

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
17 August 2006 (17.08.2006)

PCT

(10) International Publication Number  
**WO 2006/086335 A2**

(51) International Patent Classification:  
*H04M 1/64* (2006.01)

(21) International Application Number:

PCT/US2006/004174

(22) International Filing Date: 7 February 2006 (07.02.2006)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

11/053,271	7 February 2005 (07.02.2005)	US
11/053,537	7 February 2005 (07.02.2005)	US
11/053,146	7 February 2005 (07.02.2005)	US
11/053,147	7 February 2005 (07.02.2005)	US
11/053,411	7 February 2005 (07.02.2005)	US
11/053,272	7 February 2005 (07.02.2005)	US
11/053,709	7 February 2005 (07.02.2005)	US
11/053,736	7 February 2005 (07.02.2005)	US
11/053,054	7 February 2005 (07.02.2005)	US
11/053,425	7 February 2005 (07.02.2005)	US
11/053,270	7 February 2005 (07.02.2005)	US
11/053,538	7 February 2005 (07.02.2005)	US
11/053,539	7 February 2005 (07.02.2005)	US
11/053,376	7 February 2005 (07.02.2005)	US

(71) Applicant (for all designated States except US): **ADOMO, INC.** [US/US]; 20401 Stevens Creek Blvd., Cupertino, CA 95014 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **SKAKKEBAEK, Jens** [DK/US]; 46 Elston Court, San Carlos, CA 94070 (US). **FRIFELDT, Heine** [DK/US]; 575 South Rengstorff Ave., Apartment 152, Mountain View, CA 94040 (US). **SHAFFER, Anthony** [US/US]; P.O. Box 7375, Santa Cruz, CA 95061 (US). **FORNEY, David** [US/US]; 4014

Fitzpatrick Way, Santa Clara, CA 95054 (US). **POTZE, Willem, R., B.** [NL/US]; 3110 Membrillo Corte, Santa Clara, CA 95051 (US). **BIRKHAHN, Lutz** [DE/US]; 4325 Renaissance Drive, Apt. 302, San Jose, CA 95134 (US). **VAGHAR, Shahriar** [US/US]; 1036 Calboro Drive, San Jose, CA 95117 (US). **WANG, Yang** [CN/US]; 30746 Knight Court, Union City, CA 94587 (US).

(74) Agents: **WILLMAN, George, A.** et al.; Wilson Sonsini Goodrich & Rosati, 650 Page Mill Road, Palo Alto, CA 94304-1050 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: INTEGRATED MULTI-MEDIA COMMUNICATION SYSTEM

(57) Abstract: An integrated messaging system for performing various types of messaging across different types of networks, including integrated user interfaces and administrator interfaces. Embodiments include a communication server that couples among networks of different types, and an interface module that couples to the communication server. The interface module may be hosted on a messaging server of a network. The interface module pulls various user information from the messaging server, including information relevant to at least the network that includes the messaging server. A cache couples to the communication server and to the interface module to hold information from the communication server and/or the user information pulled from messaging server. The interface module directs a message from the messaging server and/or the cache to at least one device on the networks using the user information.



WO 2006/086335 A2

## INTEGRATED MULTI-MEDIA COMMUNICATION SYSTEM

### TECHNICAL FIELD

5 The disclosure herein relates generally to communication systems, and more particularly to integrated communication and messaging systems.

### BACKGROUND

10 As methods of communication continue to proliferate, enterprises continue to desire integrated systems for handling all aspects of multi-media communication for enterprise users. An enterprise can be any collection of users of communication media having some common purpose, but a typical example is a company with one or more sites and some number of employees who are users of communication media. Communication media include electronic mail ("email") messaging, Short Messaging Service ("SMS") messaging, voice messaging, and more. Users receive and send messages over a variety of wired and wireless networks via a variety of devices, such as desktop computers, wired phones, wireless devices (e.g., phones and personal digital assistants ("PDAs")), and more.

15 Enterprises currently have the ability to centralize and manage email messaging using commercially available groupware that centrally stores information about all of the users and their messages. Enterprises also have the ability to centrally manage traditional voice messaging using a Private Branch Exchange ("PBX"). However, the systems for managing email messaging and the systems for managing voice mail messaging are not at all well integrated. For example, when a new user is added to the enterprise, a system administrator for the enterprise sets up the user in the email system using the groupware application and its set methods, data and protocols. In addition, a different administrator specializing in telephony must set up the user in the voice messaging system using different methods, data and protocols. Voice data and email data are typically stored in separate databases. Both initial user setup and updating user information are complicated by the fact that the email and voice systems are so distinct.

20 The management of and access to the voice mail message information and email information in the enterprise is also complicated by the current lack of integration of the two (voice and email) systems. There are various challenges to be overcome if one were to attempt to integrate the two systems.

### INCORPORATION BY REFERENCE

30 All publications and patent applications mentioned in this specification are herein incorporated by reference to the same extent as if each individual publication or patent application was specifically and individually indicated to be incorporated by reference.

### BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a system that includes an integrated communication system ("ICS"), under an embodiment.

35 Figure 2 is a flow diagram for providing integrated communication processes using the ICS, under an embodiment.

Figure 3 is a block diagram of example information flows in a system that includes the ICS, under an embodiment.

Figure 4 is another flow diagram for providing integrated communication processes using the ICS, under an embodiment.

Figure 5 is a block diagram of an enterprise network system that includes a communication server and Interface Module ("IM") of an ICS, under an embodiment.

Figure 6 is a block diagram of an enterprise network system that includes the ICS, under an embodiment.

Figure 7 is a block diagram that shows interactions between the IM and components of a messaging server ("MSERV") environment, under an embodiment.

Figure 8 is an information flow for routing and accessing voice mail messages via the ICS when the MSERV is in an online state, under an embodiment.

Figure 9 is an alternative information flow for routing and accessing voice mail messages via the ICS when the MSERV is in an online state, under an embodiment.

Figure 10 is an information flow for routing and accessing voice mail messages via the ICS when the MSERV is in an offline state, under an embodiment.

Figure 11 is a block diagram of a system that includes the ICS with a Form-Based User Interface ("FBUI"), under an embodiment.

Figure 12 is a sample FBUI as displayed on a client device, under an embodiment.

Figure 13 is a block diagram of a system that includes multiple sites and multiple components, under an alternative embodiment.

Figure 14 is a block diagram of a system that uses a form-based interface transferred in a first type of message for controlling actions on a second type of message, along with the corresponding information flows, under an embodiment.

Figure 15 is a selection popup the ICS provides during execution of the Play on Phone action, under an embodiment.

Figure 16 is a status popup the ICS provides during execution of the Play on Phone action, under an embodiment.

Figure 17 is another status popup the ICS provides during execution of the Play on Phone action, under an embodiment.

Figure 18 is an information flow in a system that includes an ICS supporting Play on Phone operations, under an embodiment.

Figure 19 is a block diagram of an embodiment in which the enterprise includes multiple MSERVs, each of which communicates with a respective IM of an ICS, under an embodiment.

Figure 20 is a block diagram of an embodiment in which data that is particular to users of MCS, MCS Voice Applications, and Mobile Applications is stored in AD and MSERVs.

Figure 21 is a block diagram of an example of the integration of MCS Voice Applications with enterprise MSERVs, under an embodiment.

Figure 22 is a block diagram of information transfers between the MCS and/or Cache and IM, under an embodiment.

Figure 23 is a flow diagram for providing user information to the MCS from a network enterprise database, under an embodiment.

Figure 24 is an information flow diagram for incremental loading of user information, under an embodiment.

Figure 25 is a block diagram of an enterprise network that includes multiples MCSs coupled to include a "Distributed Caching System" ("DCS"), under an embodiment.

Figure 26 is another block diagram of an MCS having a DCS, under an embodiment.

Figure 27 is a block diagram of network that includes two (2) MCSs and a DCS, under an embodiment.

Figure 28 is an information flow for inter-node communications between Caching Servers of different MCSs, under an embodiment.

5 Figure 29 is a block diagram of a system that includes multiple sites and multiple components, under an alternative embodiment.

Figure 30 is a block diagram of a MCS setup/enable user process of one embodiment.

Figure 31 is a block diagram of Source Configurable Rules of one embodiment.

Figure 32 is a flow diagram showing an MCS user setup/enable process 1200 according to an embodiment.

Figure 33 is a block diagram of an embodiment of updating a VMLIST.

10 Figure 34 is a block diagram of an embodiment including an event listener for mailbox events.

Figure 35 is a block diagram of a communication system, according to an embodiment of the invention.

Figure 36 is a flow diagram of storing and receiving a private voice message, according to an embodiment of the invention.

15 Figure 37 is a block diagram of a communication system with a message and collaboration server, according to an embodiment of the invention.

Figure 38 is a flow diagram of storing and receiving a private voice message using a rights management scheme, according to an embodiment of the invention.

Figure 39 is a block diagram of a communication system with voice access to stored messages, according to an embodiment of the invention.

20 Figure 40 is a flow diagram of retrieving a private voice message from a voice interface, according to an embodiment of the invention.

Figure 41 is a block diagram of a network of communications systems with voicemail networking, according to an embodiment of the invention.

25 Figure 42 is a flow diagram of receiving and sending an email containing a voicemail message, according to an embodiment of the invention.

Figure 43 is a block diagram of two communications systems with voicemail networking and encapsulation and extraction devices, according to an embodiment of the invention.

Figure 44 is a block diagram of communications systems, some with voicemail networking, according to an embodiment of the invention.

30 Figure 45 is a block diagram of communications systems, some with voicemail networking, and a network enabled lookup device, according to an embodiment of the invention.

Figure 46 is a flow diagram of receiving and routing an email containing a voicemail message, according to an embodiment of the invention.

Figure 47 is a block diagram of a diagnostic tool, according to an embodiment of the invention.

35 Figure 48 is a block diagram of a target system with a diagnostic tool, according to an embodiment of the invention.

Figure 49 is a flow diagram of a diagnostic tool, according to an embodiment of the invention.

Figure 50 is a relationship diagram of diagnostic assemblies, according to an embodiment of the invention.

40 Figure 51 is a block diagram of a diagnostic tool that uses methods with parameters, according to an embodiment of the invention.

Figure 52 is a flow diagram of a diagnostic tool that uses methods with parameters, according to an embodiment of the invention.

Figure 53 is a block diagram of a diagnostic tool, according to an embodiment of the invention.

Figure 54 is a block diagram of an embodiment including configuration files and code on an MCS.

Figure 55 is a flow diagram for providing configuration information to a component such as an MCS according to an embodiment.

5 Figure 56 is a flow diagram for upgrading code on a component such as an MCS according to an embodiment.

Figure 57 is a block diagram of an MCS with upgradeable code according to an embodiment of the invention.

10 In the drawings, the same reference numbers identify identical or substantially similar elements or acts. To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the Figure number in which that element is first introduced (e.g., element 110 is first introduced and discussed with respect to Figure 1).

### DETAILED DESCRIPTION

15 Integrated multi-media communication systems and methods are provided below. These communication systems and methods, collectively referred to herein as "integrated communication systems" or "ICS," integrate different types of messaging so that a user of the ICS can access multiple types of messages (e.g., voice mail messages, electronic mail, email messages, instant messaging messages, SMS (Short Messaging System) messages, MMS (Multimedia Messaging System) messages, etc. with a single message interface. In providing integrated messaging functionality via a single message interface, the ICS of an embodiment relieves the dependency of a  
20 voice mail system, for example, by providing users with access to voice mail messages and capabilities of the voice mail system through the local groupware applications and email messaging system.

The ICS generally includes a communication server, a cache system, and an interface module. The ICS integrates with a messaging and collaboration system and the corresponding groupware applications in a network environment for example. In providing integrated messaging capabilities, the communication server and interface  
25 module function to route a call received from a caller to a user and, in the event the user is not available, to receive and route a voice mail message left by the caller. The ICS uses caching processes during the receiving and routing of voice mail messages that provide users with fast access to voice mail messages, user information and contact information. Using caching process, the ICS also provides access to the voice mail messaging system during periods when the messaging and collaboration system is offline. The ICS also leverages the storage capability of the  
30 messaging and collaboration system to eliminate the need for a separate voice mail database.

The message interface of the ICS includes a form-based interface for use in retrieving voice mail messages and controlling actions taken on voice mail messages received in the enterprise network system. This form-based interface enables a user to retrieve and take various actions on voice mail messages using data of a form provided to the user's client device by the enterprise network email system. Use of the form-based interface thus provides users  
35 with access to the integrated messaging functions offered by the ICS without a requirement to install or run a dedicated client application on the user's client device.

In the following description, numerous specific details are introduced to provide a thorough understanding of, and enabling description for, embodiments of the ICS. One skilled in the relevant art, however, will recognize that these embodiments can be practiced without one or more of the specific details, or with other components,  
40 systems, etc. In other instances, well-known structures or operations are not shown, or are not described in detail, to avoid obscuring aspects of the disclosed embodiments.

Figure 1 is a block diagram of a system 10 that includes an integrated communication system (“ICS”) 100, under an embodiment. ICS 100 includes a communication server 110, an interface module (“IM”) 120, and a cache system 130 (also referred to as the “cache”), but is not so limited. Communication server 110 couples to components of any number of networks 150 and 160 using any of a variety of communication protocols, where  
5 networks 150 and 160 may be of the same or of different types. Networks 150 and 160 allow for information transfers between various client devices 170 and 199, also referred to as user devices 170 and 199.

IM 120 of ICS 100 couples to transfer information or data with communication server 110. Additionally, IM 120 couples to transfer information with one or more components of a messaging server 140, where transferring information includes one or more of pulling, receiving, retrieving, polling, transmitting, and pushing operations, to name a few. As an example of an information transfer between IM 120 and messaging server 140, IM 120 pulls  
10 user information from messaging server 140 and makes the pulled user information available to other components of ICS 100, wherein the user information includes information relevant to at least network 150.

The components of messaging server 140 may include for example one or more processors 142, also referred to as “central processing units” or “CPUs,” and one or more databases 144 coupled to CPU 142. In an  
15 embodiment, IM 120 may be hosted on or running under control of messaging server 140, but is not limited to this configuration. Further, messaging server 140 may be a component of network 150 that hosts communication server 110, but is not so limited. For example, messaging server 140 may be hosting a groupware application (e.g., Microsoft Exchange, LotusNotes, etc.) of an enterprise network 150.

Cache 130 couples to communication server 110 and communicates to transfer information with one or  
20 more of communication server 110, IM 120, and one or more components of messaging server 140, as described below. Cache 130 may also couple to additional components (not shown) of network 150.

As an example of information transfers between cache 130 and communication server 110, cache 130 may receive caller information (e.g., voice mail messages, caller identification, etc.) from client devices 199 via  
25 communication server 110. An example of information transfers between cache 130 and messaging server 140 includes transfers in which cache 130 receives user information from messaging server 140, where the user information may be routed from messaging server 140 via IM 120 and/or communication server 110. Another example of information transfers between cache 130 and messaging server 140 includes transfers in which messaging server 140 receives information from cache 130 routed from cache 130 via communication server 110 and/or IM 120.

30 Examples of information transfers between cache 130 and IM 120 include transfers of user information pulled from messaging server 140 by IM 120 and directed to cache 130, and transfers in which IM 120 directs a message from at least one of messaging server 140 and cache 130 to at least one device on networks 150 and 160 using the user information. Cache 130 holds or temporarily stores the received information under the above examples.

35 Networks 150 and 160 include various network components (not shown) of one or more communication service providers or carriers, but are not so limited. Further, networks 150 and 160 and corresponding network components can be any of a number/combo of network types known in the art for providing communications among coupled devices 170 and 199 including, but not limited to, proprietary networks, local area networks (“LANs”), metropolitan area networks (“MANs”), wide area networks (“WANs”), backend networks, public  
40 switched telephone networks (“PSTN”), the Internet, and other public networks for example. Additionally, networks 150 and 160 may include hybrid networks that use a proprietary network for some portion of the

communications routing, for example, while using one or more different public networks for other portions of the communications routing.

Client devices 170 and 199 include communication devices like telephones, cellular telephones, and radio telephones. Client devices 170 and 199 also include processor-based devices like, for example, portable computers ("PC"), portable computing devices, personal digital assistants ("PDA"), communication devices, cellular telephones, portable telephones, portable communication devices, and user devices or units. Client devices can include so-called multi-modal devices, where the user can interact with the device and/or the ICS through any form of input and output, such as text input, speech recognition, text output, text-to-speech, graphics, recorded files and video. In such devices, the speech recognition and text-to-speech generation may partly take place in the device and partly in the ICS. Sound and/or video may be generated by the ICS by a continuous stream of sound and/or video data sent to the device. Client devices can include all such devices and equivalents, and are not limited to any particular type of communication and/or processor-based device. In an embodiment client devices 170 are client devices operating in a private network environment like an enterprise network, while client devices 199 are client devices operating in different private network environments or under any number of public networks.

Figure 2 is a flow diagram for providing integrated communication processes 200 using ICS 100, under an embodiment. Processes 200 include receiving data streams from networks of different types, at block 202. The data streams may include a variety of data including, for example, audio or voice data. Further, the data streams may be received from any number of networks or client devices operating on the networks. Processes 200 further include generating messages at a communication server using information of the data streams, at block 204. The generated messages may be any of a number of message types. Returning to the above example in which the received data stream includes audio data, the generated message is a voice mail message, but is not so limited. Processes 200 also include transferring the messages, at block 206. The transferring operation includes for example caching information of the messages in the ICS cache and/or forwarding the messages to a messaging server.

Continuing, processes 200 include pulling user information from a messaging server coupled to the ICS, at block 208, as described above. The user information includes information relevant to users of at least the network hosting the ICS, but is not so limited. Processes 200 also include caching pulled user information from the messaging server, at block 210. Additionally, processes 200 include use of the user information of the cache to direct a message from at least one of the messaging server and the cache to one or more client devices on any of the networks, at block 212.

The ICS of an embodiment integrates different types of messaging so that a user of the ICS can access all of the message types (e.g., voice mail messages, electronic mail or email messages, etc.) with a single message interface (also referred to as a "user interface" or "UI"). In providing integrated messaging functionality via a single message interface, the ICS of an embodiment relieves the dependency on a voice mail system with a dedicated voicemail and user database, for example, by providing users with access to voice mail messages and capabilities of the voice mail system through the local email messaging system.

Figure 3 is a block diagram of example information flows 300 in a system 30 that includes ICS 100, under an embodiment. The system also includes a messaging server 140 and any number of client devices 170 that couple to ICS 100. In addition, ICS 100 couples to a communications network 160. ICS 100, messaging server 140, and client devices 170 may be hosted under a network 150, but are not so limited. System 30 is shown with one each of ICS 100, messaging server 140, and client device 170 for purposes of this description, but may include any number of each of ICS 100, messaging server 140, and client device 170 coupled in any combination. System 30 may also couple to one or more other systems (not shown) or networks via any number of backend couplings (not shown)

Components of ICS 100 include a communication server and an interface (not shown). The interface of ICS 100 may run under control of messaging server 140, as described above, but is not so limited. Information flow 300 begins when, in response to receiving data streams from networks 160 of different types, ICS 100 generates a first message 302 and transfers first message 302 to messaging server 140 via a communication with messaging server 140. First message 302 may be a voice mail message ("Voice Mail Type" or "VMT") but is not limited to this type of message. For purposes of the description herein, a voice mail message is left by a "caller" to the ICS. For example, in an embodiment where Microsoft Exchange is the messaging server 140, the VMT may be implemented using "Message Class" and/or "Message Type" fields associated with messages in Microsoft Exchange.

Following or simultaneous with receipt of first message 302, the messaging server 140 detects or identifies a type of first message 302 using information of the first message and generates a second message 312. Second message 312 is of a different type from that of first message 302, and includes information of first message 302. Second message 312 may for example be an email message but is not limited to this type of message. Second message 312 is transferred to a client device 170 via a communication with client device 170, where the communication uses a communication protocol of network 150.

Responsive to receipt of second message 312, client device 170 determines a type of the second message and requests form data 314 that corresponds to second message 312. Messaging server 140, in response to the request for form data 314, transfers form data 314 to client device 170 via the second coupling. One or more components of ICS 100 generate and/or provide form data 314 for storage in messaging server 140, and form data 314 is generated under the communication infrastructure of network 150. The form data may be displayed to the user using the corresponding form.

Client device 170 uses form data 314 to view contents of second message 312. The client device also uses form data 314 to establish communications with communication server 110 (of ICS 100) via a third coupling. The communication protocol of the third coupling is different than the communication protocol of the second coupling, but is not so limited. An "embedded control" controls activation of the third coupling. Furthermore, the client device allows a "user" using the client device to direct actions 322 on first message 302 via the third coupling with the ICS using the form data. For purposes of the description herein, a "user" is an individual with enabled capability to use functions within the ICS.

As an example under information flows 300, Figure 4 is a flow diagram for integrated communication processes 400 using ICS 100, under an embodiment. Processes 400 include transferring a first message to a messaging server from a communication server via a first coupling, at block 402. Processes 400 also include generating a second message at the messaging server in response to a type of the first message and transferring the second message to a client device via a second coupling, at block 404. The second message may be of a different type than the first message and includes data of the first message. Processes 400 further include transferring to the client device form data that corresponds to the first message, at block 406. Additionally, processes 400 include establishing a third coupling between the client device and the communication server using the form data, at block 408. Moreover, processes 400 include directing actions on the first message from the client device using the form data, the actions directed via the third coupling, at block 410.

The ICS of an embodiment integrates messages of different types to enable a user to access a number of message types through components of the ICS. Thus, an application of the ICS of an embodiment is as a substitute for a voice mail system in an enterprise network, where the ICS enables a user to receive and/or take action on voice mail messages using the enterprise email system.



Figure 5 is a block diagram of an enterprise network system 500 that includes a communication server 110 and IM 120 of an ICS, under an embodiment. Communication server 110 couples to at least one messaging server 140 via IM 120. IM 120 runs under messaging server 140, but is not limited to running under this server. Messaging server also couples to one or more databases 144. Messaging server 140 of an embodiment supports the messaging capabilities of enterprise network system 500 using a groupware application (e.g., Microsoft Exchange) (not shown) along with other applications as appropriate to the size and type of enterprise network system 500. Messaging server 140, database 144, and groupware application (not shown) may be referred to as collectively forming a "messaging environment."

Communication server 110 couples to any number of client devices 199 external to enterprise network 500 via one or more networks (not shown), as described above with reference to Figure 1. Similarly, communication server 110 couples to any number of client devices 170 local to enterprise network 500.

Communication server 110 includes an operating system 518 as well as numerous components or subsystems. These components include but are not limited to one or more Voice Applications 512, an Execution Engine 514, and any number of Mobile Application Modules 516, as described below, or any other type of application module.

Figure 6 is a block diagram of an enterprise network system 600 that includes an ICS, under an embodiment. The ICS includes a communication server 610 as described above, also referred to as a "Messaging Communication Server" or "MCS." The MCS may be highly scalable. According to an embodiment of the invention, the MCS may be configured as a modular "appliance" that is essentially self-contained, and may be, for example, encased in a stackable, "pizza-box" style server. The ICS also includes IM 620 (also referred to herein as the "IM") and a Management Console 660. The IM, which in one embodiment runs under control of a messaging server 640 (also referred to herein as "MSERV 640" or "MSERV"), couples to components of the MCS, the MSERV, and a Database 644 (also referred to herein as a "Database") in a number of sequences as described herein and as appropriate to enterprise network system 600. The IM also couples to MCS Management Console 660. The MCS and the MSERV couple to the LAN for communication with other components (not shown) of enterprise network system 600.

The MCS of an embodiment includes an "Operating System" along with an "Execution Engine," some number of "Voice Applications," and some number of "Mobile Applications." The Operating System includes for example a Linux kernel with a journaling file system that provides integrity of file system tables and the data structure. The storage on the MCS may be configured as a RAID (Redundant Array of Independent Disks) configuration to provide high reliability access to software and data. The Operating System supports operations of numerous other components of the MCS as described below.

With regard to the Operating System, the MCS includes a "Telephony Interface" that couples calls and connects callers and users to/from the MCS. The Telephony Interface couples call information to/from a private branch exchange ("PBX") (not shown) for example, where the PBX is a component of enterprise network system 600. The Telephony Interface couples to the PBX using a variety of telephony integrations that include one or more of analog, Simplified Message Desk Interface ("SMDI"), T1/E1, Voice over Internet Protocol ("VOIP"), and Digital Set Emulation ("DSE") signals, but may couple using other signals/signaling protocols. When receiving a call from the PBX, for example, the MCS receives data of an incoming call from the PBX, where the data includes called party information, a reason for transfer of call (e.g., called party line busy, no answer by called party, called party using call forwarding, etc.), and calling parting information (caller ID, etc.).

