the digital data packets on the Internet allowing programs to exchange information with other hosts on the Internet." (*Id*., 5:15-18.) TCP/IP would have been familiar to a person of

ordinary skill in the art as a reliable end-to-end transport protocol for packet-switched networks that use the Internet protocol suite, including the Internet. As noted previously, Shinder also identifies the Internet as an example of a packet-switched network. (Shinder, p.170.) Therefore, deploying the system of Zydney using the Internet would have predictably resulted a system for instant voice messaging over a packet-switched network.

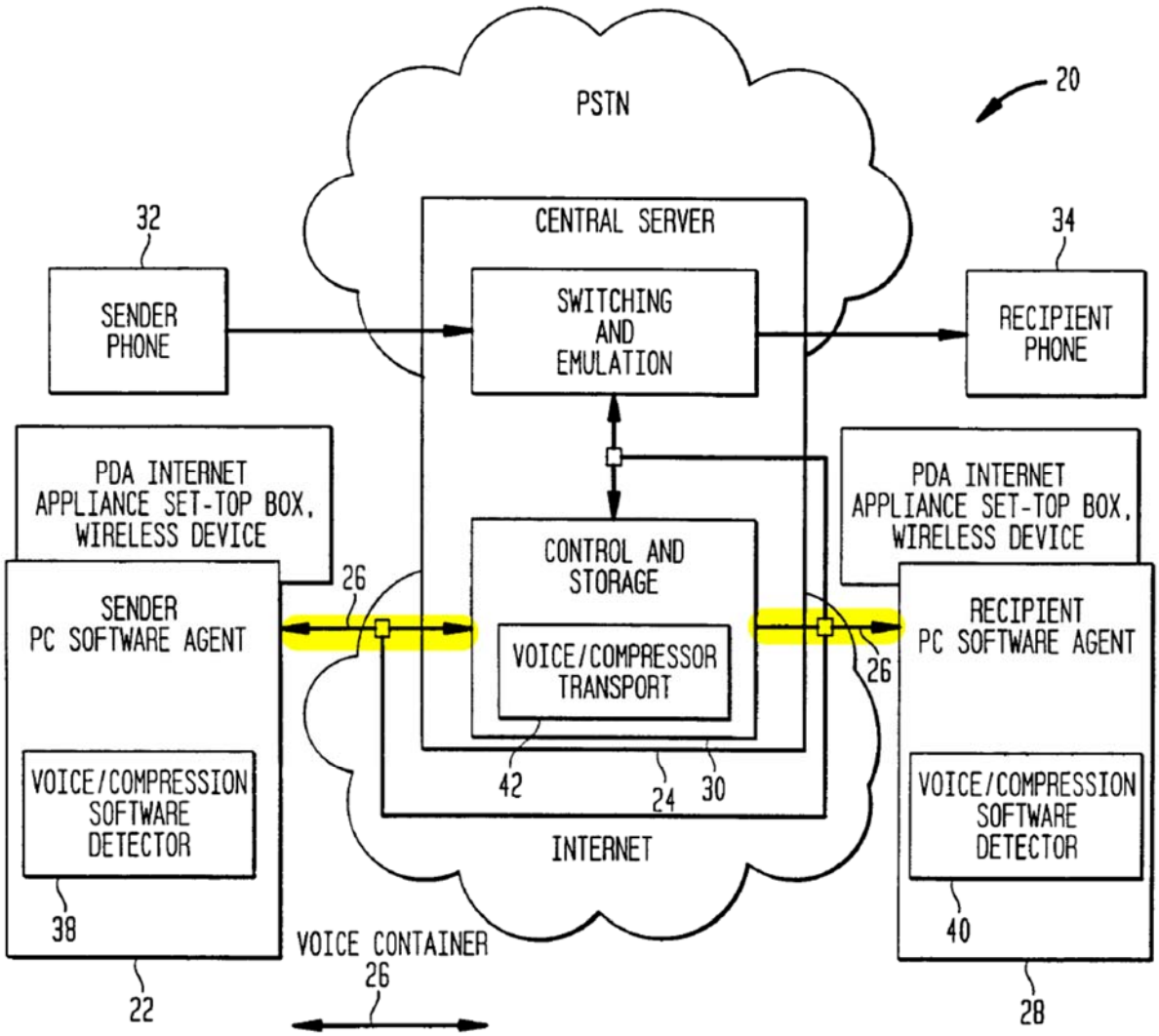
122. Furthermore, for the reasons I discussed previously, it would have been understood and obvious to a person of ordinary skill in the art that the **network interface** itself would be **connected to the packet-switched network**. Shinder explains that the network interface serves as the point of connection through which incoming and outgoing data flows to and from a networked computer. (Shinder, pp.195-196.) In particular, "[a]n important part of the network interface is the *transceiver*," which, "as its name indicates, sends and receives signals." (*Id*., p.196 (italics in Shinder).) Therefore, when the central server in Zydney's system communicates data (including instant voice messages) over the Internet using TCP/IP as Zydney describes, the network interface is connected to the packet-switched network.

(c) "a messaging system communicating with a plurality of instant voice message client systems via the network interface; and" (Claim 3[b])

123. I am informed that in pending litigation, the Patent Owner has proposed to construe "**client**" to mean "instant messaging software or hardware." In my opinion, this construction is consistent with the broadest reasonable interpretation of this term, and as such, I account for it in my analysis below.

124. The messaging system of Zydney is shown, again, in Figure 1A, which shows client systems connected to the central server through the Internet, as highlighted below. (Zydney, 10:21-23, Fig. 1A (line 26).)

FIG. 1A

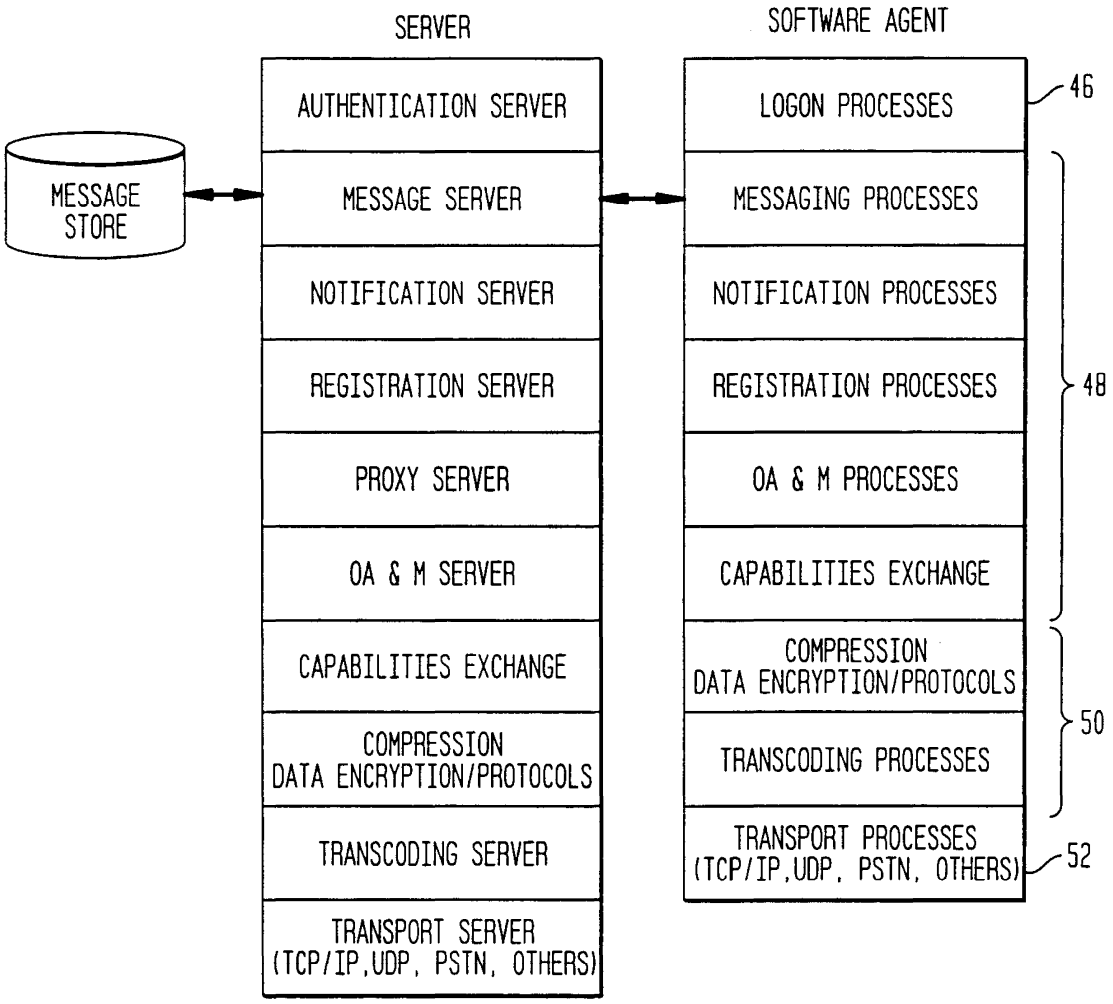


(*Id.*, Fig. 1A.)

125. The "**messaging system**" for purposes of Zydney sits within the central server (**24**) shown in the middle of Figure 1A above. Figure 2, reproduced below, shows the Central Server with its various sub-components including "a message

server 56, a notification server 58, a registration server 60, … and transport server 72" for communicating with client systems. (*Id.*, 13:7-10, Fig. 2.)

FIG. 2



(*Id.*, Fig. 2.) Zydney describes that "[t]he central server in conjunction with the software agent controls, stores and switches the voice containers to the appropriate recipients." (*Id.*, 14:6-13; *see also id.*, 13:12-18.) More specifically, the portion of

the central server that communicates instant voice messages (voice containers) with instant voice messaging client systems, which includes the **message server** within the central server (alone or in conjunction with other components of the server such as the transport server), discloses the claimed **messaging system**. For example, when a client system sends a voice container to a recipient that is "[n]ot logged on," the "[m]essage will be sent to the message server." (*Id*., 33:1-2.) Similarly, "[t]he message server will download all messages to the software agent and/or retain copies of the messages based on administrative settings from the user." (*Id*., 27:15-16; *see also id*., 25:1-9, 30:6-7, 31:1-3.) Zydney also describes that transport server which is responsible for TCP/IP transport as reflected in Figure 2 above, which a person of ordinary skill in the art would understood to disclose a component for receiving and handling the received data according to the TCP/IP protocols. (*Id*., 23:11-12 ("The voice container will be sent using standard TCP/IP transport."), 29:1-2.) Zydney thus discloses the claimed "**messaging system**."

126. Zydney's Figure 1A, reproduced above, also discloses a "**plurality of instant voice message client systems**," which are shown to the left and the right of the central server (**24**) and, more specifically, comprise the sender software agent (**22**) and the recipient software agent (**28**). (Zydney, Fig. 1A) As shown in the yellow highlighting, the client systems are connected to the central server (which

contains the "**messaging system**") over the Internet, and transmit a "voice container" (**26**) over that connection. (*Id.*) Figure 1A thus discloses "**a messaging system communicating with a plurality of instant voice message client systems**," as claimed.

127. Zydney further confirms that the clients shown in Figure 1A above are "**instant voice message client systems**." Zydney explains that the sender and recipient utilize a personal computer or other device that contains a software agent and can send and receive instant voice messages (voice containers), which is depicted as numeral **26** in Figure 1A above . (Zydney, 11:16-18, 14:2-3, Figs. 4, 6.) Each voice container discloses an **instant voice message** because a voice container "can be stored, transcoded and routed to the appropriate recipients instantaneously or stored for later delivery." (*Id*., 1:21-22.) A recipient of the voice message "can reply in a complementary way, allowing for near real-time communication." (*Id*., 16:14-15.) Zydney describes this exchange of voice containers as "a voice instant messaging session," as an alternative to the "intercom" mode that I noted previously in my summary of Zydney. (*Id*., 15:8-13, 10:19-11:3, 16:1-12.) Because the client systems in Zydney are capable of sending and receiving instant voice messages, they qualify as "**instant voice message client systems**."

128. Finally, as I explained previously for claim 3**[a]**, the central server in Zydney can include a "**network interface**" that provides the server's connection to the Internet. The central server (**24**) in Figure 1A ("messaging system"), therefore, communicates with the client systems using the network interface. Zydney and Shinder thus render obvious that the messaging system communicates with the client systems "**via the network interface**," as claimed.

(d) "a communication platform system maintaining connection information for each of the plurality of instant voice message client systems indicating whether there is a current connection to each of the plurality of instant voice message client systems," (Claim 3[c])

129. As I explained for claim 3**[b]**, Zydney's "**instant voice message client systems**" are devices containing software agents used for instant voice messaging. (Zydney, 14:2-3, 2:1-2, 11:16-18, 10:11-12, Claims 13-15.)

130. As noted previously in **Part V.C.3**, I have applied a construction of "**communication platform system**" as a "system of the server which relays communications and/or tracks client connection information." The central server of Zydney discloses such a system.

131. In particular, Zydney explains that the central server tracks the connectivity status of each software agent. (Zydney, 14:6-9 ("Central Server: … will track and maintain the status of all software agents."), 13:12-14 ("The Central

Server provides the following functionality: … maintain and provide the status of all software agents…").) The claimed "**communication platform system**" is thus the system within the central server that tracks and maintains this status. More specifically, the "**communication platform system**" comprises the combination of the "**notification server**" component of the central server, which notifies clients of the connection status of other clients, in combination with other server components that store the actual client connection information. (*Id.*, 24:15-16 ("Software agents will gain access to the system through the log on process which interfaces with the notification server."), 31:13-15 ("The software agent will send a copy of the currently logged on Internet address to the notification server for purposes of notifying other software agents of its status and receiving messages."), 25:4-7 (describing that "a software agent is on-line, i.e. has been authenticated with the system and has notified other software agents via the notification server that they are on-line…"), 32:12-15 ("The notification process will also query the server to find out the other registered software agents that are currently logged onto the system and send the Internet address of the other logged on software agents to the authenticated, newly logged on software agent.").) Zydney thus discloses a "communication platform system" that maintains status information for each of the clients.

132. Zydney also makes clear that the status information maintained by the central server is "**connection information for each of the plurality of instant voice message client systems**." Zydney explains that this "status" includes, for example, the "core state" of whether the software agent is online or offline:

> The status of all recipients entered into the software agent is frequently conveyed to the software agent by the central server. This includes the core states of whether the recipient is online or offline, but also offers related status information, for example whether the recipient does not want to be disturbed.

(Zydney, 14:20-15:1; *see also id.*, 25:4-7 (a software agent is "on-line" when it "has been authenticated with the system and has notified other software agents via the notification server that they are on-line").)

133. These disclosures also satisfy the final aspect of this limitation, which requires that the connection information "**indicat**[es] **whether there is a current connection to each of the plurality of instant voice message client systems**." Zydney's teaching above that the central server maintains the "core states" of whether the software agent is currently "online" or "offline" discloses connection information indicating whether there is "**a current connection**" between the central server and the client system containing the software agent. (*See id.*, 32:12-15 ("The notification process will also query the server to find out the other registered

software agents that are **currently** logged onto the system and send the Internet address of the other logged on software agents to the authenticated, newly logged on software agent.").)

(e) "wherein the messaging system receives an instant voice message from one of the plurality of instant voice message client systems, and" (Claim 3[d])

134. As I discussed above with regard to claim 3**[b]**, Zydney discloses that client systems transmit **instant voice messages** in the form of "voice containers." (Zydney, 1:21-22, 2:1-3, 10:20-11:3, 12:6-8, 14:2-5, Fig. 4.) This is expressly shown in Figure 1A, which shows central server (**24**) receiving a voice container (**26**) from the sending software agent (**22**). (*Id.*, Fig. 1A, above.)

135. As I explained with regard to claim 3**[b]**, the portion of the central server (including the message server) that sends and receives voice containers discloses the "**messaging system**" of claim 3. (*Id.*, 16:7-12 ("Based on status information received from the central server, the agent then decides on whether to transport the voice containers to a central file system and/or sends it directly to another software agent using the IP address previously stored in the software agent. If the intended recipient has a compatible active software agent on line after log on, the central server downloads the voice recording almost immediately to the recipient."); *see also id.*, Fig. 8, Step 1.2.3. (describing the client system "uploading

the voice container(s) to a central file server . . . when a voice recording is complete"), 15:19-21 ("[T]he voice containers are delivered to the central server to manage the ultimate delivery to the recipient."), 27:15-16 ("[t]he message server will download all messages to the software agent and/or retain copies of the messages based on administrative settings from the user"), 12:20-23, Fig. 1A.)

(f) "wherein the instant voice message includes an object field including a digitized audio file." (Claim 3[e])

136. I am informed that in pending litigation, the Patent Owner has proposed to construe "**object field**" as "**a block of data being carried by the message object such as (but not limited to) a digitized instant voice message**." This is not the broadest reasonable construction of the term. The claim language specifies that the "object field" includes a digitized audio file, and does not recite a "message object" that would be "the message object" in the Patent Owner's construction. Nevertheless, Zydney discloses this limitation even under the Patent Owner's construction, as I explain below.

137. As I explained above regarding claim 3**[b]** and **[d]**, Zydney describes a client system generating a "voice container" that discloses the claimed instant voice message. Zydney expressly refers to a voice container as an "object" that contains voice data: "The term 'voice containers' as used throughout this application refers to a container **object** that contains no methods, but contains voice data or voice data

and voice data properties." (*Id*., 12:6-8, Fig. 3.) Figure 3 of Zydney shows "an exemplary embodiment of the voice container having voice data and voice data properties components" including various information about the voice message. (*Id*., 23:1-11, Fig. 3.) Although Zydney does not use the word "**field**" in relation to storage of voice data or voice data properties in a voice container, the use of the term "field" in the claim adds nothing of patentable significance. The term "**field**" in the context of the '622 patent simply refers to a block of data, or the location where a block of data is stored. This is consistent with a standard dictionary definition of "field" in this context. *See*, *e.g.*, Microsoft Computer Dictionary (1997) **[Ex. 1012/1112]** at p.210 (defining "field" as "[a] location in a record in which a particular type of data is stored."). The term "**object field**" would thus have been understood as simply referring to a field containing content that will accompany the instant voice message. This is consistent with the specification of the '622 patent, which explains that: "A message object comprises an action field, an ID field, a source field, a destination field, and an object field." ('622, 14:6-7.) "The content of the object field is a block of data being carried by the message object, which may be, for example, a digitized instant voice message." ('622, 14:37-40.) I also note that the claim and specification do not require that the claimed "**object field**" take any particular form or be implemented in any particular way.

138. Accordingly, although Zydney does not use the word "field" in reference to the stored voice data, it clearly discloses such a field. Zydney describes a "voice container" as "a container object that… contains voice data or voice data and voice data properties." (Zydney, 12:6-8.) It would have been obvious to a person of ordinary skill in the art that the voice data inside the voice container object could have been stored in a field.[13]

[13] The term "field" as it relates to network-based data transmission does not refer to a physical "field," such as the field on **a** paper-based form. When an object is being transmitted over a network, the contents of the object (including any "fields" in the object) are generally transmitted as a series of signals over the communications medium. There are a number of known and trivially simple techniques for encoding the data into a series of "fields," which the recipient device can use to locate and obtain the different pieces of data in the fields. For example, one common technique is to simply define a "field" as the data at a predetermined location in the object (*e.g.* field X starts at byte offset Y in the object), which does not require any additional information be sent to separate the fields. Another approach involves placing identifying "tags" around blocks of data in the object to separate and identify the different fields. In any case, and regardless of the implementation chosen, it would

139. Zydney further discloses that the instant voice data in the voice container is a "**digitized audio file**," as claimed. Zydney makes clear that, in creating the instant voice message, the voice data is stored as a digital audio file: "the originator digitally records messages for one or more recipients using a microphone-equipped device and the software agent. The software agent compresses the voice and stores the **file** temporarily on the PC if the voice will be delivered as an entire message." (Zydney, 16:1-4; *see also* 21:14-16 ("The voice recordings that are made via a microphone or converted by text-to-speech software can be used for many other purposes. These voice files can be played and recorded

---

have been obvious that the voice container has an "**object field**" that contains the voice data because the recipient's software agent can receive the voice container, identify the voice data within the container, and decompress it for playback. (Zydney, 16:12-14 ("The voice is uncompressed and the recipient can hear the recording through the speakers or headset attached to their computer.").) Without some logically identified "field" in the voice container containing the voice data, in fact, the recipient device in Zydney could not separate the voice data from the other fields in the voice container (including the fields shown in Figure 3).

using voice container enabled devices."), 20:11-14 (describing "permitting the recording of one or more voice packet messages.").) As I noted previously, each voice container can contain "voice data or voice data and voice data properties." (*Id.*, 12:6-8.) Thus the voice container includes the content of the recorded voice audio file and may also include properties of the voice data. (*Id*., 12:6-8, 23:1-11.) The portion of the voice container that contains the recorded voice file therefore discloses the claimed **object field including a digitized audio file**.

140. Under the Patent Owner's construction, the voice data in the voice container discloses **a block of data being carried by the message object such as (but not limited to) a digitized instant voice message**. As noted previously, the voice container in Zydney is an "object" that carries the digitized instant voice message (the voice message spoken by the sender). (*Id*., 12:6-8, 23:1-11.) Zydney thus discloses this limitation even under the Patent Owner's proposed definition.

141. Furthermore, as a separate yet equally sufficient basis to find this limitation in Zydney, Zydney explains that the voice container can be formatted using the industry-standard Multipurpose Internet Mail Extension ("MIME") format, which "allows non-textual messages and multipart message bodies attachments to be specified in the message headers." (*Id*., 19:7-10.) The voice container formatted in MIME format provides another disclosure of the claimed

object field including a digitized audio file. In particular, as noted above, Zydney describes that the voice message is recorded as a digitized audio "file" that is then captured in a voice container. (*Id*., 16:1-4, 21:14-16.) As explained in detail below, MIME-encoded messages contain a number of fields, including a field (called the "body" in the case of a message) that can hold audio content.

142. When in MIME format, the voice container would contain the digitized audio file in an object field such as a message or body part. This is confirmed by RFC 1521 from September 1993, one of the MIME standards that Zydney cites and expressly incorporates by reference. (*Id.*, 19:13-20:9, 19:22-20:2 (RFC 1521)). Because Zydney itself discloses that voice containers can be encoded using MIME and directly cites to RFC 1521, it would have been plainly obvious to a person of ordinary skill in the art to provide the receiving software agent with the ability to format the voice container according to RFC 1521.

143. The RFC 1521 standard incorporated-by-reference in Zydney **[Ex. 1006/1106]** explains that a MIME multipart message can contain a number of different types of content, including text, audio or voice data, image data, and others that are contained in the "body" of the message. (Ex. 1006/1106, p. 2, ¶¶ 2.a, 2.b, 2.e, 2.f.; *id.*, pp. 3-4 (defining "body" as "the body of an entity, that is the body of either a message or of a body part.").) RFC 1521 specifies that "Content-Types

provide a standardized mechanism for tagging messages or body parts as audio, image, or several other kinds of data." (*Id*., p. 48; *see also id*., pp. 3-4 (defining "message," "body part," and "body").) Thus, a Content-Type header field can be "used to specify the type and subtype of data in the body of a message." (*Id*., p. 2, ¶ 2.) In particular, RFC 1521 discloses "an 'audio' Content-Type value, for transmitting audio or voice data." (*Id*., p.2, ¶ 2.f; *see also id*., p.47, ¶ 7.6 (same).) RFC 1521 thus confirms that a MIME message "**body**" (when the "Content-Type" is "audio") discloses an "**object field including a digitized audio file**" because the body is the field of the MIME message in which the digitized audio file is contained.

144. Appendix G of RFC 1521 also describes the process by which any local data in "the system's native format" such as "audio data" can be encoded into the message body of the MIME formatted message. (*Id*., pp. 66-67.) Appendix G thus provides a further detailed disclosure of how to create and populate the body of a MIME message (*i.e.* an "**object field**" in a Zydney voice container when MIME is used) to include the digitized audio file. (*Id*.)

145. Zydney therefore renders obvious claim 3.

2. Dependent Claim 6

146. I have reproduced dependent claim 6 below:

> 6. The system according to claim **3**, wherein the instant voice message includes an identifier field including a unique identifier

associated with the instant voice message.

('622, Claim 6.)

147. I observe that claim 6, under its broadest reasonable construction, does not require a unique *message* identifier, *i.e.* a unique identifier *of the message itself*. The claim only requires "a unique identifier associated with the instant voice message," not an identifier that uniquely identifies *the message itself*.

148. Zydney discloses several examples of unique identifiers that meet the claim limitation. For example, Zydney explains: "Each message will have a unique identifier that will encode," among other things, "the sending software agent[']s identifier…" (Zydney, 34:4-8; *see also id.*, 23:1-12 (describing the contents of the voice container properties and data), Fig. 3 (same).) For example, the voice container stores an "originator's code 302 (which is a unique identifier)" (*id.*, 23:2-3), which is shown in Figure 3. Zydney thus discloses that the instant voice message (voice container) includes "an identifier field including a unique identifier associated with the instant voice message." Because the unique identifier of the originator (sender) software agent is encoded as part of the voice container itself, it is "**associated with**" the instant voice message, as claimed.

149. I am informed that in the pending litigation involving the Petitioners, the Patent Owner has asserted that this claim requires a field that *uniquely identifies*

the message itself. As I explained above, this is not required under the broadest reasonable construction of the term. But even if this was a requirement, it would have been obvious over Zydney. As explained in connection with claim 3**[e]** above, Zydney discloses that voice containers may be encoded using MIME, and expressly incorporates-by-reference the RFC 1521 that describes aspects of MIME. (Zydney, 19:6-12, 19:22-20:2 (incorporation of RFC 1521).) RFC 1521 discloses that a MIME message can include a "**Content-ID**" field that uniquely identifies the MIME message/entity. (**Ex. 1006/1106**, p.19, § 6.1.) RFC 1521 explains that the Content-ID field is similar to the "Message-ID" field (a well-known field used to uniquely identify messages in the context of email), and "must be generated to be world-unique." (*Id.*) RFC 1521 explains that the "Content-ID" field is an optional field whose "value may be used for uniquely identifying MIME entities [e.g. messages] in several contexts," such as where the actual body data is in another file that is not part of the message itself. (*Id.*) The "Content-ID" field in MIME thus discloses the claimed "identifier field." Because Zydney itself discloses that voice containers can be encoded using MIME and directly cites to RFC 1521, it would have been plainly obvious to a person of ordinary skill in the art to encode a standard "Content-ID" field into the voice container, thus disclosing a unique identifier that uniquely identifies the message itself.

3. Dependent Claim 7

150. I have reproduced dependent claim 7 below:

> 7. The system according to claim **3**, wherein the instant voice message includes a source field including a unique identifier associated with at least one of a given one of the plurality of instant voice message client systems that created the instant voice message and a given one of the plurality of users using the given one of the plurality of instant voice message client systems.

('622, Claim 7.)

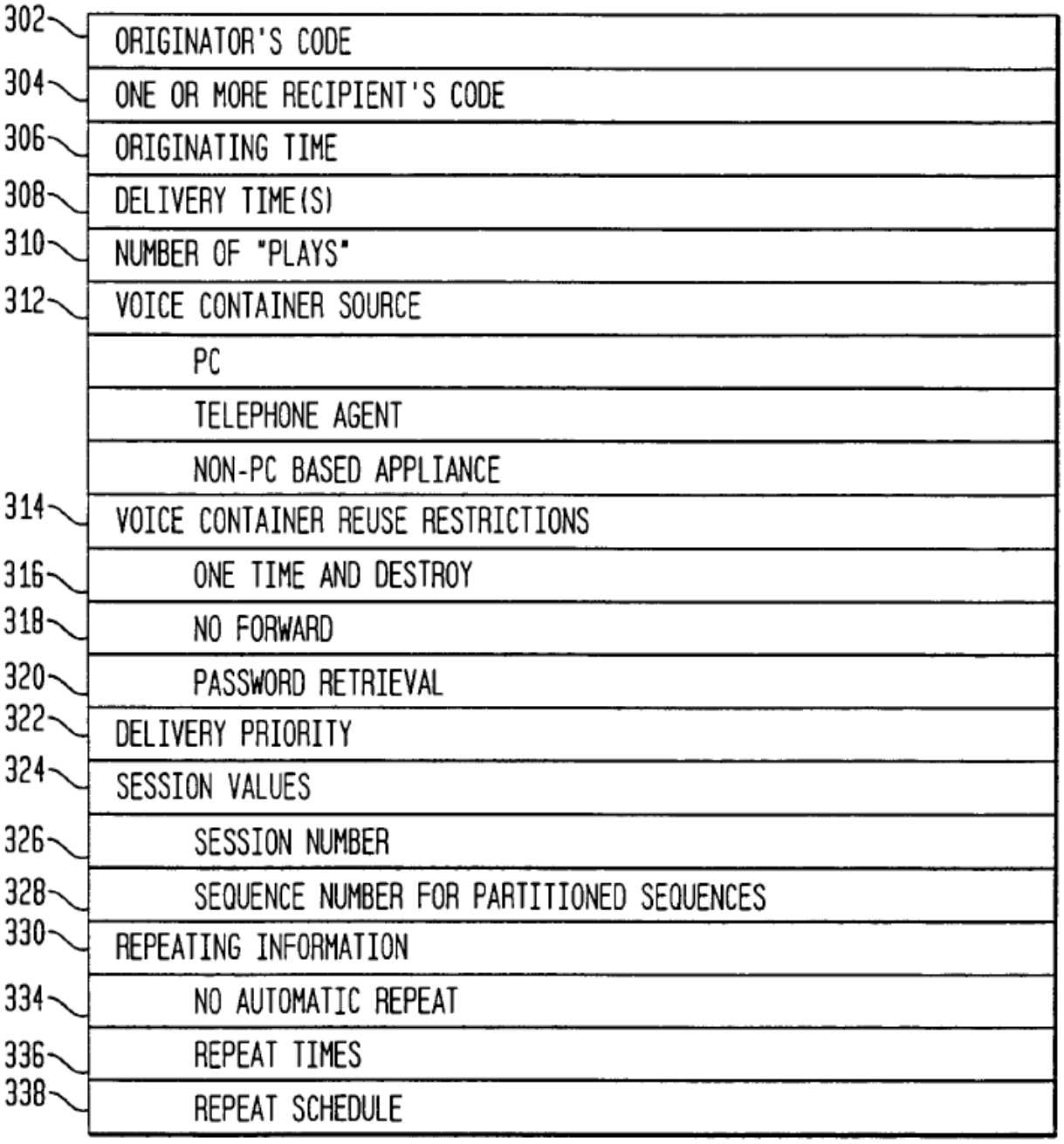
151. Zydney discloses the claimed "**source field**," which takes the form of an identifier that uniquely identifies the software agent of the originator (sender) of the message. Zydney refers to this unique identifier interchangeably as a "**unique address**" or a "**unique id**":

> The registration server assigns the software agent a unique address. This address is used for all communications from the software agent to the server, it [sic; its] components and between other software agents. The address assigned will be maintained in a data store. Each software agent may have multiple e-mail addresses, telephone numbers, name aliases, or other identifiers that may be associated with the unique id of the software agent.

(Zydney, 23:18-24:2.) Zydney also makes clear that this unique identifier can be carried in the voice container (the "instant voice message"). Zydney explains that

the "[v]oice container components include an originator's code 302 (which is a unique identifier)." (*Id.*, 23:1-3, 34:4-8.) The "originator's code" is the first of the fields of the voice container shown in Figure 3 below:

FIG. 3


302
ORIGINATOR'S CODE
304
ONE OR MORE RECIPIENT'S CODE
306
ORIGINATING TIME
308
DELIVERY TIME(S)
310
NUMBER OF "PLAYS"
312
VOICE CONTAINER SOURCE
PC
TELEPHONE AGENT
NON-PC BASED APPLIANCE
314
VOICE CONTAINER REUSE RESTRICTIONS
316
ONE TIME AND DESTROY
318
NO FORWARD
320
PASSWORD RETRIEVAL
322
DELIVERY PRIORITY
324
SESSION VALUES
326
SESSION NUMBER
328
SEQUENCE NUMBER FOR PARTITIONED SEQUENCES
330
REPEATING INFORMATION
334
NO AUTOMATIC REPEAT
336
REPEAT TIMES
338
REPEAT SCHEDULE

(*Id.*, Fig. 3.)

152. Zydney does not provide additional detail about the content of the "originator's code" (**302**) in Figure 3, but it would have been obvious to a person of ordinary skill in the art that the originator's code would have been the "unique id" (or "unique address") that the server in Zydney assigned to the sending (originating) software agent. (*Id.*, 23:18-24:2.) As noted, Zydney explains that "[t]he registration server assigns the software agent a unique address," and in the next sentence, explains that "this address is used for all communications from the software agent to the server…" (*Id.*, 23:18-19.) Zydney later states that "[e]ach message will have a unique identifier that will encode the sending software agent[']s identifier" (*id.*, 34:4-5), and the only field of the voice container in Figure 3 that meets that description is the originator code (**302**) field. Accordingly, to the extent there is any ambiguity as to whether the server-assigned unique id of the sender is used to populate the originator's code (**302**) of the voice container in Figure 3, it would have been obvious to do so because the server-assigned unique id would have served the purpose of the code as described in Zydney – providing a unique identifier for the software agent of the originator (sender) of the message. (*Id.*, 23:1-3 ("Voice container components include an originator's code 302 (which is a unique identifier)…").)

153. This claim goes on to recite that the "**source field**" is associated with "**at least one of**" two possible entities: (a) a client system "that created the instant voice message," or (b) a user "using the given one of the plurality of instant voice message client systems." For purposes of my analysis of Zydney, this is a distinction without a difference, because the sending (originating) software agent running on a client system is operated by the user of that system, and thus, Zydney meets both associations in the claim.

154. First, as noted above, Zydney discloses that the unique id is assigned to the software agent which, as noted above, runs on a client system such as a personal computer. (*Id.*, 23:18 ("The registration server assigns the software agent a unique address."), 11:16-18 ("More specifically, the agent of the present invention may be adapted to work on a personal computer, wireless handheld computer such a personal data assistant (PDA), digital telephone, or beeper.").) The "**source field**" is thus "**associated with at least one of a given one of the plurality of instant voice message client systems that created the instant voice message**," because it is associated with the originating (sending) client system.

155. Although not required by the claim, the "**source field**" in Zydney is also associated with "**a given one of the plurality of users using the given one of the plurality of instant voice message client systems**." This is because the unique

id assigned to the software agent may also be associated with a particular user. In particular, the unique id is associated with a particular "user name" and password that must be entered for authentication purposes: "The authentication server will permit or deny access to software agents based on the unique id of the software agent and a user name and password." (*Id*., 24:3-4.) "Multiple users cannot log in from the same machine with the same authentication password." (*Id*., 32:5-6.) Furthermore, the system stores additional personal identifying information in association with the unique identifier: "Each software agent may have multiple e-mail addresses, telephone numbers, name aliases, or other identifiers that may be associated with the unique id of the software agent." (*Id*., 23:21-24:2.) The unique identifier of the software agent is therefore associated with a given one of the plurality of users of the client systems who has a particular user name and password as well as one or more email addresses, telephone numbers, name aliases, and/or other information associated with the unique id.

156. I am informed that in pending litigation, the Patent Owner has proposed to construe "**source field**" as "**a block of data with a globally unique identifier of the sender**." This is not the broadest reasonable construction of the term because the claim language itself recites "a source field including a unique identifier," and does not require that the identifier be globally unique. Nevertheless, Zydney renders

this limitation obvious even under the Patent Owner's construction because, as noted above, Zydney explains that the system also maintains the Internet Protocol (IP) address of each software agent. (Zydney, 15:1-2, 15:13-14, 16:9-10, 32:9-15 (the central server will retain the Internet address of each newly logged-on software agent).) This is not surprising considering that, as **Shinder** explains, "[e]very computer, network attached printer, router, and other network device has a unique IP address." (Shinder, p.231.) It was well-known to persons of ordinary skill in the art that IP addresses of devices on the Internet qualified as "globally unique identifiers" in that no two devices on the Internet could have the same IP address.[14]

[14] Although IP addresses of devices on the Internet are globally unique, as I explained in the text, I do not mean to suggest that all IP addresses are globally unique. For example, organizations typically maintain local networks that assign "local" IP addresses to devices within the network. Those local addresses are unique within the local network, but because they are typically not used outside that network, they need not be unique across the entire Internet. In the case of Zydney, however, the client devices and the central server (as shown in Figure 1A) are connected to each other over the Internet, and as such, they have globally unique IP addresses for the reasons stated in the text.

An IP address thus discloses a globally unique identifier, as required by the Patent Owner's narrow (and incorrect) construction of "source field."

157. It would have been obvious to a person of ordinary skill in the art to adapt the system of Zydney in which the sending (originating) client sends the IP address with the voice container. Shinder confirms that the sending (originating) client device must have an IP address, and Zydney specifically discloses that the central server keeps track of those address, as noted previously. In fact, a person of ordinary skill in the art would have understood that the system of Zydney already sends the IP address of the sender with the voice container. Zydney explains that "[t]he voice container will be sent using standard TCP/IP transport." (Zydney, 23:11-12.) It was well-known to persons of ordinary skill in the art that when sending data using the standard TCP/IP transport, as specified in Zydney, that the IP address of the sender (as well as the IP address of the destination) are transmitted with the transmitted data. This is confirmed in Hunt **[Ex. 1010/1110]**, which describes TCP/IP in detail. (Hunt, pp.12-13.) Hunt describes the standard content of an IP datagram, which is "the basic unit of transmission in the Internet" and uses "the packet format defined by the Internet Protocol." (*Id*., p.12.) Hunt confirms that every IP datagram transmitted over the Internet using TCP/IP contains a header that includes the "Source Address" and the "Destination Address," which correspond to

the IP addresses of the sender and the recipient, respectively. (*Id*., p.13, Fig. 1-5.) Figure 1-5 of Hunt, reproduced below, shows the header including the Source Address and Destination address as well as the data carried in the datagram (shown in the figure as "data begins here . . .").

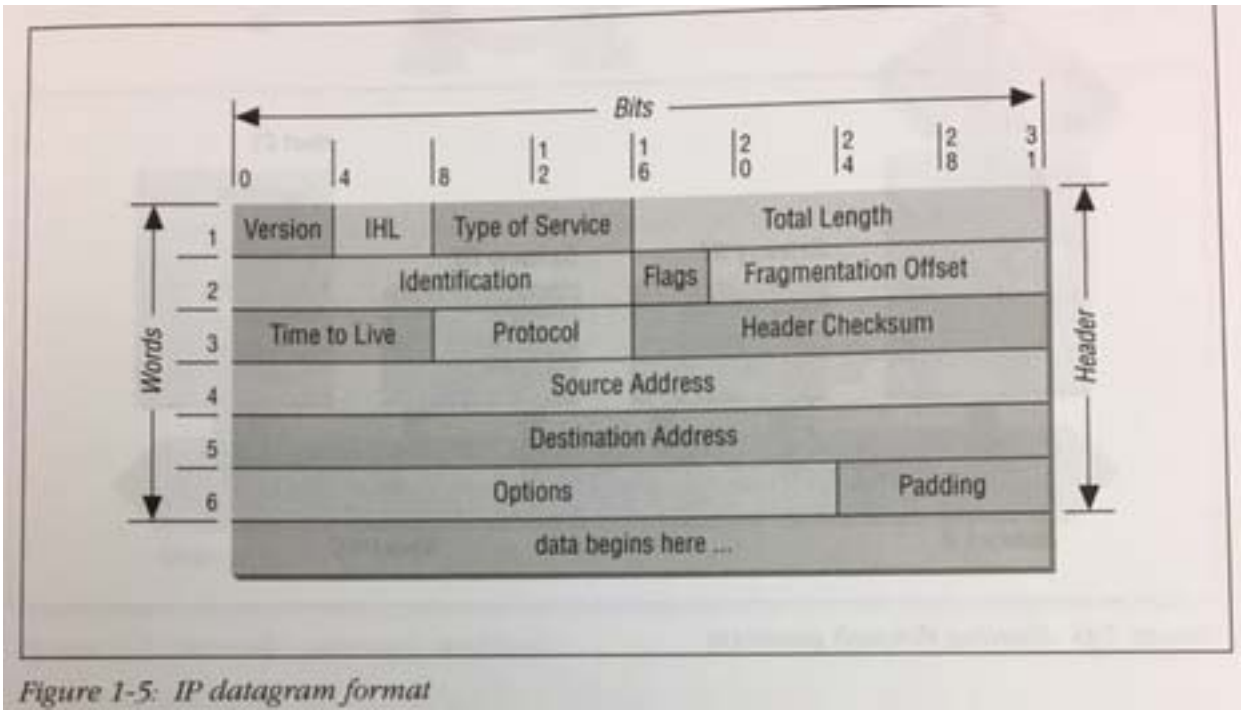


Figure 1-5: IP datagram format

(*Id*., p.13, Fig. 1-5.)

158. It would thus have been obvious to a person of ordinary skill in the art that when a voice container is sent by a sending (originating) client in Zydney, that transmission is accompanied with the IP address of the sender – a globally unique identifier – thus disclosing the claimed "source field." Under this alternative combination, the claimed "**instant voice message**" for purposes of claim 7 would include the voice container, along with the IP datagrams that accompany its

transmission pursuant to the TCP/IP protocol. Zydney thus renders claim 7 obvious even if the claimed "source field" requires a "globally" unique identifier.

4. Dependent Claim 8

159. I have reproduced dependent claim 8 below:

> **8.** The system according to claim **3**, wherein the instant voice message includes a destination field including a unique identifier associated with at least one of a given one of the plurality of instant voice message client systems identified as a recipient of the instant voice message and a given one of the plurality of users using the given one of the plurality of instant voice message client systems.

('622, Claim 8.)

160. This claim is similar to claim 7 above, the main difference being that claim 8 recites a "**destination field**" (instead of a "source field"), and thus, requires that it be associated with a recipient client system or user.

161. As I explained for claim 7 above, Zydney discloses that the server assigns a "unique id" (also called a "unique address") to each software agent. (Zydney, 23:18-24:2.) Zydney thus discloses a unique id for the recipient software agent for the same reasons as the originating (sending) software agent discussed in claim 7. Figure 3 of Zydney discussed above also confirms that the voice container

includes "one or more recipient's code" (**304**). (*Id.*, 23:1-4, 34:4-8, Fig. 3.) Zydney thus discloses the claimed destination field.

162. For the same reasons as claim 7 above, Zydney discloses that the unique id is assigned to the software agent which, as noted above, runs on a client system such as a personal computer. (*Id.*, 23:18, 11:16-18.) The "**destination field**" in Zydney is thus "**associated with at least one of a given one of the plurality of instant voice message client systems identified as a recipient of the instant voice message**," because it is associated with the recipient client system.

163. Although not required by the claim, the "**destination field**" in Zydney is also associated with "**a given one of the plurality of users using the given one of the plurality of instant voice message client systems**." This is because, as explained fully for claim 7, the unique id assigned to the software agent may also be associated with a particular user.

164. Finally, I am informed that in pending litigation, the Patent Owner has proposed to construe "**destination field**" as requiring a "globally unique" identifier, as it has with "source field." As I explained in my discussion of claim 7 above, this is not the broadest reasonable construction of the term because the claim itself does not recite any requirement that the identifier be globally unique. Nevertheless, it would have been obvious to provide a destination field in the form of "a globally

unique identifier" (*e.g.* an IP address of the recipient client) for the same reasons articulated for claim 7. As I explained for claim 7, it would have been obvious to a person of ordinary skill in the art that when a voice container is sent by a sending (originating) client in Zydney, that transmission is accompanied with the IP address of the recipient – a globally unique identifier – thus disclosing the claimed "destination field." Under this alternative combination, the claimed "**instant voice message**" for purposes of claim 8 would include the voice container, along with the IP datagrams that accompany its transmission pursuant to the TCP/IP protocol. Zydney thus renders claim 8 obvious even if the claimed "destination field" requires a "globally" unique identifier.

5. Dependent Claim 10

165. I have reproduced dependent claim 10 below:

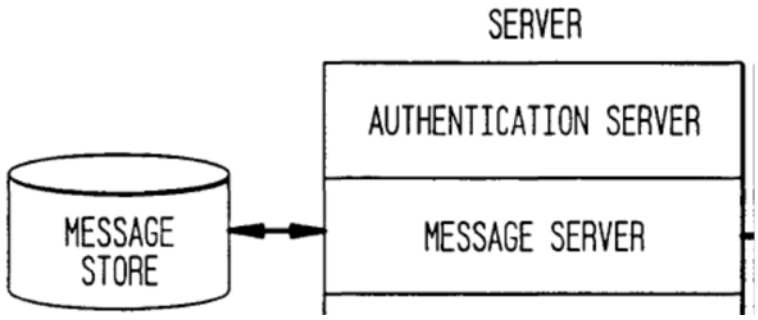
> 10. The system according to claim **3**, further comprising: a message database storing the instant voice messages received from the instant voice message client systems.

('622, Claim 10.)

166. As I explained for claims 3**[b]** and 3**[d]**, the central server in Zydney receives instant voice messages (voice containers) from the client systems. The central server also contains a **message database storing the instant voice messages received from the instant voice message client systems**. As Zydney explains, the

instant voice messages (voice containers) sent by client systems to unavailable recipients are stored in a "message store" in the central server that discloses the claimed **message database**. (Zydney, Fig. 3 ("message store"); *see also id.*, 25:1-4 (describing "repository" of voice containers at the central server), Fig. 8, step 1.2.5. (describing "voice containers in the central storage"), Fig. 4 ("if recipient is not online, client sends voice container to server file").)

167. Figure 2 shows the message store with the conventional notation of a cylinder that a person of ordinary skill in the art would have understood to denote a **database**.



(*Id*., Fig. 2 (excerpt).

168. I note, for purposes of my analysis of claim 10, that Zydney describes that at least the instant voice messages (voice containers) for unavailable recipients are stored at the central server. Therefore, the limitations of claim 10 are disclosed by Zydney's teachings that "the instant voice messages received from the instant

voice message client systems" for recipients who are unavailable are stored in the message database at the central server.

6. Dependent Claim 11

169. I have reproduced dependent claim 11 below:

> 11. The system according to claim **3**, wherein, upon receipt of an instant voice message, the communication platform system determines if there is the current connection to one of the plurality of instant voice message client systems identified as a recipient of the instant voice message, and if there is no connection with the one of the plurality of instant voice message client system identified as the recipient, the instant voice message is stored and delivered when the one of the plurality of instant voice message client systems identified as the recipient re-established a connection.

('622, Claim 11.) For clarity, I address this limitation in two parts.

(a) "wherein, upon receipt of an instant voice message, the communication platform system determines if there is the current connection to one of the plurality of instant voice message client systems identified as a recipient of the instant voice message,"

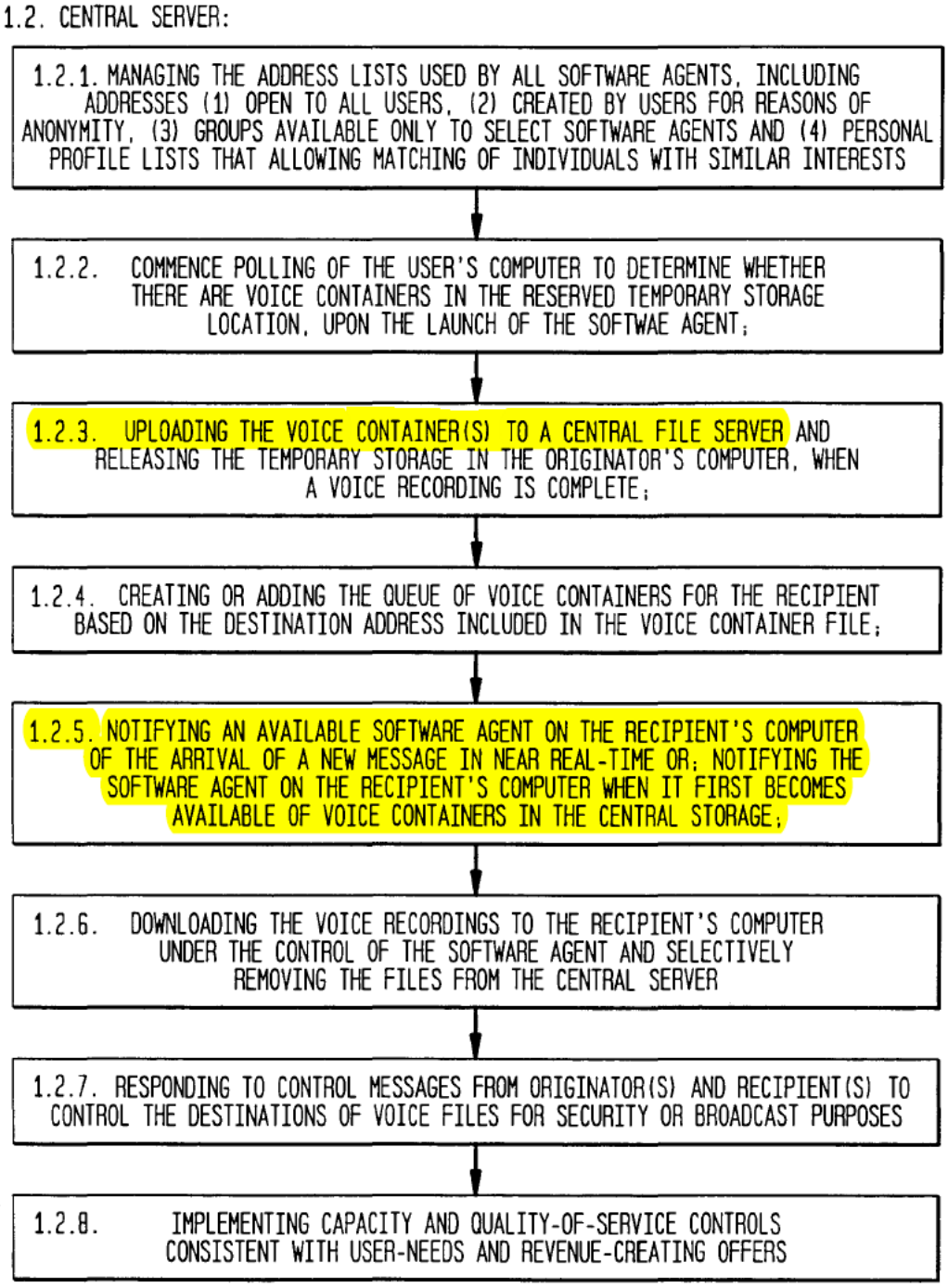
170. As I explained for claim 3**[c]**, the system of the central server in Zydney that tracks and maintains the connection information discloses the claimed "**communication platform system**."

171. As I explained previously, Zydney discloses that the central server "track[s]," *i.e.*, **determines**, "the status of all software agents." (Zydney, 14:6-9, 13:12-14.) The statuses tracked by the central server include "the core states of whether the recipient is online or offline." (*Id.*, 14:17-15:1.) Zydney therefore discloses that the central server "**determines if there is a current connection to one of the plurality of instant voice message client systems**."

172. Zydney also specifically discloses that this determination of connectivity is performed **upon receipt of an instant voice message** (voice container) by the central server. The central server performs this determination in order to determine whether to deliver the voice container to any available intended recipient(s) or maintain it in storage for one or more unavailable recipients. For example, Figure 8 of Zydney "shows a flow chart of an exemplary embodiment of the method and system for voice exchange and voice distribution with respect to the central server." (Zydney, 34:20-22.) Zydney teaches that an originator records a voice container and transmits it upon completion of the recording, so that the originator's system "upload[s] the voice container(s) to a central file server . . . when a voice recording is complete." (*Id.*, Fig. 8 (step 1.2.3.).) Upon receipt of the uploaded voice container, the central server will either "notify[] an available software agent on the recipient's computer of the arrival of a new message in near

real-time **or**; notify[] the software agent on the recipient's computer when it first becomes available of voice containers in the central storage." (*Id.*, Fig. 8 (step 1.2.5.).) Figure 8 is reproduced below.

FIG. 8


1.2. CENTRAL SERVER:
1.2.1. MANAGING THE ADDRESS LISTS USED BY ALL SOFTWARE AGENTS, INCLUDING ADDRESSES (1) OPEN TO ALL USERS, (2) CREATED BY USERS FOR REASONS OF ANONYMITY, (3) GROUPS AVAILABLE ONLY TO SELECT SOFTWARE AGENTS AND (4) PERSONAL PROFILE LISTS THAT ALLOWING MATCHING OF INDIVIDUALS WITH SIMILAR INTERESTS
1.2.2. COMMENCE POLLING OF THE USER'S COMPUTER TO DETERMINE WHETHER THERE ARE VOICE CONTAINERS IN THE RESERVED TEMPORARY STORAGE LOCATION, UPON THE LAUNCH OF THE SOFTWAE AGENT;
1.2.3. UPLOADING THE VOICE CONTAINER(S) TO A CENTRAL FILE SERVER AND RELEASING THE TEMPORARY STORAGE IN THE ORIGINATOR'S COMPUTER, WHEN A VOICE RECORDING IS COMPLETE;
1.2.4. CREATING OR ADDING THE QUEUE OF VOICE CONTAINERS FOR THE RECIPIENT BASED ON THE DESTINATION ADDRESS INCLUDED IN THE VOICE CONTAINER FILE;
1.2.5. NOTIFYING AN AVAILABLE SOFTWARE AGENT ON THE RECIPIENT'S COMPUTER OF THE ARRIVAL OF A NEW MESSAGE IN NEAR REAL-TIME OR; NOTIFYING THE SOFTWARE AGENT ON THE RECIPIENT'S COMPUTER WHEN IT FIRST BECOMES AVAILABLE OF VOICE CONTAINERS IN THE CENTRAL STORAGE;
1.2.6. DOWNLOADING THE VOICE RECORDINGS TO THE RECIPIENT'S COMPUTER UNDER THE CONTROL OF THE SOFTWARE AGENT AND SELECTIVELY REMOVING THE FILES FROM THE CENTRAL SERVER
1.2.7. RESPONDING TO CONTROL MESSAGES FROM ORIGINATOR(S) AND RECIPIENT(S) TO CONTROL THE DESTINATIONS OF VOICE FILES FOR SECURITY OR BROADCAST PURPOSES
1.2.8. IMPLEMENTING CAPACITY AND QUALITY-OF-SERVICE CONTROLS CONSISTENT WITH USER-NEEDS AND REVENUE-CREATING OFFERS

(*Id.*, Fig. 8.)

173. A person of ordinary skill in the art would have understood that step 1.2.5. shown in Figure 8 above makes a decision, based on whether or not the recipient is available, to either notify the recipient of the message (a) "in near-real-time" if the recipient is online, or if the recipient is not, (b) "when it first becomes available…" (*Id.*; *see also id.*, 14:9-11 ("It [the central server] will notify the software agent to send the voice container directly to the recipient if the recipient is available or it will store the voice container for the intended recipient if the recipient is not available.").) The central system in Zydney therefore makes a determination, upon receipt of an instant voice message, if the recipient is currently connected.

> **(b) "and if there is no connection with the one of the plurality of instant voice message client system identified as the recipient, the instant voice message is stored and delivered when the one of the plurality of instant voice message client systems identified as the recipient re-established a connection."**

174. Zydney describes that if a recipient is currently logged off (i.e., there is no current connection with the recipient client system), the central server will store the voice container and deliver it when the recipient re-establishes a connection. As Zydney explains, the message server in the central server "will be the repository for messages sent to software agents that are not logged onto the system. Once a software agent has been authenticated all messages that have been stored on the message server will be sent to the appropriate software agent." (Zydney, 25:1-4.)

This discloses that **if there is no connection with the one of the plurality of instant voice message client system identified as the recipient, the instant voice message is stored and delivered when the one of the plurality of instant voice message client systems identified as the recipient re-established a connection**.

175. This feature is shown by Figure 8, which I discussed and reproduced above. Figure 8 describes the central server "notifying the software agent on the recipient's computer when it first becomes available of voice containers in the central storage" (step 1.2.5.) followed by "downloading the voice recordings to the recipient's computer" (step 1.2.6.), which discloses that the voice containers (instant voice messages) are **stored** at the central server and **delivered** when the recipient reconnects. (*Id*., Fig. 8.)

7. Dependent Claim 13

176. I have reproduced dependent claim 13 below:

> 13. The system according to claim **3**, wherein each of the instant voice message client systems comprises an instant voice messaging application generating an instant voice message and transmitting the instant voice message over the packet-switched network to the messaging system.

('622, Claim 13.) For clarity, I address this limitation in two parts.

(a) "wherein each of the instant voice message client systems comprises an instant voice messaging application…"

177. As I explained in **Part V.C.1** above, the broadest reasonable construction of "instant voice messaging application" in the '622 patent claims at issue is **hardware and/or software used for instant voice messaging**.

178. As I explained for claim 3**[c]** above, Zydney discloses an instant voice message client system in the form of a personal computer or other device containing a software agent for instant voice messaging. The "**instant voice messaging application**" in Zydney is disclosed by the software (including a software agent) running on the computing device or hardware. (Zydney, 13:2-6 & 14:2-12 (describing "Software Agent" utilized by the sender); *id.*, 13:19-22 & 14:14-16 (describing "Software Agent" utilized by the recipient).)

179. The instant voice messaging application in Zydney, moreover, is "**used for instant voice messaging**," as required by my proposed broadest reasonable construction. As I explained previously for claim 3, Zydney explains that the software agent on the *sending* (originating) client system can generate and transmit **instant voice messages** in the form of voice containers. (*Id.*, 13:2-6 ("A Software Agent utilized by the sender of the voice container provides the following functionality … address the recipient(s) and pack message into a voice container or

multiple voice containers 50; and, enable transport 52 of the voice container to the recipient or the central server.").) Conversely, the software agent on the *receiving* client system can receive and present received voice containers. (*Id.*, 13:19-22 ("A Software Agent utilized by the recipient provides the following functionality: log on to the central server; authenticate to the central server; retrieve any undelivered voice containers; and, unpack the voice container and play the message.").)