A "Driver" couples information received at the Telephony Interface to the "Telephony Services" component of the MCS. The Driver may perform low level signaling and/or data conversion as appropriate to the received signals. The Telephony Services include one or more components for use in processing the received signals. These components include, for example, voice processing, switching /control, and PBX signaling, but are not limited to these components.

The MCS of an embodiment includes at least one "Voice Browser" that, when the MCS receives a call, receives voice information of the call. The Voice Browser controls the use of automatic speech recognition ("ASR") for speech recognition and DTMF recognition. The Voice Browser of an embodiment couples to a cache or other temporary store that holds voice recordings and/or name grammars ("Voice Recordings/Grammars") (the name grammars are cached after being generated from names in a user list, as described below). The ASR may use information of the name grammars. Further, the Voice Browser controls the use of text-to-speech ("TTS") as well as the play of any number of pre-recorded prompts (e.g., WAVE format files). The Voice Browser uses voice extensible markup language ("VXML") but is not limited to this protocol. Alternative embodiments of the MCS may not include the Voice Browser. As an alternative to a Voice Browser, the MCS may directly communicate with, or use other software or processes, for communication between the voice application and the Telephony Services and/or Driver.

The Virtual Machine, Voice Applications, and Execution Engine form a hierarchical state machine framework in which the Virtual Machine runs a number of APIs and modules. Consequently, the Voice Applications can include one component controlling the user interfaces ("UI") to the MCS, and another component handling lower-level communications with the modules. Use of a loose coupling between the modules and the Voice Browser provided by the state machine framework allows independence between the languages used in the different modules and the Voice Browser. The state machine framework may receive hypertext transport protocol ("HTTP") requests from the Voice Browser, for example, and generate VXML or Speech Application Language Tags ("SALT") (SALT extends existing mark-up languages such as hypertext markup language ("HTML"), extensible hypertext markup language ("XHTML"), and extensible markup language ("XML"), and enables multimodal and telephony-enabled access to information, applications, and web services from devices like PCs, telephones, and PDAs for example).

The Voice Applications of an embodiment include a number of components including an automatic attendant, a caller interface, a user interface, and a system main menu, but may include other types of voice applications. The automatic attendant is speech enabled, but may be dual tone multi-frequency ("DTMF") -enabled. The automatic attendant, which can be enabled or disabled, uses information of contact lists (e.g., User List) in the Cache.

The Voice Applications also include at least one voice mail application. The voice mail application uses information of the Cache (e.g., User List, Global Address List, Public Folders, Personal Contact Folders) in operations that include sending a new voice mail and/or forwarding a received voice mail. The voice mail application also uses Cache information in support of voice mail networking in which voice mails and corresponding information are exchanged with groupware applications of enterprise network system 600, as described below.

The voice mail application couples to the MCS state machine framework described above via one or more application programming interfaces ("API"). The APIs handle the different data formats/types in use by enterprise network system 600 (e.g., greeting data, PIN (Personal Identification Number) code data, voice mail message data, system parameters, etc.). Similarly, the Cache also couples to the state machine framework, where the Cache

includes one or more of local cache and distributed cache. Therefore, communications among the voice mail application, the Cache, and the MSERV take place via the state machine framework and the APIs as appropriate to the state (e.g., offline, online) of the MSERV.

5 In addition to the Voice Applications, the modules running under the Virtual Machine of an embodiment include Mobile Applications. The Mobile Applications provide access to user information via mobile devices, where the access may include transferring information of email, calendar, and/or contacts to a user's mobile client device via an electronic message (e.g., SMS, MMS, and/or pager).

10 The MCS also includes an "Administration/Configuration" manager. The Administration/Configuration manager provides access to and control of a unified configuration file of the MCS. The Administration/Configuration manager uses information of the unified configuration file to provide separate Configuration Files to one or more of the components of the MCS as appropriate. The unified configuration file can be copied from the MCS and stored for backup purposes. Additionally, a predefined configuration file may be uploaded to the MCS to provide the appropriate configuration for the MCS. A browser interface to the Administration/Configuration manager allows remote access to the MCS.

15 The MCS also includes a "Self Maintenance Supervisor" or reliability server that monitors MCS components and restarts failed processes when necessary, for example. In addition, the MCS also includes "Security Restrictions" for use in controlling MCS/port security.

20 As described above, the MCS of an embodiment interfaces with the MSERV via the IM. The MCS communicates with the IM via the Groupware Connector for example, but is not so limited. The Groupware Connector of an embodiment includes a "Web Server," but is not so limited. The MSERV functions as a messaging and collaboration server. The IM is an interface that runs under the MSERV in one embodiment to provide communications and information transfers between components of the MCS and components of the MSERV. In other embodiments, the IM may run under control of the MCS, for example. The IM includes and/or couples with Management Console 660 as well as with a diagnostics component ("Diagnostics Component") and/or a run time component ("RTC") (not shown).

25 Management Console 660 supports access to the MCS by a system administrator of enterprise network system 600 for purposes of managing user access. Consequently, Management Console 660 allows a system administrator to enable new users with integrated messaging functionality of the ICS and administer and monitor one or more MCSs.

30 The Diagnostics Component of the IM supports on-the-fly diagnostics gathering, computing, and/or compiling of pre-specified diagnostics information or parameters from the MSERV. In this manner the MCS may provide diagnostics information and a user may provide dynamically updateable diagnostics information.

35 The RTC translates communications between components of the MCS and components of the MSERV. As an example the RTC may be used to retrieve user information from the directory service (e.g., Active Directory) of a groupware application in response to a request from the MCS, as described below. Communications between the RTC and components of the MCS use for example XML and Web Services. Communications between the RTC and the MSERV may use one or more APIs of the MSERV (e.g., MAPI, Collaboration Data Objects ("CDO"), Web Distributed Authoring and Versioning ("WebDAV"), etc.).

40 The MSERV of an embodiment represents a messaging and collaboration server. The messaging and collaboration server includes a groupware application that runs on one or more servers and enables users via local client devices to send and/or receive electronic mail and other forms of interactive communication through computer networks. The MCS of an embodiment interoperates with groupware applications that include, but are not limited

to, Microsoft Exchange Server, but alternative embodiments may use other types of messaging and collaboration servers. Therefore, the MCS of an embodiment interoperates with client device applications ("client applications") such as Microsoft Outlook, as well as with other email client applications (e.g., Microsoft Outlook Express).

5 The MSERV sends and receives email messages through what is commonly referred to as a client device such as a personal computer, workstation, or a mobile device including mobile phones or PDAs. The client device typically connects to the LAN, which may include any number and/or combination of servers or mainframe computers where the email mailboxes and public folders are stored. The centralized servers connect to numerous other types of networks (e.g., private or proprietary, and the Internet) to transmit and receive email messages to other email users. Consequently, the MCS uses the MSERV for storing and forwarding email messages in an  
10 embodiment.

The MSERV also couples to a directory service (not shown), which is a database of information on each user account in the enterprise network system. Access to the directory service may use for example a Lightweight Directory Access Protocol ("LDAP").

15 With regard to client device access functionality, the MSERV provides integrated collaborative messaging features such as scheduling, contact, and task management capabilities. As an example MSERV configuration, when the MSERV is Microsoft Exchange, the MSERV runs on a version of the Microsoft Windows Server operating system. A version of Microsoft Office Outlook runs on Windows-based local client devices and communicates with the MSERV through the messaging application programming interface ("MAPI") protocol. The MSERV also accommodates other client device access by supporting one or more of Post Office Protocol 3  
20 ("POP3") and Internet Message Access Protocol 4 ("IMAP4") protocols as well as support for Simple Mail Transfer Protocol ("SMTP"). Using this same MSERV configuration example, the MCS of an embodiment, along with Microsoft Outlook Web Access (a service in Microsoft Exchange) accommodates web browser-based access clients, also referred to as thin clients.

25 The MSERV collaboration features support information sharing among users. Collaborative scenarios include maintaining shared address lists that all users can view and edit, scheduling meetings that include people and conference rooms by viewing associated free or busy schedules, the ability to grant other people, such as administrators, access to user mailboxes on behalf of the user.

30 As described above, the IM serves as an interface for the transfer of information between components of the MCS and components of the MSERV. Transferring information includes for example pulling, receiving, retrieving, polling, transmitting, and pushing operations, to name a few. As an example of information transfers between the MCS and the MSERV, the IM pulls information from one or more components of the MSERV and makes the pulled information available to, for example, the MCS Cache. The IM also pushes information from one or more components of the MCS to the MSERV.

35 In serving as an interface between the MCS and the MSERV, the components of the IM (e.g., RTC) translate communications between components of the MCS (e.g., Virtual Machine, Cache, etc.) and components of the MSERV environment. As an example the IM retrieves user information from components of the directory service (e.g., Active Directory) in response to a request from the MCS/Cache.

40 Embodiments of the IM may include one or more of the following components: an RTC, a Management Console, a desktop component, messaging actions control component, Diagnostics Component and/or a message waiting indication component. The desktop component allows the user to configure aspects of the user's integrated messaging account, such as voice message greetings, extended absence greeting, PIN code data, and presence information. The messaging actions control component receives and responds to user generated requests from the

FBUI (defined herein) to take actions such as playing, replaying to and forwarding voice messages, as well as calling the sender of a voice mail message. The message waiting indication component receives events from the user's message inbox folder and requests corresponding action from the PBX or other aspect of the telephony system, such turning on message waiting indicators on the user's device(s). The message waiting indication component may send notifications by way of SMS, MMS and/or pager.

Figure 7 is a block diagram 700 that shows interactions between the IM and components of the MSERV environment 740, under an embodiment. The components of MSERV environment 740 include the MSERV and one or more Databases as described above. The Database of an embodiment includes a directory service 742.

Directory service 742 provides a location for storage of information about network-based entities, such as applications, files, and printers to name a few. Directory service 742 also stores information about individuals, also referred to as users, and this information is referred to herein as "User Information." As such directory service 742 provides a consistent way to name, describe, locate, access, manage, and secure information about individual resources in an enterprise network environment. Directory service 742 uses the stored information to act as the main switchboard of the enterprise network operating system and is therefore the central authority that manages the identities and brokers the relationships between distributed resources of the enterprise network, thus enabling the resources to work together. Directory service 742 of an embodiment may be Microsoft Active Directory ("AD"), but is not so limited.

In embodiments including AD, there is a user object stored in an AD Database for each enterprise user. For example, the user object for enterprise USER 2 is shown as USER 2 object 702. The user object includes many fixed attributes such as user name, user phone number, user mailbox location, and user email address.

The user object further includes a number of "Custom Attributes." The number of Custom Attributes is small, for example fifteen, compared to the number of fixed attributes. The Custom Attributes are usable to store information not provided for in the predefined fixed attributes. In one embodiment, a Custom Attribute stores user-specific data that is used by the Voice Applications. Examples of such user-specific data include a class of service ("COS") for the user, a voice mail extension for the user, whether voice mail is enabled for the user, etc. The data is stored as a data stream in the Custom Attribute with a maximum size of 2048 bytes. In an alternative embodiment, the user-specific data that is used by the Voice Applications is stored as individual data items in fixed attributes by extending AD in a known manner.

The user mailbox location fixed attribute indicates where the user's email mailbox is stored in the enterprise. In some large enterprises, there may be many MSERVs, each including a database storing many user mailboxes. As shown, the mailbox location fixed attribute points to USER 2 mailbox 704 on an MSERV called MSERV 1.

User mailbox 704 stores email messages sent to the user, as well as outgoing messages and other items, for predetermined periods of time. In an embodiment, the messages can be of at least two types, one of which is a "normal" message that is routinely accessible by the user. Another message type is a "hidden" message that is not routinely accessible by the user through the normal user email interfaces. In an embodiment, a hidden message is used to store data used by the Voice Applications. In contrast to the data stored in the Custom Attribute, however, the data stored in the hidden message can be much larger than the 2048 byte limit of the custom attribute. In one embodiment, among the data stored in the hidden message are audio files stored as attachments to the hidden message, such as a "busy" greeting for the user's voice mail mailbox, a "no answer" greeting for the user's voice mail mailbox, and a recorded name for the user's voice mail mailbox.

An example of the MCS accessing the MSERV environment 740 through IM 620 is a phone caller calling the voice mail mailbox of USER 2 when USER 2 is on the phone. The MCS transmits an action via IM 620 with a request to "play busy greeting." The transmission includes information to access the USER 2 object 702 fixed attributes to determine the user's email mailbox location. In addition the transmission includes information to  
5 access the USER 2 object 702 Custom Attribute and to transfer the contents of the Custom Attribute to the MCS via IM 620. When the user's email mailbox is accessed, the hidden message is opened to transfer the appropriate audio file ("busy" greeting in this case) to the MCS for playing over the phone to the caller. In many cases, it may not be necessary to transfer either the Custom Attribute or the audio file from the MSERV environment 740 because the current custom attributes and audio file are cached on the MCS.

10 As described above, operations of the Voice Applications and the Virtual Machine couple the Cache and other components of the MCS to components of the MSERV via the IM. As such, the MCS and the IM support the transfer of information between the Cache and backend network components like the MSERV and the database. This configuration provides transparency between the Voice Applications and data stored in the database when using information of the database to support voice mail messaging functions of the MCS, as described below.

15 The information transfers between the Cache and the MSERV along with use of the Custom Attributes and Hidden Messages as described above allow the ICS to overcome the need for an external database to store information stored by a typical voice mail system. This is because the information used by the MCS in providing voice mail message capabilities integrated with the email messaging capabilities of the enterprise network is pulled by the MCS from the MSERV via the IM. The pulling or retrieving may be performed periodically, continually, on  
20 demand, and/or in response to particular events (e.g., update of the information in the MSERV) but is not so limited. The information pulled by the MCS includes information of a "Global Address List" ("GAL"), information of one or more "Public Folders," "Personal Contacts," and information of a "User List."

25 The GAL includes information of all users in the enterprise network having access privileges that include the use of email. Public Folders include information of the network enterprise (e.g., contacts, calendars, etc.) that are shared with all users. The Personal Contacts include contact information for each user.

The User List includes User Information for a subset of users in the GAL each of whom has access privileged that include the use of the ICS. The User List therefore is a subset of the GAL and is retrieved and/or  
30 cached as a separate list or stream in order to improve efficiency of communications and minimize the delays associated with having the MCS search the entire contents of the GAL for information used in executing a user-requested action on a voice mail message. The User List of an embodiment includes one or more of the following parameters corresponding to each user, but is not limited to these parameters: Site identification, mail box number, pronounceable name, office telephone extension, COS, automatic attendant state (e.g., enabled, disabled), voice mail state (e.g., enabled, disabled), Voice User Interface ("VUI") state (e.g., enabled, disabled), mobile access state (e.g.,  
35 enabled, disabled), bad logins, locked out, attendant destination, force change of PIN code, mobile gateway identification, full name, first name, last name, user name, home telephone number, office telephone number, cellular telephone number, identification, email address, department, active greeting state, time and date announcement, voice mail notification state (e.g., enabled, disabled), mail box status, PIN code in encrypted or raw form, no answer greeting, busy greeting, extended absence greeting, recorded name, and system greeting.

40 Instead of storing the information pulled from the MSERV in a separate voice mail database as would be done in a typical voice mail system, the pulled information is pushed by the IM to the MCS and held in the Cache. The MCS uses the pulled information in subsequent voice mail message manipulation operations as described below. This pulling and caching of information by the MCS improves the speed and efficiency of voice mail

message operations and prevents unnecessary loads on the MSERV resulting from the nearly continuous stream of read requests to the MSERV database in typical messaging systems.

5 The pulling of information from the MSERV by the MCS includes pulling and caching of information including the GAL, Public Folder, and User List. The pulled information is cached by the MCS on a system or non-individual basis because this information applies throughout the enterprise. This information is pulled and cached periodically, for example at 24-hour intervals (e.g., each morning at 2:00 am), or may be loaded on demand, but is not so limited.

10 In contrast the MCS pulls and caches information of the Personal Contacts on a per user basis because this information is different for each user. The Personal Contacts may be requested and cached by the MCS periodically or on demand (e.g., at the time a user logs in to the ICS, in response to modifications of the Personal Contacts, etc.).

15 In operating to provide integrated messaging capabilities, the MCS and the IM function to route a call placed by a caller to a user and, in the event the user is not available, to receive and route a voice mail message left by the caller. The MCS and the IM also function to provide a user with access to voice mail messages using the messaging server of the enterprise email system. The voice mail access supports both online and offline modes of the messaging server.

An example of call routing by the MCS, and with further reference to Figure 6, the MCS receives and detects a call at the Telephony Interface. Data of the call (e.g., called party information, calling party information, reason for call transfer, etc.) invokes the Voice Browser. The Voice Browser transfers a request to the Voice Applications in response to the call data.

20 A Dispatcher component of the Voice Applications routes the call to one or more other Voice Application components in accordance with information of the User List. As an example, the Dispatcher identifies the target user for the call, and determines whether the target user's automatic attendant is enabled. If the automatic attendant is enabled then the automatic attendant receives the call request and provides the caller with one or more call routing options (e.g., caller selects call routing by selecting and/or saying extension number, selecting and/or saying name, etc.) and routes the call according to the caller's input.

25 As an example, one or more of the Voice Applications determine an active greeting currently designated by the user for use in responding to calls (e.g., system greeting, no answer greeting, busy greeting, extended absence greeting, etc.), and retrieve the designated active greeting from one of the Cache or MSERV as appropriate to a state of the MSERV. The respective application(s) play the greeting, activate a "record mode" to record the voice mail message of the caller, and provide the caller with additional options available for call and/or message routing (e.g., message marking options, message delivery options, send message, route message to additional users, etc.). Upon completion of the recording and/or selection of a message routing option by the caller, the respective application(s) terminate the call (hangs up) and transfer the recorded voice mail message to one or more locations in the Cache and/or MSERV (e.g., a mail box) that correspond to the user, as described below with reference to Figures 8, 9, and 30 10. Alternatively, the voice mail message may be transferred before the application terminates the call.

35 As referenced above, the MCS of an embodiment in conjunction with the IM supports availability of and access to the voice mail applications when the MSERV is both "online" and "offline" through the use of the Cache. The MCS of an embodiment includes an "Offline Detector" that monitors an availability state of the MSERV and detects unavailability ("offline condition" or "offline state") of the MSERV. Upon detecting MSERV 40 unavailability, the MCS transitions to a mode that supports voice mail message recording and retrieval during the MSERV offline condition.

Caching of select information received and/or generated in the MCS, including User Information and voice mail information, enhances performance of the enterprise network voice messaging system by reducing the instances of data retrieval from the MSERV. Further, caching of select information improves the reliability of the enterprise network voice messaging system by allowing access to the voice messaging system during periods when the MSERV is offline.

Information received at the MCS is routed and held in the Cache in accordance with policies running in the state machine framework and/or the availability state of the MSERV. Examples of information held in the Cache include but are not limited to the User List, Global Address List, information of Public Folders, information of Personal Contact Folders, voice mail message information (both the text description portion and the audio message portion of the voice mail message), greetings, and other user parameters/permissions, and personal information of users (e.g., PIN codes).

Regarding actions taken by the MCS following receiving and recording of a voice mail message when the MSERV is online, the MCS generally holds information of the recorded message in the Cache. The MCS may also transfer the recorded voice mail message via the IM to the MSERV where it is stored in the Database.

As an example, Figure 8 is an information flow 800 for routing and accessing voice mail messages via the ICS when the MSERV is in an online state, under an embodiment. This information flow 800 shows one MCS and one MSERV in an enterprise network environment, but this is shown only as an example and does not limit the network environment to the types, numbers, and/or coupling of components shown as alternative embodiments may have any number of MCSs and/or MSERVs.

Information flow 800 begins when a caller places a call 802 to a user and availability of the user results in the caller leaving a voice mail message (referred to herein as the "VMSG") for the user. The voice mail message VMSG is received at the MCS and routed 804C to the Cache where it is assigned an identification (referred to herein as the "CACHEID") and held. The voice mail message VMSG may be held in the Cache for a pre-specified period of time, but the embodiment is not so limited. The voice mail message VMSG and the CACHEID are also routed 804M to the MSERV via the IM, as described above. The MSERV assigns an identification (referred to herein as the "VMSGID") to the incoming voice mail message VMSG and stores 806 the voice mail message VMSG along with the VMSGID and CACHEID in one or more areas of memory (not shown) available to the MSERV. Memory may include any various form of storage or computer-readable memories such as, but not limited to, volatile memory (random access memory ("RAM"), non-volatile memory (read-only memory ("ROM"), EEPROM, disk, and/or other storage devices that may include one or more of magnetic and optical storage media.

As described above, the MCS pulls information (e.g., periodically, on demand, etc.) from the MSERV via the IM and uses the pulled information in providing voice mail message capabilities integrated with email messaging capabilities of the enterprise network. Therefore, pulling operations by the IM include pulling of information identifying the stored voice mail message VMSG, where the information identifying the voice mail message VMSG includes but is not limited to the CACHEID. Upon request from the MCS, the IM may pull 808 a voice mail list (referred to herein as a "VMLIST" 809), which includes CACHEIDs and VMSGIDs for any stored messages from the MSERV environment. The IM pushes 810 VMLIST 809 to the MCS where it is held. VMLIST 809 may be generated from the user's inbox upon each request from the IM or may be stored and maintained in the MSERV or in the cache as a current representation of the contents of a user's voice mailbox, or inbox. If and when a time period for holding a VMSG in the Cache expires, the VMSG is still identifiable from VMLIST 809, and can be found in the MSERV if requested, using the VMSGID.



Information flow 800 continues when a user accesses 820 the enterprise network system to retrieve his/her voice mail messages. In an embodiment, the user access 820 causes the VMLIST to be pulled 808 from the MSERV and pushed 810 by the IM to the Cache, and also or alternatively to the MCS. Upon being provided with access to the MCS, the user selects one or many voice mail message(s) by selecting a VMSGID/CACHEID item from the VMLIST. In response to the user selection, MCS searches 822 the Cache for a message, using the Cache identification CACHEID of the selected message. In a scenario in which the message was left by the caller and the time period for holding the message VMSG in the Cache has not expired, the MCS will locate the CACHEID and the message contents VMSG in the Cache. Once located through use of the CACHEID, the MCS retrieves 814R the voice mail message contents VMSG from the Cache, and plays the voice mail message for the user as appropriate to the action selected by the user.

In this manner the MCS provides user access to the contents of the voice mail message VMSG via a mapping and without storing voice mail message contents in the MCS. The mapping includes a mapping of voice mail message contents to identification information of the email environment (MSERV environment), and mapping identification information of the email environment to identification information of the voice mail environment (MCS). In this embodiment, therefore, the mapping includes mapping of voice mail message contents to the message identification VMSGID, and mapping of the message identification VMSGID of the email environment to the MCS identification CACHEID.

As used herein "pushing" data or information indicates an action of a component or entity that has the affect of transferring the data or information to another component or entity. Transferring includes sending in response to a request, query or command, and sending on the initiative of the transferring component or entity. The transfer may be an internetwork transfer, an intranetwork transfer, or a transfer between a network component or entity and a non-network component or entity.

As used herein "pulling" data or information indicates a component or entity receiving transferred data or information. Receiving includes receiving in response to a request, query or command, and retrieving in response to a request, query or command. The transfer may be an inter-network transfer, an intra-network transfer, or a transfer between a network component or entity and a non-network component or entity.

Figure 9 is an alternative information flow 900 for routing and accessing voice mail messages via the ICS when the MSERV is in an online state, under an embodiment. This alternative information flow 900 describes the scenario in which the message VMSG is left by the caller and stored in the cache and in the MSERV environment, and after expiration of the time for holding the message VMSG in the cache.

Information flow 900 begins when a caller places a call 802 to a user and availability of the user results in the caller leaving a voice mail message VMSG for the user. The voice mail message VMSG is received at the MCS and routed 804C to the cache as described above, and the VMSG and CACHEID is routed 804 to the MSERV via the IM, also as described above. The MSERV assigns identification VMSGID to the incoming voice mail message VMSG and stores 806 the voice mail message VMSG along with the VMSGID in one or more areas of memory (not shown) available to the MSERV.

Information flow 900 continues when a user accesses 820 the enterprise network system to retrieve his/her voice mail messages. VMLIST 809 is pulled 808 from the MSERV and pushed 810 by the IM to the MCS. Upon being provided with access to the MCS, the user selects a voice mail message from VMLIST 809, by selecting a CACHEID/VMSGID item. The MCS searches 822 the Cache for the Cache identification CACHEID of the selected message in response to the user selection. Because the message was left by the caller and stored in the MSERV environment and expired in the cache before the user calls in, the MCS will not locate the CACHEID in the

Cache. Consequently, the MCS accesses the MSERV, identifies the message VMSG, and pulls 924R the voice mail message contents from the MSERV environment via the IM. The MCS plays the pulled voice mail message VMSG for the user as appropriate to the action selected by the user.