(b) "…[an instant voice messaging application] generating an instant voice message and transmitting the instant voice message over the packet-switched network to the messaging system"

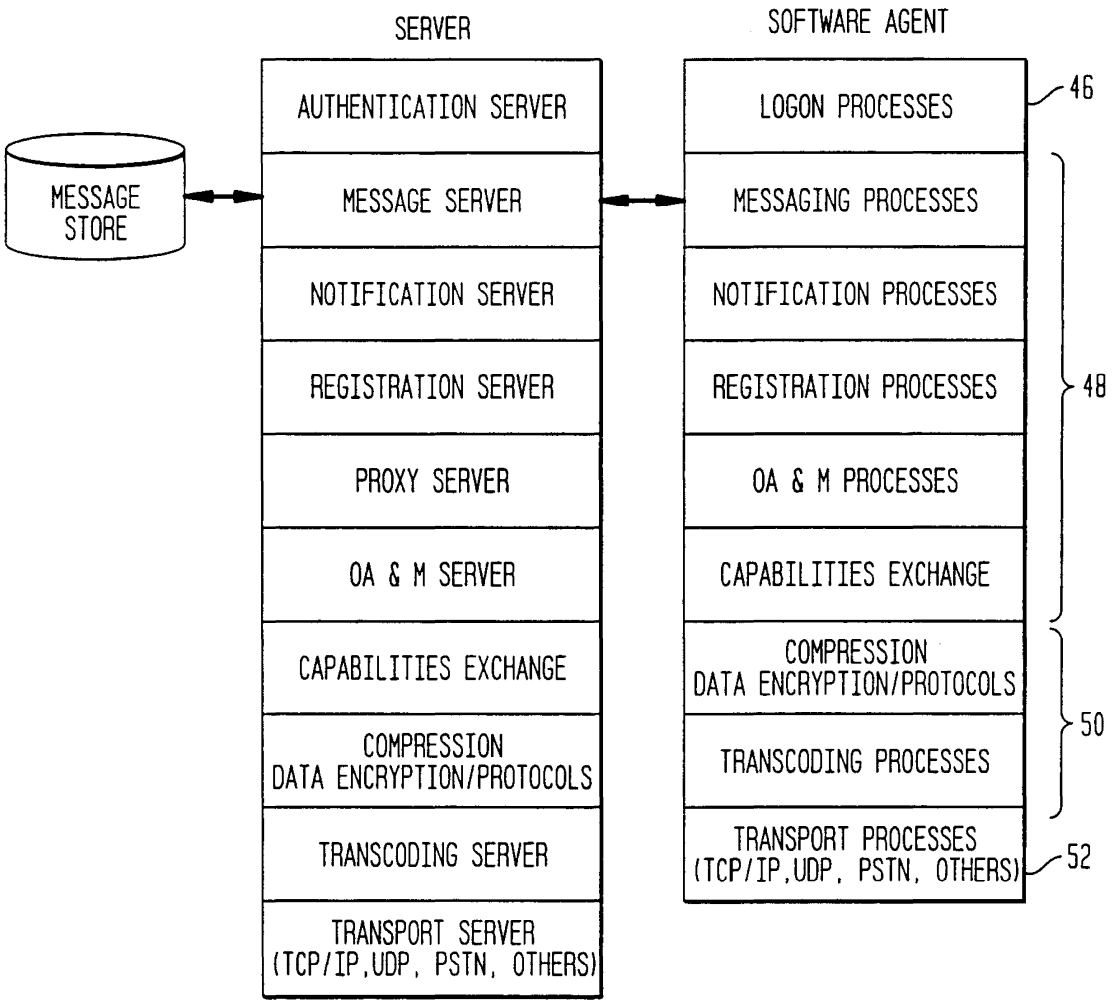
180. The "instant voice messaging application" in this limitation takes the form of the software agent running on the client device of the sending (originating) user. Zydney discloses that the software agent on the sending client system in Zydney **generates the instant voice message** (voice container): "To create a message, the software agent will address, pack and send the message in a voice container." (Zydney, 14:2-5; *id.*, 13:2-5 ("A Software Agent utilized by the sender of the voice container provides the following functionality: … address the recipient(s) and pack message into a voice container or multiple voice containers 50…").) More specifically, in order to generate a voice container, the software agent records the voice of the sender (originator) and packages it into a voice container:

Once the delivery mode has been selected, the originator digitally

> records messages for one or more recipients using a microphone-equipped device and the software agent. The software agent compresses the voice and stores the file temporarily on the PC if the voice will be delivered as an entire message.

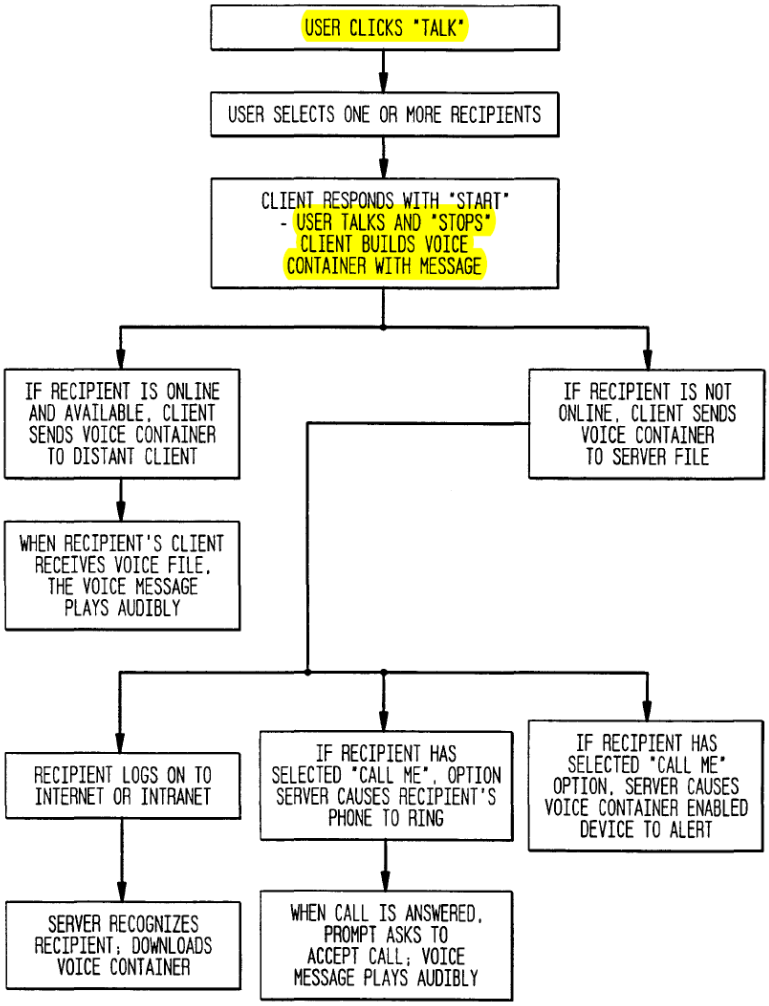
(*Id.*, 16:1-4; *see also id.*, 13:1-6 (Software Agent can "address the recipient(s) and pack message into a voice container or multiple voice containers 50"), 20:11-14, 25:10-13, 33:4-6, Claims 1, 8.) Figure 2, reproduced below, shows the Software Agent and its various components including the compression data encryption/protocols and transcoding process components (items 50). (*Id.*, Fig. 2.) The software agent's functionality is also supported by hardware including the client device (e.g., personal computer) and the microphone (which is used to record the sending (originating) user's voice). (*Id.*, 16:1-4, Figs. 4, 6, 7.)

FIG. 2



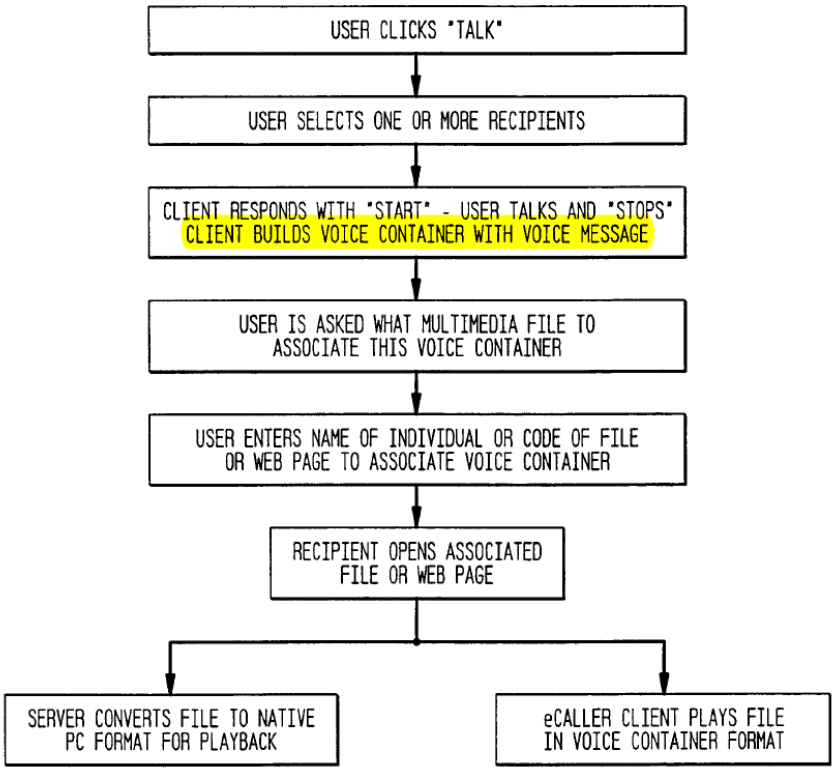
(*Id.*, Fig. 2.) Figures 4-7 and 16 of Zydney also disclose the client generating an instant voice message using the software agent. Figure 4, reproduced below, describes that the "client builds voice container with message." (*Id.*, Fig. 4, 2:20-21, 34:13-15.)

FIG. 4



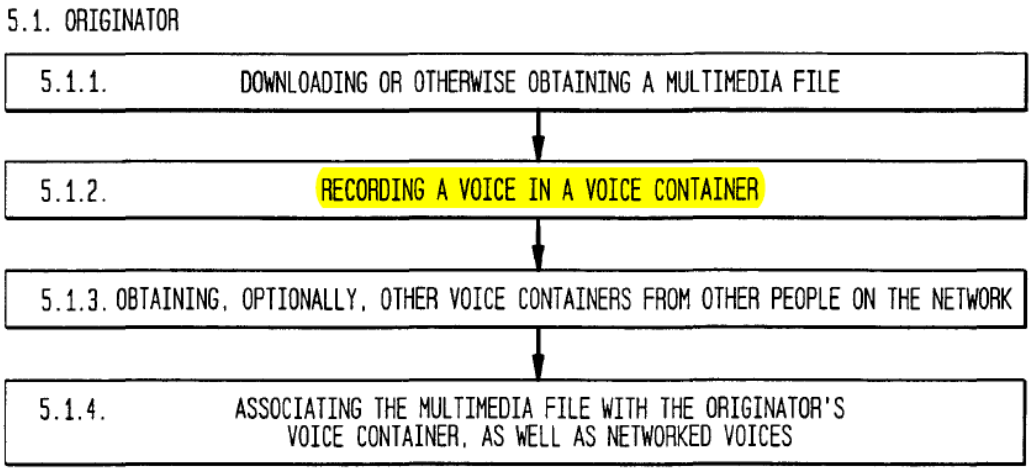
(*Id.*, Fig. 4.) Figures 6 and 16, reproduced below, further show how the client generates the voice container (and may also attach one or more files, as I explain below for claim 27**[d]**).

FIG. 6



(*Id.*, Fig. 6.)

FIG. 16



(*Id.*, Fig. 16.)

181. Zydney also discloses that the "**instant voice messaging application… transmit[s] the instant voice message over the packet-switched network to the messaging system**." (Zydney, 13:2-6 ("A Software Agent utilized by the sender of the voice container provides the following functionality: … address the recipient(s) and pack message into a voice container or multiple voice containers 50; and, enable transport 52 of the voice container to the recipient or the central server.").) More specifically, as I discussed previously, Zydney describes that the software agent creates and transmits instant voice messages using "voice containers." (*Id.*, 12:1-8, 10:20-11:3.) The instant voice messaging application messaging system in Zydney includes the portions of the software agent of the sending (originating) client responsible for transmitting the voice container. These portions are shown as "transport processes" 52 in Figure 2. (*Id.*, Fig. 2 ("Transport Processes" 52).) As explained in Zydney: "A Software Agent utilized by the sender of the voice container provides the following functionality: log on to a central server 46; authenticate to the central server 48; address the recipient(s) and pack message into a voice container or multiple voice containers 50; and, enable transport 52 of the voice container to the recipient or the central server." (*Id.*, 13:1-6.) The functionality on the client system for transmitting the voice container to the central server, particularly including the transport processes component of the software

agent, discloses the claimed "**instant voice messaging application… transmitting the instant voice message… to the messaging system**."

182. Finally, Zydney discloses that this transmission occurs "**over a packet-switched network**" for the same reasons I explained above with respect to claim 3**[a]**, as Zydney describes transmitting instant voice messages over the Internet.

8. Dependent Claim 18

183. I have reproduced dependent claim 18 below:

> 18. The system according to claim **13**, wherein the instant voice messaging application includes an audio file creation system creating an audio file for the instant voice message based on input received via an audio input device coupled to the client device.

('622, Claim 18.) For purposes of claim construction, I note that the '622 patent does not define "audio file creation system," but in this case, the meaning under the broadest reasonable construction would be reasonably clear to a person of ordinary skill in the art. The specification refers to audio file creation **312** as a component in the general-purpose programmable computer IVM client **208** and states that "[a]udio file creation **312** creates an instant voice message as audio file **210**, and is responsible for receiving input speech for the instant voice message from audio input device **212** or via network **204** and storing the input speech into audio file **210**."

('622, 12:40-44; *id.*, 12:17-21, 12:38-40.) Zydney discloses substantially similar functionality.

184. The "**audio file creation system**" in Zydney corresponds to the portion of the software agent functionality that creates the audio file. Zydney explains that the software agent creates an audio file for the instant voice message by taking input recorded from a microphone **coupled to the client device**, and storing it into a "voice" file. As Zydney explains: "the originator digitally records messages for one or more recipients using a microphone-equipped device and the software agent. The software agent compresses the voice and stores the file temporarily on the PC if the voice will be delivered as an entire message." (Zydney, 16:1-4, 21:14-16, 20:11-14, Fig. 7 ("1.1.3. recording a voice through a microphone connected to the personal computer in the voice of the originator").)

185. Zydney also further discloses that the voice files can be generated in various formats that disclose various types of audio files, such as MP3 files. (*Id.*, 12:1-13; *see also id.,* 25:10-13 ("In one embodiment, GSM is used as the default codec for the system. Other codecs, such as G.723 and G.729 are also supported."); Claims 17, 19).)

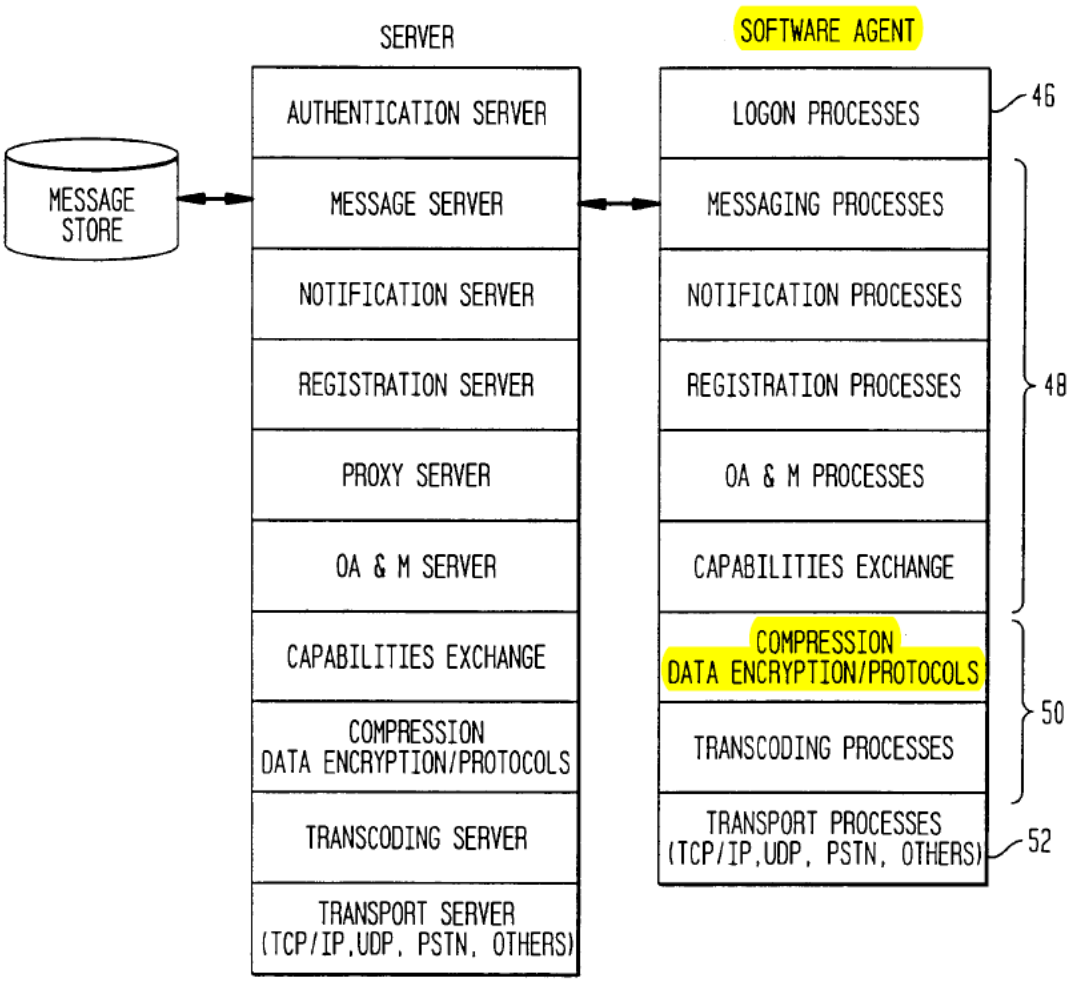
9. Dependent Claim 19

186. I have reproduced dependent claim 19 below:

19. The system according to claim **13**, wherein the instant voice messaging application includes an encryption/decryption system for encrypting the instant voice messages to be transmitted over the packet-switched network and decrypting the instant voices [sic] messages received over the packet-switched network.

('622, Claim 19.) Zydney's Figure 2 depicts the set of "encryption protocols" in the software agent, which discloses an **encryption/decryption system** in the claimed instant voice messaging application that **encrypts the instant voice message** to be transmitted and decrypts instant voice messages received over the Internet.

FIG. 2



(Zydney, Fig. 2.)

187. Although the label in Figure 2 only refers to "encryption," Zydney makes clear that those processes can perform encryption and decryption. In particular, Zydney discloses that encryption is part of the "standard codec" used in transmitting voice containers: "Each software agent that has been loaded and registered with the system will in addition to the standard codec used for the

encryption and decryption of the voice containers detail the other codecs that the software agent may have access to on the system." (Zydney, 27:1-6.) It would have been apparent and obvious to a person of ordinary skill in the art that "encryption" mentioned in Zydney is used for voice containers being *transmitted*, and that "decryption" is applied to voice containers that are *received*. This because the purpose of encryption is to protect the privacy or secrecy of information being transmitted, and thus, the "encryption" would be applied by the sending (originating) software agent, and the "decryption" by the receiving software agent. It would also make no sense to perform "decryption" on the sending (originating) client device as decryption works only on encrypted data. The fact that the Zydney system uses a "standard" codec for encryption and decryption of the voice containers further discloses encrypting an instant voice message (voice container) when it is transmitted and decrypting an instant voice message when it is received.

10. Dependent Claim 20

188. I have reproduced dependent claim 20 below:

> 20. The system according to claim **13**, wherein the instant voice messaging application includes a compression/decompression system for compressing the instant voice messages to be transmitted over the packet-switched network and decompressing the instant voice messages received over the packet-switched network.

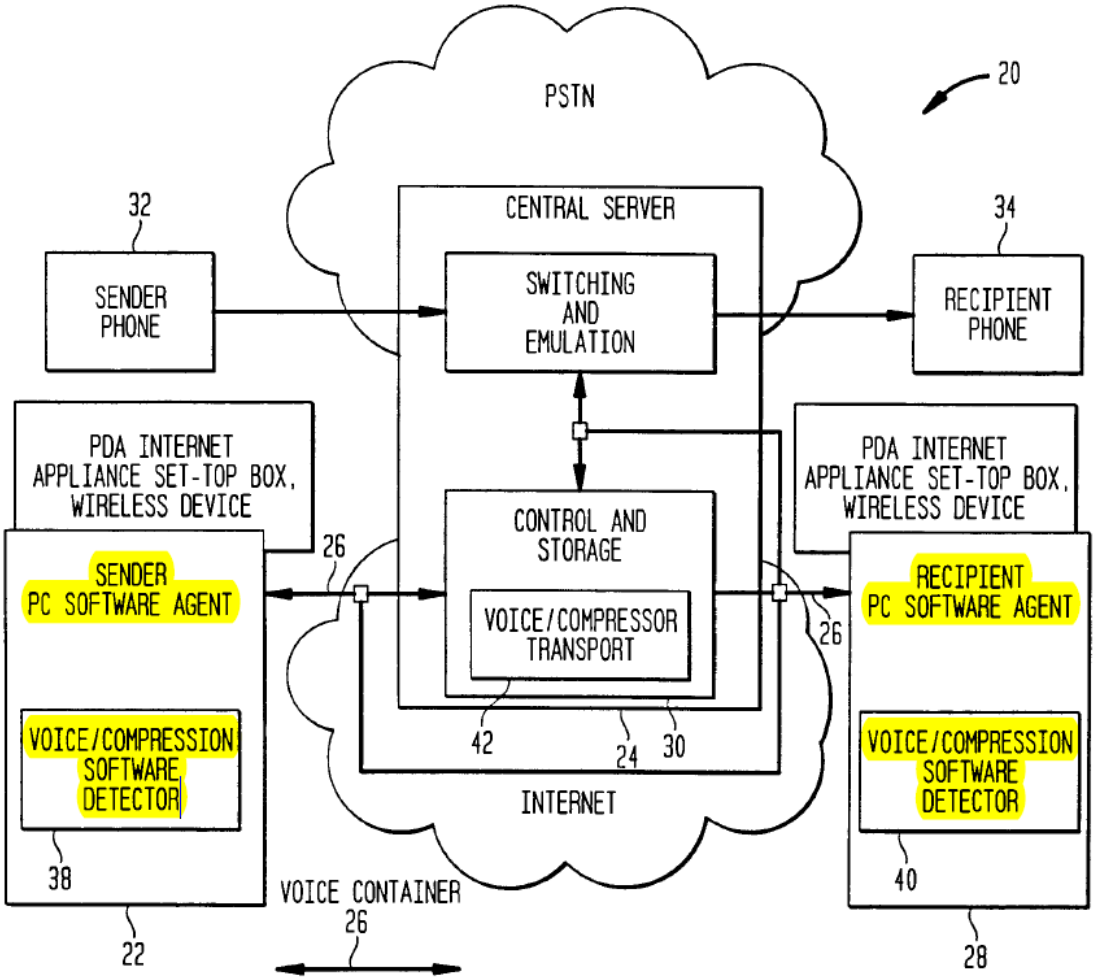
('622, Claim 20.)

189. Zydney discloses that its software agents contain software to compress and decompress instant voice messages:

> With reference to FIG. lA, the present invention is designed to adapt to the voice and data compression capabilities of the user's existing hardware and software platform.… In each case different voice and compression applications and data formats may be available as dictated by the hardware platform and software residing thereon. The present invention includes a voice/compression software detector 38 and 40 that communicates the format of the voice data to be transmitted and/or received.

(Zydney, 11:14-22.) These compression software detectors are shown in Figure 1A, reproduced below, as items 38 and 40 contained within software agents 22 and 28.

FIG. 1A

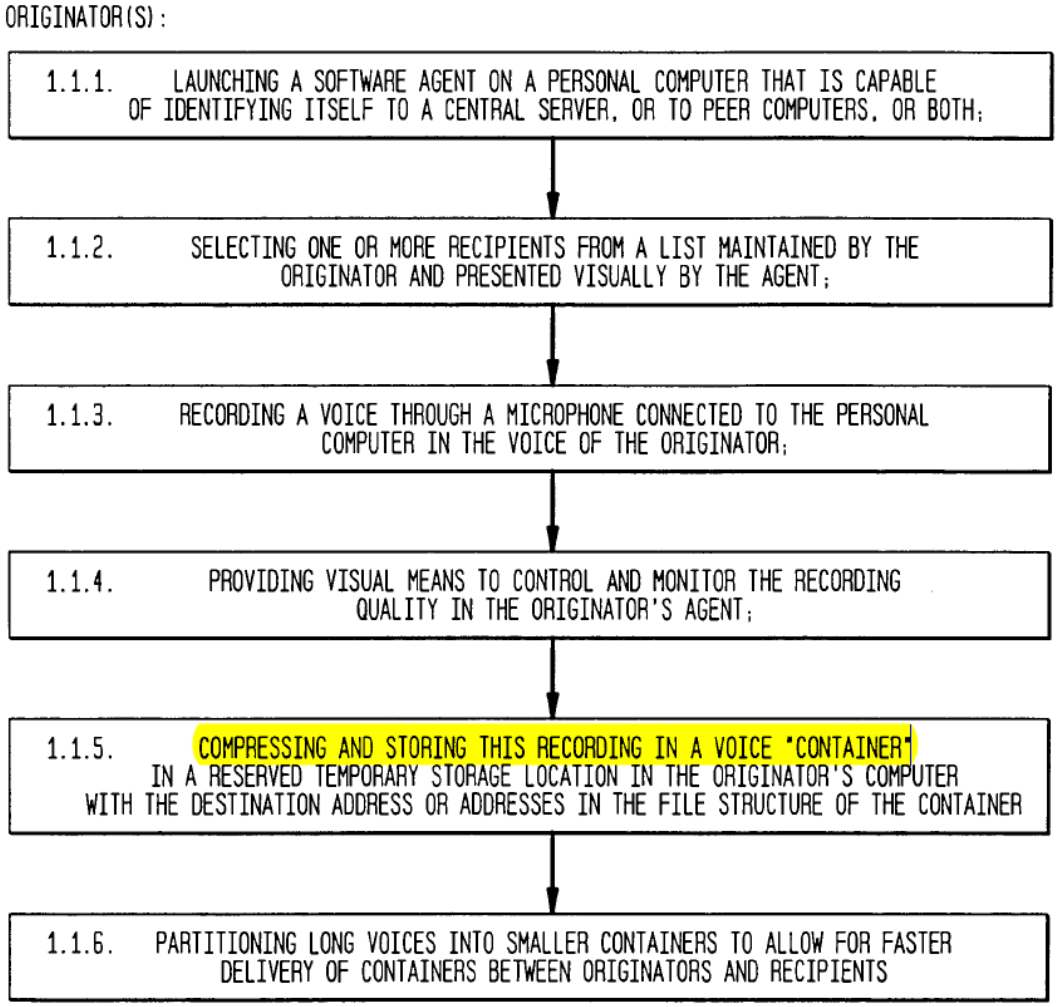


(*Id.*, Fig. 1A.)

190. Zydney further discloses that software agents compress instant voice messages as they are generated. Zydney describes a "pack and send mode of operation" as "one in which the message is first acquired, compressed and then stored in a voice container 26 which is then sent to its destination(s)." (Zydney,

11:1-3, 16:3-4.) Figure 7, reproduced below, depicts "compressing and storing this recording in a voice 'container'." (*Id.*, Fig. 7.)

FIG. 7

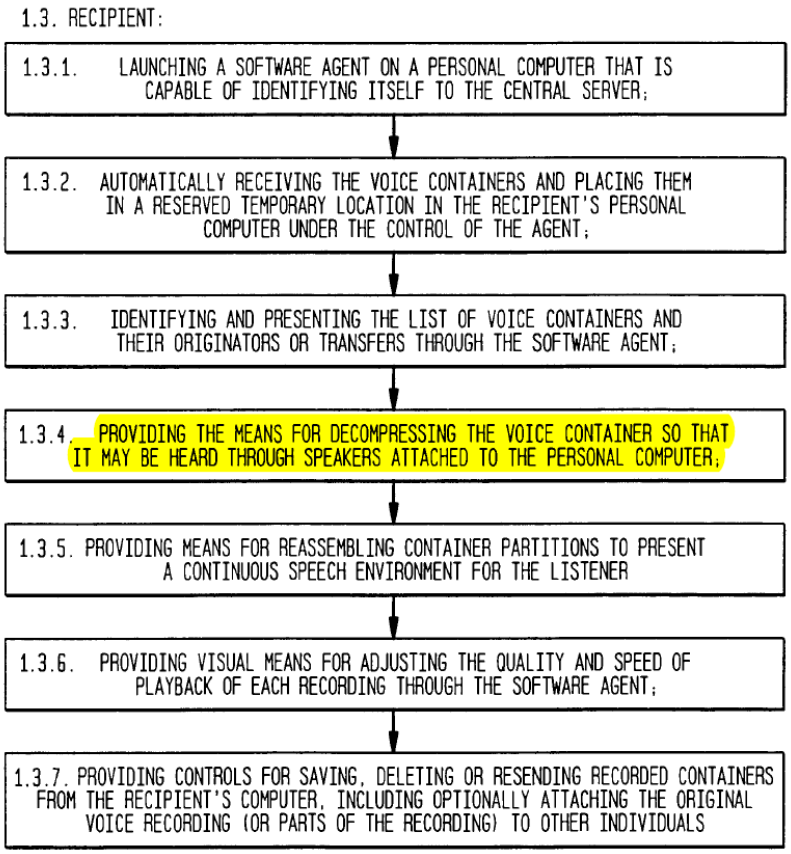


(*Id.*, Fig. 7.)

191. Similarly, Zydney discloses that when a software agent receives a compressed voice file, it will decompress the file so that it can be played: "The voice is uncompressed and the recipient can hear the recording through the speakers or

headset attached to their computer." (*Id.*, 16:10-14.) This is depicted in Figure 9, reproduced below.

FIG. 9



(*Id.*, Fig. 9.)

192. The software on the client system (including detector items 38 and 40) that compresses the voice container to be transmitted and decompresses a received voice container discloses the claimed **compression/decompression system**.

11. Dependent Claim 21

193. I have reproduced dependent claim 21 below:

> 21. The system according to claim **13**, wherein the instant voice messaging application displays a list of one or more potential recipients for the instant voice message.

('622, Claim 21.)

194. Zydney discloses that the software agent on the client system visually presents a list of potential recipients: the originator "select[s] one or more recipients from a list maintained by the originator and presented visually by the agent." (Zydney, Fig. 7 (Step 1.1.2.), 14:18-19.)