5 In addition to the online scenarios described above, the MCS of an embodiment provides offline behavior that allows for holding, storing, and retrieving voice mail messages when the MSERV is offline or unavailable for some reason, or during times when the connection between the MCS and the MSERV is unreliable. Offline behavior means absence of a coupling between the MSERV and the MCS. Regarding actions taken by the MCS following recording of a voice mail message when the MSERV is offline, a component of the MCS (e.g., Offline Detector) detects the MSERV is offline. The MCS holds the recorded voice mail message in the in response to 10 detecting the MSERV state as offline. At such time as the MCS detects the MSERV is online, the Groupware Connector pulls the voice mail message from the Cache and transfers the recorded voice mail message via the IM to the MSERV where it is stored in the Database.

As an example, Figure 10 is an information flow 1000 for routing and accessing voice mail messages via the ICS when the MSERV is in an offline state, under an embodiment. This information flow 1000 shows one MCS 15 and one MSERV in an enterprise network environment, but this is shown only as an example and does not limit the network environment to these components as alternative embodiments may have any number of MCSs and/or MSERVs.

The information flow 1000 begins when a caller places a call 802 to a user and availability of the user results in the caller leaving a voice mail message VMSG for the user. The voice mail message VMSG is received at 20 the MCS, however a component of the MCS detects an unavailable or offline condition of the MSERV. In response to detecting the offline condition, the MCS assigns a CACHEID to the incoming message VMSG, and holds 1004C the message contents VMSG along with the CACHEID in the Cache.

Information flow 1000 continues when a user accesses 820 the enterprise network system to retrieve his/her voice mail messages while the MSERV remains in an offline condition. Upon being provided with access to the 25 MCS, the user selects a voice mail message from a list of CACHEIDs generated from the collection of voice mail messages held for him/her by in the cache. In response to the user selection, the MCS searches 1022 the Cache using the Cache identification CACHEID of the selected message. Upon locating the voice mail message by its CACHEID in the Cache, the MCS pulls 1014R the voice mail message contents from the Cache, and plays the voice mail message for the user as appropriate to the action selected by the user.

30 The MCS continues to monitor the condition of the MSERV. At such time as the MCS detects a return of the MSERV to an online condition, the MCS pulls 1004P the voice mail message VMSG and its CACHEID from the Cache, and transfers 1004M the voice mail message and CACHEID via the IM to the MSERV. The MSERV assigns an identification VMSGID to the incoming voice mail message VMSG and stores 1006 the voice mail message VMSG along with the VMSGID and CACHEID in one or more areas of memory as described above.

35 In addition to the capabilities described above, the ICS of an embodiment provides a Form-Based User Interface ("FBUI"). The FBUI is a form-based messaging or communication interface for use by users in retrieving voice mail messages and controlling actions taken on voice mail messages received in the enterprise network system. This FBUI enables a user to retrieve and take various actions on voice mail messages using data of a form (referred to herein as the "FBUI FORM") that is presented to the user's client device by the enterprise network 40 email system. Use of the FBUI Form thus provides the user with access to the integrated messaging functions offered by the ICS without a requirement to install or run a dedicated client application on the user's client device.

Figure 11 is a block diagram of a system 11 that includes ICS 1100 with FBUI 1180, under an embodiment. System 11 includes an enterprise network 1101 that provides integrated voice mail and email messaging through the use of ICS 1100. Enterprise network 1101 includes a LAN that couples to components of ICS 1100 and a messaging server environment 1140. ICS 1100 includes MCS 1110 IM 1120, and FBUI 1180, but is not so limited. FBUI 1180 is presented to a user (e.g., USER Z) via one or more local devices like PCs or other processor-based devices.

Messaging server environment 1140 includes the MSERV and a Database 1144, but is not so limited. The LAN couples to any number of other networks 1150 and 1160 using any of a variety of communication protocols, where the networks 1150 and 1160 may be of the same or of different types. As an example, the networks may include a public communications network 1150 and a private communications network 1160. Private communications network 1160 may be a PBX coupled to the LAN of the enterprise network, for example. Networks 1150 and 1160 allow for information transfers between client devices 1170 that are local to enterprise network 1101 and client devices 1199 that are external to enterprise network 1101. The client devices may alternatively be referred to as "user devices" 1170 and 1199.

ICS 1100 replaces the voice mail server typically found in enterprise networks with at least one MCS 1110. MCS 1110 is coupled to the private communications network (e.g., PBX) of each network enterprise. While one MCS is shown in this example system 11, the enterprise network may include multiple MCSs 1110 coupled to enterprise network in an "N + 1" configuration, where "N" is any number 1, 2 ... X.

For security reasons, communication to and from the MCS is restricted in an embodiment. The MCS communicates with the IM servers, the private communications network, other MCSs and selected client devices. According to an embodiment of the invention, communications with the MCS may be restricted to network components having particular known addresses. Additionally or alternatively, communications with the MCS may require authentication by passcode or other security measures for certain kinds of access, for example, for access by the administrator. Security may also or alternatively be encrypted and/or provided by requiring a physical connection between the MCS and other component, such as in the case of a connection between an MCS and a private communications network through a direct cable connection.

The MCS via the FBUI generally provides a form to a client device from a first server (e.g., messaging server, MSERV, etc.) via a network connection. The form includes data or code that when executed by the receiving client device results in presentation of a FBUI on a display of the client device. The FBUI includes a number of buttons or icons that allow a user to select an action on an item via a second server (e.g., communication server, MCS, etc.), where the item is stored on the first and/or second servers, and the first and second servers are different servers. The FBUI of an embodiment uses a web browser embedded in the form as the means for coupling and/or communicating with a corresponding browser control of the second server. Communications between the client device and the second server thus avoid security and/or other network policy issues that would prohibit the client device from communicating with the second server via the network coupling between the client device and the first server.

As described above, the FBUI operates as a form-based messaging interface to transfer a first message (e.g., voice mail message) to a messaging server (e.g., MSERV) from a communication server (e.g., MCS) via a first coupling (e.g., IM). The messaging server generates a second message (e.g., email message) in response to a type of the first message and transfers the second message to a client device via a second coupling (e.g., LAN). The type of the first message is specified by the communication server using properties on the message that identify the message as a "Voice Mail Type" ("VMT") message. The second message is of a different type and includes data of the first

message, but is not so limited. The communication server also transfers to the client device form data that corresponds to the first message. The client device uses the form data to establish a third coupling (e.g., browser link) between the client device and the communication server. The user may direct actions on the first message from the client device via the third coupling using the form data.

5 The ICS of an embodiment provides the FBUI 1180 to a user via his/her local client device. The FBUI is provided to the client device through the use of a FBUI Form, where the structure of the FBUI Form conforms to the message structure of the messaging server environment. For example, when the messaging server environment includes the use of Microsoft Exchange and Microsoft Outlook, the FBUI Form is generated to comply with Microsoft formats as appropriate to Exchange and Outlook

10 Information for generation of the FBUI Form is provided to the messaging server environment by the MCS via the IM, and the code used for FBUI Form generation is hosted by the MSERV in an embodiment. The FBUI Form of an embodiment includes code that generates information of the FBUI display as well as the buttons of the display. The FBUI Form further includes an embedded browser control for use in establishing communications between the client device displaying the FBUI Form and a web server (e.g., MCS, IM, other server) for example. 15 The embedded browser control therefore allows the host client device to couple and communicate with a server that is different from the MSERV via a communication channel that is outside the enterprise network LAN. Thus, the FBUI Form enables a communication channel between the local client device currently executing the form and a component like the MCS and/or IM in spite of network policy issues that otherwise might prohibit the client device from communicating outside the enterprise network message infrastructure.

20 Using the FBUI, a user can access/view and take a variety of actions on his/her voice mail messages within an email framework of the host enterprise network system. As an example, when the MCS of an embodiment receives a voice mail message it transfers the voice mail message to the MSERV, as described above. In transferring the voice mail message to the MSERV, the MCS specifies properties on the message that identify the message as a "Voice Mail Type" ("VMT") message. The message is received and stored by the MSERV as a VMT 25 message using the same storage and retrieval structure as used with other message types like email messages.

At such time as a user wishes to access his/her messages via his/her client device, the active message browser of the client device receives the VMT message along with any other mail messages currently stored in his/her electronic mail box. The message browser corresponds to the message structure of the messaging server environment (e.g., Outlook in a Microsoft environment). Upon receipt of the message, the message browser 30 identifies the message as a VMT message. As the code that implements the FBUI Form is stored on the MSERV, implementation of the functionality and/or features associated with the FBUI Form uses communication between the user's client device and the MSERV via the LAN. For example, the client device message browser requests the FBUI Form from the MSERV in response to identifying a message as a VMT message because this is the form that corresponds to the VMT message type. The MSERV transfers the FBUI Form to the requesting client device, and 35 the client device message browser launches the form in response to the user selecting a VMT message for viewing.

The message browser uses data or code of the FBUI Form to display the FBUI on the user's client device. Figure 12 is a sample FBUI 1200 as displayed on a client device, under an embodiment. The FBUI 1200 includes three areas 1202-1206 that present information to a user. The areas include a folder area 1202, a contents area 1204, and a function area 1206, but are not limited to these areas as the UIs of alternative embodiments may present any 40 number and/or type of areas. In alternative embodiments, all three areas 1202-1206 may be presented at the same time, as shown in FBUI 1200, or various subsets of the three areas may be presented at the same time in various combinations.

Folder area 1202 presents one or more folders to which the user has access via the FBUI 1200 and the client device. The "INBOX" may contain a list of voice mail messages in the same listing as other messages, including email messages. Alternatively, the Inbox may include a subfolder ("VOICE MESSAGES") which includes the voice mail messages, and selection of this folder results in the presentation of voice mail messages of the user's mail box in the contents area 1204.

The contents area 1204 generally presents the contents of the folder selected using the folder area 1202. As an example, the contents area 1204 presents information corresponding to any number of voice mail messages in the user's mail box when the INBOX or VOICE MESSAGES folder is selected. Contents area 1204 allows the user to select a particular voice mail message by placing a cursor on "VOICE MESSAGE 1 INFORMATION" for example. By (double) clicking a message in the contents area 1204 or otherwise indicating to the message browser to display a voice message, a new window (referred to as the "ICS Window") is displayed. The ICS Window now includes function are 1206.

Function area 1206 of FBUI 1200 presents one or more "voice mail action buttons" 1206A-1206E (also referred to herein as "buttons") each of which represents an action the user may select for a voice mail message. In this example, the VOICE MESSAGES folder is selected, and selection of a message in contents area 1204 allows the user to take an action on the selected message using buttons 1206A-1206E. Placing the cursor of contents area 1204 on a particular message and choosing an action on the selected message with a button 1206A-1206E therefore invokes operations on the message via components of the ICS (e.g., MCS, Cache, IM). The buttons 1206A-1206E of an embodiment include a "Play on Phone" button 1206A, a "Play on Computer" button 1206B, a "Call Sender" button 1206C, a "Reply by Voicemail" button 1206D, and a "Forward by Voicemail" button 1206E, but the embodiment is not limited to this same number of buttons or to buttons offering the same functionality.

In other embodiments, presentation of areas or information of the FBUI may vary in many ways. For example, in one embodiment, the action buttons 1206 appear after the user has selected (for example by double clicking a particular voice message from the contents area 1204. Action buttons 1206 may also appear when the user right clicks on a particular voice message in the contents area 1204.

The folder area 1202 may also include a subfolder ("VOICE MESSAGE SYSTEM") under the Public Folder. As such, the VOICE MESSAGE SYSTEM folder may not be considered an actual folder but instead a uniform resource locator ("URL") that, when selected, sends an HTTP request to a web server and launches/displays an ICS browser inside the client device message browser. The web server may, for example, be a component of the MCS and/or IM, but is not so limited. The ICS browser is an embedded or hidden browser that displays the ICS Window in the area of the client device message browser where emails would typically appear, and the voice mail messages are displayed in the ICS Window.

As an example, the ICS Window is displayed in the contents area 1204 of an embodiment. The ICS Window may be served from the IM and may contain any information related to the voice messaging system that is user specific. In one embodiment, the ICS Window will display a user login prompt where the user enters the user name and PIN code. Subsequently, the system displays the user's configuration data, such as PIN code, attendant extension, greeting type, and other applicable information.

The hidden browser enables an HTTP link and communications with the IM, for example, which then brokers communications (via HTTP) with the MCS via the MCS Web Server (Figure 6) for example. Therefore, while typical messaging servers and LANs use security policies that restrict the use of "special" code in form data, use of the hidden browser embedded in a form structure that is native to the host system overcomes this restriction because the browser is not detected or considered as special code. Use of the hidden browser thus supports

communication with the corresponding browser control in the MCS and/or the IM, thereby allowing the integration of voice mail messaging provided by the MCS with the email messaging system of the enterprise network

A "voice mail message" in the ICS is generally any message created using a client device generating an audio stream. A "voice mail message" is also any VMT message, such as a message created using the "Reply by Voice Message" and "Forward by Voice Message" buttons of the FBUI. An "email" is any message created using buttons of a host mail message system that function to generate a reply message or to forward a message in response to receipt of a message, even if replying or forwarding a voice mail message. The ICS of an embodiment presents a voice mail message to a user in an email message system using the FBUI as the presentation form.

As described above, FBUI 1200 allows a user to take action on a voice mail message via buttons 1206A-1206E of FBUI 1200. Therefore, placing the cursor of contents area 1204 on a particular message and choosing an action on the selected message with a button 1206A-1206E invokes the action on the message via components of the MCS and/or the enterprise network environment.

As one example of an action on a voice mail message, and with further reference to Figure 11, the user may select a "Play on Phone" action using button 1206A. In response the user's client device couples to a component of the ICS (e.g., IM) using the hidden browser of the FBUI. The client device receives a pop-up message from the ICS via the browser link and the ICS Window, where the pop-up message allows the user to choose or enter a telephone number to which he/she would like the selected voice mail message routed. The pop-up message also includes a "connect" button by which the user initiates routing of the selected voice mail message to the selected telephone. In response to selection of the "connect" button, the IM couples with an MCS, and the MCS causes the PBX to initiate a call to the telephone number selected by the user via the pop-up window. Upon connection of the call from the PBX to the selected telephone, the MCS pushes the contents of the voice mail message to the selected telephone.

Another example of an action on a voice mail message includes selection of a "Play on Computer" action by the user via button 1206B. In response the user's client device couples to a component of the ICS (e.g., IM) using the hidden browser of the FBUI. In response to selection of the "Play on Computer" button, the IM couples with an MCS, and the MCS pushes a form to the user's computer that resembles a typical email. The form includes an attachment that is an audio file (e.g., WAVE, MP3, other audio formats, etc.). When the user selects the attachment the client device may launch the default audio player of the client device.

Alternatively, selection of the attachment in a "Play on Computer" action may result in the browser form controlling launch of a pre-specified audio player instead of the default audio player. This is similar to the hidden browser described above with reference to presentation of the FBUI.

The user may also select a "Call Sender" action on a voice mail message using button 1206C. In response the user's client device couples to a component of the ICS (e.g., IM) using the hidden browser of the FBUI. In response to selection of the "Call Sender" button, the IM couples with an MCS, and the MCS retrieves the selected message from the Cache or the MSERV. Using the caller information from the retrieved message, the MCS causes the PBX to connect the call to the user's local telephone. Upon connection of the call from the PBX to the user's telephone, the MCS causes the PBX to initiate a call to the sender's telephone number as determined from the caller information associated with the voice message.

Additionally, the user may select a "Reply by Voice Message" action on a voice mail message using button 1206D. In response the user's client device couples to a component of the ICS (e.g., IM) using the hidden browser of the FBUI. In response to selection of the "Reply by Voice Message" button, the IM couples with an MCS, and the MCS retrieves the selected message from the Cache or the MSERV. The MCS causes a reply message to be generated corresponding to the received message, and prompts the user to record an audio message for the reply.

The user records the audio for the reply via a microphone coupled to his/her client device. Alternatively, the user may record the audio for the reply via his/her local telephone. Upon completing the audio reply recording, the MCS causes the reply message to be transmitted to the designated addressees via the MSERV. A user is not required to listen to a message to invoke the "Reply by Voice Message" action.

5 The user may also select a "Forward by Voice Message" action on a voice mail message using button 1206E. In response the user's client device couples to a component of the ICS (e.g., IM) using the hidden browser of the FBUI. The client device receives a pop-up message from the ICS via the browser link, where the pop-up message allows the user to choose or enter a telephone number to which he/she would like the selected voice mail message routed. The pop-up message also includes a "connect" button by which the user initiates routing of the  
10 selected voice mail message to the selected telephone. In response to selection of the "connect" button, the IM couples with an MCS, and the MCS causes the PBX to initiate a call to the telephone number selected by the user via the pop-up window. Upon connection of the call from the PBX to the called telephone selected by the user, the MCS pushes the contents of the voice mail message to the called telephone and the user. During the session, and in addition to the contents of the voice mail message, the MCS may provide a verbal prompt to the user requesting  
15 information of the party to whom the message is to be forwarded, and/or a prompt to the user to record an audio message to be forwarded along with the forwarded message. A user is not required to listen to a message to invoke the "Forward by Voice Message" action.

Figure 13 is a block diagram of a system 13 that includes multiple Sites (defined herein) and multiple components, under an alternative embodiment. System 13 includes multiple Sites, some of which may have  
20 multiple MCSs, IMs, private communication networks and MSERVs. As shown, system 13 includes MSERV 1390 and MSERV 1391 communicating via a network 1392, which may comprise any of a public network, such as a PSTN, or private communications network or other network. The MSERVs are coupled to one or more IMs. For example, as shown here, MSERV 1390 is coupled to IMs 1385 (IM1 and IM2), and MSERV 1391 is coupled to IMs 1386 (IM3 and IM4). The IMs are coupled to one or more MCSs. For example, as shown here IM1 is coupled to  
25 MCS1, MCS2, and MCS3; IM2 is coupled to MCS2, MCS3, MCS4 and MCS5; IM3 is coupled to MCS6 and MCS7; and IM 4 is coupled to MCS8. The MCSs are coupled to private communications networks. As shown here, MCS1, MCS2, MCS3, MCS4 and MCS 5 are coupled to private communications network 1 1360A; MCS6, and MCS7 are coupled to private communications network 2 1360B; and MCS8 is coupled to private communications network 2 1360B and private communications network 3 1360C.

30 Thus, Figure 13 shows a system 13 that is scalable in a number of different dimensions, according to various embodiments of the invention. Two MSERVs are shown coupled by a network. This configuration allows for sharing of voicemail messages, user lists, global address lists, distribution lists and public folders between the various MSERVs that connected by a network and which may be placed at the same or different locations. Additionally, use of multiple MSERVs allows for scaling of the overall system through the increased capacity  
35 provided by the multiple MSERVs.

Multiple MCSs are shown. Increased number of MCSs can help to increase overall system capacity and/or redundancy by providing increased number of ports, storage, and processing capacity. According to an embodiment of the invention, information on the MCSs is derived from the MSERVs and automatically cached on the MCSs. This allows for easy deployment of new MCSs by which the data and configuration settings for the new MCSs are  
40 acquired from the MSERV(s) and/or caches of other MCSs. Additionally, an MCS may be coupled to more than one private communications network. In some cases an MCS may operate with multiple private communications networks simultaneously. Also, an MCS that is coupled to multiple private communications networks may continue

operation with a non-failing private communications network in the event that one of the private communications networks to which the MCS is coupled fails. In one embodiment, the MCS that is coupled to multiple private communications networks operates with at least one of the private communications networks, but begins to operate with another, non-failing private communications network in the event that a private communications network to which the MCS is coupled fails.

Multiple IMs are shown in Figure 13, which help to support the capacity of additional MCSs. The multiple IMs also may provide fail over support for each other in the event that one of the IMs fails.

In Figure 13, the equipment and users associated with a particular private communications network referred to as members of a "Site." Accordingly, a user may have a Site identification. The Site identification may be used to filter user information associated with a particular Site from the a broader set of user information stored on the MSERV servicing multiple Sites. Additionally, Sites may be combined into auto attendant groups. The auto attendant groups are Sites that share a common dial plan. For example, members of an auto attendant group may be able to place calls using extension numbers instead of full numbers.

According to an embodiment of the invention, various subsets of users may be defined from among the users in an MSERV or set of networked MSERVs. Such subsets of users may be defined by a Site identification. In this way, various subsets of users may be associated with different respective private communications networks, such that the users' access to respective Sites within a network of MSERVs depends on the users' membership in the various defined subsets of users. For example, members of a subgroup of users associated with a particular Site may be able to use functions such as message waiting indication and control of messaging actions at their associated Site but not at other Sites.

More generally, Figure 14 is a block diagram of a system 1400 that uses a form-based interface 1480 transferred in a first type of message for controlling actions on a second type of message, along with the corresponding information flows, under an embodiment. System 1400 generally includes an enterprise network 1401 coupled to one or more public communications networks 1450 via a private communication network 1460. Enterprise network 1401 includes at least one first server ("Server A") and at least one interface module 1420, both of which couple to a LAN. Server A also couples to private communication network 1460. In an embodiment, components of Server A and interface module 1420 function collectively to form an ICS, as described herein, but are not limited to ICS functionality.

System 1400 further includes a backend network environment 1440 and a user workstation 1470, each of which couple to the LAN. Backend network environment 1440 includes at least one server ("Server B") but may include any number of servers, databases, and/or other components. Components of user workstation 1470 also couple to the LAN, the workstation components including a telephone 1470P and a client device 1470C that are local to enterprise network 1401. Server A and interface module 1420 cause form-based interface 1480 to be displayed to User Z via Server B and client device 1470C. Public communications network 1450 and enterprise network 1401 allow information transfers between components that are local to enterprise network 1401 and client devices 1499 that are external to enterprise network 1401.

In response to receiving a data stream at an input of private communication network 1460, enterprise network 1401 uses the integrated communication processes described herein to cause Server A to generate and transfer a first message to Server B. The first message is transferred to Server B via interface module 1420. Server B generates a second message in response to properties of the first message, and transfers the second message to local client device 1470C via the LAN. The second message is a different type message than the first message and



includes information of the first message. In an embodiment, a type of the first message is a voice mail type, and a type of the second message is an email type.

Local client device 1470C requests form data from Server B in response to information of the first message, and Server B in turn provides the requested form data to local client device 1470C via the LAN. Local client device 1470C executes the form data resulting in presentation of form-based interface 1480 on a display of user workstation 1470.

While local client device 1470C is described as being local to enterprise network 1401, it is understood that client device 1470C is not required to reside at any particular physical location within enterprise network 1401. For example, "local" client device 1470C may be a tablet PC 1470C from which User Z (who is off site) remotely accesses components of enterprise network 1401 (e.g., Server B) via a virtual private network ("VPN") for example.

Upon display on local client device 1470C form-based interface 1480 presents a number of soft buttons 1406 or action buttons 1406 that allow User Z to control actions on data that resides at Server A and/or Server B via communications with interface module 1420 and Server A. These actions are controlled via a form browser control embedded in the form data. This form browser control thus allows the local client device (via the form-based interface) to couple to and communicate with Server A via an "Independent Access Channel" ("IAC") (e.g., browser link) that is independent of the LAN. The coupling between local client device 1470C and Server A is via a corresponding browser control of Server A and interface module 1420 but is not so limited.

Action buttons 1406 of form-based interface 1480 enable User Z to control or direct actions on data held and/or stored at Server A and/or Server B. The data may be the first message that was previously transferred to Server B, but is not so limited; this first message may be held at Server A and/or stored at Server B as appropriate to a state (e.g., online, offline) of Server B, as described above. Action buttons 1406 of form-based interface 1480 enable user control of actions on the data held and/or stored at Server A and/or Server B via the IAC between local client device 1470C, interface module 1420, and Server A.

Each of action buttons 1406, when selected by User Z, calls the form data of the form-based interface form as appropriate to the action corresponding to the selected action button 1406. The form data locates the browser control corresponding to the action selected by User Z, and the browser control directs the embedded browser of the form-based interface form to the corresponding URL of interface module 1420 and/or Server A. As such, selection of an action button establishes the IAC between local client device 1470C, interface module 1420, and Server A.

In response to selection of one or more actions via the action buttons, Server A may transfer status and/or control data (referred to as "status/control data") to local client device 1470C via the established IAC. Server A of an embodiment transfers the status/control data via interface module 1420, but is not so limited. Execution of the status/control data on local client device 1470C results in the display of popup 1482, where popup 1482 is a web browser window that may be smaller than form-based interface window 1480 and without some of the standard features such as tool bars.

Popup 1482 may present User Z with information regarding status of the selected action, prompts to select information via popup 1482, prompts to enter information via one or more fields of popup 1482, and prompts to initiate further actions, but is not limited to presentation of only this information. In response to entry of any additional information into popup 1482 in response to queries or prompts and/or selection of any further actions by User Z, local client device 1470C transfers data of the additional information and/or selection to Server A via the established IAC and interface module 1420.

In an example in which server actions are controlled via a form-based interface, with continued reference to Figure 14, Server A generates and transfers a first message to Server B via interface module 1420. Server B in turn

generates a second message in response to properties of the first message, and transfers the second message to local client device 1470C via the LAN. Local client device 1470C requests and receives via the LAN form data from Server B in response to information of the first message. Local client device 1470C executes the form data and presents form-based interface 1480 on a display of local client device 1470C.

5 Form-based interface 1480, as described above, presents one or more soft action buttons 1406 to User Z. Selection of an action button 1406 by User Z establishes communication 1406-1 between local client device 1470C and Server A via the IAC and interface module 1420. Server A determines a location of information needed to fulfill the requested action. If Server A determines the action corresponding to selected action button 1406 uses information from backend network environment 1440 (e.g., Server B), then Server A directs 1406-2 interface  
10 module 1420 to pull 1406-2 the information from components of the backend network environment as described above, and interface module 1420 pushes 1406-2 the pulled information to Server A.

Server A executes the action on the information as appropriate to selected action button 1406. As an example, the action may include establishing communication with 1406-3A and/or forwarding the information to an external client device 1499. Establishing communication 1406-3A and/or forwarding takes place via a coupling  
15 between Server A and external client device 1499 that includes one or more of private communication network 1460 and public communications network 1450, but is not so limited.

As another example, the selected action may include establishing communication with 1406-3B and/or forwarding the information to a client device 1470 within enterprise network 1401. The client device of enterprise network 1401 includes a local user telephone 1470P or client device 1470C, for example, but can include a variety  
20 of devices. Establishing communication 1406-3B and/or forwarding takes place via a coupling between Server A and local client device 1470P/1470C that includes one or more couplings via the LAN, but is not necessarily so limited.

In response to selection of one or more actions via the action buttons, Server A may transfer 1410-SC status/control data to local client device 1470C via the established IAC. Server A of an embodiment transfers 1410-  
25 SC the status/control data via interface module 1420, but is not so limited. Execution of the status/control data on local client device 1470C results in the display of a popup 1482 on a display of local client device 1470C via the form-based interface. Popup 1482 presents the status/control data to User Z. In response to entry of any additional information into popup 1482 in response to queries or prompts and/or selection of any further actions by User Z, local client device 1470C transfers 1406-1 data of the additional information and/or selection to Server A via the  
30 established IAC and interface module 1420.

As generally described above, the form-based interface of an embodiment more specifically includes the FBUI of the ICS. The FBUI operates as a form-based messaging interface to transfer a first message (e.g., voice mail message) to a messaging server (e.g., MSERV) from a communication server (e.g., MCS) via a first coupling (e.g., IM). The messaging server generates a second message (e.g., email message) in response to a type of the first  
35 message and transfers the second message to a client device via a second coupling (e.g., LAN). The type of the first message is specified by the communication server using properties on the message that identify the message as a "Voice Mail Type" ("VMT") message. The second message is of a different type and includes data of the first message, but is not so limited. The communication server also transfers to the client device form data that includes the FBUI Form, where the FBUI Form corresponds to the first message. The client device uses the FBUI Form data  
40 to establish a third coupling (e.g., browser link) between the client device and the communication server. The user may direct actions on the first message from the client device via the third coupling using the form data.

The ICS of an embodiment provides the FBUI to a user via his/her local client device. The FBUI is provided to the client device through the use of a FBUI Form, where the structure of the FBUI Form conforms to the message structure of the messaging server environment. For example, when the messaging server environment includes the use of Microsoft Exchange and Microsoft Outlook, the FBUI Form is generated to comply with Microsoft formats as appropriate to Exchange and Outlook.

Information for generation of the FBUI Form is provided to the messaging server environment by the MCS via the IM, and the code used for FBUI Form generation is hosted by the MSERV in an embodiment. The FBUI Form of an embodiment includes code that generates information of the FBUI display as well as the buttons of the display. The FBUI Form further includes an embedded browser control for use in establishing communications between the client device displaying the FBUI Form and a web server (e.g., MCS, IM, other server) for example. The embedded browser control therefore allows the host client device to couple and communicate with a server that is different from the MSERV via a communication channel IAC that is outside the enterprise network LAN. Thus, the FBUI Form enables a communication channel between the local client device currently executing the form and a component like the MCS and/or IM in spite of network policy issues that otherwise prohibit the client device from communicating outside the enterprise network message infrastructure.

The FBUI thus allows a user to access, view and take a variety of actions on his/her voice mail messages within an email framework of the host enterprise network system. As an example, when the MCS of an embodiment receives a voice mail message it transfers the voice mail message to the MSERV, as described above. In transferring the voice mail message to the MSERV, the MCS specifies properties on the message that identify the message as a VMT message. The message is received and stored by the MSERV as a VMT message using the same storage and retrieval structure as used with other message types like email messages.

At such time as a user wishes to access his/her messages via his/her client device, the active message browser of the client device receives the VMT message along with any other mail messages currently stored in his/her electronic mail box. The message browser corresponds to the message structure of the messaging server environment (e.g., Outlook in a Microsoft environment). Upon receipt of the message, the message browser identifies the message as a VMT message. As the code that implements the FBUI Form is stored on the MSERV, implementation of the functionality and/or features associated with the FBUI Form uses communication between the user's client device and the MSERV via the LAN. For example, the client device message browser requests the FBUI Form from the MSERV in response to identifying a message as a VMT message because this is the form that corresponds to the VMT message type. The MSERV transfers the FBUI Form to the requesting client device as client-side code, and the client device message browser launches the form in response to the user selecting a VMT message for viewing.

As one example of an action on a voice mail message, and with further reference to Figure 12, the user may select a "Play on Phone" action using button 1206A. In response the user's client device couples to a component of the ICS (e.g., IM) using the hidden browser of the FBUI. The client device receives a popup message from the ICS via the browser link, where the popup message allows the user to choose or enter a telephone number to which he/she would like the selected voice mail message routed (the MCS may not display the popup window if, during an active Play on Phone session, a user selects a second voice mail message for play). The popup message also includes a "connect" button by which the user initiates routing of the selected voice mail message to the selected telephone. In response to selection of the "connect" button, the IM couples with an MCS, and the MCS causes the PBX to initiate a call to the telephone number selected by the user via the popup window. Upon connection of the

call from the PBX to the selected telephone, the MCS pushes the contents of the voice mail message to the selected telephone.

In response to selection of one or more actions via the action buttons, ICS components of an embodiment may transfer status/control data in response to actions selected via the action buttons. As described above with  
5 reference to Figure 14, execution of the status/control data on the local client device results in the display of a popup window. The popup is a web browser window that may present a user with information that includes prompts to select information needed by the ICS to complete an action, prompts to enter information needed by the ICS to complete an action, status of a selected action, and prompts to initiate further actions, but is not limited to presentation of only this information.

10 As an example, Figure 15 is a selection popup 1500, or ICS Window the ICS provides during execution of the Play on Phone action, under an embodiment. The ICS presents selection popup 1500 as a means by which a user selects or enters information needed by the ICS to complete an action corresponding to a selected action button. Selection popup 1500 of this example ("Voice Message Interface Call Number") allows the user to choose a telephone number (e.g., "Work Phone" 1502, "Cell Phone" 1504, "Home Phone" 1506) to which he/she would like  
15 the selected voice mail message routed. Selection popup 1500 also allows the user to enter into a popup field 1508 a telephone number to which he/she would like the selected voice mail message routed. Popup message 1500 also includes a "Call" button 1510 by which the user initiates connection and/or routing of the selected voice mail message to the selected telephone. Selection/entry of a telephone number and activation of "Call" button 1510 by the user causes the local client device to transfer the selection/entry to the ICS.

20 As another example, Figure 16 is a status popup 1600 the ICS provides during execution of the Play on Phone action, under an embodiment. The ICS presents status popup 1600 to provide the user with status information of the selected action (e.g., "Play on Phone"). Status popup 1600 of this example ("Voice Message Call Status") informs the user that current status 1602 of the call to the number previously selected/entered (Figure 15) is "Connecting", but the status may include other status information including but not limited to "dialing", "line busy",  
25 "unable to complete call", "no answer", etc. Any amount or type of status information may be displayed using any number and/or combination of popup windows in alternative embodiments.

Figure 17 is another status popup 1700 the ICS provides during execution of the Play on Phone action, under an embodiment. The ICS presents status popup 1700 to provide the user with status information of the selected action (e.g., "Play on Phone"). Status popup 1700 of this example ("Voice Message Call Status") informs  
30 the user that current status 1702 of the call to the number previously selected/entered (Figure 15) is "Line Busy". Status popup 1700 also includes a "Close" button 1702 by which a user terminates presentation of popup 1700.

Components of the ICS of an embodiment provide for storage of voice mail messages in an email messaging structure of an enterprise network as described above. The ICS components also provide a form-based interface to a client device via the messaging server. The ICS and form-based interface also provide the user with a  
35 system for accessing and directing action on the voice mail messages outside of the communication policies of the enterprise network. The form-based interface also eliminates the requirement for a dedicated client application on the user's client device.

One action the ICS allows on voice mail messages is the Play on Phone action described above. Play on Phone allows a user accessing the enterprise network via a local client device to select a telephone to which voice  
40 mail messages can be forwarded. As an example, a user who is away from the office on vacation may access the enterprise network using a portable computer (e.g., via VPN) and use the Play on Phone action of the FBUI to direct his/her voice mail messages to be routed to his/her cellular telephone. In response, components of the enterprise

network including the ICS initiate a call to the user's cellular telephone number, retrieve the selected voice mail messages, and play the messages over the call.

The operation of routing voice mail messages to a user-selected device like a telephone during the Play on Phone action includes a number of information transfers among components of the ICS and the host enterprise network. Figure 18 is an information flow in a system 1800 that includes an ICS supporting Play on Phone operations, under an embodiment. System 1800 includes an enterprise network 1801 coupled to one or more public communications networks 1850 via a PBX. Enterprise network 1801 includes at least one MCS and at least one IM, both of which couple between the PBX and a messaging and collaboration server MSERV. In an embodiment, components of the MCS and IM function collectively to form the ICS, as described herein, but operation of the MCS and/or IM is not limited to ICS functionality. System 1200 further includes at least one local client device 1870 that couples to enterprise network 1801.

The MCS and IM operate to cause FBUI 1880 to be displayed to User Z via the MSERV and local client device 1870. Public communications network 1850 and enterprise network 1801 allow information transfers between client devices 1870 that are local to enterprise network 1801 and client devices 1899 that are external to enterprise network 1801. In this example external client device 1899 is a cellular telephone of User Z, but is not so limited.

The information flow begins when the PBX transfers 1802 a data stream to the MCS. The data stream may for example be an audio stream of an incoming call from a caller to User Z. Enterprise network 1801 uses processes of the ICS described herein to cause the MCS to generate and transfer 1803 a first message to the IM. The first message includes information of the received data stream, and a type of the first message is a voice mail message, but is not so limited. The IM in turn transfers 1804 the first message to the MSERV. The MSERV generates a second message in response to properties of the first message, and transfers 1805 the second message to local client device 1870 via communications over the enterprise network. The type of the second message is an email message, where the email message includes information of the first message, but is not so limited.

Local client device 1870 receives the second message and requests 1806 form data from the MSERV in response to information of the second message. The MSERV retrieves 1807 the requested form data, and transfers 1808 the retrieved form data to local client device 1870 via the enterprise network. Local client device 1870 executes the form data resulting in presentation of FBUI 1880 on a display of user workstation 1870.

While local client device 1870 is described as being local to enterprise network 1801, it is understood that client device 1870 is not required to reside at any particular physical location within enterprise network 1801. For example, "local" client device 1870 may be a portable PC 1870 from which User Z (off site) remotely accesses components of enterprise network 1801 (e.g., MSERV) via a VPN.

FBUI 1880, as described above, presents one or more soft action buttons 1882 to User Z. This example assumes selection of Play on Phone button 1882 by User Z. Selection of Play on Phone establishes communication 1810 between local client device 1870 and the MCS via the IAC and the IM.

As an example of establishing communication 1810, selecting Play on Phone via the FBUI causes the client device local browser to call the function in the form data that corresponds to the Play on Phone operation. The form data when called generates a URL request (code) corresponding to the Play on Phone action, and the URL request expresses the ICS data that is to be transferred to the IM and the MCS. The ICS data of the URL request includes for example a message identification corresponding to the selected voice mail message (for use by the IM/MCS in accessing the voice mail message stored in the MSERV environment), user identification, and identification

information of the action desired by "User Z (Play on Phone). The URL request subsequently directs the FBUI embedded browser to the specified URL.

In response to receiving the URL request via communication 1810 from local client device 1870, the MCS determines a location of information (e.g., user information, voice mail message contents, etc.) needed to fulfill the Play on Phone operation. If the MCS determines the Play on Phone uses information from the MSERV environment, then the MCS directs 1803 the IM to pull 1812 the information from components of the MSERV environment as described above (e.g., directory service), and the IM returns 1803 the pulled information to the MCS.

In response to selection of Play on Phone, the MCS generates and transfers 1814 status/control data (e.g., HTML) to local client device 1870 via the IAC. Execution of the status/control data on local client device 1870 causes the FBUI embedded browser to display a popup window 1884.

Popup window 1884 corresponding to status/control data transfer 1814 for example is selection popup 1500 described with reference to Figure 15. This example assumes User Z directs 1816 routing of his/her voice mail messages to his/her cellular telephone 1899 by selecting "Cell Phone" in popup 1884, but the embodiment is not so limited. The selection of "Cell Phone" along with activation of the "Call" button via popup 1884 causes the form data to generate another URL request (code) that is transferred 1816 to the IM and the MCS.

Upon receiving the URL request with the selected cellular telephone number from local client device 1870, the IM uses the message identification of the selected voice mail message, user identification, and the selected cellular telephone number to transfer a request to the MCS for initiation of the Play on Phone call.

The MCS initiates a call to cellular telephone 1899 via communications 1802 with the PBX. The MCS, which manages call functions of the ICS, initiates the call by directing the PBX to place the call to cellular telephone 1899. The MCS enforces any User Z COS parameters of enterprise network 1801. The call is made via any number of telephony interfaces including T1/E1, DSE, VOIP, and analog, but is not so limited.

During the process of initiating 1802 the call, the MCS transfers 1818 status information of the call (e.g., from the PBX) to local client device 1870. The MCS transfers 1818 via the IM the status information using a status message in popup window 1884 to provide the user with status information of the Play on Phone call to his/her cellular telephone. An example of status message/popup 1884 is described above with reference to Figure 16. The status information is continually updated during the process of making the call, but is not so limited.

When the call to cellular telephone 1899 is successfully connected by the PBX, the MCS transfers 1820 the selected voice mail messages to cellular telephone 1899. Successful connection of the call causes status message/popup 1884 to close. In contrast, when the call is not successfully connected for any reason (e.g., error, line busy, no answer, call not authorized, etc.), then a status message indicating failure of the call is displayed in popup 1884 along with the "close" button.

MSERV 640 and database 644 are typically part of an enterprise MSERV environment that includes multiple MSERVs and multiple databases in the same or various locations. Typically included in the MSERV environment is a directory service that includes a database. In some configurations, the directory service may be included in database 644. Database 644 can represent storage capability for the enterprise, and can be distributed in any manner. Directory services are available from several vendors, for example Microsoft Corporation offers the Active Directory ("AD") directory service, and IBM offers a Distributed Computing Environment ("DCE") directory service.

Figure 19 is a block diagram of an embodiment in which the enterprise includes multiple MSERVs 640A through 640D, each of which communicates with a respective IM of IMs 620A through 620D. In one embodiment,

the enterprise includes an AD 701, which communicates with each MSERV 640 through a network 703 as shown. Network 703 can include one or more networks, including but not limited to local area networks (LANs) and the Internet.

AD 701 includes many objects each of which includes data for one particular network-based entity. For example, AD 701 includes user objects for each user, shown here as User 1, User 2, etc. Similarly, AD 701 includes computer objects for each computer, shown here as computer 1, computer 2, etc.

Figure 20 is a block diagram of an embodiment in which data that is particular to users of MCS 610, MCS Voice Applications, and Mobile Applications is stored in AD 701 and MSERVs 640. This facilitates integration of the users into an existing enterprise using a directory service such as AD, both in terms of integrating the user experience and integrating the administration experience, as will be further described below.

User 2 object is shown in Figure 20. A user object in AD 701 includes many "fixed" attributes for storing predefined information about User 2. There are hundreds of AD fixed attributes, such as name, phone number, mailbox location, email address, etc. However, multiple attributes of User 2 are specific to the Voice Applications of MCS 610, and are not provided for in AD, or other off-the-shelf directory services. For convenience of reference, these MCS 610 user attributes will be referred to as voice mail (VM) attributes. According to one embodiment, the VM attributes are stored in another portion of AD set aside for "custom attributes". Currently fifteen (15) custom attributes are supplied with AD and each can be used to store up to 2048 bytes of data. In an embodiment, the VM attributes are collected in a string and stored in one custom attribute. String of VM attributes provides information specific to User 2's relationship to the Voice Applications and MCS 610. For example, VM attributes include: ClassOfService (COS), which includes levels of telephone and VM service; whether an auto attendant is enabled for User 2's phone number; whether User 2 is locked out of the VM system; whether an active greeting is on; User 2's work phone extension; whether User 2's VM notification is enabled, etc. As discussed more fully below, each of the VM attributes is generated, and assembled in the string for storage in the custom attribute. Any of the available custom attributes may be used to store VM attributes.

Typically, the data included in VM attributes is infrequently changed after a user is set up and enabled by a system administrator. However, VM attributes are easily modified by regenerating them and storing a new VM attribute string in one of custom attributes. An alternative method of storing the VM attributes is to extend the AD schema by populating unused fixed attributes. The fixed attributes accommodate significantly less data, so one VM attribute might be stored in one fixed attribute. Although this is an alternative to the scheme shown in Figure 20, it is difficult or impossible to "undo" the AD schema extension once it is done, and this may be a factor in the choice of a scheme for storing the VM attributes. In addition, in enterprises that include more than one instance of AD, it is something of a challenge to keep all instances identical over time as data is updated in the extended attributes.

To further integrate the Voice Applications and other functionality of MCS 610 into the enterprise with MSERVs 640, data that is too large to store in the custom attributes (or in the fixed attributes) is stored in a database of MSERV 640. In one embodiment, each MSERV 640 includes an email database that stores user emails in designated locations. A user's email data store is sometimes referred to as a user mailbox or inbox. A user mailbox typically contains incoming email messages, sent email messages, archived email messages, etc. Retention policies for the user mailbox can be set by a combination of the user and the system administrator.

As shown in Figure 20, there may be multiple MSERVs 640A-640D. The mailbox for User 2 can be on any of MSERVs 640. In this case, User 2's mailbox, mailbox (MB) 2008, is stored on MSERV 640A. User 2

object on AD 701 includes a pointer to the location of MB 2008 in the attribute "mailbox location." This directs any inquiries or actions related to MB 2008 to the appropriate MSERV 640, database, and mailbox. MB 2008, as previously described, includes email messages 2010. MB 2008 further stores hidden messages 2012. In an embodiment, the MSERV supports a "normal" type, including email messages 2010, as well as a "hidden" message type. Hidden messages are not routinely accessible by the user through the normal user email interfaces. In an embodiment, a hidden message 2012 is used to store data used by the Voice Applications, also referred to as VM-related information. In one embodiment, the VM-related information is stored as one or more attachments to a particular user VM-related hidden message 2014. The attachments include audio files with various greetings, such as a "busy" greeting for the user's voice mail mailbox, and a "no answer" greeting for the user's voice mail mailbox.

An example of the integration of Voice Applications of MCS 610 with enterprise MSERVs 640 according to an embodiment is shown in the block diagram of Figure 21. MCS 610 accesses MSERVs 640 through IM 620. One example of a voice application is a phone caller calling the voice mail mailbox of User 2 when User 2 is on the phone. MCS 610 transmits an action via IM 620 with a request to "play no answer greeting." The transmission includes information to access User 2 object fixed attributes 2002 to determine User 2's email mailbox location. In addition, the transmission includes information to access User 2 object custom attribute 2006 and to transfer the contents of the custom attribute to MCS 610 via IM 620. When User 2 email mailbox 2008 is accessed, VM-related hidden message 2014 is opened to transfer the appropriate audio file ("no answer" greeting in this case) to the MCS for playing over the phone to the caller.

In some cases, it may not be necessary to transfer either the custom attribute or the audio file(s) from MSERV 640 because the current custom attributes and audio files are stored on the MCS. In various embodiments, the custom attributes and hidden message data are cached on the MCS, as will be discussed in more detail below.

Figure 22 is a block diagram 2200 of information transfers between the MCS and/or Cache and IM, under an embodiment. Information of the GAL, User List, Public Folder, and Personal Contacts may collectively be referred to herein as "user information," but user information is not necessarily limited to this information. As an example, the Public Folder can include shared contacts and/or other information like calendars that are applicable to all members of the enterprise.

The pushing and caching of user information by the IM and/or MCS serves to reduce the impact that losses of data would have on the ICS in providing voice mail message functions. The typical voice mail system uses database storage that is separate and independent from the database of the email system. As such, periodic synchronization operations are needed to synchronize the voice mail system database with that of the email system. If the voice mail system database becomes corrupt for any reason or fails to receive updated information during a synchronization operation with the email system database, the result is that the voice mail database is left in an unknown state regarding the validity of the data. The pushing and caching provided by the ICS reduces if not eliminates this issue because any loss of data in the MCS is efficiently overcome by the periodic and/or on-demand pushing of the user information to the MCS.

The pushing of information from the MSERV by the IM includes pushing of information including the GAL, Public Folder, and User List. The pushed information is cached by the MCS on a system or non-individual basis because this information applies throughout the enterprise. This information is pushed by the IM and cached periodically, for example at 24-hour intervals (e.g., each morning at 2:00 am), but is not so limited.

In contrast the IM pushes and the MCS caches information of the Personal Contacts on a per-user basis because this information is different for each user. The Personal Contacts pushed by the IM and cached by the MCS



periodically or on demand (e.g., at the time a user logs in to the ICS, in response to modifications of the Personal Contacts, etc.).

Cache of an embodiment may include local cache that is local to the MCS. Additionally, Cache may include a distributed cache system in which the user information is distributed among Caches of multiple MCSs. As an example, the configuration of an embodiment includes four (4) MCSs where each MCS includes components of and/or is coupled to a distributed cache system in a configuration that allows for caching of the same information on one or more of the MCSs in addition to caching information on a particular MCS and allowing other MCSs to access the cached information of the particular MCS.

The MCS may hold user information in the local cache and or distributed cache in any number of combinations. For example, the MCS may hold all user information in the local cache reserving the distributed cache for other information. Alternatively, the MCS may hold all user information in the distributed cache reserving the local cache for other information. Further, the MCS may hold the user information in both the local and distributed cache.

One scenario under which the MCS holds user information in both local and distributed cache systems is when all user information received from the IM is held in local cache while a subset of user information is held in distributed cache. This scenario may be used for example to store select user information in distributed cache, where the select user information includes information like user PIN codes and/or other parameters that may be changed by the user via the MCS. Under this scenario, the MCS pulls these user-modifiable parameters from received user information from the IM, and places these parameters in distributed cache; all other information received from the MCS is placed in local cache.

Figure 23 is a flow diagram for providing user information to the MCS from a network enterprise database, under an embodiment. The process includes interfacing with the enterprise network using the IM, at block 1102. As described above, the network enterprise includes a Database storing a groupware application and a directory service, and the directory service stores user information for use in email messaging among client devices coupled to the network.

The process continues by detecting and retrieving user information in the network enterprise, at block 2304. The detection and retrieval of user information includes detecting and retrieving user information that has been updated or modified. The process further includes pushing user information from the Database to the MCS, at block 2306. The pushing includes regular pushing at pre-specified intervals, on demand pushing performed in response to a request, and as needed pushing. The process also includes caching of user information received as a result of the pushing operations, at block 2308. The MCS and/or IM provide voice mail messaging among the client devices using cached user information, at block 2310.

The ICS of an embodiment also includes processes for caching user information received via IM push operations, where caching includes updating of user information previously cached at the MCS/Cache. The updating of cached user information in an embodiment includes detecting updated information using the IM and pushing detected information to the MCS and/or Cache as appropriate to a network configuration. The detection of updated user information includes detecting modifications to user information performed by a network administrator (e.g., administrator changes a telephone extension for a voice mail system user), for example. Once detected, the IM of an embodiment may incrementally push updated user information to the MCS, as described above. The pushing includes pushing of all user information corresponding to a user for whom the administrator has changed any component of his/her user information. Alternative embodiments however may push only revised information or information of differences (e.g., delta files/streams or difference files/streams) resulting from updates.