12. Dependent Claim 23

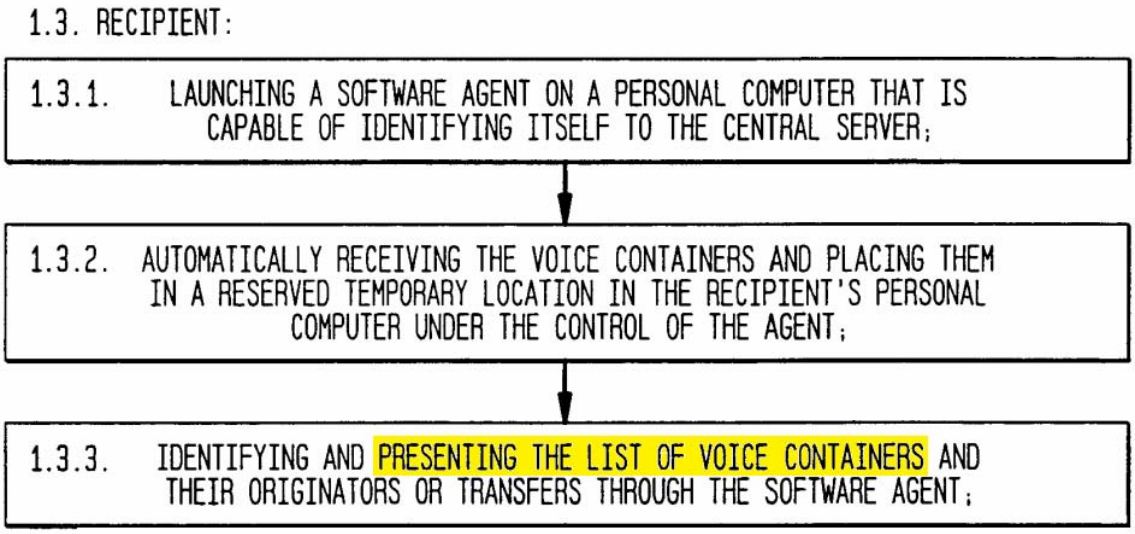
195. I have reproduced dependent claim 23 below:

> 23. The system according to claim **13**, wherein the instant voice message application generates an audible or visual effect indicating receipt of an instant voice message.

('622, Claim 23.)

196. Zydney discloses a visual effect indicating receipt of an instant message. Step 1.3.2. in Figure 9 discloses the steps of a "software agent" being launched, then "automatically receiving the voice containers," which is followed in Step 1.3.3. by "presenting the list of voice containers." (Zydney, Fig. 9 (step 1.3.3.).)

FIG. 9



(Zydney, Fig. 9 (partial figure shown).)

197. The "list of voice containers" presented in Step 1.3.3. discloses the software agent on the client system providing the claimed "**visual effect**" because it indicates to the recipient that an instant voice message has been received.

13. Independent Claim 27

198. I have reproduced independent claim 27 below using bracketed notations (e.g. "**[a]**," "**[b]**," etc.):

27. A system comprising:

[a] a client device; and

[b] a network interface coupled to the client device and connecting the client device to a packet-switched network; and

[c] an instant voice messaging application installed on the client

device, wherein the instant voice messaging application includes

[c1] a client platform system for generating an instant voice message and

[c2] a messaging system for transmitting the instant voice message over the packet-switched network via the network interface,

[d] wherein the instant voice messaging application includes a document handler system for attaching one or more files to the instant voice message.

('622, Claim 27.) Each limitation of claim 27 is disclosed and rendered obvious by Zydney.

(a) Preamble of claim 27: "A system comprising:"

199. To the extent the preamble is limiting, Zydney discloses "[a] system" with the features discussed in my analysis of the limitations of claim 27 below.

(b) "a client device" (Claim 27[a])

200. Zydney discloses **a client device** in the form of a personal computer or other device that contains a software agent and can send and receive instant voice messages, for the same reasons I discussed previously for claim 3**[b]** regarding a client system. (Zydney, 11:16-18, 14:2-3, Figs. 4, 6.)

(c) "a network interface coupled to the client device and connecting the client device to a packet-switched network; and" (Claim 27[b])

201. As I explained with respect to claim 3**[a]**, Zydney discloses a client system and a server connected over the Internet, *i.e.*, a packet-switched network. I also explained for claim 3**[a]** that Zydney, alone or in combination with Shinder, discloses that the central sever in Zydney includes a "**network interface**" connecting the server to the Internet.

202. The same rationale I provided for claim 3**[a]** applies here. A person of ordinary skill in the art would have appreciated that, in order for the client in Zydney to connect to the Internet, it would have needed a network interface (such as a network interface controller (NIC)). As explained for claim 3**[a]**, Shinder itself confirms that "[s]ome sort of network interface is always required to communicate over a network." (Shinder, pp.195, 196.) This requirement of a network interface applies equally to the *client* in Zydney as it does to the *server* in Zydney. The claimed "network interface" in claim 27**[b]** is therefore obvious for the same reasons as the network interface of claim 3**[a]**.

203. Additionally, a person of ordinary skill in the art reading Zydney would also have found it plainly obvious to use a network interface such as a cable modem or other standard interface component to provide the claimed network interface for

the client system. For example, Zydney describes the use of cable modems and specifically motivates the use of a cable modem to take advantage of higher bandwidth deployments for better quality instant voice messaging. (Zydney, 17:5-9 ("as bandwidth deployment increases via cable modems, high-speed subscriber lines, and other techniques, the conversational gaps are reduced and an even more natural sounding conversation results.").) A person of ordinary skill in the art therefore would have understood and found it obvious that the client system would have contained a network interface, such as a cable modem to enable higher bandwidth and higher quality, in order to provide connectivity to the network.

204. As noted previously, **Shinder** further explains that various types of network interface were well-known in the prior art and could be selected as a matter of implementation design choice based on considerations such as data transfer speed, network architecture, media type, and available bus type. (Shinder, pp.196, 197.) A person of ordinary skill in the art would have thus found it obvious that the client device would be coupled to a network interface for communicating over the Internet, and would have further found it trivially obvious to use a cable modem for high-speed bandwidth as suggested in Zydney or various other types of network interfaces.

(d) "an instant voice messaging application installed on the client device, wherein the instant voice messaging application includes" (Claim 27[c])

205. Element 27**[c]** is similar to claim 13, which recites in part "**wherein each of the instant voice message client systems comprises an instant voice messaging application**." As I explained previously for claim 13, Zydney discloses that each client device contains hardware and/or software used for instant voice messaging that constitute an "instant voice messaging application." Accordingly, Zydney discloses and renders obvious element 27**[c]** for the same reasons explained with respect to claim 13.

(e) "a client platform system for generating an instant voice message and," (Claim 27[c1])

206. Element 27**[c1]** is similar to claim 13, which recites in part "**an instant voice messaging application generating an instant voice message**." As I explained previously, Zydney discloses that the client device contains hardware and/or software used for generating an instant voice message that constitutes an instant messaging application. (Zydney, 14:2-5, 12:6-8, 16:1-4, 20:11-14, 25:10-13, 33:4-66, 13:1-6, Fig. 2, 4-7, 16, Claims 1, 8.) This hardware and/or software forms the "client platform system" disclosed in the '622 patent.

207. As explained previously, the "client platform system" is part of the claimed "instant voice messaging application." The term "client platform system"

under its broadest reasonable construction is "**hardware and/or software on a client for generating an instant voice message**," as I explained in **Part V.C.2**. The client platform system in Zydney is disclosed by the portions of the software agent on the client of the sending (originating) user responsible for creating the instant voice message for the same reasons I explained for claim 13.

208. As I noted in my discussion of claim construction in **Part V.C.2** above, I have been informed that in pending litigation, the Patent Owner has proposed to construe "**a client platform system**" to mean "the system of the client engine which controls other components used to generate an instant voice message." Although I disagree with this construction for the reasons I provided in **Part V.C.2** above, in my opinion, Zydney would disclose the claimed "client platform system" even under the Patent Owner's definition. This is because, as explained above, the software agent in Zydney generates a voice container by controlling various other components of the client system. These components in Zydney include, for example, a microphone (for capturing the sender's voice), processor (used to pack the recorded voice into a file), and various software components including the compression/encryption protocols 50 (Fig. 2) that are used in creation of the voice container. (Zydney, Fig. 2 (item 50), 13:1-6 ("A Software Agent utilized by the sender of the voice container provides the following functionality… pack message

into a voice container or multiple voice containers 50…."), 16:1-4, Figs. 4, 6, 7, 16.) Zydney therefore discloses "**the instant voice messaging application includes a client platform system for generating an instant voice message**," as claimed, even under the Patent Owner's narrower construction.

(f) "a messaging system for transmitting the instant voice message over the packet-switched network via the network interface," (Claim 27[c2])

209. To avoid confusion, I observe that this element of claim 27 requires that the claimed "**messaging system**" reside on the client, whereas the term "messaging system" in claim 3 refers to a system on a central system (e.g. server) that communicates with the clients. Accordingly, the "messaging system" of element 27**[c]** is actually very similar to claim 13, which recites in part "**an instant voice messaging application… transmitting the instant voice message over the packet-switched network to the messaging system**." As I previously explained with respect to claim 13, Zydney discloses that the client device contains hardware and/or software used to **transmit an instant voice message over the packet-switched network**. (Zydney, 12:1-8, 10:20-11:3, 13:1-6, Fig. 2.) The functionality of the client system for transmitting the voice container, such as the transport processes subcomponent of the software agent, discloses the claimed **messaging system for transmitting the instant voice message**. Figure 1A of Zydney

illustrates that the communication between the central server and the software agent occurs over the Internet, which evidences that the messaging system transmits instant messages **via the network interface** as I explained above with regard to claim 27**[b]**. (*Id.*, Fig. 1A.)

(g) "wherein the instant voice messaging application includes a document handler system for attaching one or more files to the instant voice message." (Claim 27[d])

210. For purposes of claim interpretation, I note that the '622 patent specification does not define the term "document handler system," but its meaning under the broadest reasonable construction is reasonably clear to a person of ordinary skill in the art. The written description refers to document handler **306** contained within the general-purpose programmable computer IVM client **208**. ('622, 12:11-21.) The patent states that to attach files to an instant voice message, the system "invokes the document handler **306** to make the appropriate linkages to the one or more files." (*Id.*, 13:33-38.) The specification also states that "[t]he document handler **306** oversees the retrieving, sending, receiving and storing of one or more documents (or files) attached to instant voice messages from/to the one or more selected IVM recipients that may be communicating with the IVM client **208**." (*Id.*, 12:26-30.) Under the broadest reasonable interpretation, a person of ordinary skill

in the art would have understood the term to have its plain and ordinary meaning to refer to a software component that handles documents.

211. I am informed that in pending litigation, the Patent Owner has proposed to construe "**document handler**" to mean a "component of the client platform that oversees the retrieving, sending, receiving and storing of one or more documents (or files) attached to instant voice messages from/to the one or more selected IVM recipients that may be communicating with the IVM client." In my opinion, this construction is not the broadest reasonable interpretation of this term. The written description of the preferred embodiment does not appear to strictly limit the claimed "document handler" to these specific functions and features under the broadest reasonable construction, as discussed above, to the exclusion of other types of software functionality for handling documents. More importantly, the claim language itself does not require the document handler perform the steps of "retrieving, sending, receiving and storing" files. The claim only requires "**a document handler system for attaching one or more files to the instant voice message**." Zydney clearly discloses this.

212. The instant voice messaging application in Zydney **attaches one or more files** to the voice container (the "instant voice message"), such as a "digitized

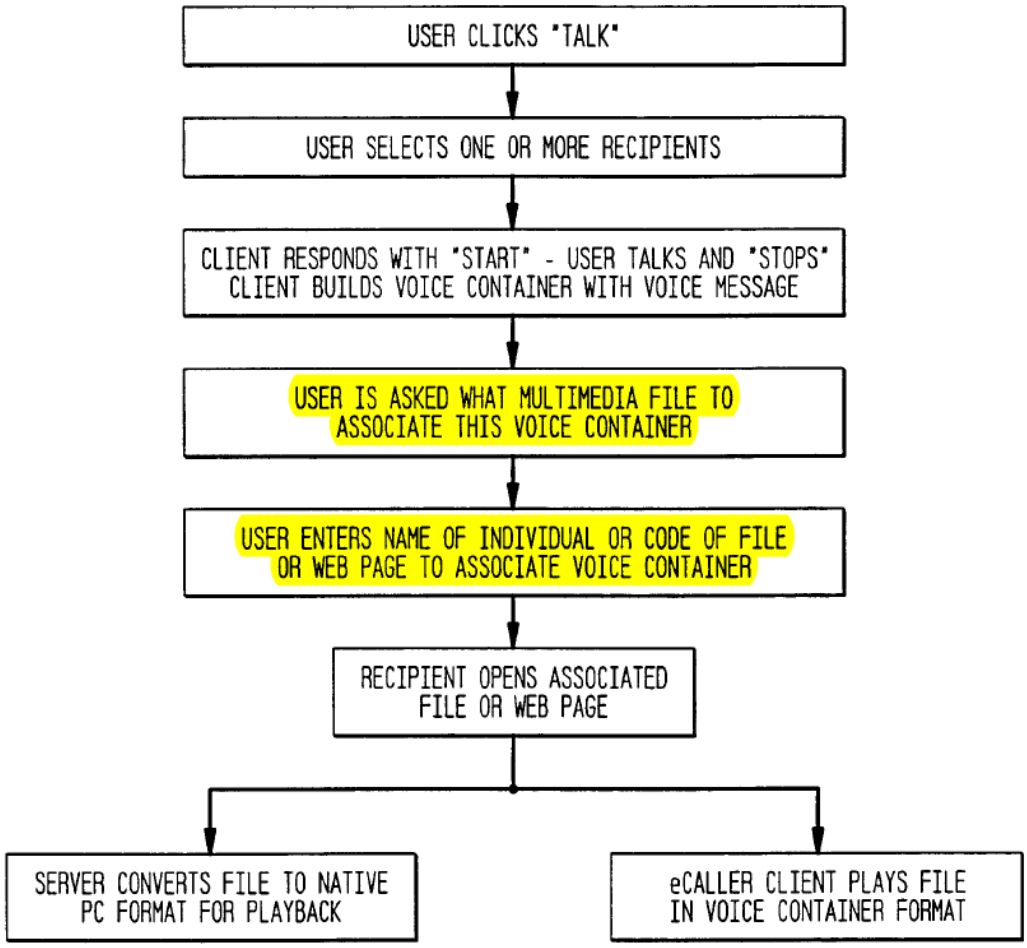
greeting card" or "other data types" to each instant voice message (voice container) to be "transported to the recipient":

> Example: Multimedia Attachments
>
> Another important application of the present invention system and method for voice exchange and voice distribution is attaching other media to the voice containers to provide a richer communications environment. For example, voice containers may have digitized greeting cards appended to them to present a personalized greeting.
>
> The voice container has the ability to have other data types attached to it and thus be transported to the recipient.

(Zydney, 19:1-7.) Controls may also be implemented to specify "the number and type of attachments that can accompany a voice container message." (*Id.*, 22:19-20.) The "document handler system" in Zydney therefore takes the form of the software functionality in the client for attaching files to a voice container.

213. Figure 6, reproduced below, depicts a process for generating an instant voice message that includes attaching a multimedia "file" to the voice container audio file. The user clicks "talk" to record a voice message in a voice container and then can specify a "file" to attach to the voice container.

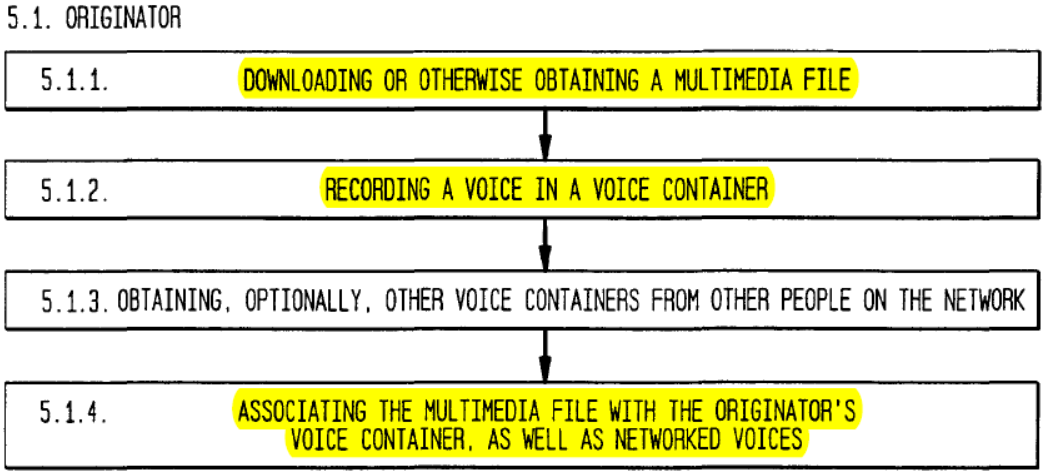
FIG. 6



(*Id.*, Fig. 6.)

214. Figures 16-18 similarly provide a three-part description of the generation and transmission of a "voice container with multimedia attachments." (*Id.*, 35:15-22, Figs. 16-18.) As shown in Figure 16, reproduced below, the "originator" can obtain a "multimedia file," record a voice container, and "associate" (attach) the multimedia file to the voice container.

FIG. 16



(*Id.*, Fig. 16.) Figure 17 continues the description from Figure 16 ("5.1. Originator"), showing "5.2. Central Server" including a step of "receiving the voice container and associated media file." (*Id.*, Fig. 17.) These figures therefore confirm that the multimedia file is attached to the voice container on the originator's client system before the voice container and attachment are transmitted to the central server.

215. Zydney also describes attaching files to voice containers using the industry-standard Multipurpose Internet Mail Extension (MIME) format, which allows attachments including "binary, audio, and video" files to be specified in message headers. (*Id.*, 19:6-12.) MIME refers to a well-known standard originally developed in the context of sending email messages containing file attachments. MIME specifies a technique for encoding a message into multiple parts (a "multipart message") with each part capable of containing a different type of data. (*Id.*)

Different aspects of the MIME protocols were defined in IETF's RFC 2045, RFC 2046, RFC 2047, and RFC 2049 from November 1996. The MIME protocol was supported by almost every mail server and mail client application in the market.

216. Zydney does not appear to explicitly describe which particular part of the software on the client system attaches files to voice containers. In my opinion, however, this is a trivial omission, because the client system operated by the user to generate and transmit the voice container (which discloses the claimed "instant voice messaging application") performs the attachment, as I explained previously. (*Id*., 19:1-7, 22:19-20, Figs. 16-17.) Moreover, a person of ordinary skill in the art would have found it obvious that the software agent that generates and transmits the voice container could also be responsible for the attachment of files to the voice container, given that the software agent performs the various other functions for generating and transmitting voice containers as shown in Figure 2. Zydney does not identify any other software on the client that performs this function.

217. In any event, it would have been obvious to implement the system of Zydney in which the software agent on the client performs the function of attaching files to the voice container prior to transmission. As explained in Zydney, "[t]o create a message, the software agent will address, pack and send the message in a voice container." (*Id.*, 14:4-5; *see also id.*, 13:2-5 ("A Software Agent utilized by

the sender of the voice container provides the following functionality… pack message into a voice container or multiple voice containers 50…").) A person of ordinary skill in the art would have found it plainly obvious that attaching files to a voice container could have been included as part of the overall process of "pack[ing]" the message into a voice container, a process that Zydney confirms is performed by the software agent on the sending client device. (*Id.*)

218. Even under the Patent Owner's proposed construction, the software agent would disclose and render obvious the claimed "document handler." Zydney describes that the client system contains functionality for retrieving, sending, receiving and storing sent and received instant voice messages (voice containers) including any attached files. (*Id.*, 1:21-2:5, 10:19-11:6, 11:14-22, 13:1-6, 13:19-22, 16:1-14, 19:2-12, Figs. 4, 6, 16-18.) Zydney describes the software agent as responsible for sending and receiving the voice containers and otherwise communicating with the central server, as illustrated in Figure 2. It would have been understood and obvious that the software agent is a component of the client platform that oversees the retrieving, sending, receiving and storing of one or more documents (or files) attached to instant voice messages from/to the one or more selected IVM recipients that may be communicating with the IVM client. Zydney does not explicitly describe another software program or module on the client system that

would perform this "overseeing" function, and therefore it would have been understood and obvious that the software agent would perform it.

219. Claim 27 is therefore obvious based on Zydney.

14. Dependent Claim 32

220. I have reproduced dependent claim 32 below:

> 32. The system according to claim **27**, wherein the instant voice messaging application includes an audio file creation system creating an audio file for the instant voice message based on input received via an audio input device coupled to the client device.

('622, Claim 32.) The additional limitations of claim 32 are identical to the additional limitations of claim 18, which I discussed previously. Zydney discloses claim 32 for the same reasons I previously explained as to claim 18.

15. Dependent Claim 33

221. I have reproduced dependent claim 33 below:

> 33. The system according to claim **27**, wherein the instant voice messaging application includes an encryption/decryption system for encrypting the instant voice messages to be transmitted over the packet-switched network and decrypting the instant voices messages received over the packet-switched network.

('622, Claim 33.) The additional limitations of claim 33 are identical to the additional limitations of claim 19, which I discussed previously. Zydney discloses

claim 33 for the same reasons I previously explained as to claim 19.

16. Dependent Claim 34

222. I have reproduced dependent claim 34 below:

> 34. The system according to claim **27**, wherein the instant voice messaging application includes a compression/decompression system for compressing the instant voice messages to be transmitted over the packet-switched network and decompressing the instant voice messages received over the packet-switched network.

('622, Claim 34.) The additional limitations of claim 34 are identical to the additional limitations of claim 20, which I discussed previously. Zydney discloses claim 34 for the same reasons I previously explained as to claim 20.

17. Dependent Claim 35

223. I have reproduced dependent claim 35 below:

> 35. The system according to claim **27**, wherein the instant voice message application generates an audible or visual effect indicating receipt of an instant voice message.

('622, Claim 35.) The additional limitations of claim 35 are identical to the additional limitations of claim 23, which I discussed previously. Zydney discloses claim 35 for the same reasons I previously explained as to claim 23.

18. Independent Claim 38

224. I have reproduced independent claim 38 below using bracketed notations (e.g. "**[a]**," "**[b]**," etc.):

38. A system comprising:

[a] a client device;

[b] a network interface coupled to the client device and connecting the client device to a packet-switched network; and

[c] an instant voice messaging application installed on the client device, wherein the instant voice messaging application includes

[c1] a client platform system for generating an instant voice message and

[c2] a messaging system for transmitting the instant voice message over the packet-switched network via the network interface,

[d] a display displaying a list of one or more potential recipients for an instant voice message.

('622, Claim 38.) Each limitation of claim 38 is disclosed and rendered obvious by Zydney.

(a) "A system comprising:" (Claim 38 Preamble)

"a client device; and" (Claim 38[a])

"a network interface coupled to the client device and connecting the client device to a packet-switched network; and" (Claim 38[b])

"an instant voice messaging application installed on the client device, wherein the instant voice messaging application includes" (Claim 38[c])

"a client platform system for generating an instant voice message and" (Claim 38[c1])

"a messaging system for transmitting the instant voice message over the packet-switched network via the network interface," (Claim 38[c2])

225. The preamble and elements 38**[a]-[c2]** are identical to the preamble of claim 27 and elements 27**[a]-[c2]**. Zydney discloses elements 38**[a]-[c2]** for the same reasons I previously explained as to elements 27**[a]-[c2]**.

(b) "a display displaying a list of one or more potential recipients for an instant voice message." (Claim 38[d])

226. Zydney discloses that the software on the user device visually presents a list of potential recipients: the originator "select[s] one or more recipients from a list maintained by the originator and presented visually by the agent." (*Id.*, Fig. 7, 14:18-19.) A person of ordinary skill in the art would have understood that this list would be "presented visually" on the **display** of the client device.

C. Zydney in View of Shinder and Appelman Renders Obvious Claims 22 and 39

1. Dependent Claim 22

227. I have reproduced dependent claim 22 below:

> 22. The system according to claim **21**, wherein the instant voice messaging application displays an indicia for each of the one or more potential recipients indicating whether the potential recipient is currently available to receive an instant voice message.

('622, Claim 22.) Zydney in view of Appelman discloses and renders obvious the additional limitation of claim 22. As I discussed above regarding element 3**[c]**, Zydney's central server "track[s] and maintain[s] the status of all software agents." (Zydney, 14:6-9, 13:12-14.) The statuses tracked by the central server include "the core states of whether the recipient is online or offline, but also offers related status information, for example whether the recipient does not want to be disturbed." (*Id.*, 14:17-15:1.) Zydney further explains that recipients are "available" and accepting messages when identified as "Available," "Do Not Disturb," "Will return," or "Out to Lunch," and "unavailable" (i.e., not accepting messages) when identified as "Not Available" or "Not logged on." (*Id.*, 32:18-33:2)

228. Zydney further discloses that the sending software agent receives two pieces of information: (1) one or more potential recipients in "a list of names that

have been previously entered into the software agent" (*id.*, 14:17-19), and (2) the "status of all recipients entered into the software agent," which "is frequently conveyed to the software agent by the central server." (*Id.*, 14:20-22.) The sending software agent in Zydney then offers the sender different communication options for each recipient based on its connectivity status received from the server. (*Id.*, 14:22-23, 15:3-4 ("Considering just the two core states [of online and offline], the software agent offers the originator alternative ways to communicate with the recipient.").)

229. Zydney further describes that the client system (disclosing the claimed "instant voice messaging application") displays to the user a list of potential recipients for a voice container (an instant voice message). (*Id.*, Fig. 7 ("selecting one or more recipients from a list maintained by the originator and presented visually by the agent"), 14:17-18 ("the originator selects one or more intended recipients from a list of names that have been previously entered into the software agent").)

230. Zydney does not appear to disclose that the client system also displays an indicia for each recipient indicating whether the potential recipient is currently available to receive an instant voice message. However, this feature would have been obvious in view of **Appelman [Ex. 1004/1104]**.

231. Appelman teaches the use of "buddy lists" that identify particular users and, for each user, specify its connectivity status for use in an instant messaging

system that can be implemented by devices on the Internet, as I discussed previously. (Appelman, Abstract, 3:44-46 (each user can be identified by "Internet address").)

232. Figure 3, reproduced below, shows a display of the buddy list in buddy list window **40**, as it would appear to the user. As shown, "the 'Home List' includes three co-users and their status as 'IN' (i.e., currently logged into the system) or 'OUT' (i.e., currently logged out of the system)." (*Id.*, 4:4-7.)

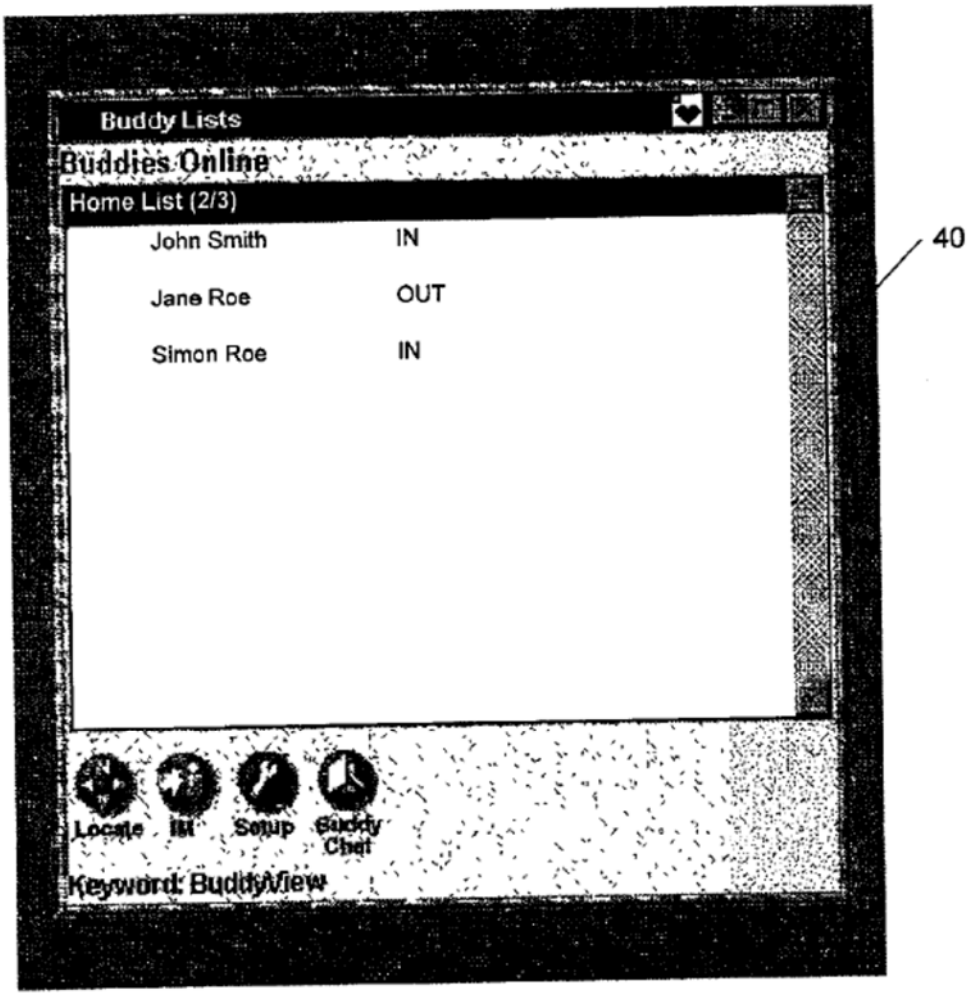


FIG. 3

(*Id*., Fig. 3.) Appelman teaches that "when the user first logs into the system, the Buddy List window 40 opens, informing the user which of the user's buddy list members are currently online." (*Id*., 4:29-32; *see also id*., 6:66-7:2.)

233. Appelman further makes clear that the "IN" and "OUT" statuses correspond, respectively to an "available" and "unavailable" connectivity status. As I explained in my summary in **Part VI.A.3** above, the "IN" and "OUT" status indicates whether or not a co-user is logged into the system. (Appelman, 3:43-47, 4:4-7.) This in turn determines whether or not that co-user is available for instant messaging communication. For example, Appelman explains that "[o]nce a co-user is displayed on a user's buddy list, indicating that the co-user is currently logged into the network system, the preferred embodiment of the invention enables a simple way of communicating with that co-user." (*Id.*, 6:1-5.) For example, the bottom of Figure 3 shows a button called "IM" that allows the user to send an instant message to another user. (*Id.*, 6:1-16.) Because each name in the Buddy List can have a status of "IN" or "OUT," Appelman discloses that the list "indicat[es] whether the potential recipient is currently available to receive … [a] message" as claimed.

234. ***Rationale and Motivation to Combine.*** A person of ordinary skill in the art would have had ample motivation to combine Zydney's system with Appelman's buddy list disclosures, predictably resulting in Zydney's instant voice

messaging system where the software agent displaying the list of recipients also displays an indication as to whether each potential recipient is currently available to receive an instant voice message (*e.g.*, "IN" or "OUT").