The IM detects updates to user information made through a user interface of the ICS, but is not so limited. In an alternative embodiment, for example, the IM detects updates made by an administrator using an interface of the directory service or other email system interface (e.g., AD interface).

Updates to user information may include any number of changes to user information. Examples of updates therefore include updating user information, enabling new users, and disabling existing users to name a few. Descriptions follow of operations for updating user information. While updates of user information are described below in the context of specific types of user information (e.g., Personal Contacts), the embodiment is not so limited. Alternative embodiments may update various other collections or sets of user information (e.g., Global Address List) using processes similar to those described herein.

One example of the IM pushing updated user information to components of the ICS (e.g., MCS) occurs when the network administrator updates user information corresponding to an existing user. The IM detects the updates to user information and pushes the user information including the updated information to the MCS. The IM may push updated user information of a single user in one push transaction and/or one data stream. Alternatively, the IM may push updated user information of any number of multiple users in one push transaction and/or one data stream.

The MCS, when receiving updated user information, identifies a user to whom the user information corresponds. The MCS also uses an entry identification assigned to user information by the MSERV to relate received user information to user information currently held in Cache. When the MCS determines that user information in Cache corresponds to received user information, the MCS updates user information held in Cache using received user information. The MCS adds received user information to Cache when the MCS fails to identify user information corresponding to received user information in Cache.

Another example of the IM pushing updates of user information to components of the ICS (e.g., MCS) occurs when the network administrator adds user information corresponding to a new user. The IM detects new user information and pushes the new user information to the MCS. The IM may push new user information of a single new user in one push transaction. Further, the IM may push new user information of any number of new users in one push transaction.

The MCS, when receiving new user information, uses the entry identification assigned by the MSERV to attempt to relate the new user information to user information currently held in the cache as described above. The MCS will be unable to identify cached user information corresponding to the updated user information because the received user information is new user information (for a new user). Consequently, the MCS adds the new user information to the Cache.

The user information may be pushed by the IM and cached by the MCS periodically and/or on an "as needed" basis (e.g., at the time a user logs in to the ICS, in response to modifications of the Personal Contacts, etc.) as described above. Upon receipt of user information, the MCS of an embodiment incrementally loads the user information for holding in the Cache. Figure 24 is an information flow diagram 1200 for incremental loading of user information, under an embodiment. The incremental loading example of this flow diagram 2400 assumes loading of Personal Contacts but may be used for any type of user information and/or other information of the enterprise network Database.

This example begins with a user's first time login to an MCS of the enterprise network. The MCS in response to initiation of the first user login event detects no cached Personal Contacts corresponding to the user, and generates a request ("GetPersonalContactsAll") for all Personal Contacts of the user. The MCS transfers the request to the directory service Database via the IM. The IM retrieves the Personal Contacts from the Database in response

to the request and pushes the Personal Contacts to the MCS along with a time stamp "TS." The MCS caches information of the Personal Contacts in an area of Cache corresponding to a user (e.g., "User Z List") along with the time stamp information (e.g., "TS").

5 The example continues when the user subsequently logs in to the MCS. The MCS in response to initiation of the subsequent user login detects cached Personal Contacts corresponding to the user, and generates a request ("GetPersonalContactsTS") for all Personal Contacts of the user modified since the date/time specified by the cached time stamp information. In contrast to a first login event, this request includes the time stamp information corresponding to the cached Personal Contacts.

10 The MCS transfers the request to the directory service Database via the IM. The IM retrieves and pushes updated Personal Contacts modified since the date/time specified in the time stamp information of the request, along with an updated time stamp. Additionally, the IM includes in the pushed information a total number ("Total") of the user's Personal Contacts found in the Database. The MCS merges the updated Personal Contacts with the cached Personal Contacts and Caches the updated time stamp. The updated Personal contacts include but are not limited to modified contacts, new contacts, and deleted contacts.

15 In responding to a request for Personal Contacts, the ICS uses the time stamp information to ensure that unsynchronized clocks between the MCS and the database system for example do not result in the exclusion of some Personal Contacts from subsequent Personal Contact update operations. In so doing the IM generates the time stamp at such time as the IM receives the request from the MCS for Personal Contacts. The IM generates the time stamp by reading the server time of the MSERV before Personal Contacts are accessed (e.g., at the time the IM  
20 begins to generate the Personal Contact stream).

In contrast, if the IM were to generate the time stamp at the time the Personal Contacts were pushed to the MCS, a scenario might arise in which the contacts are updated after retrieval of the Personal Contacts by the IM but before push of the Personal Contacts to the MCS and generation of the time stamp. Thus, generation of the time stamp before accessing the Personal Contacts prevents the scenario in which an update by the user in the interim  
25 period between retrieving an update request and time stamping the Personal Contacts results in updated Personal Contacts being missed by the IM and thus not cached in the MCS.

In addition to the time stamp, the IM includes in a response the total number of the user's Personal Contacts identified in the database as appropriate to the request. The MCS uses the total number of Personal Contacts in one or more types of incremental loading scenarios as described below.

30 One example showing use of the total number of Personal Contact is an incremental loading scenario in which a new contact has been added to the Personal Contacts. To begin, User Z's Personal Contact list includes three (3) contacts. User Z logs in to the ICS for the first time, and the MCS detects no cached Personal Contacts corresponding to User Z. The MCS requests (GetPersonalContactsAll) all Personal Contacts for User Z from the IM. In response to the request, the IM generates a time stamp (TS = 0900), retrieves the three Personal Contacts,  
35 generates a data stream including the Personal Contacts and the time stamp, and pushes the data stream to the MCS. Upon receiving the data stream the MCS caches the three Personal Contacts and the time stamp information (TS = 0900).

A new Personal Contact is subsequently added to User Z's Personal Contacts at 0930 hours. User Z again logs in to the ICS at a later time (1000 hours), and the MCS finds three cached Personal Contacts corresponding to  
40 User Z. The MCS requests updated Personal Contacts (GetPersonalContactsTS) for User Z from the IM, where the time stamp indicates the time (TS = 0900) corresponding to the currently cached Personal Contacts. In response to the request, the IM generates a time stamp (TS = 1000), determines a total number of contacts (Total = 4), retrieves

the new Personal Contact added since the time stamp of the request (0900), and generates a data stream including the new Personal Contact, the time stamp, and the total number of contacts. The IM pushes the data stream to the requesting MCS. Upon receiving the data stream the MCS reads the cache and determines the new contact of the data stream is not present in cached Personal Contacts. In response, the MCS updates the cache to include the new  
5 Personal Contact, and updates the cached time stamp (TS = 1000).

Another example showing use of the total number of Personal Contact is a scenario in which information of a contact has been modified. To begin, User Z's Personal Contact list includes three (3) contacts. User Z logs in to the ICS for the first time, and the MCS finds no cached Personal Contacts corresponding to User Z. The MCS requests (GetPersonalContactsAll) all Personal Contacts for User Z from the IM. In response to the request, the IM  
10 generates a time stamp (TS = 0900), retrieves the three Personal Contacts, generates a data stream including the Personal Contacts and the time stamp, and pushes the data stream to the MCS. Upon receiving the data stream the MCS caches the three Personal Contacts and the time stamp information.

A Personal Contact (contact #2) is subsequently updated in User Z's Personal Contacts at 1100 hours. User Z again logs in to the ICS at a later time (1230 hours), and the MCS finds three cached Personal Contacts  
15 corresponding to User Z. The MCS requests updated Personal Contacts (GetPersonalContactsTS) for User Z from the IM, where the time stamp indicates the time (TS = 0900) corresponding to the currently cached Personal Contacts. In response to the request, the IM generates a time stamp (TS = 1230), determines a total number of contacts (Total = 3), retrieves the Personal Contact updated since the time stamp of the request (0900), and generates a data stream including the updated Personal Contact (contact #2), the time stamp (TS = 1230), and the total number  
20 of contacts (Total = 3). The IM pushes the data stream to the requesting MCS. Upon receiving the data stream the MCS reads the cache and determines a Personal Contact has been updated, updates the cache to include the updated Personal Contact (contact #2), and updates the cached time stamp (TS = 1230).

An additional example shows use of the total number of Personal Contact in a scenario in which a contact has been deleted. To begin, User Z's Personal Contact list includes three (3) contacts. User Z logs in to the ICS for  
25 the first time, and the MCS finds no cached Personal Contacts corresponding to User Z. The MCS requests (GetPersonalContactsAll) all Personal Contacts for User Z from the IM. In response to the request, the IM generates a time stamp (TS = 0900), retrieves the three Personal Contacts, generates a data stream including the Personal Contacts and the time stamp, and pushes the data stream to the MCS. Upon receiving the data stream the MCS caches the three Personal Contacts and the time stamp information.

A Personal Contact (contact #3) is subsequently deleted from User Z's Personal Contacts at 1000 hours. User Z again logs in to the ICS at a later time (1100 hours), and the MCS finds three cached Personal Contacts  
30 corresponding to User Z. The MCS requests updated Personal Contacts (GetPersonalContactsTS) for User Z from the IM, where the time stamp indicates the time (TS = 0900) corresponding to the currently cached Personal Contacts. In response to the request, the IM generates a time stamp (TS = 1100) and determines a total number of contacts (Total = 2). The IM determines no contacts have been modified in the database since the time stamp of the request (0900) and in response generates a data stream including the TS and the total number of contacts (two). The IM pushes the data stream to the requesting MCS. Upon receiving the data stream the MCS reads the cache and determines a Personal Contact has been deleted by comparing the total number (two) of contacts received from the IM with the number of contacts (three) currently in the cache.

The MCS requests (GetPersonalContactsAll) all Personal Contacts for User Z from the IM in response to  
40 determining a contact has been deleted. In response to the request, the IM generates a time stamp (TS = 1100),

retrieves the two Personal Contacts, generates a data stream including the Personal Contacts and the time stamp, and pushes the data stream to the MCS. Upon receiving the data stream the MCS caches the two Personal Contacts.

Yet another example shows use of the total number of Personal Contact in a scenario in which no contacts have been added, deleted, or modified. To begin, User Z's Personal Contact list includes three (3) contacts. User Z  
5 logs in to the ICS for the first time, and the MCS finds no cached Personal Contacts corresponding to User Z. The MCS requests (GetPersonalContactsAll) all Personal Contacts for User Z from the IM. In response to the request, the IM generates a time stamp (TS = 0900), retrieves the three Personal Contacts, generates a data stream including the Personal Contacts and the time stamp, and pushes the data stream to the MCS. Upon receiving the data stream the MCS caches the three Personal Contacts and the time stamp information.

10 User Z again logs in to the ICS at a later time (1000 hours), and the MCS finds three cached Personal Contacts corresponding to User Z. The MCS requests updated Personal Contacts (GetPersonalContactsTS) for User Z from the IM, where the time stamp indicates the time (TS = 0900) corresponding to the currently cached Personal Contacts. In response to the request, the IM generates a time stamp (TS = 1000) and determines a total number of contacts (Total = 3). The IM determines no contacts have been modified in the database since the time stamp of the  
15 request (0900) and in response generates a data stream including the TS and the total number of contacts (three). The IM pushes the data stream to the requesting MCS. Upon receiving the data stream the MCS reads the cache, determines from absence of contact information that no contacts have been modified, and determines by comparing the total number (three) of contacts received from the IM with the number of contacts (three) currently in the cache that no contacts have been added or deleted. The MCS updates time stamp information of the cache (TS = 1000)  
20 using the received time stamp.

The MCS of an embodiment provides availability of voice mail applications and consequently access to voice mail messages when the MSERV is online and offline through the use of Cache. An offline condition of the MSERV is described herein to generally include conditions under which the groupware servers of the coupled network are not responding for different reasons.

25 The MCS may use multiple caching strategies to cache data based on a type of the data. Information received at the MCS is routed and held in the Cache in accordance with policies running in the state machine framework and/or the availability condition of the MSERV. Examples of information held in the Cache include but are not limited to the User List, Global Address List, information of Public Folders, information of Personal Contact Folders, voice mail message information (both the text description portion and the audio message portion of the  
30 voice mail message), recorded name of user, user greetings, and other user parameters/permissions and personal information of users (e.g., passwords or PIN codes).

Cache of an embodiment may include local cache that is local to the MCS. Additionally, Cache may include a distributed cache system in which the user information is distributed among Caches of multiple MCSs. As an example, Figure 25 is a block diagram of an enterprise network 2500-703 that includes multiples MCSs coupled to include a "Distributed Caching System" ("DCS"), under an embodiment. Figure 26 is another block diagram of an MCS having a DCS, under an embodiment. The network 2500 provides integrated voice mail and email  
35 messaging through the use of three MCSs. The number "M" of MCSs however is not limited to three and may be any number in an "N + 1" configuration. Each MCS couples to a PBX and messaging and collaboration server environment MSERV of network 1600, as described above, but is not so limited. The MSERV includes at least one  
40 Database.

Each MCS of network 2500 includes at least one "Voice Mail Application" ("VMA") that is a component of the MCS Voice Applications, as described above with reference to Figure 6. The VMA uses information of the

Cache in operations that include sending a new voice mail and/or forwarding a received voice mail. The VMA also uses Cache information in support of voice mail networking in which voice mails and corresponding information are exchanged with groupware applications of the MSERV as described herein.

The MCS includes a Voice Browser that couples to transfer communications from the PBX to the VMA.

5 The VMA couples to the Cache via a "Data Layer" and one or more APIs (collectively referred to herein as the "state machine framework"). The APIs handle the different data formats/types in use by network 2500 (e.g., greeting data, PIN code data, voice mail message data, system parameters, etc.) using any number of protocols (e.g., XML, HTTP, SOAP, etc.). Communications among the VMA, DCS, and MSERV take place via the state machine framework as appropriate to the condition (e.g., offline, online) of the MSERV.

10 The Cache includes one or more of the DCS and local cache (not shown). The DCS allows for caching of the same information on one or more of the MCSs in addition to caching information on a particular MCS and allowing other MCSs to access the cached information of the particular MCS. Information held in the DCS of an embodiment includes voice mail information (both the text description of the voice mail and the audio message), greetings (busy, extended absence, no answer, etc.), COS and other user parameters and/or permissions, and  
15 personal information of the user (PIN code, etc.).

The MCS uses the DCS to support operation of the voice mail system during periods when the MSERV is offline or otherwise unavailable. In so doing information received at the MCS is generally routed and/or stored by the MCS in accordance with policies running in the Data Layer. When the MSERV is online, information received at the MCS is held in the DCS and routed to the MSERV where it is stored in the Database. However, when the  
20 MSERV is offline information received at the MCS is held in the DCS and later transferred via the data layer to the MSERV upon its return to online mode.

An example of information received at the MCS includes a user logging in to the ICS and changing his/her greeting while the MSERV is online, the MCS routes the new greeting to the DCS and the MSERV. When the new greeting is received at the MCS and the MSERV is offline, the MCS Data Layer routes the new greeting to the DCS  
25 where it is held; the Data Layer transfers the new greeting from the DCS to the MSERV at such time as the MSERV returns online.

In addition to information described above, the MCS uses the DCS in providing voice mail services during periods when the MSERV is offline by also routing voice mail messages to the DCS and the MSERV. When the voice mail message is received at the MCS and the MSERV is offline, the MCS Data Layer routes the voice mail  
30 message to the DCS where it is held; the Data Layer transfers the voice mail message from the DCS to the MSERV at such time as the MSERV returns online. Caching received voice mail messages in the DCS allows the MCS to receive and to allow retrieval of voice mail messages regardless of a condition of the MSERV.

The MCS of an embodiment also tracks attributes (e.g., message has been retrieved by user; message not retrieved) of MCS data during MSERV offline periods. At such time as the MSERV returns online following a  
35 period of offline operation, the MCS Offline Detector identifies any messages received for a user during the period of the offline condition, retrieves the identified messages along with the message attributes, and forwards the messages to the MSERV for storage.

The DCSs of different MCSs are coupled to communicate with each other so that replication of all information of a first DCS is not required to be broadcast to every other DCS of an enterprise. Therefore, with  
40 reference to Figure 25, a voice mail message left in DCS of MCS 1 is not replicated in DCS of MCS 2 and MCS 3. Instead, the voice mail message and other information cached in an MCS are provided and/or replicated on demand.

An example of on-demand replication involves a user changing user information via an MCS. Using this example, a caller accesses MCS 1 and leaves a voice mail message for User Z, and the message is cached in DCS of MCS 1. User Z subsequently accesses the network via MCS 3 to retrieve his messages. In response to initiation of a session by User Z with MCS 3, MCS 3 communicates with MCS 1 to locate, retrieve and hold the message. MCS 3 provides the retrieved message to User Z.

In another example, a user accesses the network using MCS 2 by which she changes her personal greeting. A subsequent caller attempting to reach the user accesses the network via MCS 4. MCS 4 communicates with MCS 2 to identify and retrieve the latest personal greeting for the user. Upon determining the most current version of the user's personal greeting is cached at MCS 2, MCS 4 retrieves and caches the personal greeting of MCS 2. MCS 4 plays the retrieved greeting to the caller.

The MCS provides callers with access to the voice mail system during offline conditions by allowing callers to receive a user's personal recordings (e.g., greetings and recorded names) and to record voice mails for the user. In so doing the MCS caches the personal recordings of the user while the MSERV is online, and uses the cached recordings during MESERV offline periods.

Regarding recorded names and greetings available to callers, the personal recordings cached by the MCS include recorded names and current active greetings if recorded by the user. The MCS of an embodiment makes use of the user's mailbox number to cache and retrieve recorded names and greetings. Therefore the DCS APIs store and retrieve recorded names and greetings from the cache using the user's mailbox number.

If a user has recorded a name during the online mode, the most recent version of the name recording is played for the caller during the offline mode, regardless of the MCS to which the caller has been assigned by the PBX. Thus, if each of multiple MCSs has different versions of a user's recorded name, the DCS will locate and retrieve the most recent recording for play to the caller. When a user has no recorded name, and consequently the recorded information is not in the DCS, the caller will hear the TTS version of user's name.

If a user has recorded personal greetings (e.g., No Answer, Busy greeting) or an extended absence greeting during the online mode, the most recent version of these greetings is played for the caller during the offline mode, regardless of which MCS has been assigned to the caller. Thus, if each of multiple MCSs has different versions of a user's active greeting, the DCS will locate and retrieve the most recent recording for play to the caller. When a user has no recorded greeting and consequently a greeting can not be found in the DCS, the caller will hear the enterprise standard (default) greeting.

In support of callers wanting to leave voice mail messages during MSERV offline conditions, the DCS caches the voice mail messages. The DCS caches the voice mail message along with one or more other information items including, but not limited to, user mailbox number for use as an identifier, a timestamp for use in determining expiration, a "keep" flag that may override the expiration time to prevent expiration of a new message not yet played to a user, cache identification ("CACHEID"), and a flag to distinguish between voice mail status (e.g., played, unplayed).

The MCS of an embodiment uses the DCS to hold both the XML data and audio portions of a voice mail message for example. In addition, the MCS adds two additional parameters to the voice mail XML format. A first added parameter includes the DCS CACHEID of the audio portion of the message, while a second added parameter includes the DCS CACHEID of the XML data of the message, but the embodiment is not so limited.

During the offline mode, new voice mail messages are cached by the MCS with the "keep" flag set to true. The set "keep" flag prevents the DCS from deleting new but un-played voice mail messages from the DCS at such time as the voice mails expire. Upon return of the MSERV to the online state, all voice mail messages with the

“keep” flag set to true (along with all new voice mail messages recorded during offline mode) are transferred to the IM and the corresponding “keep” flag is set to false.

In addition to the caller access provided during offline conditions, the MCS of an embodiment allows users to login to the voice mail system to access and take actions on voice mail messages. In so doing the MCS caches the current PIN codes in the DCS for support of voice mail message access by users during offline conditions. The PIN codes are received as components of the user information pushed from the MSERV by the IM during periods when the MSERV is online as described above. The MCS authenticates the user during offline conditions using PIN codes and/or other user information held in the Cache. Once authenticated, a user may perform operations in the voice mail system like retrieving and manipulating voice mail messages.

The MCS supports user voice mail access during offline conditions through caching of voice mail messages in the DCS during the online and offline conditions of the MSERV. The MCS supports actions including but not limited to send, reply, and/or forward actions on voice mail messages to contacts of the Global Address List and Public Folders.

In the online mode received voice mails are cached in the DCS and also transferred to the IM for storage in the MSERV. The MCS caches and retrieves voice mail messages using information including user mailbox numbers. The MCS APIs thus allow caching of voice mail messages using unique mailbox numbers.

Voice mail messages recorded during the offline mode are marked as “keep” and “unread” by the MCS. The “keep” and “unread” status is a status property supported by the DCS. The DCS uses the “keep” flag to prevent deletion of voice mail messages cached during an offline mode until they have been transferred to the IM following a return to the online mode. The status property of a cached data item is transferred along with the cached item as it moves between DCS File Systems of different MCSs. If users login during the offline condition and listen to their voice mail messages, those voice mail messages are marked as “read.” Voice mail messages marked as “read” subsequently remain available for listening during the offline mode. The MCS delivers voice mail messages marked as “read” to the IM during the online mode so that the “read” messages appear as read voice mail messages in users’ inboxes.

The DCS sorts voice mail messages of a user chronologically and provides the sorted messages to the MCS. The sorting is based for example on a timestamp, but is not so limited. Further the DCS provides a list of all “unread” voice mail messages and a list of all “keep” voice mail messages for a user (identified by a mailbox number) to the MCS.

Voice mail messages cached by the MCS during the online mode or previous offline modes may expire based on corresponding timestamp and voice mail message expiration time information. The expiration time of an embodiment is configurable and may be set to a time period as appropriate to a network configuration (e.g., 72 hours). The MCS provides a copy of expired voice mail messages in a user’s desktop inbox folders when the expired messages have not been previously deleted by the user.

The MCS Offline Detector detects the MSERV return to online condition and in response switches to the online mode. The MCS transfers all voice mail messages cached during the offline mode to the IM in response to the MSERV returning online.

Figure 27 is a block diagram of network that includes two (2) MCSs and a DCS, under an embodiment. The network includes MCS 1 and MCS 2, but is not limited to two MCSs. Each MCS includes a “Caching Server” that couples to a “File System.” Each Caching Server communicates with a server of the host MCS as well as with Caching Servers of other MCSs as appropriate to the network configuration. The Caching Servers handle caching



requests, cache management, caching algorithms, and inter-server communication, and in so doing collectively form the DCS of an embodiment.

Local operations of the DCS include information transfers between the Caching Server and the host MCS via a "DCS Client" and "DCS Stub" of the host MCS and a "DCS Skeleton" and "DCS Protocol Server" of the Caching Server. The DCS Client resides on the MCS server (not shown) and couples to the DCS Stub, which is a library that enables communication between the Caching Server and applications of the MCS using communication protocols as appropriate to the network configuration. The DCS Stub couples to the DCS Skeleton of the Caching Server. The DCS Skeleton handles the network communication layer and couples to the DCS Protocol Server, where the DCS Protocol Server handles logic of communications between the MCS and DCS.

The Caching Server further includes a "Cache Manager" that couples to a "File System" via a "File Cache Manager." The Cache Manager, which couples to the DCS Protocol Server, manages information held in the File System using knowledge of the structure of the File System. Therefore, the Cache Manager directs information into the File System as appropriate to the information type. The Cache Manager performs for example indexing, cache chaining, locking, and crash recovery functions. The File Cache Manager performs physical file layout management and directory management of the File System, but is not so limited.

The DCS of an embodiment also includes a "Garbage Collector" that couples to the Cache Manager and the File Cache Manager. The Garbage Collector performs on-demand and/or periodic purging of expired caches from the File System.

Distributed operations of the DCS include information exchanges and file transfers between the Caching Server of a host MCS and the Caching Servers of one or more different MCSs. Figure 28 is an information flow 2800 for inter-node communications between Caching Servers ("CS") of different MCSs, under an embodiment. Generally a Caching Server receiving a request for information ("Requesting CS") generates a query or request 2812 for an object list corresponding to a specified mailbox. The request for information comes, for example, from a user connecting to the MCS that hosts the Requesting CS in order to retrieve voice mail messages. The Requesting CS transfers 2802 the request to one or more Caching Servers as appropriate to the network configuration ("Responding CS"). Responding CSs caching information corresponding to query 2812 respond to the Requesting CS. Responding ones of the Responding CSs return 2804 a stream 2814 that includes a list of item identifications, URLs, file sizes, and timestamps corresponding to the user's voice mail messages currently cached in the Responding CS.

The Requesting CS processes stream 2814 and returns 2806 a fetch request 2816 to the Responding CS. Fetch request 2816 includes identification information (e.g., CACHEID) for an item requested by the user. The Responding CS in turn transfers 2808 object data stream 2818 corresponding to fetch request 2816 to the Requesting CS.

Taking two MCSs as an example, and with reference to Figure 27, each MCS hosts a Caching Server, and the two Caching Servers form a DCS. Caching Server 1 ("CS1") of MCS 1 couples for distributed communications with Caching Server 2 ("CS2") of MCS 2 using a Server-to-Server Communication System ("SSCS") to form the DCS. The SSCS of a Caching Server generally includes a Server-to Server Skeleton ("SSS"), a Server-to Server Stub ("SSST"), a Group Communication Server ("GCS"), and an Inter-node Protocol Server ("IPS").

Each Caching Server may couple to the Caching Server of one or more other MCSs of a configuration using the respective SSS and SSST components. During distributed communications for example, the SSS of CS2 couples to communicate with the SSST of CS1. The SSS of a Caching Server (CS2) couples communications with other Caching Servers locally to the Cache Manager (CS2) using the IPS (CS2). The GCS (CS2) couples the DCS

Protocol Server (CS2) to the SSST (CS2), where the SSST (CS2) may couple to the Caching Server of yet another MCS (not shown).

Communications between the Caching Servers in support of information retrieval from the DCS include scenarios under which an MCS requests information from other MCSs of a configuration in order to provide cached information to users and callers. DCSs of an embodiment use at least two types of CS-to-CS communication for information retrieval, but are not so limited. A CS of a requesting MCS for example uses a first type of communication to retrieve identification information of a message from other CSs, and uses a second type of communication to retrieve message contents from a File System of the other CSs.

For example, User Z accesses the host network in order to retrieve her voice mail messages, and the PBX routes her call to MCS 1. MCS 1 in response requests information from other MCSs (MCS 2) as to voice mail messages received and cached for User Z at each of those other MCSs. The DCS Protocol Server (CS1) of MCS 1 generally receives and interprets the request from the requesting host MCS 1 and in turn retrieves a list of messages from the local Cache Manager (CS1). The DCS Protocol Server (CS1) also couples the request through a local communication layer to the GCS (CS1). The GCS has knowledge of the number of other MCSs in the host network configuration and as such generates requests to the other MCSs for lists of messages. The GCS (CS1) sends the request messages to the other MCSs of the configuration using the local SSST (CS1). The local SSST (CS1) therefore couples the request to the SSS (CS2) of MCS 2 in this example.

The SSS (CS2) of other MCSs receives the request for a list of messages cached in CS2 for User Z, and the SSS (CS2) transfers the message using local communications to the Cache Manager (CS2) via the IPS (CS2). The Cache Manager (CS2) retrieves the list via communications with the File Cache Manager (CS2) and File System (CS2). The retrieved message list includes metadata including a list of cache identifications (CACHEID) of User Z's voice mail messages but is not so limited. The retrieved list is transferred to the requesting DCS Protocol Server (CS1) via the IPS (CS2) and SSS (CS2) of the Caching Server receiving a request, and the SSST (CS1) and GCS (CS1) of the requesting Caching Server. The retrieved message information from other MCSs (MCS 2) is cached in the File System (CS1) of the requesting Caching Server (via the local Cache Manager and File Cache Manager) and forwarded to the requesting MCS (MCS 1).

At such time as User Z requests play of a message cached in a remote CS (CS 2), the DCS uses a second type of communication to retrieve message content from a File System of other CSs. The local DCS Protocol Server, which is "local" to the MCS receiving the request (MCS 1), generally receives and interprets the request from the requesting host MCS 1. The DCS Protocol Server (CS1) couples the request to a local "Fetching Server" (CS1). The Fetching Server couples the request to the remote Cache Manager (CS2) via a "Web Server" of MCS 2, where "remote" components are components of a different MCS than the MCS at which the request is received. The remote Cache Manager (CS2) retrieves the message contents via communications with the remote File Cache Manager (CS2) and File System (CS2). The retrieved message content is transferred to the local DCS Protocol Server (CS1) via the remote Web Server (MCS 2) and the local Fetching Server (CS1). The retrieved message content from the remote DCS (MCS 2) is cached in the local DCS (MCS 1) and forwarded to the requesting component (MCS 1).

The cache identification (CACHEID) described above is used as metadata for storing and identifying voice mail messages and as such is the identification that uniquely identifies one item of information in the DCS. The CACHEID allows the DCS to determine the type of an item and when the item was created to name a few. The DCS generates CACHEID using a "gen\_id" function.

One example of CACHEID includes the identification "ADOMO-VMX:1.0\_100\_foo\_192.168.1.132\_1234\_1086982006\_15." Referring to this example, CACHEID of an embodiment includes information like item type ("VMX" represents voicemail item, XML portion; item types can also represent PINs, greetings, etc.), version number ("1.0"), IP address representing the MCS that generates the  
5 CACHEID (used to prevent multiple-MCS collisions) ("192.168.1.132"), mailbox number ("foo"), process identification representing the DCS that receives the request ("1234"), numeric representation of item type ("100"), timestamp indicating time when VM message generated ("1086982006"), and counter value ("15") for use in avoiding collisions. The CACHEID however is not limited to the information of this example.

The MCS of an embodiment may alert users to any transition between the online and offline modes in  
10 support of the respective online and offline states of the MSERV. The alert may take any number of forms including audio notification played to the user during the current active session, but is not so limited as the MCS may use any of its notification functions to notify the user. As an example, a user accessing the MCS during the online mode would be alerted by a voice application of the MCS when the MSERV transitioned to the offline mode during the user's current active session. Following notification of this transition, the user may continue to access  
15 and retrieve voice mail messages from the cache for example. The MCS of an embodiment may provide full integrated message functionality during the offline mode, but alternative embodiments may provide limited access to features as appropriate to a network configuration. For example, in one embodiment the user may not be able to carry out all user configuration actions as well as listen to messages that are not held in cache.

Figure 29 is a block diagram of a system 29 that includes multiple Sites (defined herein) and multiple  
20 components, under an alternative embodiment. System 29 includes multiple Sites, some of which may have multiple MCSs, IMs, private communication networks and MSERVs. As shown, system 29 includes MSERV 2990 and MSERV 2991 communicating via a network 2992, which may comprise any of a public network, such as a PSTN, or private communications network or other network. The MSERVs are coupled to one or more IMs. For example, as shown here, MSERV 2990 is coupled to IMs 2985 (IM1 and IM2), and MSERV 2991 is coupled to IMs  
25 2986 (IM3 and IM4). The IMs are coupled to one or more MCSs. For example, as shown here IM1 is coupled to MCS1, MCS2, and MCS3; IM2 is coupled to MCS2, MCS3, MCS4 and MCS5; IM3 is coupled to MCS6 and MCS7; and IM 4 is coupled to MCS8. The MCSs are coupled to private communications networks. As shown here, MCS1, MCS2, MCS3, MCS4 and MCS 5 are coupled to private communications network 1 2960A; MCS6, and MCS7 are coupled to private communications network 2 2960B; and MCS8 is coupled to private communications  
30 network 2 2960B and private communications network 3 2960C.

Thus, Figure 29 shows a system 29 that is scalable in a number of different dimensions, according to various embodiments of the invention. Two MSERVs are shown coupled by a network. This configuration allows for sharing of voicemail messages, user lists, global address lists, distribution lists and public folders between the various MSERVs that connected by a network and which may be placed at the same or different locations.  
35 Additionally, use of multiple MSERVs allows for scaling of the overall system through the increased capacity provided by the multiple MSERVs.

Multiple MCSs are shown. Increased number of MCSs can help to increase overall system capacity and/or redundancy by providing increased number of ports, storage, and processing capacity. According to an embodiment of the invention, information on the MCSs is derived from the MSERVs and automatically cached on the MCSs.  
40 This allows for easy deployment of new MCSs by which the data and configuration settings for the new MCSs are acquired from the MSERV(s) and/or caches of other MCSs. Additionally, an MCS may be coupled to more than one private communications network. In some cases an MCS may operate with multiple private communications

networks simultaneously. Also, an MCS that is coupled to multiple private communications networks may continue operation with a non-failing private communications network in the event that one of the private communications networks to which the MCS is coupled fails. In one embodiment, the MCS that is coupled to multiple private communications networks operates with at least one of the private communications networks, but begins to operate with another, non-failing private communications network in the event that a private communications network to which the MCS is coupled fails.

Multiple IMs are shown in Figure 29, which help to support the capacity of additional MCSs. The multiple IMs also may provide fail over support for each other in the event that one of the IMs fails.

In Figure 29, the equipment and users associated with a particular private communications network referred to as members of a "Site." Accordingly, a user may have a Site identification. The Site identification may be used to filter user information associated with a particular Site from the a broader set of user information stored on the MSERV servicing multiple Sites. Additionally, Sites may be combined into auto attendant groups. The auto attendant groups are Sites that share a common dial plan. For example, members of an auto attendant group may be able to place calls using extension numbers instead of full numbers.

According to an embodiment of the invention, various subsets of users may be defined from among the users in an MSERV or set of networked MSERVs. Such subsets of users may be defined by a Site identification. In this way, various subsets of users may be associated with different respective private communications networks, such that the users' access to respective Sites within a network of MSERVs depends on the users' membership in the various defined subsets of users. For example, members of a subgroup of users associated with a particular Site may be able to use functions such as message waiting indication and control of messaging actions at their associated Site but not at other Sites.

As described herein, the ICS includes: a network that includes a database storing a groupware application and a directory service, wherein the directory service stores user information for use in providing messaging of a first type among client devices coupled to the network; a messaging communication server (MCS) that couples to the network and to at least one communication network, wherein the MCS provides messaging of a second type among client devices coupled to the network; and an interface module that couples the MCS to the network, the interface module pushing user information from the database to the MCS, wherein the MCS uses the pushed user information to provide the second type of messaging, wherein the MCS and the interface module provide integration of the messaging of the second type into the network.

In an embodiment, the ICS further comprises a cache that couples to the MCS, the cache holding the user information pushed by the interface module.

The interface module may detect changes to the user information and incrementally pushes the changes to the MCS.

In an embodiment, the MCS includes at least one voice application and various application programming interfaces ("APIs"); and messaging of the second type includes voice messaging, and wherein the cache further couples to a state machine framework, wherein communications among the at least one voice application, the cache, groupware application and a directory service take place via the state machine framework and the various APIs as appropriate to a state of the groupware application.

The state of the groupware application includes an online state in which the groupware application communicates with the MCS, and an offline state in which the groupware application does not communicate with the MCS.

In an embodiment, the voice applications include a voice mail messaging application, and wherein the user information includes user information particular to the voice messaging application.

When a new member enters the enterprise, the enterprise administrator must setup the new enterprise member in the groupware application (e.g., MS Exchange). In this context, the enterprise administrator is an information technology ("IT") professional familiar with the groupware application and directory service of the enterprise. The new member may or may not be enabled as a user of the MCS and voice applications as described herein. As part of the usual setup process, for example in an enterprise with MS Exchange and AD, the administrator interacts with an AD setup interface, which is a graphical user interface ("GUI") supplied by AD to populate a user object for the new member. According to an embodiment, the administrator may wish to make the new member an MCS "user" as described herein, with access to the MCS and its voice applications. In that case, the process for setting up and enabling a user is integrated with AD such that the administrator can use the AD setup interface in the accustomed manner.

In one embodiment, the user object is populated first and the new member is enabled as a member of the enterprise. Then, the administrator may use the AD setup interface to enable and setup the member as a user of the MCS and voice applications as described herein. In one embodiment, an option to setup the member as an MCS user is included in the normal member setup process, and the administrator merely selects the option to initiate an MCS setup/enable user process through the AD interface. In one embodiment, the administrator selects a user, and an MCS setup/enable menu comes up for the user, including a wizard to guide the administrator through the process.

Figure 30 is a block diagram of the MCS setup/enable user process of one embodiment. A user interface (UI) 3010 is shown with the option to "setup/enable User 2." When the option to "setup/enable User 2" is selected, as shown at 3012, a source configurable rules framework ("SCR FW") 3016 is initiated. SCR FW 3016 requests User 2 object from AD 701 at 3014. User 2 object is returned to SCR FW 3016 so that the information about User 2 can be used to generate MCS specific information needed to setup and enable User 2. SCR FW 3016, as further described below, receives source code file and/or a compiled code file. After processing the received file(s), SCR FW 3016 executes the code contained therein, which includes multiple methods. Each of the methods generates an MCS user attribute according to a rule. The MCS user attributes include user information that is specific to the MCS and voice applications, and therefore is not included in the typical AD user object. However, embodiments allow the MCS-specific user information to be stored and accessed in the same way as the typical AD user attributes.

Referring again to SCR FW 3016, the methods executed include GenerateVMailboxNumber, GenerateExtensionNumber, GenerateSiteID, and more. As stated, the methods each generate a result according to a rule. GenerateSiteID method 3018 is shown as an example. GenerateSiteID method 3018 includes "SiteID = AreaCode." The area code information is obtained from User 2 object. GenerateSiteID method 3018 uses User 2 attributes to generate a result, but the invention is not so limited. The rules may be written to derive a result from any available information.

When all of the methods have been executed and all of the results have been obtained, User 2 attributes are assembled at 3024 into a custom attribute. For example, in one embodiment, the custom attribute is an XML stream such as <ClassOfService = "0" AutoAttendantEnabled = "False" LockedOut = False ActiveGreeting = "System" WorkExtension = "1234" VoiceMailNotificationEnabled = "True" VoiceMailBoxInitStatus = "0"...>. The custom attribute is then stored 3026 in AD 701 in User 2 object. SCR FW 3016 then sends a "done" indication to UI 3010. In various embodiments, the UI may prompt the administrator to enter information that cannot or should not be generated by SCR FW 3016, such as an initial password for User 2.

Figure 31 will now be described in order to explain embodiments of SCR FW 3016 in more detail. Figure 31 is a block diagram of SCR FW 3116, UI 3110, and two of the file types accepted by SCR FW 3116. The file types accepted by SCR FW 3116 include a dynamic link library (“.dll”) file, “user.dll” 3102 and a source file 3104. Compiled files may also be in alternative formats to .dll. Source files can be in a variety of formats, such a C Sharp (“CS”) or Visual Basic.net (“vb.net”). In one embodiment, SCR FW 3116 is a .net framework that performs reflection on assemblies to determine various metadata, and also invokes the assemblies. In one embodiment, user.dll file 3102 is an MCS user setup/enable file that is shipped with UI 3110 and SCR FW 3116 to an enterprise that is installing an MCS as described here. User.dll file 3102 includes all of the methods and parameters anticipated to be needed for the enterprise. For situations in which the enterprise cannot complete the MCS user setup/enable with user.dll file 3102 as written, the administrator can write a source code file 3104 to perform the “missing” or “defective” method. Source code file 3104 is compiled by SCR FW 3116 and thereafter reflected upon and invoked in a similar manner to user.dll file 3102. In various situations, either one of files 3102 and 3104 can be used by the administrator in setting up and enabling a user, or both of them can be used.

Allowing the administrator to code methods with a commonly used language makes it simple for the administrator on-site at the enterprise, or alternatively a field technician visiting the site, to modify or customize the setup/enable process to suit the enterprise. This is much easier and more efficient than, for example, requesting that a new .dll file be created for one enterprise. Creating a new .dll file for one enterprise has the disadvantage of possibly causing the proliferation of multiple incompatible versions of the user setup/enable .dll file. In addition, requesting and receiving a new .dll file takes additional time and expense. Because SCR FW 3116 can receive and process both .dll files and source code files, the administrator has enhanced capability and flexibility in designing the setup/enable process for MCS users. This is in addition to the advantage of accessing this capability through a known and previously used interface, e.g., an AD administrator interface.

In one embodiment, SCR FW 3116 further includes the capability to detect a requirement for run-time parameters, and the capability to request, receive and use the run-time parameters, as will be explained in more detail with reference to Figure 32. Figure 32 is a flow diagram showing an MCS user setup/enable process 3200 according to an embodiment. At 3202, the administrator selects “setup/enable User 2” from UI 3110. SCR FW 3116 gets User 2 object from AD 701 as shown at 3204. The dashed line around 3205 indicates that if there is a source file 3104, it is compiled at his time. An example file “Admin\_Specific.net” is shown, but the invention is not so limited.

At 3206, SCR FW 3116 reflects on user.dll file 3102 or source file 3104 as applicable. SCR FW 3116 determines during reflection whether there are any attributes required and/or not present for use of the methods. If there are any “missing” or “defective” attributes, SCR FW 3116 queries the administrator at 3210 for the parameter, and receives the parameters from the administrator. “Missing” parameters could be parameters missing because of some inconsistency or error, and/or they could be parameters that cannot be known until run time and must be supplied at that time.

At 3212, SCR FW 3116 invokes the methods contained in the file just reflected upon, using the supplied parameters, if applicable. At 3214, a first method of the file is run, and a result is output at 3216. Then, SCR FW 3116 determines whether all of the methods have been run at 3218. If all of the methods have not been run, the next method is run at 3220, and 3216 and 3218 are repeated until all of the methods have been run. The results are assembled into the custom attribute at 3222, and the custom attribute is stored to AD 701 at 3224.

Class of service (“COS”) is one of the MCS or voice application-specific attributes assigned to an MCS user. In telephony, a COS includes values for various permissions or levels of service, such as dialout privileges

(none, local, long distance, international, etc.), voice mailbox privileges, voice mailbox capacity, length of recorded greeting, and so on. Traditionally, several telephony COSs are defined by a telephone administrator (who is distinguished from the enterprise IT administrator as described herein), and each enterprise telephone system user is assigned one of the defined COSs. In traditional enterprises, this assignment is accomplished when the telephone administrator sets up a new user of the enterprise telephone system. This is a relatively rigid scheme. For example, in order to grant a telephone user one additional privilege not included in his or her current COS typically requires the creation of a new COS.

In the enterprise software world, groupware applications typically have a different method for assigning enterprise network resource using a host of capabilities and privileges, including specific login times for the user's computer, the network printers to which the user is allowed to print, the shared files and applications the user is allowed to access, etc. In various embodiments, the MCS user setup/enable process integrates with the existing groupware application schemes for assigning permissions, levels of service, privileges, etc. pertaining to telephony and voice applications, which are not traditionally provided for. In one embodiment, the MCS user setup/enable process integrates with the MS Exchange/AD Group Policy ("GP") schema, which is considerably more flexible and expandable than the telephony COS method. As such, the enterprise IT administrator is able to assign Group Policies in the accustomed manner, but the Group Policies effectively encompass all of the required telephone COS concepts. According to the embodiments described, a telephone administrator is not needed to define or assign COSs to users of the enterprise telephone system.

In AD, a Group Policy Object ("GPO") is a policy setting. The setting of a default desktop for instance, is configured in a GPO. A Group Policy Container is an AD object where GPOs are linked. Only sites, domains, and organizational units ("OUs") can have GPOs linked to them. The Group Policy ("GP") schema is hierarchical, with multiple OUs to a domain, and multiple domains to a site. There is typically some overlap resulting from the ability to create multiple settings in multiple areas for an enterprise structure. There are therefore inheritance rules in GP to avoid conflicts and assure proper function. GP is applied to a user or computer object in a specific order, e.g., local computer, site, domain, domain controller(s), and OU. In the case of nested OUs, GP is applied in order from the highest level OU, or parent, down to the lowest level OU. Examples of OUs include geographical locations and organizational units (e.g., sales and development). Very generally, OUs lower in the hierarchy inherit their GPO(s) from above in such a manner that once a privilege is restricted at a point in the hierarchy, it is restricted the rest of the way down. For example, if executive management is above sales, which is above development, executive management may have unlimited dialout privileges. Sales may have dialout to exclude international calls. Development therefore cannot have international call privileges, but may have an even lesser dialout privilege, such as local only.

Thus, a member of the organization has an effective GPO (EGPO) that needs to be calculated from the schema. In one embodiment, the AD GP schema is extended to include telephony-related and voice mail-related permissions, levels of service, and privileges. This allows the GPO process for the administrator to remain unchanged. In one embodiment, the enterprise is supplied with a compliant GP text file template that is specifically written to include all of the telephony-related and voice mail-related information currently lacking in groupware applications. The template is mapped into a UI that prompts the administrator to "add GP." When the template is brought up in the UI, it lists the various telephony-related and voice mail-related items, or COS parameter. Each item includes its own selection box or button which the administrator can check or click to select all the values to be set for different groups. In one embodiment, the COS parameters include: MaxNumber of Greetings; ExtendedAbsenceGreetingAllowed; ExtendedAbsenceGreetingLength; MaxGreeting Length;

ExtendedAbsenceGreetingBlockMsgRecord; GeneralDialoutPrivileges; PasscodeAgingEnabled; AccessToSystemDistLists; AllowToSendBroadcastMessage; ActiveMobileNumber; MobilePhoneNumber; ExtendedAbsenceText; MailboxLanguage; CallerFirstLanguage; CallerSecondLanguage; CallerThirdLanguage; and AttendantSchedule.

5 This process is a one-time setup process. The administrator receives the template and goes through the process of "extending" GP once. After that, the user setup process is the same as before for the administrator.

In one embodiment, the EGP is retrieved from AD and stored in a cache on the MCS as described in more detail below. In another embodiment, as part of the process of populating the custom attribute as described with reference to Figure 32, the EGP is retrieved, and stored in the custom attribute as "COS." In yet another  
10 embodiment, the enterprise is relatively simple, and each user has a GPO assigned. In that case, the GPO for the user can also be cached and retrieved in the same manner as previously described with reference to the EGPO, but the amount of time required to retrieve the GPO will be reduced.

When a user logs into the MCS, the most recent version of the VMLIST should be available to the user. With reference to Figure 33, in one embodiment, the user accesses his or her voice mailbox by logging into MCS  
15 3313. The user is prompted to enter his or her mailbox number and password. The information entered by the user is authenticated by comparison with user data. The user data may be held in the Cache or stored on a database 3344 of the MSERV. Upon request from the MCS, the IM can generate a VMLIST 3309 from the list of messages in the user's mailbox. In order to minimize the time that the user may wait for the VMLIST to be returned from the database 3344, a fetch of the VMLIST may be performed in parallel with the authentication process. As shown in  
20 Figure 33, when the user logs into MCS 3313, the user enters his or her mailbox number and password. The mailbox number is sent to IM 3311 for it to fetch the VMLIST in parallel with the password query/entry process and authentication process. Alternatively, the mailbox number can be derived from the caller identification information provide by the private communications network to the MCS. The VMLIST may be returned before the authentication process is complete, such that the user does not have to wait additional seconds to begin manipulating  
25 the voice mailbox. Even if the VMLIST is not returned by the time the authentication process is complete, it is still useful to the user, since the time from calling in to presentation of the VMLIST is shorter than it would have been, if the VMLIST was not fetched until the user was authenticated. This is helpful because phone callers are very intolerant of delay. This is in contrast to, for example, computer users performing similar log on or authentication functions.

30 When user logs into the MCS 3313, various information about the user is immediately collected by the MCS in anticipation of performing actions that may be directed by the user. In various circumstances, the user information may be held on the MCS 3313 or may be stored on the MSERV. In any case, the MCS 3313 often requests user information, which can include as the VMLIST, and appointment list etc. from the MSERV.

One of the MCS 3313 requests is to retrieve calendar appointments and meetings for the specific date on  
35 which the user logs in. This information may take some time to look up in the MSERV and transfer to the MCS 3313. Typically, particular information, such as calendar information is cached or indexed by the MSERV speed up lookups of this information. Information of a transient character, such as daily calendar appointments, is typically deleted periodically. For this reason, an initial request to look up such information is usually slower than subsequent requests to look up the information.

40 Based on this knowledge, in one embodiment, the MCS 3313 initiates a request for transient-type information, such as calendar appointments, as soon as the MCS 3313 has received enough information about a user to "guess" who the user is. This guess may be based on a user ID, or the user saying his/her name. Using the



example of calendar data as the transient data," at this point the MCS 3313 makes a request to the IM 3311 to "warm up" the calendar. In an embodiment, this means that the IM 3311 makes a request to the MSERV for appointments on the date of the login, but the result is ignored. No information is actually returned to the MCS 3313 at this point due to security reasons (user is not yet authenticated), but the MCS 3313 has a "head start" in accessing the  
5 calendar. If the user takes longer to authenticate than the time required for the warm up request, the actual calendar information for the day is going to be the second request for the same information. Because the MSERV has now cached information about this query, the results of subsequent queries are obtained within a fraction of a second.