235. Appelman provides an express motivation to use its disclosed buddy list system with an online instant messaging system such as Zydney. Appelman teaches that an "important aspect" in "online" communication systems is "knowledge of the people/users/processes on that system." (Appelman, 1:12-16.) Appelman explains that "[a] problem with networks on any size [sic] is tracking personal relationships and maintaining knowledge of the people/users/processes on that system." (*Id*., 1:37-39.) Appelman teaches that its disclosed buddy list system "addresses the problem of tracking personal relationships and maintaining knowledge of the people/users/processes on that system with a unique way of establishing and maintaining user definable on-line co-user lists." (*Id*., 1:46-49.)

236. A person of ordinary skill in the art would also have regarded Appelman and Zydney as closely analogous references in the same field of providing instant messaging communication over a computer network. In fact, like Appelman, Zydney itself discloses the use of a "buddy list," and also discloses a technique for conveying the "available" and "unavailable" status of each potential recipient. (Zydney, 30:13-15, 14:17-15:7 ("The status of all recipients entered into the

software agent is frequently conveyed to the software agent by the central server. This includes whether the core states of whether the recipient is online or offline"); 13:12-14, 14:6-13, 30:13-15, 32:18-33:2 (describing statuses including "Available – available for messages or live talking" and "Not logged on – Message will be sent to the message server[.]").) Appelman and Zydney are both directed at solving the common problem of maintaining a list that tracks availability of other users on the system, in order to determine which other users are available for communication. In light of the fact that both systems have common goals and seek to address the same problem, it would have been natural for a person of ordinary skill in the art to apply the more detailed "Buddy List" disclosures in Appelman to Zydney to provide the claimed indication as to whether each potential recipient is currently available to receive an instant voice message.

237. A person of ordinary skill in the art would also have readily appreciated Appelman's motivation to use its buddy list system with Zydney to provide a convenient and straightforward interface for each user to quickly view the online/offline status of the users in her buddy list. (Appelman, 4:33-36 ("If the Buddy List window **40** [in Fig. 3] is left open, the user has a current, real-time list of all the user's buddies in who are online at any particular moment.").)

238. Finally, buddy lists showing users' online and offline statuses, as disclosed by Appelman, were so ubiquitous in instant messaging systems prior to 2003 that basic market considerations would have compelled a person of ordinary skill in the art to have been aware of and have considered them. The Appelman patent originated with AOL, as indicated on the face of the patent, and Appelman discusses AOL in its Background section. As I discussed previously in this Declaration, AOL's Instant Messenger program, which included the use of buddy lists to keep track of "online buddies" and "offline buddies," had more than 100 million registered users by 2002. (Young **[Ex. 1005/1105]** at 336-38, Fig. 14-4.) Zydney's Background of the Invention section similarly discusses known text-based messaging systems including "instant messaging, where text is typed and exchanged between computers when a 'buddy' address (or group address) is present in an address field," and explains that a need existed for a convenient voice messaging technique. (Zydney, 1:7-17.) Zydney also contemplates that each user has a "buddy list," as I noted above. (*Id*., 30:13-15.)

239. With hundreds of millions of instant messaging users familiar with "buddy lists" to quickly indicate which friends are online and offline, a person of ordinary skill in the art considering Zydney's instant messaging system would have been amply motivated to use the known and straightforward buddy list features

disclosed by Appelman in order to meet the expectations of likely instant messaging users. A person of ordinary skill in the art would have understood that adapting "buddy list" techniques of Appelman could have reduced the learning curve of users of the IM system of Zydney by leveraging an existing and exceedingly well-known buddy list user interface.

240. Accordingly, Zydney in view of Appelman discloses "wherein the instant voice messaging application displays an indicia for each of the one or more potential recipients indicating whether the potential recipient is currently available to receive an instant voice message." Claim 22 is therefore obvious based on the prior art.

2. Dependent Claim 39

241. I have reproduced dependent claim 39 below:

> 39. The system according to claim **38**, wherein the display includes an indicia for each of the one or more potential recipients indicating whether the potential recipient is currently available to receive an instant voice message.

('622, Claim 39.) Claim 39 is similar to claim 22, which recites "wherein the instant voice messaging application displays an indicia for each of the one or more potential recipients indicating whether the potential recipient is currently available to receive an instant voice message." As I explained previously for claim 22, Zydney in view

of Appelman discloses that the list of recipients includes connectivity status for each recipient. Accordingly, Zydney in view of Appelman discloses and renders obvious claim 39 for the same reasons I previously explained with respect to claim 22.

D. Zydney in View of Shinder and Clark Renders Obvious Claims 14-17 and 28-31

1. Dependent Claim 14

242. I have reproduced dependent claim 14 below:

> 14. The system according to claim **13**, wherein the instant voice messaging application includes a message database storing the instant voice message, wherein the instant voice message is represented by a database record including a unique identifier.

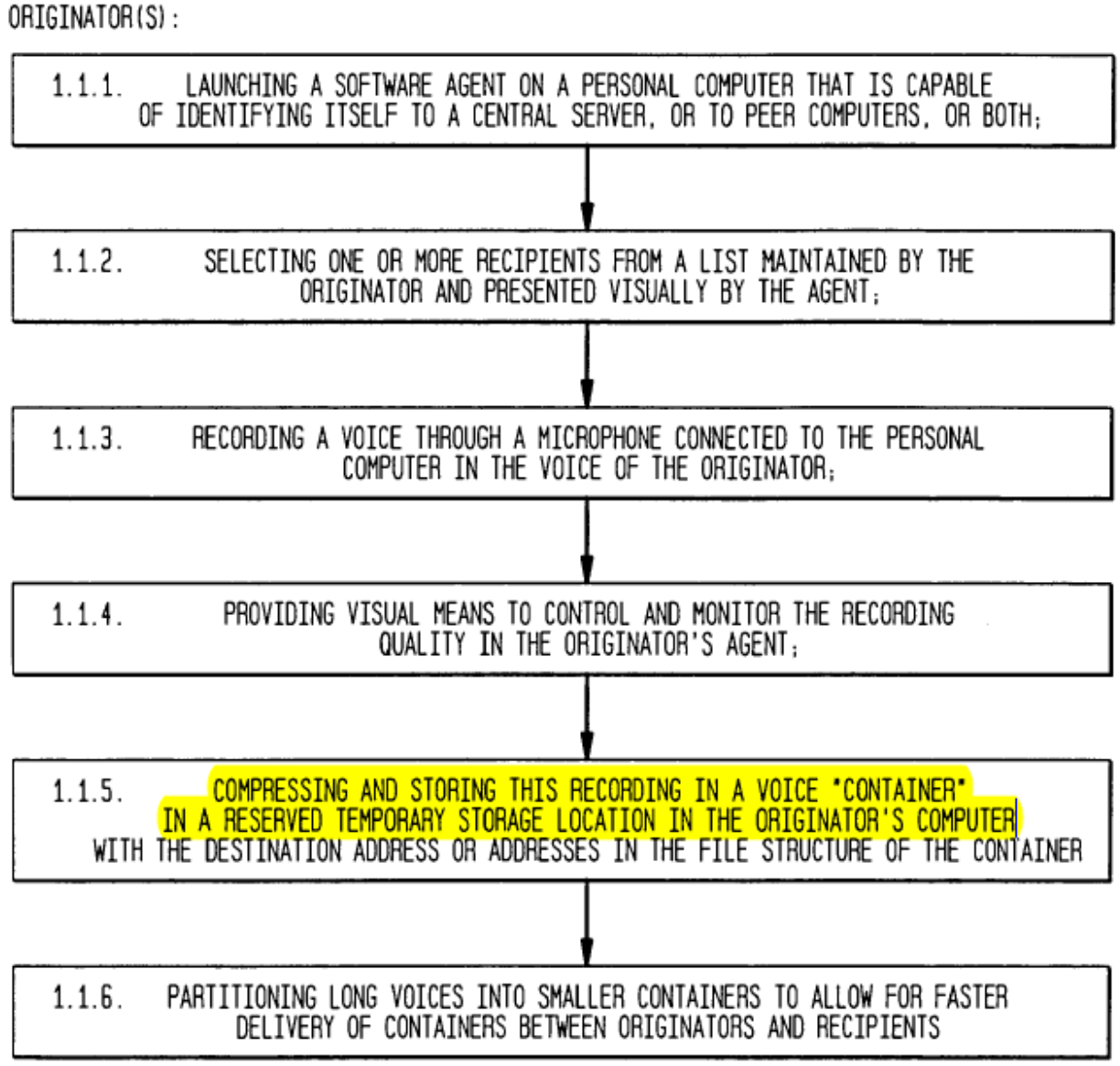
('622, Claim 14.) As I explained previously, Zydney and Shinder disclose and render obvious claim 13. These references, in further in view of Clark, disclose the additional limitations of claim 14. For clarity, I address this claim in two parts.

(a) "wherein the instant voice messaging application includes a message database storing the instant voice message,"

243. Zydney discloses that the instant voice messaging client system stores both outgoing and incoming instant voice messages. In particular, the messages to be sent to an unavailable recipient can be stored at the sending user's device until the recipient becomes available. (Zydney, 2:3-5 ("[T]he present invention for voice exchange and voice distribution provides the ability to store messages both locally

and centrally at the server whenever the recipient is not available for a prescribed period of time."), 11:3-6 (same).) Step 1.1.5. of Zydney's Figure 7 (shown below) likewise discloses that outgoing messages are "compress[ed] and stor[ed] … in a voice 'container' in a reserved temporary storage location in the originator's computer." (*Id*., Fig. 7.) Outgoing instant voice messages are also stored locally when a software agent is located behind a firewall. (*Id.*, 30:11-16 ("When a software agent is located behind a Firewall . . . all agent-stored messages will be delivered to the server."), 2:3-5, 11:3-7.)

FIG. 7



(*Id.*, Fig. 7.)

244. Accordingly, Zydney discloses "the instant voice messaging application … storing the instant voice message."

245. Zydney also describes storing messages that are received by a software agent, which can be saved on the recipient's computer for example. (*Id.*, Fig. 9 (describing "receiving the voice containers" and "controls for saving, deleting or resending recorded containers from the recipient's computer.").) A user's successful login will result in "all of the user messages waiting in the message server being downloaded to the software agent." (*Id*., 30:6-7.) Zydney also discloses playing the received instant voice messages on the user's device. (*Id.*, 13:19-22, 14:14-16, 16:10-14, 20:14-17, Fig. 4.) It would have been understood to a person of ordinary skill in the art that downloading the messages to the software agent would involve storing the messages on the user's device so that the user could listen to them.

246. Zydney also describes an alternative "guest log-in" feature where a guest may log-in to a computer and receive messages that are then deleted when the guest user logs off instead of being retained in storage. (*Id*., 31:1-6.)

247. Zydney does not use the term "**message database**" to describe storage of instant voice messages on the client system, but in my opinion, the storage in Zydney meets this definition under its broadest reasonable construction. The claim does not, for example, require that the "message database" be a particular type of database, such as a relational database. Nevertheless, in the event it is argued that

the client system storage in Zydney is insufficient to disclose a "message database," this limitation would have been obvious in view of **Clark [Ex. 1008/1108]**.

248. Clark discloses a system for storing and organizing sent and received electronic messages, such as instant messages, in a message store **23** that discloses a **message database**. "A collection of electronic messages **22** is stored in one or more message stores **23**. Each message store **23** comprises a memory, file or **database** structure that provides temporary or permanent storage for the contained messages **22**." (Clark, 9:11-15.) Clark specifically describes the message store as a "database." (*Id.*, 11:1-5 ("As shown in FIGS. 5A and 5B catalog database **28** and message store **23** may be separate from one another or may be integrated in a single integrated message store. Each of these components is preferably provided in the form of a database comprising a plurality of related tables.").)

249. This database can also be located on a client system. For example, Figure 4A shows an embodiment in which a user's computer **18** contains the message client **27** and message store **23**. (*Id.*, 10:27-33, Fig. 4A.)

250. The message database system in Clark "can be applied to organizing any sort of electronic messages which are to be temporarily or permanently stored," including "instant messages," "voice mail messages," or "any other present or future

types of electronic messages," which may include attachments of various types. (*Id*., 8:31-44.)

251. Message store **23** stores both outgoing (sent) and incoming (received) messages. The disclosed system uses a MessageSummary table **52** (shown in relevant part below) that records information about the underlying message objects in message store **23**. (*Id*., 16:50-53.) The table **52** confirms that the stored messages include "sent" messages as well as "received" messages:

IsCorresp	Indicates whether the message is correspondence from or to a recognized correspondent. The logic to set this value is described below.
MessageDateTime	The date and time to be displayed to the user when listing messages in a folder. For unsent messages this contains the creation date/time, for received messages this contains the receive date/time, and for sent message this contains the send date/time.
DisplayNames	The sender or recipient names to be displayed to the user when listing messages in a folder. For received messages this contains the sender's name, and for sent or unsent messages this contains is the names of the primary recipients.

(*Id*., 17:12-22.) As shown in the table, the messages in the message store **23** include both "sent" and "received" (and "unsent") messages, with fields indicating the time

the message was sent or received, the name or the sender or recipient, and other information.

252. When Clark's message database system is applied to Zydney's system that stores instant voice messages on the client system, Zydney in view of Clark discloses and renders obvious "wherein the instant voice messaging application includes a message database storing the instant voice message."

253. ***Rationale and Motivation to Combine.*** A person of ordinary skill in the art would have had ample motivation to combine Zydney's system with Clark's message database disclosures, predictably resulting in Zydney's system where the sent and received instant voice messages are stored in a message database on the client system. By using the Clark message database, as explained below, outgoing and incoming messages can be stored on the client system in Zydney and more effectively organized using Clark's techniques.

254. Clark provides express motivations to use its message database system with an instant messaging system such as Zydney's system. Clark explains that existing prior art electronic message systems did not provide sufficiently effective and efficient ways to store, organize, and search electronic messages. (Clark, 1:20-4:8.) For example, in some systems, a message might only be accessible through one folder at a time, and various existing message search techniques did not provide

satisfactory results. (*Id*., 1:57-2:4, 2:52-3:12.) According to Clark, attempts had been made to "overcome problems associated with the current folder/message model," but still there was "a need for systems and methods which can automatically organize stored electronic messages, such as e-mail messages, instant messages, voice messages and fax messages." (*Id*., 3:19-4:13.)

255. Clark addresses this need with its "computer-based system for cataloging, retrieving and manipulating electronic messages saved in a message store." (*Id*., 4:25-39.) In particular, Clark expressly motivates the use of its message database system with "instant message" systems, as noted in the previous paragraph, and teaches that "[t]he invention can advantageously be integrated with messaging client software . . . to facilitate the organization of electronic messages." (*Id*., 4:36-38.) A person of ordinary skill in the art would have recognized that the software agent in Zydney is an example of "messaging client software," which thus could have utilized the message database of Clark.

256. Clark further explains that its invention "can be applied to organizing any sort of electronic messages which are to be temporarily or permanently stored," including messages with attached files. (Clark, 8:31-44.) As I noted previously, Clark also discloses that its message database stores both sent and received messages, consistent with the well-known messaging system model that stores both

sent and received messages, such as "sent messages in a Sent Messages folder" in prior art email systems. (*Id*., 1:43-48, 2:57-61, 17:9-21, 32:23-25.)

257. These teachings would have expressly motivated a person of ordinary skill in the art to implement Clark's message database system "integrated with messaging client software" in Zydney to store and organize sent and received instant voice messages (voice containers), including attachments. As I discussed previously, Zydney discloses that the client system stores both incoming and outgoing instant voice messages and also describes that a user can view a list of received messages. (Zydney, 2:3-5, 11:3-7, 30:15-16, Figs. 6, 9, 16-18.) Zydney also discloses that the client system may be a personal computer, which would have been well-known to have local storage capacity such as a hard drive. (*Id*., 11:16-18.) However, Zydney does not explicitly describe the details of the underlying storage structure or details about techniques for organizing, searching, or otherwise managing the stored messages. A person of ordinary skill in the art would have readily appreciated Clark's express motivation to use its message database system to improve the Zydney client system. In particular, given that Zydney discloses storing both sent and received messages, and that storing sent messages was a well-known feature of existing electronic message systems as Clark explains, it would have been plainly obvious to a person of ordinary skill in the art to store both the

sent and received messages in the message database to meet users' expectations that they could access their own previously-sent messages, to know what they had sent, as well as accessing messages received from others.

258. Further, Clark expressly motivates that "the invention could also be applied to the organization of messages which already exist in a message store." (Clark, 8:47-50.) A person of ordinary skill in the art would have recognized that Clark's teachings directly complement Zydney, where incoming and outgoing voice containers already exist in storage on the client system. A person of ordinary skill in the art would have appreciated that the combination would have advantageously provided a suitable structure to store and organize the instant voice messages, while also providing the obvious benefits to the user of Zydney's instant voice messaging system to be able to organize and retrieve the messages and attachments in the efficient way that Clark teaches.

259. A person of ordinary skill in the art also would have recognized that it would have been obvious to store Zydney's instant voice messages locally on the client computer system as an alternative to storing messages at the central server. As I have previously explained, Zydney describes that messages can be stored "both locally and centrally at the sever" when a recipient is unavailable, such as in a "server file." (Zydney, 2:2-5, 10:20-11:6, Fig. 2 (showing "message store" connected to

server), Fig. 4.) Clark's teachings would have motivated a person of ordinary skill in the art to implement the message database on the client system (*e.g.*, personal computer hard drive) as an obvious design choice, which would have conveniently permitted users to access their stored sent and received messages on their local computers without needing to access the central server, for example at times when the computer was not connected to the Internet or other network providing access to the central server.

260. Furthermore, Clark expressly motivates the use of its message database system based on the particular advantageous features its system provides. For example, Clark teaches that each electronic message can be associated with multiple folders based on attributes of each message such as sender/recipient, date, attachments, and keywords, unlike prior art systems where each message would be confined to one folder. (Clark, 1:57-2:4, 4:27-35.) Clark explicitly touts this feature as an advantage: "Advantageously a user can manually organize the same message into multiple folders without making multiple copies of the message." (*Id*., 34:4-6.) Similarly, "[a]n advantage of the invention is that such associations may be made simultaneously on the basis of a wide range of criteria." (*Id*., 10:4-10.) A person of ordinary skill in the art would have easily appreciated that these same criteria apply to Zydney's voice containers, which include identifying information such as the

sender, recipient, "originating time," delivery time, and other information as shown in Figure 3. Zydney describes "presenting the list of voice containers and their originators" to a recipient of voice containers, but does not appear to explicitly describe grouping the displayed voice containers according to a particular organization. (Zydney, Fig. 9.) It would have been readily apparent that organizing Zydney's instant voice messages based on criteria such as the sender, date, presence of attachments, and other features would have improved the user experience, allowing the user to organize messages into folders by a particular sender or recipient (*e.g.*, a certain friend or family member), by time sent or received, by the presence of attachments, and other criteria as taught by Clark.

261. A person of ordinary skill in the art also would have appreciated that Clark and Zydney are naturally complementary references in the same field of handling electronic messages between clients over a computer network. As I discussed previously, Clark contemplates a client/server messaging system architecture where users operate computers running messaging software, just like Zydney where each user operates a device such as a personal computer running a software agent in coordination with a central server. (Clark, 7:65-8:30, Fig. 1A; Zydney, 11:16-18, Fig. 1A.) As noted previously, just like Zydney, Clark makes clear that its system database can be used to store "instant messages."

262. A person of ordinary skill in the art also would not have expected any particular technical difficulties in implementing the combination, such as by applying the message database to Zydney's voice containers. On the contrary, Clark expressly motivates that "[t]he invention does not rely on any specific message format (such as RFC822 or MAPI) or any specific messaging protocol (such as SMTP or X.400), but can be readily adapted to the set of information made available by any practical message format and protocol." (Clark, 8:50-54.) Using Clark's message database system to store Zydney's sent and received instant voice messages on the client system therefore would have been expected to provide nothing more than predictable results.

(b) "wherein the instant voice message is represented by a database record including a unique identifier."

263. Zydney in view of Clark discloses and renders obvious this limitation. Clark discloses that each electronic message in the message store **23** is represented by a database record including a unique identifier. Specifically, "when a message is added to a message store **23**, the message store server **24** assigns a unique StoreMessageId to the message and generates an event which informs catalog server **29** of the newly added message." (Clark, 11:50-54.) "StoreMessageId and StoreAttachId may comprise numbers, or other identifiers, assigned to the messages and attachments respectively by message store server **24**." (*Id*., 11:21-24.)

264. The unique identifier (StoreMessageId) is stored in a database record: "Catalog database **28** also has a MessageSummary table **52** which contains the StoreMessageId **52**A of messages in message store **23**." (*Id*., 11:31-32.) Clark also teaches that "catalog database **28** and message store **23** may be separate from one another or may be integrated in a single integrated message store." (*Id*., 11:1-5.) The unique identifier represents the underlying stored message and can be used to retrieve it: "Using the StoreMessageId **52**A and the related StoreId **51**A, catalog server **29** can make requests to the message store server **24** to read messages from message store **23**." (*Id*., 11:38-40.) The StoreMessageId represents the message, while the StoreID identifies the message store. (*Id*., 11:8-12.)

265. Accordingly, when the Clark message database system is applied to Zydney's stored instant voice messages, Zydney in view of Clark discloses "wherein the instant voice message is represented by a database record including a unique identifier."

266. ***Rationale and Motivation to Combine.*** As I discussed above, a person of ordinary skill in the art would have found it obvious to modify Zydney in view of the disclosures in Clark. That discussion applies equally here, and the combination would have predictably resulted in the Zydney system implementing the Clark

message database system, where each instant voice message is identified by a database record including a unique identifier.

267. A person of ordinary skill in the art also would have specifically found it obvious to modify Zydney to represent each message with a unique identifier as taught by Clark as part of Clark's disclosed message database implementation, as a naturally convenient way to organize and store the messages for searching and retrieval. Assigning unique identifiers to represent objects in database records was well-known long before the '622 patent in the prior art, and Clark's teaching is consistent with known database implementation design choice in this regard.

2. Dependent Claim 15

268. I have reproduced dependent claim 15 below:

> 15. The system according to claim **14**, wherein the message database includes a plurality of instant voice messages recorded by a user of the client device and instant voice messages received over the packet-switched network.

('622, Claim 15.) Each limitation of claim 15 is disclosed and rendered obvious by the prior art.

269. As I explained previously, Zydney in view of Clark discloses and renders obvious claim 14. (*See* **Part VI.D.1** above.) Zydney in view of Clark discloses and renders obvious the additional limitations of claim 15 for the same

reasons. As I explained for claim 14, the message database disclosed and rendered obvious by Zydney in view of Clark stores both outgoing instant voice messages (voice containers) and instant voice messages received over the Internet (packet-switched network). The outgoing voice containers are recorded by a user of the client device, such as by using a microphone. (Zydney, 16:1-3, Fig. 7 (describing "recording a voice through a microphone connected to the personal computer in the voice of the originator" in step 1.1.3. and "compressing and storing this recording in a voice 'container' in a reserved temporary storage location in the originator's computer" in step 1.1.5.).)

270. Zydney discloses saving instant voice messages "received over the packet-switched network." As I discussed above regarding claim 3**[a]**, Zydney discloses transmitting messages over the Internet, which a person of ordinary skill in the art would have known is a packet-switched network.

271. Accordingly, Zydney in view of Clark discloses and renders obvious "wherein the message database includes a plurality of instant voice messages recorded by a user of the client device and instant voice messages received over the packet-switched network." Claim 15 is therefore obvious based on the prior art.

3. Dependent Claim 16

272. I have reproduced dependent claim 16 below:

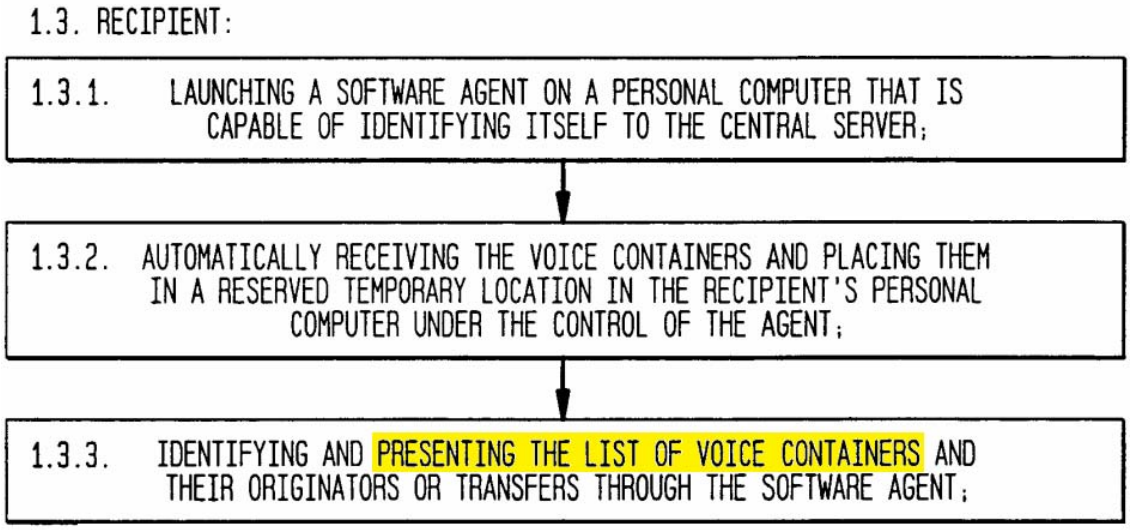
16. The system according to claim **15**, wherein the instant voice messaging application displays at least one of the plurality of instant voice messages stored in the message database.

('622, Claim 16.) Each limitation of claim 16 is disclosed and rendered obvious by the prior art.

273. As I explained previously, Zydney in view of Clark discloses and renders obvious claim 15. (*See* **Part VI.D.2** above.) And Zydney in view of Clark discloses and renders obvious the additional limitation of claim 16.

274. Zydney discloses displaying an instant voice message. Step 1.3.2. in Figure 9, reproduced below, discloses the step of "automatically receiving the voice containers," which is followed in Step 1.3.3. by "presenting the list of voice containers." (Zydney, Fig. 9.)

FIG. 9



(Zydney, Fig. 9 (partial figure shown).)

275. Zydney does not explicitly describe displaying "at least one of the plurality of instant voice messages stored in the message database," but **Clark** discloses and renders obvious this limitation. As I discussed previously for claim 14, Clark discloses a message database and renders obvious the use of the message database to store Zydney's instant voice messages, including the messages sent and received by a particular user. My discussion for claim 14 applies equally here, rendering obvious the predictable result of Zydney's instant voice messaging system using Clark's database where the displayed instant voice message is stored in the database.

276. Clark also specifically describes displaying the messages stored in the message database. Clark describes that "[t]he user interface comprises a display **60** which could be used to allow a user to access messages at a desktop or laptop computer." (Clark, 12:8-10.) Figure 6, reproduced below, shows an example user interface screen display for viewing individual messages. As shown in Figure 6, "[d]isplay **60** includes a message header display panel **66** and a message contents display panel **67**. When the interface detects that a user has selected a specific message, for example by clicking on a row in the list in panel **64** then the interface displays selected information about the associated message in message header

panel **66** and displays the body of the associated message in the message contents panel **67**." (*Id.*, 12:63-13:2.)

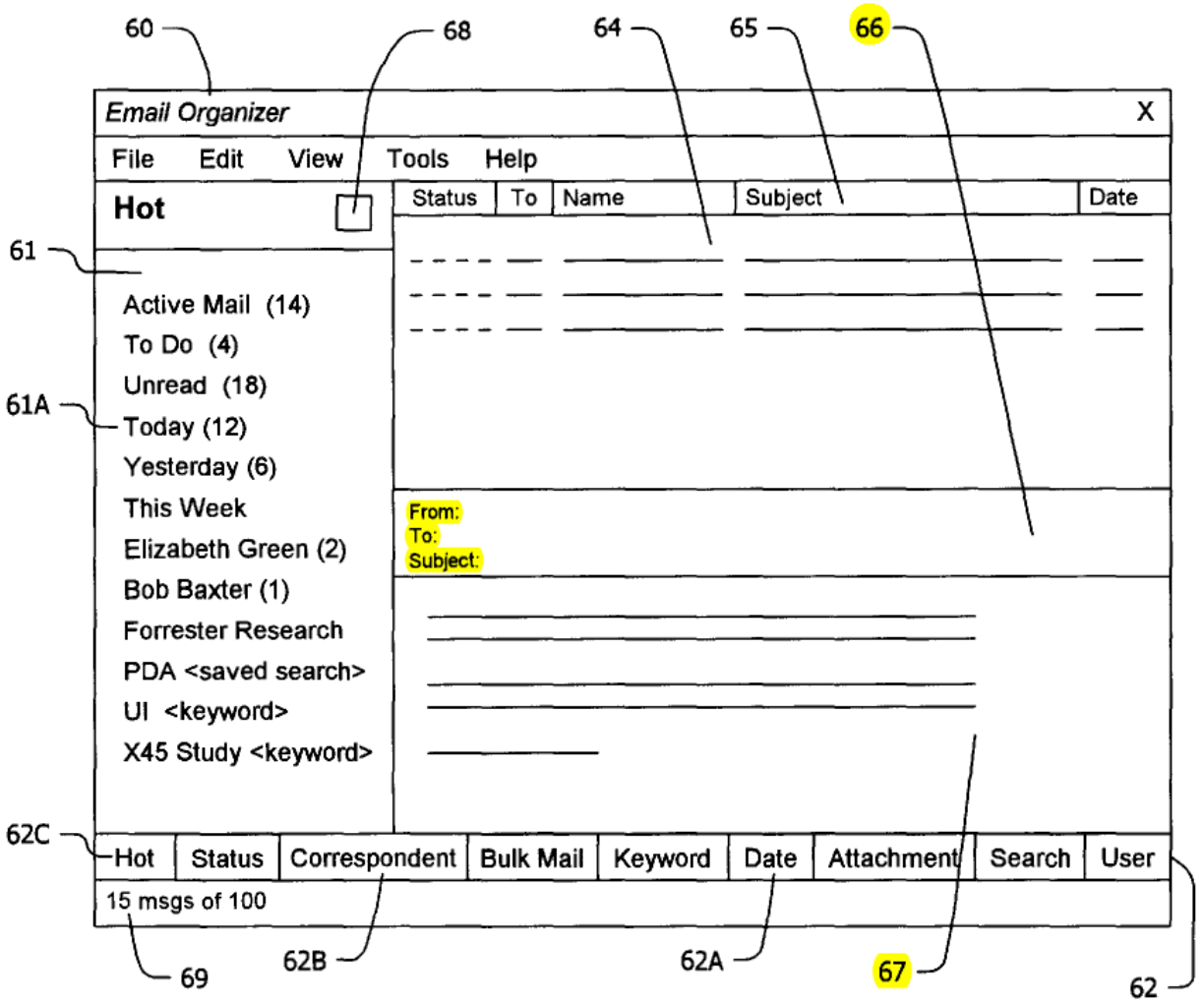


FIG. 6

(*Id.*, Fig. 6 (showing "message display panel **66**" and "message contents display panel **67**").) Clark notes that this user interface illustrates an embodiment in which the user accesses emails from the message store **23** (*id.*, 13:24-28), but as I noted previously, the message store **23** can be used to store instant messages or other types

of messages, and it would have been obvious to use to store the instant voice messages of Zydney.