In other embodiments, the VMLIST is held in the Cache. One such embodiment is shown in the block diagram of Figure 34. A messaging server environment, MSERV environment 3440, includes a User N mailbox  
10 3441 and an Event Listener 3442. When an event occurs in User N mailbox 3441, it is recorded in Event Listener 3442. For example, events recorded include, but are not limited to events such as the user deleting a message, reading a message, saving a message, moving a message, etc. In an embodiment, Event Listener 3442 communicates with IM 3411 each time a new event is recorded. Alternatively, Event Listener 3442 may choose to collect events over a given time period and communicate the event collection to IM 3411. IM 3411 passes the event  
15 to an Event List Service. In addition, the receipt of an event by MCS 3413 alerts MCS 3413 that it should request an updated VMLIST from MSERV environment 3440, and cache the returned VMLIST. Alternatively, the Event List Service sends only the updates to the VMLIST to the Cache. In either case, a current VMLIST is almost certain to be held in the Cache at any time the user accesses MCS 3413.

When user logs into the MCS 3313, various information about the user is immediately collected by the  
20 MCS in anticipation of performing actions that may be directed by the user. In various circumstances, the user information may be held on the MCS 3313 or may be stored on the MSERV. In any case, the MCS 3313 often requests user information, which can include as the VMLIST, and appointment list etc. from the MSERV.

Other functions of the Event List Service include sending "message waiting indicator off" and "message  
25 indicatory on" notifications to client devices 3470, sending an email notification to client devices 3470, sending an SMS message notification to client devices 3499, and sending an email notification to client devices 3499. In one embodiment, client devices 3499 are on a public communications network 3450, and MCS 3413 communicates with public communications network 3450 through a private communication network 3460, but the invention is not so limited.

An example of a message waiting indicator in an embodiment is a light emitting diode ("LED") on a user's  
30 office phone 3470. For various other client devices 3470 and 3499, different message waiting indicators may be used. In one embodiment, when a new voice message is left in the user's voice mailbox, the event is detected, and a signal is sent to the client to turn the message waiting indicator on. Similarly, when the user accesses a voice mail message, that event is detected, and a signal is sent to the client device to turn the message waiting indicator off.

In various embodiments described herein, a user may access his or her voice mailbox from several sources,  
35 such as the enterprise email application (e.g., MS Outlook), the MCS 3413, web applications such as Outlook Web Access ("OWA"), and other client devices, to name a few. Therefore, the user may not access his or her voice mailbox through the MCS 3413 to access and manipulate existing voice mail messages. For this reason, the MCS 3413 may not have direct information about an event that it has not participated in. However, because the event listener detects all such manipulations (e.g., read, delete, save, etc.) and the MCS 3413 is informed as described  
40 herein, the client devices will reflect the current state of messages in the voice mailbox.

In another embodiment, IM 3411 polls the MSERV environment periodically to determine whether updates to the VMLIST have occurred. If an update has occurred, a new VMLIST is transferred to and held in the Cache.

In another embodiment, the user goes through the VMLIST item by item (VMSG by VMSG) and performs an action for each voice mail item. For example, for each VMSG, the user may delete, save, forward, etc. All of the VMSGs on the VMLIST may be held in the Cache, a subset of the VMSGs on the VMLIST may be held in the Cache, or none of the VMSGs on the VMLIST may be held in the Cache. In order to minimize delays that may occur when a VMSG must be fetched from the MSERV (in the manner previously described), a concurrent fetch is performed on one embodiment. Concurrent with the user performing an action on a VMSG, the process of obtaining the next VMSG is initiated. As previously described, this begins with a lookup in the Cache to determine whether the VMSG is held there, and if it is not, a fetch to the MSERV is performed.

Regarding actions taken by the MCS following recording of a voice mail message when the MSERV is offline, a component of the MCS (e.g., Offline Detector) detects the MSERV is offline. The MCS holds the recorded voice mail message in the Cache for a pre-specified period of time in response to detecting the MSERV state as offline. At such time as the MCS detects the MSERV is online, the Groupware Connector pulls the voice mail message from the Cache and transfers the recorded voice mail message via the IM to the MSERV where it is stored in the Database.

An embodiment of the invention is directed to a voicemail system. A caller using the system is able to leave a voicemail message request that the voicemail be private. When the voicemail is private, this may mean various things. For example, the system may be configured such that the voicemail may be read by only the recipient. The voicemail system uses an e-mail system to store the voicemail and provide the voicemail to the recipient. The system prepares an e-mail message containing the voicemail. Before placing the voicemail message into the e-mail system and providing the message to the recipient, the e-mail containing the voicemail is protected using a protection scheme of the e-mail system. The protection scheme enforces the privacy requested by the caller. Then the e-mail containing the voice message is provided to the recipient. The e-mail system's protection scheme prevents the recipient of the e-mail from taking an action not allowed for the private message. For example, if the recipient attempts to forward the e-mail, the e-mail protect scheme will not take this action with the e-mail.

Figure 35 is a block diagram of a communication system connected to an e-mail system. A caller is able to leave a private voicemail for a user. The voicemail is converted to an e-mail in the e-mail system and the privacy of the message is maintained in the e-mail system.

Figure 35 includes a telephone device 3502, public communications network 3503, private communications network 3504, voicemail system 3505, voicemail system to e-mail connector 3507 and e-mail system 3510. Also shown are user A, user B, user C and user D. Voicemail system 3505 includes voice message 3506. Voicemail system to e-mail system connector 3507 includes voicemail to e-mail conversion block 338 and protect e-mail block 339. E-mail system 3510 includes e-mail 3511 and policy enforcement block 3512. User B is associated with computer system 3513 and telephone device 3514. Also shown is message 3519 displayed from computer system 3513. Message 3519 includes play button 3512 and disabled forward button 3521. Also included in message 3519 is a text to the user 3522. User C is associated with computer system 3515 and telephone device 3516, and user D is associated with computer system 3517 and telephone device 3518.

Private communications network 3340 is coupled to the public communications network 3303. This allows for the private communications network 3304, such as a PBX found in an enterprise to connect users of the private communications network to communicate with others in the outside world. Private communications network 3304 is also coupled to the voicemail system 3305 and to various users of the private communications network 3304, such as user B, user C and user D. According to various embodiments of the invention, private communications network 3304 may couple to telephone devices of these users such as telephone devices 3514, 3516 and 3518. According to

other embodiments of the invention, private communications network 3304 may also be coupled to the computer systems of these users, such as computer system 3513, computer system 3515 and computer system 3517.

Voicemail system 3305 is coupled to voicemail system to e-mail system connector 3307, which is coupled to e-mail system 3510. E-mail system 3510 is coupled to the various computer systems of users, such as computer system 3513 of user B. Thus, the various users are able to receive the voice communications from private communications network 3304 over the respective telephone devices and e-mail and other data communication from e-mail system 3510 over the respective computer systems. Various forms of integration between private communications network 3304, voicemail system 3305 and e-mail system 3510 are possible according to various embodiments of the invention.

Figure 35 shows a caller, user A, leaving a message to be stored in the communications system. The caller leaves the message and marks the message private. The e-mail system then enforces the privacy of the message. Thus, using telephone device 3302, the caller, user A, leaves voicemail message 3306 on voicemail system 3305 while communicating through public communications network 3503 and private communications network 3504. Note that the caller, shown here as user A, is not necessarily a subscriber on voicemail system 3505 or e-mail system 3510. User A indicates that the message is private, for example, by pressing a particular key such as key 3501 of telephone device 3502. The message may be indicated private by other forms of user input. For example, user A may provide a spoken command that indicates that the message is to be maintained as private. The voice message 3506 is provided to a voicemail system to e-mail system connector 3507, which will provide the message to e-mail system 3510.

Voicemail system to e-mail system connector 3507 may exist in various forms according to various embodiments of the invention. For example, such a connector may comprise software on e-mail system 3510. Alternatively, voicemail system to e-mail system connector 3507 may comprise software on a messaging and collaboration server that works in connection with e-mail system 3510. For example, voicemail system to e-mail system connector 3507 may comprise software running on a system such as Microsoft Exchange, which also provides messaging capability to support e-mail system 3510. Alternatively, voicemail system to e-mail system connector 3507 may be included within voicemail system 3505, or may be a separate system, according to other embodiments of the invention.

Voice message 3506 is converted to an e-mail in voicemail system to e-mail system connector 3507. This conversion takes place in voicemail to e-mail conversion block 3508. The e-mail is protected in block 3509, which provides protection of the e-mail to enforce the privacy requested by user A. This is protection that is recognized by e-mail system 3510. Thus, when e-mail system 3510 receives the resulting e-mail 3511, e-mail system 3510 can enforce the privacy requested by user A.

In this example, e-mail 3511 contains a voicemail intended for user B. Thus e-mail 3511 is delivered to computer system 3513 which is associated with user B. Computer system 3513 displays e-mail 3511 as a message 3519 on the user interface. The e-mail system 3510 enforces the privacy of the message through a policy enforcement mechanism 3512 of e-mail system 3510. Policy enforcement may be implemented in various forms. For example, software code or logic may be included in e-mail system 3510 as shown with block 3512. Alternatively, a portion of e-mail system 3510 running on computer system 3513 may include the appropriate software code or logic to provide the policy enforcement. The requested form of privacy is enforced for the message. For example, here the message may be played by user B, for example, by pressing play button 120. However, the message may not be forwarded to other users such as to user C or user D, as illustrated by the disabled forward button 3521.

Figure 36 is a flow diagram of storing and receiving a private voice message, according to an embodiment of the invention. A voicemail is received (block 3601). This voicemail is received from a caller into a communication system. A request is received to make a voicemail private (block 3602). For example, the caller may press a key or indicate by a voice command that the voicemail is to be made private. The meaning of private will depend on the particular voicemail system. For example, in one embodiment of the invention, private will mean that the voicemail is to be listened to only by the recipient of the voicemail and not to be forwarded to other recipients.

The voicemail is converted to an e-mail (block 3603). This e-mail is protected in a scheme that is recognized by the e-mail system (block 3604). For example, the e-mail system may have a mechanism that prevents certain types of e-mails to be forwarded under certain circumstances. This may be a protection system that is used to protect the e-mail in order to achieve the privacy that was requested. The protected e-mail is sent through the e-mail system (block 3605). The e-mail may be displayed to the user (block 3606). The display may indicate that the e-mail contains a voicemail message to the user.

The user is not allowed to take an action prohibited by the protection system for this message, given that the message has been marked private by the caller. A command is received from the user (block 3607). The command may be a command to take an action on the e-mail, such as to play the voicemail message. If the user is authorized to have the command executed under the protection scheme that protects the e-mail (block 3608), then the requested action is taken (block 3609). For example, in a case where the user is the recipient of the voicemail, and the user is intended to be able to play the voice message, the voice message is played (block 3609). If the user is not authorized to execute a particular command with respect to the particular message, then such action is not undertaken (block 3610). For example, the message may have been marked private in a system in which marking private means that the message is only to be listened to by the intended recipient and not to be forwarded to others. In such a system, the protection scheme does not allow the message to be forwarded to another user.

The protection scheme may be implemented in various ways. For example, if the user is not allowed to forward the message, the option of forwarding may not appear in the user interface for the particular message. Alternatively, a button for forwarding may be grayed out to indicate that this option is not available. In another example, when the user attempts to forward the message, the user may receive a message that the message is private and/or that the user is not allowed to forward this message.

Figure 37 is a block diagram of a communication system with a message and collaboration server, according to an embodiment of the invention. In this communication system, a caller is able to leave a voicemail and request a level of privacy for the voicemail when the voicemail is stored and accessed in the e-mail system. The privacy is enforced by a rights management scheme. The system includes voicemail system 3701, voicemail system to e-mail system connector 3703, message and collaboration server 3711 and user B's computer system 3716. Also included are rights management application 3709, encryption block 3710, and rights management application 3715. Shown within voicemail system 3701 is a voicemail message 3702. Voicemail system to e-mail system connector 3703 includes a voicemail to e-mail conversion block 3704. Message and collaboration server 3711 includes e-mail storage 3712 and e-mail application 3714. Associated with user B are computer system 3716 and telephone device 3717.

Voicemail system 3701 is coupled to messaging and collaboration server 3711 through voicemail system to e-mail system connector 3703. Voicemail system to e-mail system connector 3703 may be implemented in various ways, for example, as a module contained on the same computer system as messaging and collaboration server 3711 or as a separate block or system. Computer system 3716 communicates with e-mail application 3714, which is

located on messaging and collaboration server 3711. Voicemail system to e-mail system conversion block 3704 communicates with rights management application 3709, which uses encryption block 3710. Although shown separately, according to an embodiment of the invention, rights management application 3709 and rights management application 3715 may be included with a common software application or suite of applications that is called by each of voicemail system to e-mail system connector 3703 and e-mail application 3714.

The caller leaves a voicemail and requests that the voicemail be protected or made private. The recipient of the voicemail is then not able to take actions prohibited by the protection. As shown here, user A, the caller, leaves voice message 3702 on voicemail system 3701. User A may be a subscriber, or an outside caller who is not a subscriber on voicemail system 3701. In an example where user A is using equipment connected to private communications network, user A may leave voice message 3702 by communicating through a private communications network. In another example, user A makes a call from outside, for example where user A is a caller, such as a customer or other unrelated individual who is not a subscriber on the system. User A indicates that the voice message 3702 is to be protected, requesting, for example, that the voicemail can only be listened to by the intended recipient. This intended recipient may be the subscriber whom user A was calling when user A encountered the voicemail system.

Voice message 3702 is converted to an e-mail in voicemail to e-mail conversion block 3704 of voicemail system to e-mail connector 3703. First, according to one implementation, a plain text e-mail version of the voicemail is created. The e-mail may be either passed internally within the same system or via a network to another machine. According to an embodiment of the invention, the message headers (e.g., to:, from:, cc: and bcc: fields) are converted into an access control list. In the event that a message and collaboration server such as Microsoft Exchange is used as messaging and collaboration server 3711, active directory user identities are used to convert the message headers into the appropriate access control list. As shown, the message headers from 3705 and to 3706 are passed to rights management application 3709. Additionally, the audio information 3707, which contains the voice message, is also passed to rights management application 3709. Rights are assigned to viewers of the message. The rights may depend on input from the system user and/or administrator and may be set through the administrator's user interface. The access control list, the desired rights and the audio message itself are packaged into various structures defined by the rights management application 3709 and encryption system 3710. Encryption block 3710 returns an encrypted version of the e-mail, encrypted e-mail 3708. Encrypted e-mail 3708 is returned to e-mail conversion block 3704 and stored in the data store 3712 of messaging and collaboration server 3711 as e-mail 3713.

When the recipient, user B, accesses e-mail 3713 through computer system 3716, user B is prevented from taking actions prohibited by the rights management application 3715 for this message. Rights management application 3715 may exist on messaging and collaboration server 3711. Alternatively, appropriate portions of rights management application 3715 may be included in other parts of the system, such as in a portion of e-mail application 3714 that runs on computer system 316. According to an embodiment of the invention, if user B attempts to save the message to disk, an encrypted copy of the message is saved, but not a plain text version. User B is able to take allowed actions with the message, such as playing the message through play button 3719 on message display 3718. Prohibited actions are not allowed, such as forwarding in the event that the requested protection includes a prohibition against forwarding the message. For example, as shown here, forwarding button 3720 is disabled. The user interface may include an indication that the message is a voicemail message, as shown here with the indication "You have a voice message" 3721. The message may include an icon 3722 that symbolizes the audio content of the message. According to an embodiment of the invention, the user may click icon 3722 to take certain

actions on the message such as a playing, saving, deleting or other action on the message. These actions may be limited by the rights management scheme.

A certain level of protection is provided for voicemail messages marked private. However, this does not mean that completely hack-proof protection is provided in all embodiments of the invention. There may exist some  
5 embodiments in which the protection can be defeated under certain circumstances.

Figure 38 is a flow diagram of storing and receiving a private voice message using a rights management scheme, according to an embodiment of the invention. A voicemail is received (block 3801). This voicemail may be received, for example, from a caller who is not a subscriber to the system. The caller requests that the voicemail be made private, and the system receives this request to make the voicemail private (block 3802). Since the  
10 voicemail is received by a voicemail system and not an e-mail system, the voicemail is converted into an e-mail (block 3803).

The message is prepared for the e-mail system so that the privacy requested by the caller may be implemented. For example, access control, rights and the voice audio message are packaged into structures defined by the e-mail system and rights management system (block 3804). Depending on the rights management scheme,  
15 the e-mail may be encrypted in a form accessible by the rights management system (block 3805). The encrypted e-mail is sent through the e-mail system (block 3806). A command is received from the user (block 3807). This user is ordinarily the intended recipient of the voice message. If the command is authorized under the rights management system for this message given that the message has been marked private (block 3808), then the requested action is taken (block 3809). For example, the user may request to play the voice message, or to forward  
20 it to an authorized recipient. If the rights management system authorizes this action with respect to the particular message, the system may take such action. If the user attempts an action which is not authorized under the rights management system, the system does not take such unauthorized action (block 3810). For example, if the message is marked private, and the system does not offer forwarding of private messages, then the system does not take this action.

Figure 39 is a block diagram of a communication system with voice access to stored messages, according to an embodiment of the invention. In addition to some of the components shown in Figure 37, the system shown in Figure 39 includes a system user 3953. Figure 39 shows an example in which user B accesses, through telephone  
25 device 3950 connected to public communications network 3951, a message that was left and marked private by user A. The message is protected. However, user B is not directly accessing the message as an e-mail through computer system 3916. Rather, user B is accessing the message through a voice interface through telephone device 3950. In order to allow user B to access encrypted e-mail 3908 through a voice interface, user B communicates with system user 3953, which has access to encrypted e-mail 3908. To allow system user 3953 to access encrypted  
30 e-mail 3908, the protected message, as the voicemail message is converted to e-mail in block 3904, system user 3953 is added to the access control list for the particular message. Appropriate information is passed to rights management application 3909 to effect this. For example, system user 3953 may be added to the "to:" list for encrypted e-mail 3908.

When user B accesses e-mail 3913 in message storage 3912, user B does so through system user 3953. System user 3953 is thus able to access e-mail 3913 and enforces the protection that was implemented for the particular message to effect the privacy. Voice application 3952 communicates with system user 3953 in order to  
40 access the message and play the message for user B. Thus, the rights management scheme is enforced for the message for a voice user using the rights management protection scheme of the e-mail application.

Figure 40 is a flow diagram of retrieving a private voice message from a voice interface, according to an embodiment of the invention. This process shows how a user may access a voice message that is protected using a rights management scheme of an e-mail system when the user is communicating with the system through a voice interface, such as through a typical telephone. First, a user voice call is received (block 4001). A command is received from the user, via the voice call, to access a protected e-mail containing a voicemail (block 4002). The e-mail may be protected through various mechanisms, such as through encryption and/or rights management. According to an embodiment of the invention, a system user is given access to the protected e-mail containing the voicemail in advance. The system user has the appropriate rights to allow the appropriate access for a user calling in over a voice interface. The system user is used to access the protected e-mail containing the voicemail (block 4003).

A command is received from the user via the voice call (block 4004). For example, the user may request to listen to the message. The system user passes the request to the e-mail system (block 4005). The rights management system helps to determine whether the command is authorized under the protection scheme under which the e-mail was stored (block 4006). If the user command is authorized under the protection scheme, the requested action is taken (block 4007). For example, the user may play the voicemail or forward the voicemail to an authorized recipient, if these actions are allowed under the protection scheme. If the action is not authorized, the system does not take the respective action (block 4008). For example, if the message has been marked private and private means the message cannot be forwarded to another recipient or to a particular recipient, such action is not taken.

According to an embodiment of the invention, the rights of a user to take actions on a voicemail depend on the user's Class of Service. Additionally or alternatively, the rights of a user to take actions on a voicemail depend on the Group Policy associated with a user. Additional protection may be applied to voicemails, such as temporal expiration--the voicemail expires and cannot be accessed and/or listened to after a particular time period or after a particular time. The voicemail system and/or the messaging and collaboration server include logic to implement the expiration, according to various embodiments. According to another embodiment, security logic or a rights management system coupled with the messaging collaboration server or the email system applies various restrictions on who can listen to a particular message. For example, only members of a particular organization may be able to access and/or listen to a voicemail. Other restrictions may include that the voicemail can be accessed and/or listed to only once or a limited number of times, or that the voicemail may be accessed and/or listed to only through the local voice mail system.

The Class of Service may be stored in the active directory (AD) on the messaging and collaboration server as an attribute associated with a user. Class of service ("COS") is one of the MCS or voice application-specific attributes assigned to an MCS user. In telephony, a COS includes values for various permissions or levels of service, such as dialout privileges (none, local, long distance, international, etc.), voice mailbox privileges, voice mailbox capacity, length of recorded greeting, and so on. Traditionally, several telephony COSs are defined by a telephone administrator (who is distinguished from the enterprise IT administrator as described herein), and each enterprise telephone system user is assigned one of the defined COSs. In traditional enterprises, this assignment is accomplished when the telephone administrator sets up a new user of the enterprise telephone system. This is a relatively rigid scheme. For example, in order to grant a telephone user one additional privilege not included in his or her current COS typically requires the creation of a new COS.

In the enterprise software world, groupware applications typically have a different method for assigning enterprise network resource using a host of capabilities and privileges, including specific login times for the user's computer, the network printers to which the user is allowed to print, the shared files and applications the user is

allowed to access, etc. In various embodiments, the MCS user setup/enable process integrates with the existing groupware application schemes for assigning permissions, levels of service, privileges, etc. pertaining to telephony and voice applications, which are not traditionally provided for. In one embodiment, the MCS user setup/enable process integrates with the MS Exchange/AD Group Policy ("GP") schema, which is considerably more flexible and expandable than the telephony COS method. As such, the enterprise IT administrator is able to assign Group Policies in the accustomed manner, but the Group Policies effectively encompass all of the required telephone COS concepts. According to the embodiments described, a telephone administrator is not needed to define or assign COSs to users of the enterprise telephone system.

In AD, a Group Policy Object ("GPO") is a policy setting. The setting of a default desktop for instance, is configured in a GPO. A Group Policy Container is an AD object where GPOs are linked. Only sites, domains, and organizational units ("OUs") can have GPOs linked to them. The Group Policy ("GP") schema is hierarchical, with multiple OUs to a domain, and multiple domains to a site. There is typically some overlap resulting from the ability to create multiple settings in multiple areas for an enterprise structure. There are therefore inheritance rules in GP to avoid conflicts and assure proper function. GP is applied to a user or computer object in a specific order, e.g., local computer, site, domain, domain controller(s), and OU. In the case of nested OUs, GP is applied in order from the highest level OU, or parent, down to the lowest level OU. Examples of OUs include geographical locations and organizational units (e.g., sales and development). Very generally, OUs lower in the hierarchy inherit their GPO(s) from above in such a manner that once a privilege is restricted at a point in the hierarchy, it is restricted the rest of the way down. For example, if executive management is above sales, which is above development, executive management may have unlimited dialout privileges. Sales may have dialout to exclude international calls. Development therefore cannot have international call privileges, but may have an even lesser dialout privilege, such as local only.

Thus, a member of the organization has an effective GPO (EGPO) that needs to be calculated from the schema. In one embodiment, the AD GP schema is extended to include telephony-related and voice mail-related permissions, levels of service, and privileges. This allows the GPO process for the administrator to remain unchanged. In one embodiment, the enterprise is supplied with a compliant GP text file template that is specifically written to include all of the telephony-related and voice mail-related information currently lacking in groupware applications. The template is mapped into a UI that prompts the administrator to "add GP." When the template is brought up in the UI, it lists the various telephony-related and voice mail-related items, or COS parameter. Each item includes its own selection box or button which the administrator can check or click to select all the values to be set for different groups. In one embodiment, the COS parameters include: MaxNumber of Greetings; ExtendedAbsenceGreetingAllowed; ExtendedAbsenceGreetingLength; MaxGreeting Length; ExtendedAbsenceGreetingBlockMsgRecord; GeneralDialoutPrivileges; PasscodeAgingEnabled; AccessToSystemDistLists; AllowToSendBroadcastMessage; ActiveMobileNumber; MobilePhoneNumber; ExtendedAbsenceText; MailboxLanguage; CallerFirstLanguage; CallerSecondLanguage; CallerThirdLanguage; and AttendantSchedule.

This process is a one-time setup process. The administrator receives the template and goes through the process of "extending" GP once. After that, the user setup process is the same as before for the administrator.

In one embodiment, the EGP is retrieved from AD and stored in a cache on the MCS as described in more detail below. In another embodiment, as part of the process of populating the custom attribute, the EGP is retrieved, and stored in the custom attribute as "COS." In yet another embodiment, the enterprise is relatively simple, and each user has a GPO assigned. In that case, the GPO for the user can also be cached and retrieved in the same



manner as previously described with reference to the EGPO, but the amount of time required to retrieve the GPO will be reduced.