277. Clark also specifically discloses that the user can access and display messages sent by that user that are saved in the message database. In particular, Clark's message store contains both "sent" and "received" messages as identified in the MessageSummary table **52**. (*Id*., 16:50-52, 17:9-22 ("IsCorresp[:] Indicates whether the message is correspondence from or to a recognized correspondent.").) When the user accesses and displays a selected message as Clark describes, the message can be any message in the database, including the "sent" messages, and indeed Clark describes a folder specifically for "Sent" messages. (*Id*., 12:8-10, 12:63-13:2, 32:23-25.)

278. Furthermore, Clark teaches that the stored messages can be automatically organized and displayed based on criteria such as their date (*e.g.*, "Today" or "Yesterday"), which is identified based on the MessageDateTime field for the message. (*Id*., 32:45-58, 33:31-37, Fig. 6 (display interface showing "Today" and "Yesterday" folders).) The MessageDateTime field applies to both sent and received messages, so that a user could select the "Today" folder to display messages sent or received with today's date in the MessageDateTime field. (*Id*., 17:12-17.)

279. When the Clark database system was used to store and retrieve Zydney's instant voice messages, as I previously explained for claim 14**,** the feature described in Zydney Figure 9 of "presenting the list of voice containers" would obviously be modified to include displaying the voice containers (instant voice messages) from the message database, including the voice containers sent and received by a particular user.

280. Accordingly, Zydney in view of Clark discloses "wherein the instant voice messaging application displays at least one of the plurality of instant voice messages stored in the message database." Claim 16 is therefore obvious based on the prior art.

4. Dependent Claim 17

281. I have reproduced dependent claim 17 below:

> 17. The system according to claim **14**, wherein the instant voice messaging application includes a file manager system performing at least one of storing, deleting and retrieving the instant voice messages from the message database.

('622, Claim 17.) Each limitation of claim 17 is disclosed and rendered obvious by the prior art.

282. As I explained previously, Zydney in view of Clark discloses and renders obvious claim 14. (*See* Part VI.D.1 above.)

283. I am informed that in pending litigation, the Patent Owner has proposed to construe "**a file manager system**" to mean a "system that services requests regarding files." I account for this construction in my analysis below.

284. Zydney in view of Clark discloses and renders obvious this limitation. As I discussed above regarding claims 14 and 15, Zydney discloses **storing** instant voice messages at the user's computer. (Zydney, 30:15-16, Fig. 9.) For example, a sending (originating) user can specify that the message will be delivered as part of a single instant voice message, which causes the voice container to be stored. (*Id.*, 16:1-4 ("Once the delivery mode has been selected, the originator digitally records messages for one or more recipients using a microphone-equipped device and the software agent. The software agent compresses the voice and stores the file temporarily on the PC if the voice will be delivered as an entire message.").) Because Zydney discloses that the instant message delivery mode can be selected by the user, and that selection causes the storage of the voice container, the storage can occur in response to a user request. (*Id.*, 15:4-6 ("This choice can either be dictated by the originator or automatically selected by the software agent, according to rules that are stored.").) It would have been obvious to a person of ordinary skill that in order for the software agent in Zydney to store the voice container file, the client

would have included a system that services requests from the agent to create and write files.

285. Zydney also discloses "**retrieving**" instant voice messages. For example, a recipient's software agent "provid[es] visual means for adjusting the quality and speed of playback of each recording through the software agent." (*Id.*, Fig. 9.) A person of ordinary skill in the art would have understood that playing a recorded voice message would require **retrieving** that message from storage. Further, a person of ordinary skill in the art would have understood from the disclosure of "visual means" that the user has controls that respond to user requests.

286. Zydney also describes controls on the client computer for "**deleting**" instant voice messages, as well as saving or resending them to additional recipients. (*Id.*, Fig. 9 (describing "controls for receiving the voice containers" and "controls for saving, deleting or resending recorded containers from the recipient's computer.").)

287. As I noted previously, Zydney does not appear to explicitly describe a "message database" and therefore does not explicitly describe "a file manager system performing at least one of storing, deleting and retrieving the instant voice messages from the message database." However, this limitation would also have been obvious in view of **Clark**. As I discussed previously for claim 14, Clark

discloses a message database system for storing and organizing both sent and received messages, which can be instant voice messages. Among other things, the system includes the capability to **add** messages to the message database (i.e., **store** messages in the message database) and **delete** messages from the message database in response to a user request: "Message client **27** will typically generate requests in response to user input such as requests to message store server **24** to add, change or delete a message." (Clark, 18:25-29.)

288. Figure 2 of Clark, reproduced below with highlighting, shows the computer system including a user interface that interacts through message client **27** to make requests (such as the delete message request) to the message store server **24**, which "manages the messages **22** in the message store **23**." (*Id*., 9:15-16.)

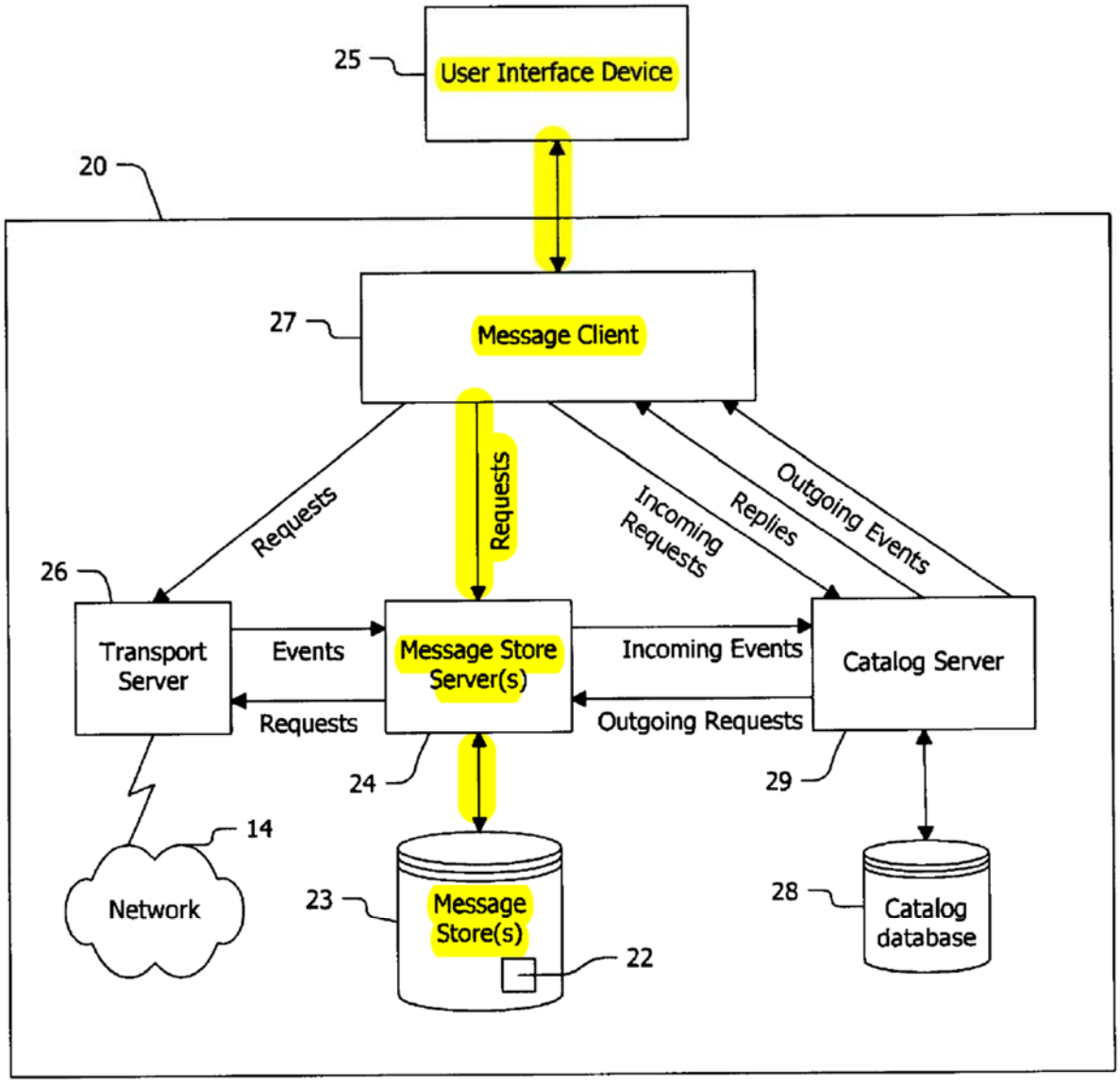


FIG. 2

(*Id.*, Fig. 2.)

289. Clark's system also permits a user to **retrieve** messages from the message database (message store), as part of its "computer-based system for

cataloging, retrieving and manipulating electronic messages saved in a message store." (*Id*., 4:25-27.) The user can retrieve any message from the database, including sent messages and received messages, as I discuss further below. In particular, Clark discloses a user interface that permits a user to select and view messages: "A user interface **15** equipped with a suitable input device **17** permits a user to select a folder and to view and manipulate messages which have shortcuts in the selected folder." (*Id*., 8:65-9:1.) Clark explains that message store server **24**, shown in Figure 2 reproduced above, handles requests for messages from the message store **23**: "Among other tasks, message store server **24** receives requests for messages **22** from other parts of system **20** and locates and provides the requested messages **22**." (*Id*., 9:17-19.)

290. Figure 6, reproduced below, shows an example user interface screen display for viewing individual messages.

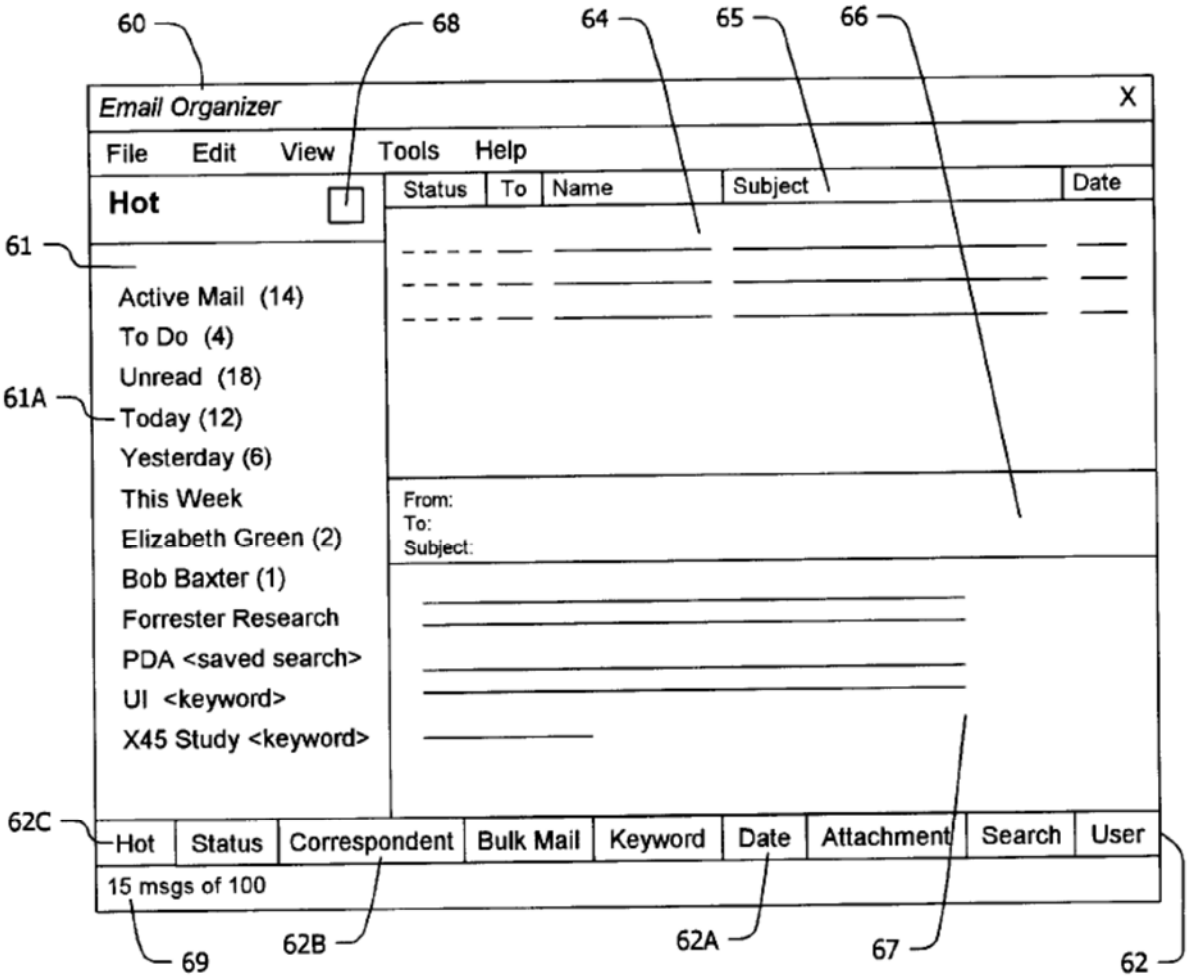


FIG. 6

(*Id*., Fig. 6.) Clark notes that this user interface illustrates an embodiment in which the user accesses emails from the message store **23** (*id*., 13:24-28), but as I noted previously the message store **23** can be used to store instant messages or other types of messages, and it would have been obvious to use to store the instant voice messages of Zydney. Clark describes that "[t]he user interface comprises a display **60** which could be used to allow a user to access messages at a desktop or

laptop computer." (*Id*., 12:8-10.) As shown in Figure 6, "[d]isplay **60** includes a message header display panel **66** and a message contents display panel **67**. When the interface detects that a user has selected a specific message, for example by clicking on a row in the list in panel **64** then the interface displays selected information about the associated message in message header panel **66** and displays the body of the associated message in the message contents panel **67**." (*Id*., 12:63-13:2.) A person of ordinary skill in the art would have understood that when a user selects and views a message stored in the database, the system is **retrieving** the message from the message store **23** (message database) to be displayed.

291. Clark also specifically discloses that the user can retrieve messages sent by that user that are saved in the message database. In particular, as I explained previously for claim 14, Clark's message store contains both "sent" and "received" messages as identified in the MessageSummary table **52**. (*Id*., 16:50-52, 17:12-22.) When the user retrieves a selected message as Clark describes (*id*. at 12:8-10, 12:63-13:2), the message can be any message in the database, including the "sent" messages. In fact, Clark describes that a folder can be specifically created for "Sent" items. (*Id*., 32:23-25.) Furthermore, messages can be automatically organized and displayed based on criteria such as their date (e.g. "Today" or "Yesterday"), which is identified based on the MessageDateTime field for the message. (*Id*., 32:45-58,

33:31-37, Fig. 6 (display interface showing "Today" and "Yesterday" folders).) The MessageDateTime field, in turn, applies to both sent and received messages, so that retrieving messages dated "Today" (such as by selecting the "Today" folder in the user interface of Figure 6) will retrieve messages sent or received with today's date in the MessageDateTime field. (*Id*., 17:12-17.)

292. The components of the Clark system that carry out the functions including deleting and retrieving messages from the message database in response to user requests disclose the claimed **file manager system**. Using Patent Owner's proposed construction, Clark in view of Zydney discloses a "system that services requests regarding files." For example, the components shown in Figure 2, reproduced above, including the message client **27** in communication with the message store server **24** that accesses message store **23**, disclose a file manager system with the functionality that I discussed in the preceding paragraphs. Clark describes that the message store **23** may contain any type of file, including attachments to messages such as "images, sound media, video, executable files, word processing files" and other files, as well as any type of electronic message comprising any type of file. (*Id*., 8:35-44, 11:14-24.)

293. I also note that, when used with the Zydney instant voice message system, the instant voice messages stored in the Clark message database would also

disclose **files** to be managed by the file manager system. Zydney describes the transmission of "files" including audio files and attached files. (Zydney, 16:1-4, 21:14-16 (describing voice "files" used for instant voice messages).) Figures 6 and 16 of Zydney further describe attaching a "file" to the instant voice message, and Zydney also describes attaching files to voice containers using the industry-standard Multipurpose Internet Mail Extension (MIME) format, which allows attachments including "binary, audio, and video" files to be specified in message headers. (*Id.*, Figs. 6, 16, 19:6-12.)

294. Therefore, when Clark's database is implemented with Zydney's voice message files and attachment files, the management of requests regarding the files in the database discloses the claimed **file manager system** under the Patent Owner's proposed construction.

295. Accordingly, Zydney in view of Clark discloses and renders obvious "wherein the instant voice messaging application includes a file manager system performing at least one of storing, deleting and retrieving the instant voice messages from the message database," including in response to a user request.

296. ***Rationale and Motivation to Combine.*** As I discussed above regarding claim 14, a person of ordinary skill in the art would have found it obvious to modify Zydney in view of the disclosures in Clark. Those discussions apply equally here,

and the combined disclosure would have predictably resulted in the Zydney system using the Clark message database system including its file manager system to perform at least one of storing, deleting and retrieving the instant voice messages from the message database in response to a user request.

297. In addition, the functions of storing, deleting, or retrieving objects from a database were exceedingly well-known in the prior art and would have been considered basic features of any database management system. Any user of a database system for storing messages, including the Clark system as implemented with Zydney's system, would have expected the ability to store, delete, and retrieve the messages, as a fundamental purpose of the database, which would have further motivated a person of ordinary skill in the art implementing the Clark database system with Zydney to enable the user to easily manage the instant voice messages in the database. Claim 17 is therefore obvious based on the prior art.

5. Dependent Claim 28

298. I have reproduced dependent claim 28 below:

> 28. The system according to claim **27**, wherein the instant voice messaging application includes a message database storing the instant voice message, wherein the instant voice messages is represented by a database record including a unique identifier.

('622, Claim 28.) The additional limitation of claim 28 is identical to claim 14,

which I discussed previously. Zydney in view of Clark discloses and renders obvious claim 28 for the same reasons I previously explained as to claim 14.

6. Dependent Claim 29

299. I have reproduced dependent claim 29 below:

> 29. The system according to claim **28**, wherein the instant voice message stored in the message database include a plurality of instant voice messages recorded by a user of the client device and instant voice messages received over the packet-switched network.

('622, Claim 29.) Claim 29 is substantially the same as claim 15, which recites "**wherein the message database includes a plurality of instant voice messages recorded by a user of the client device and instant voice messages received over the packet-switched network**." Zydney in view of Clark discloses and renders obvious claim 29 for the same reasons I previously explained with respect to claim 15.

7. Dependent Claim 30

300. I have reproduced dependent claim 30 below:

> 30. The system according to claim **29**, further comprising: a display displaying at least one of the plurality of instant voice messages stored in the message database.

('622, Claim 30.) Claim 30 is similar to claim 16, which recites "**wherein the**

instant voice messaging application displays at least one of the plurality of instant voice messages stored in the message database." Zydney in view of Clark discloses and renders obvious claim 30 for the same reasons I previously explained with respect to claim 16.

8. Dependent Claim 31

301. I have reproduced dependent claim 31 below:

> 31. The system according to claim **28**, wherein the instant voice messaging application includes a file manager system storing, deleting and retrieving the instant voice messages from the message database in response to a user request.

('622, Claim 31.) Claim 31 is similar to claim 17, which recites "**wherein the instant voice messaging application includes a file manager system performing at least one of storing, deleting and retrieving the instant voice messages from the message database**." As I explained previously, Zydney in view of Clark discloses a file manager system storing, deleting and retrieving the instant voice messages from the message database in response to a user request. Zydney in view of Clark discloses and renders obvious claim 31 for the reasons I explain here and the same reasons I previously explained with respect to claim 17.

E. Zydney in View of Shinder and Hethmon Renders Obvious Claims 4, 5, and 24-26

1. Dependent Claim 4

302. I have reproduced dependent claim 4 below:

> 4. The system according to claim **3**, wherein the instant voice message includes an action field identifying one of a predetermined set of permitted actions requested by the user.

('622, Claim 4.)

303. I am informed that in pending litigation, the Patent Owner has proposed to construe "**action field**" as "**a block of data identifying permitted actions**." This construction is consistent with the broadest reasonable interpretation, and thus, I account for it in my analysis. (*See also* '622, 14:7-10 ("The content of the action field is selected from a list of permitted actions, which among other actions includes: connect, disconnect, subscribe, unsubscribe, and post message.").)

304. Zydney does not appear to explicitly describe that the instant voice message contains a "field" that identifies one of a predetermined set of permitted actions requested by the user. However, this feature would have been obvious over Zydney in view of **Hethmon**, which confirms that the claimed "**action field**" is a well-known and built-in feature of the Hypertext Transfer Protocol (HTTP) 1.1.

305. As I discussed previously in my summary in **Part VI.A.5**, Hethmon describes HTTP/1.1, a protocol for sending requests and responses between client

and server computers over the Internet. Hethmon explains that HTTP is often known as a "request-response" protocol, meaning that a client (such as software on a personal computer) can send a "request message" to a server, to which the server responds by sending back a "response message" to the client. (Hethmon, pp.10-11.)

306. The format and syntax for a "request message" in HTTP/1.1 was well-documented and widely known. Hethmon provides the following summary of a request message that a client would send to a server using HTTP/1.1:

```
Request         = Request-Line
                  *( General-Header
                   | Request-Header
                   | Entity-Header )
                  CRLF
                  [ Entity-Body ]
Request-Line    = Method SP Request-URI SP HTTP-Version CRLF
```

(*Id.*, p.54.)

307. As shown above, the "Request" contains a number of elements including a "**Request-Line**," which is further described in the last line of the excerpt quoted above. Hethmon explains that "[t]he request line is the message sent by the client to the server to request a resource or an action to take place." (*Id.*, p.54.) As shown, the "Request-Line" contains at least three items of information, each separated by a space character ("SP"): **(1)** a "Method" that identifies an action to be taken on a resource, **(2)** a "Request-URI" (which is often a Uniform Resource

Locator (URL) identifying the name of the resource that is the subject of the request), and **(3)** the "HTTP-Version" for the request (such as "HTTP/1.1"). (*Id.*) The "CRLF" at the end of the "Request-Line" is a well-known sequence of control characters known as "carriage return, line feed," which simply indicate the end of the line of text. Any subsequent text following the "CRLF" sequence, therefore, will appear on a new line.

308. For purposes of my analysis of claim 4, the critical field here is the "Method" of the "Request-Line," which discloses the claimed "**action field**" of claim 4. Hethmon explains that the Method is a keyword "to indicate the type [of] action the request is asking the server to execute." (*Id.*, p.55.) Furthermore, in HTTP/1.1, the Request can specify any one of a predetermined set of seven different methods, namely OPTIONS, GET, HEAD, POST, PUT, DELETE, and TRACE. (*Id.*, p.55-61 ("With HTTP/1.1, there are seven basic methods . . .").) The Request-Line therefore discloses **an action field identifying one of a predetermined set of permitted actions requested by the user**.

309. Hethmon illustrates how the Method specified by the action field (Request-Line) identifies a permitted action requested by the user. For example, "[t]he POST method is used as a way for a client application to submit data to a resource on a server application." (*Id.*, p.78.) The data to the transmitted (effectively

the "payload" of the message") is contained in the "Entity-Body" field in the request message, as shown above. (*Id*., p.54.) Specifically, "[u]sing the POST method, the client sends an entity body to the server for processing." (*Id*., p.78.) "This allows for data submission via HTTP to accomplish various goals, such as database updating or order entry." (*Id*., p.58.) POST may be used to transmit data of various types, as the POST Request message can include a field specifying "Content-Type" for the Entity-Body field carrying the "payload" of the message, and a "Content-Length" indicating the size of the payload. (*See id*., p.78.) In the exemplary POST message below from Hethmon, for example, the client has sent 23 bytes of plain text information to the "/cgi-sin/survey" resource on the server.

```
POST /cgi-bin/survey HTTP/1.1
Host: www.example.com
From: phethmon@hethmon.com
Content-Type: text/plain
Content-Length: 23

week=ToMuch&access=Psycho
```

(*Id*., p.78.)

310. Another method similar to POST is known as PUT. (*Id*., pp.59-60, 80-81.) "The PUT method is analogous to [] sending a file via FTP. The client requests the server to accept the enclosed entity body, and store it as the Request-URI in the request line." (*Id*., p.59.) Thus, upon completion of a successful PUT, "[e]ither a

new resource is created or an old resource is replaced" on the server. (*Id*., p.80.) Hethmon provides the following example of a PUT message:

```
Request:  PUT /users/phethmon/welcome.html HTTP/1.1
          Content-Type: text/html
          Content-Length: 3109

          [ 3109 bytes of entity ]
```

(*Id*., p.59.) In this example, "/users/phethmon/welcome.html" points to a location on the server, "HTTP/1.1" specifies that the sender is using version 1.1 of HTTP, "Content-Type: text/html" specifies the type of content being transmitted, and "Content-Length: 3109" specifies that 3109 bytes are being transmitted.

311. The POST and PUT features of HTTP/1.1 therefore disclose examples of **an action field identifying one of a predetermined set of permitted actions requested by the user**, as claimed. In fact, the '622 patent expressly refers to a "**post message**" as an example of one of the permitted actions that can be identified in the "action field." ('622, 14:6-10.)

312. ***Rationale and Motivation to Combine***. It would have been obvious to a person of ordinary skill in the art to combine Zydney with Hethmon, with no change in their respective functions. As noted previously, Zydney explains that an instant voice message can be sent by the sender to a central server. (Zydney, 16:7-12, 15:19-21, 27:15-16; Fig. 1A (showing voice container transmission path through

the central server), Fig. 8 (Step 1.2.3.).) The combination of Zydney and Hethmon would have predictably resulted in the voice container of Zydney is transmitted by the sending client to the central server using HTTP/1.1, resulting in the voice container being carried in an HTTP/1.1 message that includes an "**action field**," such as the "POST" or "PUT" method described above. Because the "POST" or "PUT" message in this combination was the result of the Zydney user's decision to send an instant voice message, the action field identifies a "**permitted action**[] **requested by the user**," as claimed.

313. A person of ordinary skill in the art would have had ample motivation to combine Zydney and Hethmon in this manner. To begin with, Zydney expressly encourages skilled artisans to use HTTP as a mechanism for transmitting objects from clients to servers. Zydney explains that HTTP is "a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods (commands). A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred." (Zydney, 7:21-8:3.)[15] A person of ordinary skill in the art would have understood that the

[15] Although HTTP is commonly used to request and deliver pages from the World

voice containers of Zydney could have been transported within using HTTP/1.1 requests, such as by using the "POST" or "PUT" methods described above. A person of ordinary skill in the art would have been motivated to do so to exploit the beneficial features of HTTP/1.1 as identified in Zydney, such as the use of a "generic" protocol that "allow[s] systems to be built independently of the data being transferred." (*Id.*)

314. A person of ordinary skill in the art would also have been motivated to use HTTP/1.1 request messages described in Hethmon because they would have simplified the implementation of the system of Zydney. For example, to implement the system of Zydney, a person of ordinary skill in the art would have to answer a basic technical question: what mechanism should I use to transport the voice container from the client to the central server? HTTP/1.1, as suggested by Zydney, would have provided an obvious and straightforward answer to that question. The format of HTTP/1.1 requests was exceedingly well-documented and the subject of published industry standards, including the draft cited in Zydney. Zydney specifically cites to and incorporates by reference a draft of the HTTP/1.1 standard

---

Wide Web, HTTP is a more generic protocol that is not limited to the delivery of web pages, as expressly confirmed by Zydney. (Zydney, 7:21-8:3.)

that describes the operation and format of HTTP/1.1 requests. (*Id.*, 8:3-6 (incorporating by reference Ex. **1011/1111**).) Software for forming, sending, receiving, and processing HTTP/1.1 messages was widely well-available by the late 1990s. Accordingly, using well-known HTTP requests (such as POST or PUT) to transport the voice containers would have eased implementation burdens by leveraging an existing and very mature technology. This would have obviated the need for the developer to develop his or her own command protocol to run atop TCP/IP to handle delivery of voice containers to the central server. Use of standard HTTP requests would thus also have increased interoperability because clients and servers typically had built-in abilities to process HTTP/1.1 requests.

315. A person of ordinary skill in the art would have also recognized that using HTTP would have eased the administrative burdens associated with client devices that operate within an intranet, behind a network "firewall" – an arrangement explicitly discussed in Zydney. (Zydney, 9:3-5, 28:12-18.) The term "firewall" in the context of networked computing generally refers to a mechanism for protecting client computers from hackers, viruses, and other undesired Internet content by preventing certain types of communications with computers on the Internet. Because of the desirability of accessing the World Wide Web, however, firewalls were often configured to allowed clients to engage in HTTP-based

communications (associated with particular TCP ports such as port 80) with computers on the Internet. Configuring the clients in Zydney to communicate with the central server using HTTP, therefore, would have made it easier for those communications to pass through an organization's network firewalls.

316. This benefit is expressly recognized in Zydney: "Where the Firewall is administered to limit ports accessible to an external server, the software agent can be changed to use other available ports, most notably the ports used for generic request-response traffic for the World Wide Web." (Zydney, 28:16-18.) The term "port" in this context refers to a TCP port, which is a numerical value used to identify the type of network services being used. One of the most well-known TCP ports (port 80) specifies use of HTTP network traffic, as Hethmon describes. (Hethmon, p.10 (describing that the client connects to the server "on the default port of 80 (unless otherwise specified).").) Hethmon shows the basic client-server message architecture through port 80 as follows:

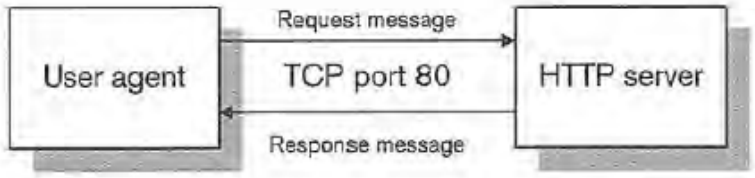


Figure 2.2 Basic client to server HTTP operation

(Hethmon, p.11.)

317. Thus, the reference in Zydney to "the ports used for generic request-response traffic for the World Wide Web," would have been understood by a person of ordinary skill in the art as a reference to the ports used for HTTP communications. (*See id.*; *see also id.*, pp.8-10 ("HTTP is the protocol used to send and receive messages between Web clients and servers. . . . HTTP is a request-response type of protocol.").) Not only does this statement from Zydney confirm that HTTP techniques could have been used to facilitate communication between the client and central server, but it expressly suggests use of HTTP techniques to prevent the undesirable result of having the firewall block permissible voice instant message communications.

318. Hethmon also motivates the use of HTTP as a means of transporting information, and specifically "more than simple text," between clients and servers. (Hethmon, p.14.) In fact, "HTTP/1.0 developed from the need to exchange more than simple text information. It became a way to build a distributed hypermedia information system adapted to many needs and purposes." (*Id.*) As HTTP was enhanced in subsequent revisions, "HTTP developed from a simple information retrieval system into a general purpose transaction system capable of building quite complex systems with standard applications across multiple platforms." (*Id.*, p.16.) Among other enhancements, the POST method provided "a way to send substantial

information to a server for processing"—"with POST, virtually unlimited entity bodies could be sent in a request message." (*Id.*, p.15.) HTTP version 1.1 introduced additional improvements, as Hethmon describes. (Hethmon, pp.16-24.) In view of these teachings, a person of ordinary skill in the art would have readily appreciated that HTTP could be used as the transport protocol for transmitting instant voice messages in Zydney's instant voice messaging system, such as using the POST method, consistent with Zydney's explicit discussion of HTTP as an available protocol. (Zydney, 7:20-8:6.)

319. Finally, a person of ordinary skill in the art would have perceived no significant technical obstacle in implementing the combination of Zydney and Hethmon described above. Zydney already discloses using standard TCP/IP techniques to transport a voice container to the server. (Zydney, 23:11-12 ("The voice container will be sent using standard TCP/IP transport), 5:15-18 (describing TCP/IP).) Because HTTP is built on top of TCP/IP, it would have been straightforward to use HTTP to facilitate voice container delivery from clients to the central server, and as I explained in detail above, a skilled artisan would have had many compelling reasons to do so.

2. Dependent Claim 5

320. I have reproduced dependent claim 5 below:

> 5. The system according to claim **4**, wherein the predetermined set of permitted actions includes at least one of a connection request, a disconnection request, a subscription request, an unsubscription request, a message transmission request, and a set status request.

('622, Claim 5.) As discussed previously for claim 4, Zydney in view of Hethmon discloses and renders obvious that the client transmits the voice container to the central server via either an HTTP 1.1 POST or PUT message. These same teachings also disclose and render obvious the claimed "**message transmission request**" of claim 5. As I discussed previously, in the Zydney system using HTTP 1.1, the user transmits the instant voice message to its intended recipient(s) by uploading it to the central server, which then delivers it to the intended recipient(s). The POST or PUT method specified in the HTTP request message, as discussed previously, therefore discloses a **message transmission request**, as claimed, because it is used (in the combination of Zydney and Hethmon) to transmit the voice container to the central server in Zydney.

3. Independent Claim 24

321. I have reproduced independent claim 24 below using bracketed notations (e.g. "**[a]**," "**[b]**," etc.):

> 24. A system comprising:

[a] a network interface connected to a packet-switched network;

[b] a messaging system communicating with a plurality of instant voice message client systems via the network interface; and

[c] a communication platform system maintaining connection information for each of the plurality of instant voice message client systems indicating whether there is a current connection to each of the plurality of instant voice message client systems,

[d] wherein the messaging system receives connection object messages from the plurality of instant voice message client systems,

[d1] wherein each of the connection object messages includes data representing a state of a logical connection with a given one of the plurality of instant voice message client systems.

('622, Claim 24.) Each limitation of claim 24 is disclosed and rendered obvious by the prior art.

(a) Preamble of claim 24: "A system comprising:"

322. To the extent the preamble is limiting, Zydney discloses "[a] system" with the features discussed in my analysis of the limitations of claim 24 below.

(b) "a network interface connected to a packet-switched network;" (Claim 24[a])

"a messaging system communicating with a plurality of instant voice message client systems via the network interface; and" (Claim 24[b])

"a communication platform system maintaining connection information for each of the plurality of instant voice message client systems indicating whether there is a current connection to each of the plurality of instant voice message client systems," (Claim 24[c])

323. Elements 24**[a]-[c]** are identical to elements 3**[a]-[c]**. Zydney discloses and renders obvious elements 24**[a]-[c]** for the same reasons I previously explained as to elements 3**[a]-[c]**.

(c) "wherein the messaging system receives connection object messages from the plurality of instant voice message client systems," (Claim 24[d])

324. As I explained above, I have been asked to apply a construction of "**connection object messages**" as "messages containing data representing the state of the connection and code (one or more methods) for establishing and maintaining the logical connections between an instant voice messaging server and instant voice messaging clients."

325. Zydney discloses that the central server receives status information from each client system regarding its connectivity status, including "online" and "offline" statuses. (Zydney, 14:8-9, 14:20-23.) In particular, Zydney explains that

the sender with a software agent loaded on a computer or other device "will log on, authenticate, and notify the central server of its status." (*Id*., 14:2-4.) Furthermore, "[w]hen a software agent has been authenticated all other software agents that are in the specific group or community of the authenticated software agent will be notified that the other agent(s) are on line." (*Id.*, 24:17-20.) Likewise, "[s]hould a software agent log off the system then a notification of such will be sent to all interested software agents." (*Id*., 24:20-21.)

326. Zydney thus discloses "**messages containing data representing the state of the connection**," under the first part of the definition of "**connection object messages**." But because Zydney does not describe format and content of the messages the client sends to central server to "notify the central server of its status" (*id*., 14:2-4), it is not clear if those messages also include "**code (one or more methods) for establishing and maintaining the logical connections between an instant voice messaging server and instant voice messaging clients**." Zydney also does not appear to explicitly describe whether or not the status information from the client systems is specifically received by the components of the central server that disclose the **messaging system** as claimed, including the message server and the transport server as discussed for claim 3**[b]**. But these features would have been obvious over Zydney in view of **Hethmon**, as I will explain below.

327. In particular, as I explained previously for claims 4 and 5, Zydney describes that various different types of transports may be used in its system, and specifically discusses HTTP as one such transport protocol. (Zydney, 26:16-19, 7:21-8:6.) As I explained for claims 4 and 5, Zydney in view of Hethmon renders obvious the use of HTTP 1.1 for communications between the client and server in Zydney's system. As I explained previously for claim 4, a person of ordinary skill in the art would be particularly motivated to use the POST method to transmit substantial data, in view of Hethmon's teaching that POST provided "a way to send substantial information to a server for processing" because "with POST, virtually unlimited entity bodies could be sent in a request message." (Hethmon, p.15.)

328. The same rationale and motivation to combine applies equally here. When using HTTP 1.1 under this combination, the client would need to send the status information (*e.g.*, "ONLINE") to the Zydney central server through one of the seven available HTTP request method types. A person of ordinary skill in the art would have understood and found it obvious to use a POST request, which I discussed previously for claim 4, as the vehicle to provide the client's status information to the central server.

329. As noted previously, messages in HTTP 1.1 generally conform to the following structure:

```
Request          = Request-Line
                   *( General-Header
                    | Request-Header
                    | Entity-Header )
                   CRLF
                   [ Entity-Body ]
Request-Line     = Method SP Request-URI SP HTTP-Version CRLF
```

(*Id.*, p.54.)

330. The Request-Line identifies the Method (i.e., POST or one of the other six available methods). An exemplary POST request message is illustrated below, where the Request-Line at the top provides the "POST" instructions:

```
POST /cgi-bin/survey HTTP/1.1
Host: www.example.com
From: phethmon@hethmon.com
Content-Type: text/plain
Content-Length: 23
```

(*Id.*, p.78.)

331. Under this combination, when the client in Zydney transmits an HTTP POST message to the central server to report the client's status (*e.g.*, "ONLINE"), the POST message contains **data representing the state of the connection**, i.e., data indicating the client's status (*e.g.*, ONLINE). The POST message also contains **code (one or more methods)** (*e.g.*, the "POST" method) for establishing and maintaining the logical connections between an instant voice messaging server and instant voice messaging clients. In particular, as shown above, the message contains

the POST instructions that disclose the **code**, and specifically the **method**, i.e., POST. In the example illustrated above, the message instructs the server to POST the content at location /cgi-bin/survey. (*Id*., p.78; *see also* pp.58-59 (discussing the content of the POST message).)

332. The "POST" code in the POST messages from the clients to the central server reporting their statuses in Zydney are also **for establishing and maintaining the logical connections** between the server and the clients. As noted previously, Zydney explains that when the software agent logs onto the system, it will "notify the central server of its status." (Zydney, 14:4.) This status includes, as noted above, whether the software agent is online or offline. (*Id.*, 14:22-23.) Under the combination of Zydney and Hethmon, the client provides this notification via an HTTP 1.1 POST message method to the central server to convey the client's "ONLINE" status. The client's "online" status discloses a **logical connection** between the client and the server in this context because it indicates that the central server can communicate with the client.

333. Although the '622 patent does not explicitly define a "logical connection," a person of ordinary skill in the art would have recognized that a logical connection in this context refers to a connection between devices in a network that is defined by the devices' functional relationship to each other, regardless of the

physical hardware (*e.g.*, wires, routers, etc.) physically connecting them. (*See*, *e.g.*, Microsoft Computer Dictionary (1997) [**Ex. 1012/1112**] at p.288 ("logical . . . Of or pertaining to a conceptual piece of equipment or frame of reference, regardless of how it may be realized physically. *Compare* physical.").)

334. Thus, when the client system in Zydney logs on and communicates to the central server its "online" status, the client and central server have established a logical connection indicating that the client is online and able to communicate with the server. (Zydney, 32:9-12 ("After the software agent has logged onto the system and has been authenticated they [sic; it] will have access to the system. During the authentication process the Internet address of the newly authenticated software agent will be made known to all other interested software agents and retained in the proxy server [of the central server].").) Indeed, Zydney describes the process of a client logging on to the central server as a "connection request," indicating that the client is seeking to go "online" and therefore obtain a logical connection with the server. (Zydney, 31:20-21.) Likewise, Zydney describes the overall communication system between the clients and central server, including clients' log-on authentication techniques and connectivity statuses, as "The Connection Service Description," reflecting that the on-line clients are "connected" to the central server through the logical connection. (*Id*., 14:1-16.)

335. The feature is also disclosed in at least two additional separate ways. *First*, HTTP 1.1 provides for "persistent connections" between the client and the server, so that a connection established between the client and the server remains open until it is closed. "Starting with HTTP/1.1, the protocol implements, as a default behavior, the practice of persistent connections. This means that once a client and server open a connection, the connection remains open until one or the other specifically requests that it be closed." (Hethmon, p.18.) A Connection header in each message "allows either the client or server to specify options to apply to the current session." (*Id*., p.89.) For example, the client may provide a value of "close" in the Connection field, "indicating the sender's desire to close the current connection, once the response is sent." (*Id*.) If the "close" value is not provided, "the connection remains open," and thus this is the default. (*Id*., p.18.) The presence and content of the Connection header is checked each time a message is received to determine whether to maintain or close the connection. (*Id*., p.203 ("If we find the Connection header, we check to see if its value is close.").)

336. Thus, when the Zydney client logs on and transmits an "ONLINE" status in a POST method message, a persistent HTTP connection (which is a **logical connection** between the client and server, as it constitutes a functional communication relationship between them, for the same reasons as I explained

previously) is established and maintained between the client and the central server. The connection remains open until it is specifically closed, as discussed previously.

337. *Second*, the POST request message also contains a Content-Length field whose contents are used to establish and maintain the logical connection (the socket) at the TCP layer to complete the transfer of the data carried by the POST method. When HTTP is implemented atop TCP/IP, "[t]he socket is the basic mechanism used by programs to communicate in the TCP/IP world. In simple terms, it is a communications link between two programs across a TCP/IP network." (Hethmon, p.129.) The socket discloses a **logical connection** between the client and server, as the client and server may be located anywhere on the TCP/IP network and are connected by virtue of the established communication link. Hethmon provides further details regarding establishing socket connections and transmitting data through them. (*Id.*, pp.129-143.) The code in the POST method request in HTTP, in turn, specifies the length of a socket connection. Hethmon explains:

> When a client uses the POST method, it must include a Content-Length header as part of the request. This must be included as a way for the server to determine the end of the entity body. Since the socket connection must remain open for the server to send a response, the client cannot simply close the connection to mark the end of the data, as is done for FTP transfers.

(Hethmon, pp.58-59.) Hethmon provides examples of Content-Length fields containing the value "3189" for "3819 bytes of data" and "6082" for "6082 bytes of data." (*Id*., p.59.) The following illustration, reproduced previously, similarly shows the code of the POST method including the Content-Length field, this time specifying 23 bytes of data.

```
POST /cgi-bin/survey HTTP/1.1
Host: www.example.com
From: phethmon@hethmon.com
Content-Type: text/plain
Content-Length: 23
```

(*Id*., p.78.)

338. Thus, the code contained in the POST messages from the clients, including the Content-Length field, further discloses code for establishing and maintaining the logical connections between the clients and the server. In particular, as I have explained, the Content-Length specifies the length of the socket connection that must remain open to accommodate the message contents.

339. When the client systems in Zydney used HTTP/1.1 to transmit these POST messages to the central server, a person of ordinary skill in the art would have understood and found it obvious that the messages would be received specifically by the **messaging system** in Zydney's central server, which includes the message server and transport server shown in Figure 2 of Zydney, as I discussed for claim

3**[b]**. In particular, as noted previously, Zydney discloses that the message server and transport server are responsible for receiving messages using TCP/IP from the client systems. (Zydney, 33:1-2, 23:11-12, 29:1-2, Fig. 2.) Just as the clients' instant voice messages sent over the Internet using TCP/IP would be received by the transport server in accordance with the TCP/IP protocols, so too the POST messages sent over the Internet using HTTP/1.1 atop the TCP/IP protocols would likewise be received by the TCP/IP transport server. To the extent this is not explicitly described in Zydney, a person of ordinary skill in the art would have found it plainly obvious that the transport server receives the TCP/IP message transmission, because it is the only one of the components of the central server depicted in Figure 2 that is described as being involved with "TCP/IP" transport. (*Id*., Fig. 2 ("Transport Server (TCP/IP, UDP, PSTN, Others").) Similarly, it would have been obvious that the messaging system could also receive the same messages, just as it receives voice containers, because it is the only component of the central server depicted in Figure 2 described as having a "message" function, and Figure 2 shows a bi-directional arrow between the message server and the "messaging processes" in the software agent indicating the communication between the client and the server. (*Id*., Fig. 2.) Finally, in Zydney's system, the central server and clients disclose an **instant voice messaging server and instant voice messaging clients**, respectively, as discussed previously.

340. I note one potential ambiguity in mapping the claim limitations to the prior art. Claim 24 is a system claim that recites that the messaging system receives "connection object messages," plural, "from the plurality of instant voice message client systems," plural. The broadest reasonable interpretation is that the messaging system receives at least one connection object message from at least two client systems; that is, at least two client systems each provide at least one connection object message. Zydney discloses and this limitation insofar as every software agent notifies the central server of its status, as discussed above.

341. The claim does not under its broadest reasonable construction require that the messaging system receive a plurality of connection object messages from each one of a plurality of client systems (i.e., at least two client systems each transmit at least two connection object messages). However, even if interpreted in such a manner, this feature also would have been obvious for reasons similar to those I have explained above. Specifically, Zydney discloses that each client system provides its status both when it logs on and when it logs off. Specifically, Zydney describes that the central server will "track and maintain the status of all software agents." (Zydney, 14:6-9; *see also* 13:12-14 (the central server will "maintain and provide the status of all software agents").) The possible statuses for each agent include "online or offline" and may also include additional status information such as

"whether the recipient does not want to be disturbed." (*Id*., 14:23-15:1.) Zydney describes that, just as the software agent provides its status when it logs on, it also provides its status when it logs off. (*Id*., 24:20-23 ("Should a software agent log off the system then a notification of such will be sent to all interested software agents. The software agent will notify the server with the Internet address that they are currently using for the session to identify where the messages should be sent.").)

342. Although Zydney does not appear to explicitly describe that the client logging off transmits its "OFFLINE" status to the central server, it would have been understood and trivially obvious, and perhaps imperative, that it would do so that the central server can accurately track and maintain the status for that client and therefore would not attempt to deliver messages addressed to that recipient. Thus, when using HTTP, just as the client system would transmit a POST message with its ONLINE status upon logon, it would also transmit a POST message with its OFFLINE status when logging off, and would naturally also transmit messages for any changes in other status information such as "DO NOT DISTURB" as Zydney suggests.

343. ***Rationale and Motivation to Combine***. The same rationale and motivation to combine the teachings of Zydney and Hethmon for claims 4 and 5 applies equally here, predictably resulting in Zydney's instant voice messaging

system in which the clients report their statuses using POST messages that disclose the claimed connection object messages. As I explained previously, Zydney explicitly suggests HTTP as a transport protocol and incorporates a draft HTTP 1.1 specification. (Zydney, 7:20-8:6, Ex. 1011/1111.) Hethmon likewise describes the HTTP 1.1 protocol and motivates its use, including the POST method for the client to transmit information to the server, as I explained previously. As noted, Hethmon teaches that the POST method can flexibly accommodate information or data that the client wishes to transmit to the server, providing "a way to send substantial information to a server for processing" because "with POST, virtually unlimited entity bodies could be sent in a request message." (Hethmon, p.15.) Hethmon also specifically teaches that the POST method "allows for data submission via HTTP to accomplish various goals, such as database updating or order entry." (*Id*., p.58.) A person of ordinary skill in the art would have readily appreciated that, similar to updating a database, the POST method would be well-suited to update clients' statuses that the central server will "track and maintain" as Zydney describes. (Zydney, 14:7-9.) It would have been equally apparent that the Zydney client could use the POST method to transmit instant voice messages to the server for processing and delivery to the recipient, because Hethmon motivates that POST is "a way to

send substantial information to a server for processing" including "virtually unlimited entity bodies." (Hethmon, p.15.)

344. Accordingly, Zydney in view of Hethmon discloses and renders obvious "wherein the messaging system receives connection object messages from the plurality of instant voice message client systems" as recited in claim 24**[d]**.

(d) "wherein each of the connection object messages includes data representing a state of a logical connection with a given one of the plurality of instant voice message client systems." (Claim 24[d1])

345. For the same reasons discussed above for claim 24**[d]** above, Zydney in view of Hethmon discloses and renders obvious that the client system transmits an HTTP 1.1 POST message that includes data representing a state of the central server's logical connection with that client. In particular, as explained previously, the POST message would contain the "ONLINE" (or "OFFLINE") status of the client's logical connection with the server. It would also contain the status of the HTTP persistent connection specified by the Connection header, which is open unless it contains a "close" value, in which case it is closed. It would also contain the Content-Length field that specifies a state of the TCP socket connection, which is a logical connection that must remain open for a certain data size specified by the Content-Length value and then may be closed, as I explained previously. Claim 24 is therefore obvious.

4. Dependent Claim 25

346. I have reproduced dependent claim 25 below:

> 25. The system according to claim **24**, wherein the connection object messages identifies at least one of a socket, a size of data to be transferred and a priority of the data.

('622, Claim 25.) The POST message that I discussed for claim 24 identifies **a size of data to be transferred** in its Content-Length field. (Hethmon, p.86 (the Content-Length field "is used to specify the byte length of the entity body being sent").) For example, in the POST message illustration I discussed previously, "Content-Length: 23" specifies that 23 bytes are being transmitted. (*Id*., p.78.)

347. The same motivations to combine Zydney and Hethmon that I previously explained for element 24**[d]** apply equally here.

5. Dependent Claim 26

348. I have reproduced dependent claim 26 below:

> **26.** The system according to claim **24**, wherein the communication platform system populates a connection list for the plurality of instant voice message client systems with the data in the connection object messages received from each of the plurality of instant voice message client systems.

('622, Claim 26.)

349. As I explained for claim 3**[c]** and 24**[c]**, Zydney describes that the central server includes a communication platform system, as claimed, that tracks and maintains connection status information for all client systems. Zydney also describes that the central server "will maintain the current list of agents" identifying correspondents for each software agent. (Zydney, 26:10-14.) Based on tracking the connectivity status of all software agents, the system also tracks "who else is on line in the users 'buddy list.'" (*Id*., 30:14-15.)

350. Zydney does not appear to explicitly describe that the status information provided by client systems to the central server, as discussed for claim 24**[d]**, is used to populate a "connection list" in the communication platform system as claim 26 states. However, this feature would have been obvious. As I discussed previously for claim 24**[d]**, Zydney expressly describes that the central server tracks and maintains the status information for all software agents. (*Id*., 14:6-9; 13:12-14.) As I also explained previously for claim 24**[d]**, it would have been obvious that this status information is provided from the client systems in connection object messages, thereby populating the set of status information maintained at the server.

351. While Zydney does not appear to explicitly describe that the status of all software agents is maintained in a "list" that would be populated with the client's status messages (e.g., "ONLINE" and "OFFLINE"), it would have been trivially

obvious to maintain the agents' connection status information in "list" form, thereby disclosing a "**connection list**" as claimed. Specifically, the concept of using a "list" to store, organize and convey information was well within the knowledge of persons of ordinary skill in the art. Introductory computer science courses in universities, for example, typically teach a variety of techniques for organizing information into lists, such as arrays, queues, stacks, trees, graphs, linked lists, and other types of data structures. In many universities these computer science data structures are taught in a course called "Introduction to data structures" or a similar course. For example, since at least the 1970s, well-known programming languages (such as C) provided built-in features for enabling programmers to create and maintain lists. (*See also* **Ex. 1012/1112**, Microsoft Computer Dictionary (1997), at p.30 (defining "array" as "a list of data values, all of the same type… Arrays are part of the fundamentals of data structures, which, in turn, are a major fundamental of computer programming.").)[16]

[16] For example, an array can be used to store a group of elements such as integer values, strings, or more complex data structures (such as structures or "structs," each of which can contain multiple different types of data). Arrays allow computer programs to organize data so related values can be quickly identified, and arrays are

A person of ordinary skill in the art would have understood that storing the connection status information for each software agent in a "list" (such as an array), thereby disclosing a "**connection list**" as claimed, was one of a finite number of well-known and predictable techniques for organizing status information and making it available to the client. The choice of a "list" would have been particularly obvious here considering that, as noted, Zydney discloses that its server already maintains the identity of software agents in "lists." (Zydney, 26:10-12, 30:14-15.) This would have made it plainly obvious to use a connection status "list" to track and maintain the status information for all software agents (i.e., a connection list), which would be populated using the status information in the status messages received from clients. Claim 26 is therefore obvious.

F. Zydney in View of Shinder and Microsoft (1991) and Moghe Renders Obvious Claim 12

1. Dependent Claim 12

352. I have reproduced dependent claim 12 below:

> 12. The system according to claim **3**, wherein the communication platform system updates the connection information for each of the instant voice message client systems by periodically transmitting a connection status request to the given one of the

---

one of the most elementary storage techniques taught in computer science courses.

plurality of instant voice message client systems.

('622, Claim 12.) As I explained with regard to claim 3**[c]**, Zydney discloses **"the communication platform system**" within the central server that will "track and maintain the status of all software agents" (*i.e.*, the status of each instant voice messaging client system). (Zydney, 14:6-9.)

353. Zydney does not appear to explicitly describe the underlying details of how the central server tracks and maintains the status of all software agents. Zydney describes, for example, that the client system will notify the central server of its status when it initially logs on. (*Id*., 14:2-4.) But Zydney does not explicitly describe how the central server becomes aware of other changes in the client system's status, including when a client computer system unintentionally becomes disconnected, *e.g.*, unexpectedly loses its power or wireless connection or its operating system suddenly "crashes."[17] However, this is not a significant omission. A person of ordinary skill in the art would have been familiar with several well-known ways of

[17] I note that Zydney mentions various versions of the Microsoft Windows operating system that could be used for a client PC system. (*Id*., 12:2-3, 12:12-13 (noting Windows 95, 98, 2000, and NT).) Those versions of Windows were well-known to be susceptible to crashes.

updating the connectivity status that would have been obvious as a matter of implementation design choice. One well-known technique was **polling**, where one system periodically polls other systems (*e.g.*, periodically requests the status from the other systems) to determine and update the status of each system. For example, **Microsoft (1991) [Ex. 1018/1118]** provides the following definition of "polling" (also known as "autopolling"):

> **Autopolling** Also called polling. The process of periodically determining the status of each device in a set so that the active program can process events generated through each device. The process can be used to determine the status of a range of events such as whether a key or a mouse button was pressed or whether new data is available at a serial port. Autopolling can be compared with event-driven processing, in which a low-level routine in the operating system alerts a program or routine to an event occurring in a device with an interrupt or message, rather than requiring the program to check each device in turn.

(Microsoft (1991), pp. 26-27 (boldface in original).)

354. The concept of "polling" is a universally-known concept that cuts across numerous areas of device technology including computer networking, as reflected in Microsoft (1991). (*See id*., p.272 (defining "polling cycle" as "[t]he time and sequence required for a program to poll each of its devices or network nodes. *See also* autopolling.") (italics in original, underlining added). In the context of

computer networking, "polling" often involves a first system periodically sending a status request to a second system at some specific interval in order to check on the status of the second system. As Microsoft (1991) describes, polling is the process of "periodically determining the status of each device in a set," which involves a process to "check each device in turn." (*Id*.)

355. This technique is further described in **Moghe**, which confirms that the concept of "polling" applies to determining the status of network-connected devices and resources. The Background of Moghe explains that "polling" provides a means for requesting the status of other devices or resources on a network:

> Typically one host on the network is assigned the task of network manager ("NM") **10**, running appropriate software, while the remaining hosts and resources are identified as agents. The manager **10** will periodically request information from the agents using one of a variety of protocols, e.g. Simple Network Manager Protocol ("SNMP") at the application layer, or Packet Internet Groper ("PING") at the IP layer, and expect a response from each agent using the same protocol. This process is referred to as "polling."

(Moghe, 1:14-22.) Moghe further explains that "[e]fficient polling is becoming increasingly important with new bandwidth-intensive applications such as conferencing and web-push applications." (*Id.*, 1:23-24.)

356. ***Rationale and Motivation to Combine.*** It would have been obvious to adapt well-known "polling" techniques, as described in Microsoft (1991) and Moghe, to the system of Zydney. This combination would have predictably resulted in the instant voice messaging system of Zydney in which the system of the central server responsible for tracking connection statuses (disclosing the claimed "communication platform system," as I discussed above for claim 3**[c]**) periodically transmits a connection status request to the software agent on each client inquiring about its current status, in order to update the system's connection information. A person of ordinary skill in the art would have found this to be a trivial combination.

357. To begin with, the use of "polling" to obtain status information from other devices would have been basic and elementary knowledge to a person of ordinary skill in the art; so much so that citation to Microsoft (1991) was arguably not even warranted. The definition from Microsoft (1991) above, in fact, contrasts polling with "event-processing," reflecting a well-known dichotomy of techniques for obtaining status information from other devices. As noted previously, Microsoft (1991) describes polling as "requiring the program to check each device in turn," *e.g.*, by issuing a request to each device, whereas event-processing involves an alert or message sent upon detection of an event. (Microsoft (1991), pp. 26-27.) Adapting the central server in Zydney to use polling by periodically issuing "connection status

requests" to the client would have been recognized as one of the two clearly identified, predictable solutions for obtaining the status information from the client, and a skilled artisan would have had every expectation of success.

358. Moghe further confirms that "polling" is also used to ascertain the status of networked devices, and that "[e]fficient polling is becoming increasingly important with new bandwidth-intensive applications such as conferencing and web-push applications." (Moghe, 1:23-25.) A person of ordinary skill in the art would have recognized voice instant messaging as another example of a potentially "bandwidth-intensive" application, and would have thus been motivated to adapt Zydney to perform periodic polling.

359. A person of ordinary skill in the art would have understood that, in choosing between a "polling" and an "event-processing" model, each model has potential advantages and disadvantages that are well-known in the art. In the context of client-server based systems such as Zydney, a person of ordinary skill in the art would have appreciated the potential advantages of using the polling model. To begin with, polling would have made implementation significantly easier on the client side. The client would have required nothing more than basic programming to detect the "connection status request" from the central server and respond with its current status information. The server would have merely needed similarly simple

logic to periodically transmit the request and process the response. Polling would also have been useful for Zydney's communication platform system to detect unexpected losses of client connectivity that the client is unable to actively report, such as a sudden loss of power or network connectivity. In such a circumstance, the communication platform system would simply receive no response to one or more connection status requests and could accordingly update the client's status to "offline." By contrast, to track and maintain clients' connectivity statuses in an event-processing model, the client and server would need to include more complex logic because the updates to the status information would be provided only when a certain specified change is detected, and thus, would be potentially more sporadic or unpredictable. For example, using an event-processing model, a suddenly-disconnected client might not be able to report its disconnection, leaving the server unaware of the disconnection.

360. A person of ordinary skill in the art also would have found it obvious that the same system within the central server that discloses the **communication platform system**, as claimed, would issue the connection status requests. As I discussed previously for claim 3**[c]**, Zydney discloses that the central server contains a system that tracks and maintains the status of all software agents, including the notification server and other components. As noted previously, Zydney describes

that the notification server communicates with the software agents, including to report agents' on-line statuses. (Zydney, 25:4-7 (describing that software agent "has notified other software agents via the notification server that they are on-line"), 31:13-15 ("The software agent will send a copy of the currently logged on Internet address to the notification server for purposes of notifying other software agents of its status and receiving messages."), 24:15-16 ("Software agents will gain access to the system through the log on process which interfaces with the notification server.").) A person of ordinary skill in the art would have found it obvious that the same system in the central server responsible for tracking and maintaining the software agents' statuses would also issue the connection status requests used to track and maintain the statuses. A person of ordinary skill in the art also would have found it obvious that the notification server could issue the requests, given that Zydney describes that the notification server communicates with the software agents regarding their on-line statuses and does not describe that a different one of the components of the central server would perform this type of status notification and tracking function.

361. A person of ordinary skill in the art would have understood that the polling and event-processing approaches have tradeoffs that, depending on the implementation, may militate in favor of one approach over the other (or a hybrid

approach combining aspects of both). In fact, these two approaches were so well established that deciding to adapt the central server of Zydney to implement either one would have been entirely predictable and obvious.

VII. ENABLEMENT OF THE PRIOR ART

362. I am informed that in an *inter partes* review, the petitioning party does not have a burden to show that the prior art is enabling. Nevertheless, in my opinion, the Zydney, Shinder, Clark, Appelman, Hethmon, and Microsoft (1991) references provide sufficient detail to enable a person of ordinary skill in the art to practice the limitations of the claims to which they apply without undue experimentation.

363. As I have explained in **Part IV** above, the technological underpinnings of the challenged '622 patent claims were firmly in place well before December 2003. As the '622 patent acknowledges, instant text messaging and instant voice messaging were well-known by December 2003. ('622, 2:22-46.) Software clients such as AIM provided both instant text messaging and computer-to-computer voice communications over the Internet. (Bogard, 1:25-48.)

364. Each reference I have cited pre-dates the '622 patent, and those references themselves treat instant messaging and buddy lists, and instant voice messaging over networks such as the Internet, as firmly established in the prior art. Zydney also explains that instant messages can be encoded using well-known

standard encoding techniques such as MIME, which were well-known in the art. As I explained above, a person of ordinary skill in the art would have been motivated to combine their teachings and could have done so, due to the maturity of those technologies.

365. In short, by December 2003, each aspect of the disclosures that I have cited was already well-known and the subject of extensive public documentation. A person of ordinary skill in the art would not have required disclosures any more detailed than the disclosures in the prior art to apply the prior art teachings in the manner described in this Declaration.

VIII. CONCLUSION

366. In signing this Declaration, I recognize that the Declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in this proceeding. If required, I will appear for cross-examination at the appropriate time. I reserve the right to offer opinions relevant to the invalidity of the '622 patent claims at issue and/or offer testimony in support of this Declaration.

367. I hereby 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 further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Dated: June 22, 2017

Respectfully submitted,

Tal Lavian
June, 22-2017

Tal Lavian, Ph.D.
Berkeley, California

EXHIBIT A

Tal Lavian, Ph.D.

http://telecommnet.com
http://cs.berkeley.edu/~tlavian
tlavian@telecommnet.com

1640 Mariani Dr.
Sunnyvale, CA 94087
(408)-209-9112

Research and Consulting:Telecommunications,Network Communications, and Mobile Wireless Technologies

Scientist, educator, and technologist with over 25years of experience; co-author on over 25 scientific publications, journal articles, and peer-reviewed papers; named inventor on over 100 issued and filed patents; industry fellow and lecturer at UC Berkeley Engineering–Center for Entrepreneurship and Technology (CET)

EDUCATION

- **Ph.D.**, Computer Science specializing in networking and communications, UC Berkeley
- **M.Sc.**, Electrical Engineering, Tel Aviv University
- **B.Sc.**, Mathematics and Computer Science, Tel Aviv University

EXPERTISE

Network communications, telecommunications, Internet protocols, and mobile wireless:

- **Communication networks:** Internet protocols; TCP/IP suite; TCP; UDP; IP; VoIP; Ethernet; network protocols; network software applications; data link, network, and transport layers (L2, L3, L4)
- **Internet software:** Internet software applications; distributed computing; cloud computing; Web applications; FTP; HTTP; Java; client server; file transfer; multicast; streaming media
- **Routing/switching:** LAN; WAN; VPN; routing protocols; RIP; BGP; MPLS; OSPF; IS-IS; DNS; QoS; switching; packet switching; network infrastructure; network communication architectures
- **Mobile wireless:** wireless LAN; 802.11; cellular systems; mobile devices; smartphone technologies

ACCOMPLISHMENTS

- Selected as principal investigator for three US Department of Defense (DARPA) projects
- Directed research project on networking computation for the US Air Force Research Lab (AFRL)
- Led and developed the first network resourcescheduling service for grid computing
- Administered wireless research project for an undisclosed US federal agency
- Managed and engineered the first demonstrated transatlantic dynamic allocation of 10Gbs Lambdas as a grid service
- Spearheaded the development of the first demonstrated wire-speed active network on commercial hardware
- Invented over 100 patents; over 50 prosecuted *pro se* in front of the USPTO
- Created and chaired Nortel Networks' EDN Patent Committee

PROFESSIONAL EXPERIENCE

University of California, Berkeley, Berkeley, California 2000-Present
Berkeley Industry Fellow, Lecturer, Visiting Scientist, Ph.D. Candidate, Nortel's Scientist Liaison
Some positions and projects were concurrent, others sequential

- Serves as an industry fellow and lecturer at the Center for Entrepreneurship and Technology (CET).
- Studied network services, telecommunication systems and software, communications infrastructure, and data centers
- Developed long-term technology for the enterprise market, integrating communication and computing technologies
- Conducted research projects in data centers (RAD Labs), telecommunication infrastructure (SAHARA), and wireless systems (ICEBERG)
- Acted as scientific liaison between Nortel Research Lab and UC Berkeley, providing tangible value in advanced technologies
- Earned a Ph.D. in Computer Science with a specialization in communications and networking

TelecommNet Consulting, Inc.(Innovations-IP) Sunnyvale, California 2006-Present
Principal Scientist

- Consults in the areas of network communications, telecommunications, Internet protocols, and smartphone mobile wireless devices
- Provides architecture and system consultation for projects relating to computer networks, mobile wireless devices, and Internet web technologies
- Acts as an expert witness in network communications patent infringement lawsuits

VisuMenu, Inc., Sunnyvale, California 2010-Present
Co- Founder and Chief Technology Officer (CTO)

- Designs and develops architecture and system of visual IVR technologies for smartphones and wireless mobile devices in the area of network communications
- Designs crawler/spider system for IVR / PBX using Asterisk, SIP, and VoIP
- Deploys the system as cloud networking and cloud computing utilizing Amazon Web Services

Ixia, Santa Clara, California 2008 - 2008
Communications Consultant

- Researched and developed advanced network communications testing technologies:
  • IxNetwork/IxN2X — tested IP routing and switching devices and broadband access equipment. Provided traffic generation and emulation for the full range of protocols: routing, MPLS, layer 2/3 VPNs, carrier Ethernet, broadband access, and data center bridging
  • IxLoad — quickly and accurately modeled high-volume video, data, and voice subscribers and servers to test real-world performance of multiservice delivery and security platforms
  • IxCatapult — emulated a broad range of wireless access and core protocols to test wireless components and systems that, when combined with IxLoad, provides an end-to-end solution for testing wireless service quality
  • IxVeriWave — employed a client-centric model to test Wi-Fi and wireless LAN networks by generating repeatable large-scale, real-world test scenarios that are virtually impossible to create by any other means

• Test automation — provided simple, comprehensive lab automation to help test engineering teams create, organize, catalog, and schedule execution of tests

Nortel Networks, Santa Clara, California 1996 - 2007

Originally employed by Bay Networks, which was acquired by Nortel Networks

Principal Scientist, Principal Architect, Principal Engineer, Senior Software Engineer

- Held scientific and research roles at Nortel Labs, Bay Architecture Labs, and in the office of the CTO

Principal Investigator for US Department of Defense (DARPA) Projects

- Conceived, proposed, and completed three research projects: active networks, DWDM-RAM, and a networking computation project for Air Force Research Lab (AFRL)
- Led a wireless research project for an undisclosed US federal agency

Academic and Industrial Researcher

- Analyzed new technologies to reduce risks associated with R&D investment
- Spearheaded research collaboration with leading universities and professors at UC Berkeley, Northwestern University, University of Amsterdam, and University of Technology, Sydney
- Evaluated competitive products relative to Nortel's products and technology
- Proactively identified prospective business ideas, which led to new networking products
- Predicted technological trends through researching the technological horizon and academic sphere
- Designed software for switches, routers, and network communications devices
- Developed systems and architectures for switches, routers, and network management
- Researched and developed the following projects:
  - Data-Center Communications: network and server orchestration 2006-2007
  - DRAC: SOA-facilitated L1/L2/L3 network dynamic controller 2003-2007
  - Omega: classified wireless project for undisclosed US Federal Agency 2006-2006
  - Open platform: project for the US Air Force Research Laboratory (AFRL) 2005-2005
  - Network resource orchestration for Web services workflows 2004-2005
  - Proxy study between Web/grids services and network services 2004-2004
  - Streaming content replication: real-time A/V media multicast at edge 2003-2004
  - DWDM-RAM: US DARPA-funded program on agile optical transport 2003-2004
  - Packet capturing and forwarding service on IP and Ethernet traffic 2002-2003
  - CO2: content-aware agile networking 2001-2003
  - Active networks: US DARPA-funded research program 1999-2002
  - ORE: programmable network service platform 1998-2002
  - JVM platform: Java on network devices 1998-2001
  - Web-based device management: network device management 1996-1997

Technology Innovator and Patent Leader

- Created and chaired Nortel Networks' EDN Patent Committee
- Facilitated continuous stream of innovative ideas and their conversion into intellectual property rights
- Developed intellectual property assets through invention and analysis of existing technology portfolios

Aptel Communications, Netanya, Israel 1994-1995

Software Engineer, Team Leader

Start-up company focused on mobile wireless CDMA spread spectrum PCN/PCS

- Developed a mobile wireless device using an unlicensed band [Direct Sequence Spread Spectrum (DSSS)]
- Designed and managed a personal communication network (PCN) and personal communication system (PCS), which are the precursors of short text messages (SMS)
- Designed and developed network communications software products (mainly in C/C++)
- Brought a two-way paging product from concept to development

Scitex Ltd.,Herzeliya, Israel 1990-1993

Software Engineer, Team Leader

Software and hardware company acquired by Hewlett Packard (HP)

- Developed system and network communications (mainly in C/C++)
- Invented Parallel SIMD Architecture
- Participated in the Technology Innovation group

Shalev, Ramat-HaSharon, Israel 1987-1990

Start-up company

Software Engineer

- Developed real-time software and algorithms (mainly in C/C++ and Pascal)

PROFESSIONAL ASSOCIATIONS

- IEEE senior member
- IEEE CNSV co-chair, Intellectual Property SIG (2013)
- President Next Step Toastmasters (an advanced TM club in the Silicon Valley) (2013-2014)
- Technical co-chair, IEEE Hot Interconnects 2005 at Stanford University
- Member, IEEE Communications Society (COMMSOC)
- Member, IEEE Computer Society
- Member, IEEE Systems, Man, and Cybernetics Society
- Member, IEEE-USA Intellectual Property Committee
- Member, ACM, ACM Special Interest Group on Data Communication (SIGCOM)
- Member, ACM Special Interest Group on Hypertext, Hypermedia, and Web (SIGWEB)
- Member, IEEE Consultants' Network (CNSV)
- Global Member, Internet Society (ISOC)
- President Java Users Group – Silicon Valley Mountain View, CA,1999-2000
- Toastmasters International

ADVISORY BOARDS

- Quixey –search engine for wireless mobile apps
- Mytopia – mobile social games
- iLeverage – Israeli Innovations

PROFESSIONAL AWARDS

- Top Talent Award – Nortel
- Top Inventors Award – Nortel EDN
- Certified IEEE-WCET - Wireless Communications Engineering Technologies
- Toastmasters International - Competent Communicator (twice)
- Toastmasters International - Advanced Communicator Bronze

Patents and Publications

(Not an exhaustive list)

Patents Issued

US 9,184,989	Grid proxy architecture for network resources	Link
US 9,083,728	Systems and methods to support sharing and exchanging in a network	Link
US 9,021,130	Photonic line sharing for high-speed routers	Link
US 9,001,819	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,949,846	Time-value curves to provide dynamic QoS for time sensitive file transfers	Link
US 8,929,517	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,903,073	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,898,274	Grid proxy architecture for network resources	Link
US 8,880,120	Device and method for providing enhanced telephony	Link
US 8,879,703	System method and device for providing tailored services when call is on-hold	Link
US 8,879,698	Device and method for providing enhanced telephony	Link
US 8,867,708	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,787,536	Systems and methods for communicating with an interactive voice response system	Link
US 8,782,230	Method and apparatus for using a command design pattern to access and configure network elements	Link
US 8,762,963	Translation of programming code	Link
US 8,762,962	Methods and apparatus for automatic translation of a computer program language code	Link
US 8,745,573	Platform-independent application development framework	Link
US 8,731,148	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,688,796	Rating system for determining whether to accept or reject objection raised by user in social network	Link
US 8,619,793	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link
US 8,572,303	Portable universal communication device	Link
US 8,553,859	Device and method for providing enhanced telephony	Link

US 8,548,131	Systems and methods for communicating with an interactive voice response system	Link
US 8,537,989	Device and method for providing enhanced telephony	Link
US 8,341,257	Grid proxy architecture for network resources	Link
US 8,161,139	Method and apparatus for intelligent management of a network element	Link
US 8,146,090	Time-value curves to provide dynamic QoS for time sensitive file transfer	Link
US 8,078,708	Grid proxy architecture for network resources	Link
US 7,944,827	Content-aware dynamic network resource allocation	Link
US 7,860,999	Distributed computation in network devices	Link
US 7,734,748	Method and apparatus for intelligent management of a network element	Link
US 7,710,871	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link
US 7,580,349	Content-aware dynamic network resource allocation	Link
US 7,433,941	Method and apparatus for accessing network information on a network device	Link
US 7,359,993	Method and apparatus for interfacing external resources with a network element	Link
US 7,313,608	Method and apparatus for using documents written in a markup language to access and configure network elements	Link
US 7,260,621	Object-oriented network management interface	Link
US 7,237,012	Method and apparatus for classifying Java remote method invocation transport traffic	Link
US 7,127,526	Method and apparatus for dynamically loading and managing software services on a network device	Link
US 7,047,536	Method and apparatus for classifying remote procedure call transport traffic	Link
US 7,039,724	Programmable command-line interface API for managing operation of a network device	Link
US 6,976,054	Method and system for accessing low-level resources in a network device	Link
US 6,970,943	Routing architecture including a compute plane configured for high-speed processing of packets to provide application layer support	Link
US 6,950,932	Security association mediator for Java-enabled devices	Link
US 6,850,989	Method and apparatus for automatically configuring a network switch	Link

US 6,845,397	Interface method and system for accessing inner layers of a network protocol	Link
US 6,842,781	Download and processing of a network management application on a network device	Link
US 6,772,205	Executing applications on a target network device using a proxy network device	Link
US 6,564,325	Method of and apparatus for providing multi-level security access to system	Link
US 6,175,868	Method and apparatus for automatically configuring a network switch	Link
US 6,170,015	Network apparatus with Java co-processor	Link
US 8,687,777	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,681,951	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,625,756	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,594,280	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,548,135	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,406,388	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,345,835	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,223,931	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,160,215	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,155,280	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,054,952	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,000,454	Systems and methods for visual presentation and selection of IVR menu	Link
EP 1,905,211	Technique for authenticating network users	Link
EP 1,142,213	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link
EP 1,671,460	Method and apparatus for scheduling resources on a switched underlay network	Link
CA 2,358,525	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link

Patent Applications Published and Pending
(Not an exhaustive list)

US 20150058490	Grid Proxy Architecture for Network Resources	Link
US 20150010136	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20140379784	Method and Apparatus for Using a Command Design Pattern to Access and Configure Network Elements	Link
US 20140105025	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
US 20140105012	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
US 20140012991	Grid Proxy Architecture for Network Resources	Link
US 20130080898	Systems and Methods for Electronic Communications	Link
US 20130022191	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20130022183	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20130022181	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20120180059	Time-Value Curves to Provide Dynamic QOS for Time Sensitive File Transfers	Link
US 20120063574	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20110225330	Portable Universal Communication Device	Link
US 20100220616	Optimizing Network Connections	Link
US 20100217854	Method and Apparatus for Intelligent Management of a Network Element	Link
US 20100146492	Translation of Programming Code	Link
US 20100146112	Efficient Communication Techniques	Link
US 20100146111	Efficient Communication in a Network	Link
US 20090313613	Methods and Apparatus for Automatic Translation of a Computer Program Language Code	Link
US 20090313004	Platform-Independent Application Development Framework	Link
US 20090279562	Content-aware dynamic network resource allocation	Link
US 20080040630	Time-Value Curves to Provide Dynamic QoS for Time Sensitive File	Link

	Transfers	
US 20070169171	Technique for authenticating network users	Link
US 20060123481	Method and apparatus for network immunization	Link
US 20060075042	Extensible Resource Messaging Between User Applications and Network Elements in a Communication Network	Link
US 20050083960	Method and Apparatus for Transporting Parcels of Data Using Network Elements with Network Element Storage	Link
US 20050076339	Method and Apparatus for Automated Negotiation for Resources on a Switched Underlay Network	Link
US 20050076336	Method and Apparatus for Scheduling Resources on a Switched Underlay Network	Link
US 20050076173	Method And Apparatus for Preconditioning Data to Be Transferred on a Switched Underlay Network	Link
US 20050076099	Method and Apparatus for Live Streaming Media Replication in a Communication Network	Link
US 20050074529	Method and apparatus for transporting visualization information on a switched underlay network	Link
US 20040076161	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
US 20020021701	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
WO 2006/063052	Method and apparatus for network immunization	Link
WO 2007/008976	Technique for authenticating network users	Link
WO2000/0054460	Method and apparatus for accessing network information on a network device	Link
US 20140156556	Time-variant rating system and method thereof	Link
US 20140156758	Reliable rating system and method thereof	Link

Publications

(Not an exhaustive list)

- "R&D Models for Advanced Development & Corporate Research" Understanding Six Models of Advanced R&D - Ikhlaq Sidhu, Tal Lavian, Victoria Howell - University of California, Berkeley. Accepted paper for 2015 ASEE Annual Conference and Exposition- June 2015
- "Communications Architecture in Support of Grid Computing",Tal Lavian,Scholar's Press 2013ISBN 978-3-639-51098-0.
- "Applications Drive Secure Lightpath Creation across Heterogeneous Domains, Feature Topic Optical Control Planes for Grid Networks: Opportunities, Challenges and the Vision." Gommans L.; Van Oudenaarde B.; Dijkstra F.; De Laat C.; Lavian T.; Monga I.; Taal A.; Travostino F.; Wan A.; IEEE Communications Magazine, vol. 44, no. 3, March 2006, pp. 100-106.
- *Lambda Data Grid: Communications Architecture in Support of Grid Computing*. Tal I. Lavian, Randy H. Katz; Doctoral Thesis, University of California at Berkeley. January 2006.
- "Information Switching Networks." Hoang D.B.; T. Lavian;The 4th Workshop on the Internet, *Telecommunications and Signal Processing, WITSP2005*, December 19-21, 2005, Sunshine Coast, Australia.
- "Impact of Grid Computing on Network Operators and HW Vendors." Allcock B.; Arnaud B.; Lavian T.; Papadopoulos P.B.; Hasan M.Z.; Kaplow W.; *IEEE Hot Interconnects at Stanford University 2005*, pp.89-90.
- *DWDM-RAM: A Data Intensive Grid Service Architecture Enabled by Dynamic Optical Networks*. Lavian T.; Mambretti J.; Cutrell D.; Cohen H.J; Merrill S.; Durairaj R.; Daspit P.; Monga I.; Naiksatam S.; Figueira S.; Gutierrez D.; Hoang D.B., Travostino F.; *CCGRID 2004*, pp. 762-764.
- *DWDM-RAM: An Architecture for Data Intensive Service Enabled by Next Generation Dynamic Optical Networks*. Hoang D.B.; Cohen H.; Cutrell D.; Figueira S.; Lavian T.; Mambretti J.; Monga I.; Naiksatam S.; Travostino F.; Proceedings IEEE Globecom 2004, Workshop on High-Performance Global Grid Networks, Houston, 29 Nov. to 3 Dec. 2004, pp.400-409.
- *Implementation of a Quality of Service Feedback Control Loop on Programmable Routers*. Nguyen C.; Hoang D.B.; Zhao, I.L.; Lavian, T.; Proceedings, 12th IEEE International Conference on Networks 2004. (ICON 2004) Singapore, Volume 2, 16-19 Nov. 2004, pp.578-582.
- *A Platform for Large-Scale Grid Data Service on Dynamic High-Performance Networks*. Lavian T.; Hoang D.B.; Mambretti J.; Figueira S.; Naiksatam S.; Kaushil N.; Monga I.; Durairaj R.; Cutrell D.; Merrill S.; Cohen H.; Daspit P.; Travostino F; GridNets 2004, San Jose, CA., October 2004.
- *DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks*. Figueira S.; Naiksatam S.; Cohen H.; Cutrell D.; Daspit, P.; Gutierrez D.; Hoang D. B.; Lavian T.; Mambretti J.; Merrill S.; Travostino F; Proceedings, 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, USA, April 2004, pp. 707-714.
- *DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks*. Figueira S.; Naiksatam S.; Cohen H.; Cutrell D.; Gutierrez D.; Hoang D.B.; Lavian T.; Mambretti J.; Merrill S.; Travostino F.; 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, USA, April 2004.
- *An Extensible, Programmable, Commercial-Grade Platform for Internet Service Architecture*. Lavian T.; Hoang D.B.; Travostino F.; Wang P.Y.; Subramanian S.; Monga I.; IEEE Transactions on Systems, Man, and Cybernetics on Technologies Promoting Computational

Intelligence, Openness and Programmability in Networks and Internet Services Volume 34, Issue 1, Feb. 2004, pp.58-68.

- *DWDM-RAM: An Architecture for Data Intensive Service Enabled by Next Generation Dynamic Optical Networks*. Lavian T.; Cutrell D.; Mambretti J.; Weinberger J.; Gutierrez D.; Naiksatam S.; Figueira S.; Hoang D. B.; Supercomputing Conference, SC2003 Igniting Innovation, Phoenix, November 2003.
- *Edge Device Multi-Unicasting for Video Streaming*. Lavian T.; Wang P.; Durairaj R.; Hoang D.; Travostino F.; Telecommunications, 2003. ICT 2003. 10th International Conference on Telecommunications, Tahiti, Volume 2, 23 Feb.-1 March, 2003 pp. 1441-1447.
- The SAHARA Model for Service Composition Across Multiple Providers. Raman B.; Agarwal S.; Chen Y.; Caesar M.; Cui W.; Lai K.; Lavian T.; Machiraju S.; Mao Z. M.; Porter G.; Roscoe T.; Subramanian L.; Suzuki T.; Zhuang S.; Joseph A. D.; Katz Y.H.; Stoica I.; Proceedings of the First International Conference on Pervasive Computing. ACM Pervasive 2002, pp. 1-14.
- *Enabling Active Flow Manipulation in Silicon-Based Network Forwarding Engines*. Lavian T.; Wang P.; Travostino F.; Subramanian S.; Duraraj R.; Hoang D.B.; Sethaput V.; Culler D.; Proceeding of the Active Networks Conference and Exposition, 2002.(DANCE) 29-30 May 2002, pp. 65-76.
- *Practical Active Network Services within Content-Aware Gateways*. Subramanian S.; Wang P.; Durairaj R.; Rasimas J.; Travostino F.; Lavian T.; Hoang D.B.; Proceeding of the DARPA Active Networks Conference and Exposition, 2002.(DANCE) 29-30 May 2002, pp. 344-354.
- *Active Networking on a Programmable Network Platform*. Wang P.Y.; Lavian T.; Duncan R.; Jaeger R.; Fourth IEEE Conference on Open Architectures and Network Programming (OPENARCH), Anchorage, April 2002.
- *Intelligent Network Services through Active Flow Manipulation*. Lavian T.; Wang P.; Travostino F.; Subramanian S.; Hoang D.B.; Sethaput V.; IEEE Intelligent Networks 2001 Workshop (IN2001), Boston, May 2001.
- *Intelligent Network Services through Active Flow Manipulation*. Lavian T.; Wang P.; Travostino F.; Subramanian S.; Hoang D.B.; Sethaput V.; Intelligent Network Workshop, 2001 IEEE 6-9 May 2001, pp.73 -82.
- *Enabling Active Flow Manipulation in Silicon-based Network Forwarding Engine*. Lavian, T.; Wang, P.; Travostino, F.; Subramanian S.; Hoang D.B.; Sethaput V.; Culler D.; Journal of Communications and Networks, March 2001, pp.78-87.
- *Active Networking on a Programmable Networking Platform*. Lavian T.; Wang P.Y.; IEEE Open Architectures and Network Programming, 2001, pp. 95-103.
- *Enabling Active Networks Services on a Gigabit Routing Switch*. Wang P.; Jaeger R.; Duncan R.; Lavian T.; Travostino F.; 2nd Workshop on Active Middleware Services, 2000.
- *Dynamic Classification in Silicon-Based Forwarding Engine Environments*. Jaeger R.; Duncan R.; Travostino F.; Lavian T.; Hollingsworth J.; Selected Papers. 10th IEEE Workshop on Metropolitan Area and Local Networks, 1999. 21-24 Nov. 1999, pp.103-109.
- *Open Programmable Architecture for Java-Enabled Network Devices*. Lavian, T.; Jaeger, R. F.; Hollingsworth, J. K.; IEEE Hot Interconnects Stanford University, August 1999, pp. 265-277.
- *Open Java SNMP MIB API*. Rob Duncan, Tal Lavian, Roy Lee, Jason Zhou, Bay Architecture Lab Technical Report TR98-038, December 1998.
- *Java-Based Open Service Interface Architecture*. Lavian T.; Lau S.; BAL TR98-010 Bay Architecture Lab Technical Report, March 1998.

- *Parallel SIMD Architecture for Color Image Processing*. Lavian T. Tel – Aviv University, Tel – Aviv, Israel, November 1995.
- *Grid Network Services, Draft-ggf-ghpn-netservices-1.0*. George Clapp, Tiziana Ferrari, Doan B. Hoang, Gigi Karmous-Edwards, Tal Lavian, Mark J. Leese, Paul Mealor, Inder Monga, Volker Sander, Franco Travostino, Global Grid Forum(GGF).
- *Project DRAC: Creating an applications-aware network*.Travostino F.; Keates R.; Lavian T.; Monga I.; Schofield B.; Nortel Technical Journal, February 2005, pp. 23-26.
- *Optical Network Infrastructure for Grid, Draft-ggf-ghpn-opticalnets-1*. Dimitra Simeonidou, Reza Nejabati, Bill St. Arnaud, Micah Beck, Peter Clarke, Doan B. Hoang, David Hutchison, Gigi Karmous-Edwards, Tal Lavian, Jason Leigh, Joe Mambretti, Volker Sander, John Strand, Franco Travostino, Global Grid Forum(GGF) GHPN Standard GFD-I.036 August 2004.
- *Popeye - Using Fine-grained Network Access Control to Support Mobile Users and Protect Intranet Hosts*. Mike Chen, Barbara Hohlt, Tal Lavian, December 2000.

Presentations and Talks

(Not an exhaustive list)

- Lambda Data Grid: An Agile Optical Platform for Grid Computing and Data-intensive Applications.
- Web Services and OGSA
- WINER Workflow Integrated Network Resource Orchestration.
- Technology & Society
- Abundant Bandwidth and how it affects us?
- Active Content Networking(ACN)
- DWDM-RAM:Enabling Grid Services with Dynamic Optical Networks
- Application-engaged Dynamic Orchestration of Optical Network Resources
- A Platform for Data Intensive Services Enabled by Next Generation Dynamic Optical Networks
- Optical Networks
- Grid Optical Network Service Architecture for Data Intensive Applications
- Optical Networking & DWDM
- OptiCal Inc.
- OptiCal & LUMOS Networks
- Optical Networking Services
- Business Models for Dynamically Provisioned Optical Networks
- Business Model Concepts for Dynamically Provisioned Optical Networks
- Optical Networks Infrastructure
- Research Challenges in agile optical networks
- Services and Applications' infrastructure for agile optical networks
- Impact on Society
- TeraGrid Communication and Computation
- Unified Device Management via Java-enabled Network Devices
- Active Network Node in Silicon-Based L3 Gigabit Routing Switch
- Active Nets Technology Transfer through High-Performance Network Devices
- Programmable Network Node: Applications
- Open Innovation via Java-enabled Network Devices
- Practical Considerations for Deploying a Java Active Networking Platform
- Open Java-Based Intelligent Agent Architecture for Adaptive Networking Devices

- Java SNMP Oplet
- Open Distributed Networking Intelligence: A New Java Paradigm
- Open Programmability
- Active Networking On A Programmable Networking Platform
- Open Networking through Programmability
- Open Programmable Architecture for Java-enabled Network Devices
- Integrating Active Networking and Commercial-Grade Routing Platforms
- Programmable Network Devices
- To be smart or not to be?