An embodiment of the invention is directed to allowing users in different organizations to exchange voicemail messages through electronic mail. The voicemail system stores voicemails as emails and uses a special format to store information about the voicemail. This format allows the user to access special voicemail features when the user accesses the email containing the voicemail. Under certain circumstances, a user, for example, a user in company A, may want to forward this email to an individual in another organization, for example, in company B. Ideally, the user would want for the recipient in company B to also be able to access some of the special voicemail features associated with the email. However, a standard email system may not support sending of the rich format of the email that was provided. Thus, the user in company B would not be able to have the benefit of the special voicemail features associated with the email.

An embodiment of the invention is directed to helping the recipient in company B to access some of the special voicemail features associated with the email. When the user in company A sends the email to the recipient in company B, the special voicemail formatting is encapsulated into a format that can be included in the email sent from company A to company B. When the email is received at company B, the special voicemail formatting can be extracted. As a result, the recipient in company B will be able to access some of the special voicemail features associated with the email. According to an embodiment of the invention, the system in company A checks whether the recipient's system supports emails with the special voicemail formatting. If the recipient's system supports this kind of email, the special voicemail formatting is encapsulated and sent in the email as described above, and if the recipient's system does not support this kind of email, the special formatting is not included with the email.

Figure 41 is a block diagram of a network of communications systems with voicemail networking, according to an embodiment of the invention. In the combination of systems shown in Figure 41, emails containing voicemail messages can be forwarded from one email system to various other email systems and retain particular features not generally part of an email communication system.

Figure 41 includes system A 4101, system B 4102, system C 4103, system D 4104 and internet 4114. System A includes private communication network 4105, system A voicemail system 4106, system A email system 4107, user A 4108 and user B 4109. System B includes system B voicemail system 4116, system B email system 4115 and user C 4117. User A 4108 includes computer system 4110 and telephone device 4112, and user B 4109 includes computer system 4111 and telephone device 4113. System A email system 4107 includes a rich format voicemail email 4121, encapsulation block 4122, and email with encapsulated data 4123. System B email system 4115 includes email with encapsulated data 4118, extraction block 4119, and rich format voicemail email 4120.

Each of system A 4101, system B 4102, system C 4103, and system D 4104 is coupled to Internet 4114. The various systems include various components of communications and computer network systems. For example, in system A 4101, private communications network 4105, system A voicemail system 4106, system A email system 4107, user A 4108 and user B 4109 are coupled together through one or several networks. Similarly, the other systems shown include elements of voice and data communications networks. For example, system B 4102 includes system B voicemail system 4116 coupled to system B email system 4115 and user C 4117, which is also coupled to system B email system 4115. The various components of the systems shown in Figure 41 are coupled through Internet 4114. The blocks shown in Figure 41 and other descriptions herein of the invention may be implemented as software, hardware or combinations of software and hardware.

According to an embodiment of the invention, system A 4101, system B 4102, system C 4103, and system D 4104 each represents communications systems of different organizations, such as different companies. Users of

these different systems of the various organizations would like to be able to exchange voicemail messages to others having similar equipment even though they are at different organizations. In the system, voicemail messages may be stored as a rich format voicemail email. This may mean that the email containing the voicemail message has various particular features beyond those that are included in a standard email. This would allow a user, for example, to see certain special choices or buttons on the user interface that would allow the user to take certain actions that relate to voicemail messages, such as playing, forwarding or other unique voicemail actions. Such rich format voicemail email may not be supported by a typical email that is sent over the Internet, such as an email that is sent over Internet 414. Thus, a recipient on another organization's system may not be able to use some of the particular features of the voicemail system for the received email if the rich format is not retained. In order to allow for the rich format to be used on another system, certain data from rich format voicemail email 4121 is encapsulated in encapsulation block 4122. The result is email with encapsulated data 4123.

Email with encapsulated data 4123 is passed to another email system, such as system B email system 4115. This email is received in system B email system 4115, as email with encapsulated data 4118 as shown in Figure 41. The rich format of the email is extracted in extraction block 4119. The result is a corresponding rich format voicemail email 4120, which corresponds to the rich format of rich format voicemail email 4121 from system A email system 4107. Thus, user C 4117 is able to view rich format voicemail email 4120 and experience some of the unique voicemail actions and particular features of the user interface that were available on system A email system 4107 to users such as user A 4108 or user B 4109.

If system C 4103 is similarly enabled to extract and use the rich format of the email with encapsulated data 4123, a user on system C 4103 will similarly be able to experience the unique voicemail actions and particular features of the received email. However, other systems not equipped with the requisite extraction logic and/or voicemail system may not necessarily be able to display the rich format email, or extract the rich format of the email, and therefore not be able to provide the corresponding unique voicemail actions or particular features of the received email. System A email system 4107 may account for this by not encapsulating the data from the rich format voicemail email and instead by sending a simplified form of the email with the included voicemail audio data. For example if system D 4104 is not able to extract and use the rich format of the email, according to an embodiment of the invention, system A email system 4107 does not encapsulate the data from the rich format voicemail email and instead sends a simplified form of the email with the included voicemail audio data to system D 4104.

Figure 42 is a flow diagram of receiving and sending an email containing a voicemail message, according to an embodiment of the invention. A user receives a rich format voicemail email (block 4201). The rich format may mean that the email containing the voicemail message has various particular features beyond those that are available with a standard email, which would allow the user, for example, to see certain special choices or buttons on the user interface that would allow the user to take certain actions that relate to voicemail messages, such as playing, forwarding or other unique voicemail actions.

The user requests to forward the rich format voicemail email to a recipient on a remote voicemail system, for example to a recipient at a different organization with its own voicemail system. Such request may be made by an action on a user interface, such as by a command in a voice interface or selection of a button on a graphical user interface. The rich format data is encapsulated (block 4202) and packaged into the outgoing email (block 4203). The email with the encapsulated data is sent to the remote system, which is a different communication system that is coupled to the intended recipient of the email (block 4204). The remote system extracts the rich format data from the received email (block 4205). The remote system then provides a rich format voicemail email from the received

email and the extracted data (block 4206). The resulting rich format voicemail email is then displayed to the user that was the intended recipient of the email containing the voicemail (block 4207). The user is then able to experience some of the same unique voicemail actions and particular features that were available to the user sending the rich format voicemail email.

5 Figure 43 is a block diagram of two communications systems with voicemail networking and encapsulation and extraction devices, according to an embodiment of the invention. In the systems shown in Figure 43, a rich format voicemail email is transmitted from one system to the other system, while preserving the rich format data through the use of encapsulation logic and extraction logic applied when the email is sent and received, respectively.

10 Figure 43 includes a system A 4301 and system B 4302. System A includes system A email system 4303, and system B 4302 includes system B email system 4304 and user C 4305. System A email system includes a rich format voicemail email 4306, encapsulation block 4307, and forwarded email 4308. Rich format voicemail email 4306 includes voicemail properties 4309, and forwarded email 4308 includes encapsulated data 4310 and standard email attributes 4311. System B email system 4304 includes received email 4312, extraction block 4313, and rich format voicemail email 4314. Received email 4312 includes encapsulated data 4315 and standard email attributes 15 4316, and rich format voicemail email 4314 includes voicemail properties 4317. User C 4305 includes computer system 4318 and telephone device 4319. Also shown is message 4320 displayed from computer system 4318. Message 4320 includes play button 4321 and forward button 4322.

20 System A 4301 and System B 4302 are each part of various communications and computer network systems. System A email system 4303 and system B email system 4304 are coupled by way of Internet 4323. User C 4305 is coupled to system B email system 4304.

25 In the system, a rich format voicemail email may be forwarded to a user on a different email system. So long as the email system from which the rich format voicemail email is forwarded and the email system to which the rich format voicemail email is forwarded both have the capability to recognize and use the particular format, the rich format properties will be preserved. The rich format may mean that the email containing the voicemail message has various particular features beyond those that are available with a standard email, which would allow the user, for example, to see certain special choices or buttons on the user interface that would allow the user to take certain actions that relate to voicemail messages, such as playing, forwarding or other unique voicemail actions, or other voicemail properties.

30 In one embodiment of the invention, Messaging Application Programming Interface (MAPI) is used as a method of preserving the rich format data contained in the rich format email by assigning MAPI properties to corresponding rich format data. MAPI is used by various industry-standard email clients, such as the Microsoft Exchange client. MAPI is a mechanism to access information in Exchange and enables different email systems to work together to distribute email. In Exchange, email message properties are stored as MAPI properties, which contain general email message properties, including recipient, sender, email body and email subject, and Exchange 35 specific email message properties, including unique identifiers for the sender, each recipient and the message itself. However, MAPI does not work well outside an intranet. Thus, if the sending and receiving email systems utilize Exchange to handle emails, and are each part of the same organization using one intranet, all MAPI properties, which preserve the rich format data of an email, will be preserved by Exchange when sent from the sending system to the receiving system. However, when the sending system is a member of one organization using one intranet, and 40 the receiving system is a member of a different organization not using the same intranet, even if they both utilize Exchange to handle emails, when the sending system sends the email to the receiving system, Exchange may strip off the Exchange specific email message MAPI properties from the email. In this situation, Exchange provides a

simplified form of the email with only the general email message MAPI properties, which may include voicemail audio data, but none of the Exchange specific email message MAPI properties, which include voicemail properties.

In an embodiment of the invention, Exchange specific MAPI properties are assigned for each of the voicemail properties 4309 contained in rich format voicemail email 4306, resulting in voicemail properties 4309 being stored in MAPI format. If system A 4301 and system B 4302 are each part of the same organization, using the same intranet that uses Exchange to handle emails, the MAPI format is preserved when a user in system A 4301 sends rich format voicemail email 4306 to a user in system B 4302. However, if system B 4302 is part of a different organization not using the same intranet as system A 4301, Exchange may strip off the Exchange specific MAPI properties associated with voicemail properties 4309.

Thus, given that system A 4301 and system B 4302 are each part of a common network of voicemail systems, rich format voicemail email 4306, containing the voicemail properties 4309, is encapsulated in block 4307. System A email system 4303 uses a MAPI transport provider to convert MAPI properties into a Transport-Neutral Encapsulated Format (TNEF) serial data stream. TNEF defines several TNEF-specific attributes, each of which corresponds to a particular MAPI property. These attributes are used to encode the corresponding MAPI properties into the TNEF serial data stream. TNEF is primarily used by transport providers that need to encode MAPI message properties for transmission through a messaging system that does not support those properties directly. Thus, before an email containing MAPI properties is sent, the MAPI transport provider converts the MAPI Properties into a TNEF data stream, and encapsulates the TNEF data stream into a special attachment on the outgoing email, using the underlying email system's attachment model. For example, Simple Mail Transfer Protocol (SMTP)-based transport, which is a protocol for sending email messages between email systems, uses a MAPI transport provider to encode MAPI properties into a TNEF data stream. Exchange, which is utilized by system A email system 4303 to handle email, sends message 4308, which includes standard email attributes 4311, such as "to" and "from" fields, and the TNEF data stream attachment, as encapsulated data 4310, which is usually named "winmail.dat."

The received email 4312, which contains the encapsulated data 4315 and the standard email attributes 4316, is delivered to system B email system 4304. Exchange, which is utilized by system B email system 4304 to handle email, uses the MAPI transport provider in extraction block 4313 and extracts the MAPI properties from the TNEF serial data stream contained in encapsulated data 4315. Exchange recognizes the MAPI properties corresponding to rich format data. The result is a corresponding rich format voicemail email 4314, containing voicemail properties 4317, which corresponds to the rich format voicemail email 4306, containing voicemail properties 4309, from system A email system 4303. As a result, the users in the organizations associated with system A 4301 and system B 4302 may get the experience of having their voicemail systems networked.

Thus, user C 4305 is able to view rich format voicemail email 4314 on computer system 4318, along with the voicemail properties 4317, in a similar manner as displayed in message 4320. User C 4305 can then take certain actions that relate to voicemail messages, such as playing, forwarding or other unique voicemail actions, by pressing the appropriate button displayed on the user interface. For example, user C 4305 may play the voicemail message by pressing play button 4321, or forward the voicemail message by pressing the forward button 4322. One feature supported by the voicemail system based on the special voicemail formatting may be that a voicemail email message can be played on the user's telephone. Thus user C 4305 may be able to play voicemail email message 4317 on telephone device 4319 using the special voicemail formatting in voicemail email 4314.

Figure 44 is a block diagram of communications systems, some with voicemail networking, according to an embodiment of the invention. In the system shown in Figure 44, a rich format voicemail email is routed and sent according to the network properties of the recipient system. Figure 44 includes system A 4401, system B 4402,

system C 4403, system D 4404 and Internet 4405. System A 4401 includes system A email system 4406, rich format voicemail email 4407, routing block 4408, network enabled lookup block 4409, dividing block 4410, encapsulation block 4411, no encapsulation block 4412 and send email block 4413.

Each of system A 4401, system B 4402, system C 4403, and system D 4404 is coupled to Internet 4405.

5 The various systems may or may not be members of a common network of voicemail systems. For example, system A 4401, system B 4402 and system C 4403 have corresponding voicemail capabilities that allow for special voicemail features of inbound emails to be used. The various systems may contain various components of communications and computer network systems. For example, system A includes system A email system 4406.

10 In the system, a rich format voicemail email, such as rich format voicemail email 4407, may be forwarded to a user on a different email system, such as a user coupled to system B 4402, a user coupled to system C 4403, or a user coupled to system D 4404. When a user of system A email system 4406 forwards a rich format voicemail email 4407, before email 4407 is sent, it is analyzed in routing block 4408. Routing block 4408 will use network enabled lookup 4409 to determine if the recipient system is able to receive and use the special voicemail formatting of voicemail email 4407.

15 If network enabled lookup 4409 determines the recipient network is a member of the common network of voicemail systems, routing block 4408 causes dividing block 4410 to provide rich format voicemail email 4407 to be encapsulated in encapsulation block 4411. If network enabled lookup 4409 determines the recipient network is not a member of the common network of voicemail systems, routing block 4408 causes dividing block 4410 to provide rich format voicemail email 4407 to no encapsulation block 4412 and rich format voicemail email 4407 is not encapsulated. If rich format voicemail email 4407 is sent to encapsulation block 4411, the rich format data is encapsulated. The sent email block 4413 then sends the forwarded email, whether or not the rich format data has been encapsulated, to the recipient system.

20 If the recipient system is a member of the common network of voicemail systems to which system A 4401 is a member, it will receive the forwarded email with encapsulated rich format data. If the recipient system is not a member of the common network of voicemail systems, it will receive a simplified form of the email with the included voicemail audio data. For example, given that system A 4401, system B 4402 and system C 4403 are all members of a common network of voicemail systems, system B 4402 and system C 4403 receive the forwarded email with encapsulated rich format data. Given that system D 4404 is not a member of the common network of voicemail systems, system D 4404 only receives a simplified version of the email. The email system corresponding to system B 4402 or system C 4403, as the case may be, then extracts the encapsulated rich format data from the forwarded email, and provides to its users a rich format voicemail email corresponding to the rich format voicemail email 4407 in system A email system 4406.

30 Figure 45 is a block diagram of communications systems, some with voicemail networking, and a network enabled lookup device, according to an embodiment of the invention. Before a user on a system sends a rich format voicemail email outside the system, a network enabled lookup device checks to see if the remote system of the intended recipient is registered to receive emails in a format that preserves the rich format properties of the email. If the remote system of the intended recipient is registered, the two systems are compatible with one another, and each is part of a virtual network of voicemail systems. Figure 45 includes system A 4550, system B 4551, system C 4552, registration service 4560, and Internet 4562. System A 4550 includes system A voicemail system 4553 and system A email system 4554, system B 4551 includes system B email system 4555 and system B voicemail system 4556, and system C 4552 includes system C email system 4557. Also shown is registration service 4560, which includes database 4561, and the entry chart 4565.

Each of system A 4550, system B 4551, system C 4552 and registration service 4560 is coupled to Internet 4562. The various systems may or may not include voice mail systems, and may or may not be registered to be compatible with one another. For example, system A 4550 and system B 4551 contain system A voicemail system 4553, and system B voicemail system 4556, respectively. Company A, which uses system A 4550, and company B, 5 which uses system B 4551, may then be registered to receive and use emails in a format that preserves the rich format properties of the voicemail email, but company C, which uses system C 4552, is not registered, as shown in entry chart 4565. By registering, company A and company B each becomes part of a virtual network.

Thus, before a user from company A sends a rich format voicemail email to a user from company B and a user from company C, network enabled lookup 4558 checks with database 4561 of registration service 4560, and 10 since company B is registered, routing block 4559 sends email 4563 containing the rich format data to the user from company B; however, since company C is not registered, routing block 4559 sends email 4564 not containing the rich format data to the user from company C. Therefore, the users from company A may forward rich format voicemail emails to users from company B, and the company B users experience the voicemail properties in a similar manner experienced by the company A users. However, since they are not part of the virtual network, the 15 company C users only receive a simplified form of the email with the included voicemail audio data delivered to system C email system 4557.

Optionally, system A 4550 may have local cache 4566, and the registration data corresponding to a remote system is stored locally on cache 4566. Thus, before a user from company A sends a rich format voicemail email to a user from a different company, for example a user from company B, network enabled lookup checks with database 20 4561 of registration service 4560, and stores company B's registration data in cache 4566. Thereafter every time before a user from company A sends a rich format voicemail email to a user from company B, network enabled lookup 4558 checks registration data from cache 4566. Network enabled lookup 4558 may periodically check registration data in database 4561 for updates.

In one embodiment of the invention, network enabled lookup 4558 uses a Domain Name System (DNS) 25 lookup of the domain name recipient service (SRV) record. DNS translates domain names into Internet Protocol (IP) addresses, and an SRV record provides information on available services for a given domain name. When an email system sends out an email, it uses DNS to convert the domain name corresponding to the email address to an IP address. DNS also performs a lookup of the domain name and determines where to send the email, based on the SRV record of the domain name. In such an embodiment of the invention, the SRV record for a domain name of a 30 system, which is part of the virtual network, will include some identification that the system is part of the virtual network. DNS also determines if the domain name is part of the virtual network. For example if a user from company A desires to forward a rich format voicemail email to a user on company B, whose email address is "user@companyB.com," before the voicemail email is sent, network enabled lookup 4558 looks up "companyB.com" using a DNS lookup and, given that company A and company B are members of the virtual 35 network, network enabled lookup 4558 finds that the SRV record of companyB.com contains an identification that companyB.com is part of the virtual network of voicemail systems. Given that company D is not a member of the virtual network of voicemail systems, if a user from company A sends a rich format voicemail email to a user from company B, the DNS lookup of company B's domain finds that the SRV record of companyD.com will not contain any identification that company D is a member of the virtual network.

40 In another embodiment of the invention, a system administrator of system A 4550 system specifies which remote systems belong to the virtual network, and stores such data in a database or file that may be located on optional cache 4566, or in a database outside system A 4550. Network enabled lookup 4558 then checks

compatibility of the recipient system against the database or file. In such an embodiment of the invention, the system may not include registration service 4560. Alternatively, registration service 4560 may be used in combination with information maintained by the system administrator.

5 In another embodiment of the invention, compatibility data normally stored in database 4561, which is located outside system A 4550, may be downloaded by system A email system 4554 and be stored in local cache 4566. Network enabled lookup 4558 then checks compatibility of the recipient system against the downloaded data.

10 In another embodiment of the invention, web services are used to check compatibility recipient system to receive the rich formatted voicemail. Such web services may be used as an alternative to the DNS lookup described herein. Embodiments of web services may include integration of Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. According to various embodiments, XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. According to an embodiment of the invention, a service is used to check compatibility recipient system to receive the rich formatted voicemail, such as web services, which share business logic, data and processes through a programmatic interface across a network.

15 Figure 46 is a flow diagram of receiving and routing an email containing a voicemail message, according to an embodiment of the invention. A user receives a rich format voicemail email (block 4501). The rich format may mean that the email containing the voicemail message has various particular features beyond those that are available with a standard email, which would allow the user, for example, to see certain special choices or buttons on the user interface that would allow the user to take certain actions that relate to voicemail messages, such as playing, forwarding or other unique voicemail actions.

20 The user desires to forward the rich format voicemail email to another individual on a recipient system. Prior to the email being forwarded, lookup logic determines if the recipient system is a part of the common network of voicemail systems to which the sending system is a member (block 4502). If the recipient system is a member of the common network of voicemail systems, then both systems are part of a virtual network.

25 In one embodiment of the invention, the lookup logic uses a DNS lookup of the domain name SRV record. When an email system sends out an email, DNS performs a lookup of the recipient domain name and determines where to send the email, based on the SRV record of the domain name. In such an embodiment, the SRV record for a domain name of a system, which is part of the virtual network of voicemail systems, will include some identification that the system is part of the virtual network. In the same manner DNS lookup allocates where to send the email within a domain name, it will also determine if the domain name is part of the virtual network of voicemail systems.

30 In another embodiment of the invention, the system administrator of the sending system specifies which recipient systems belong to the virtual network of voicemail systems, and stores such data in a database or file that may be located on sending system local cache, or in a database outside the sending system. The lookup logic would then check the virtual network status of the recipient system against the database or file.

35 In another embodiment of the invention, virtual network status of recipient systems normally stored in a media outside the sending system may be downloaded by the email system of the sending system and be stored in local cache. The lookup logic checks virtual network status of the recipient system against the downloaded data.

40 If the receiving system is part of the virtual network of voicemail systems, the rich format data is encapsulated (block 4503) and packaged into the outgoing email. If the receiving system is not a member of the virtual network of voicemail systems, the rich format data is not encapsulated, and a simplified form of the email with the included voicemail audio data is queued for transmission.

The email, whether or not the rich format data has been encapsulated, is sent to recipient system (block 4504). Recipient systems that are part of the virtual network of voicemail systems extract the rich format data from the received email and provide a rich format voicemail email from the received email and the extracted data. The resulting rich format voicemail email is then displayed to the user that was the intended recipient of the rich format voicemail email. The user is then able to experience some of the same unique voicemail actions and particular features of the user interface that were available to the user sending the rich format voicemail email. Recipient systems that are not part of the virtual network of voicemail systems only receive the simplified form of the email with the included voicemail audio data.

The following is a non-exhaustive description of various embodiments of systems and methods with which the methods and systems described herein may be combined. For example, the systems and methods described above for networking of voicemail systems may be employed in combination with the form-based interface for use in retrieving voice mail messages and controlling actions taken on voice mail messages received in an enterprise network system. Thus, according to an embodiment, a rich-format email is used to allow a user to use a form-based interface even when the email is sent over a network to a different system.

An embodiment of the invention is directed to a diagnostic tool. The tool provides a framework for running diagnostics. The diagnostics may be for internal or external troubleshooting and resolution. The tool supports source code compilation to extend the suite of diagnostics in the field. In an embodiment of the invention, the tool has support for runtime configurable parameters to the diagnostic methods.

An embodiment of the tool supports source code compilation to compliment diagnostics that are precompiled before deployment. This allows field engineers, technical staff, or end users to enhance the suite of diagnostics. In addition, since the compiled source runs within the preexisting framework provided by the tool, the advantages of the framework can be available for the newly provided source code without needing to recreate a framework. This can provide advantages in capturing or otherwise recording the results of a diagnostic run. In an embodiment of the invention, when the newly provided source diagnostics are run within the preexisting framework, the framework automatically causes the output of the new diagnostics to be gathered and stored.

In an embodiment of the invention, the tool supports runtime configurable parameters. Over time, new diagnostic modules may be provided that require new parameters. These may be precompiled diagnostic modules or source code of diagnostic modules. The framework automatically recognizes the requirement for values for parameters in the new diagnostic modules, and the values for the new parameters are provided automatically at runtime.

Figure 47 is a block diagram of a diagnostic tool, according to an embodiment of the invention. Shown in Figure 47 are front end 4701, diagnostic framework 4702, compiled diagnostic modules 4703 and source code diagnostic modules 4704. Front end 4701 is coupled to diagnostics framework 4702. Compiled diagnostic modules 4703 and source code modules 4704 are coupled to diagnostic framework 4702. Diagnostic framework 4702 loads the diagnostic modules that perform diagnostics. According to an embodiment of the invention, diagnostic framework loads these modules in response to a request from front end 4701. Front end 4701 may comprise a user interface by which a user requests that certain diagnostic modules be run. For example, in one implementation front end 4701 includes a user interface with pull down menus which provides a user a choice of various diagnostics that may be run. After the user selects the diagnostics to be run and request that they be run, front end 4701 requests that diagnostic framework 4702 run the respective diagnostics.

Diagnostic framework 4702 loads the respective diagnostic modules from among compiled diagnostic modules 4703 and source code diagnostic modules 4704. Compiled diagnostic modules may be diagnostic modules



that are provided along with the target system that is to be tested. Alternatively, the compiled diagnostic modules may be diagnostic modules that are provided in some other way, for example, as updates provided by the seller of the target system which is to be tested. Source code diagnostic modules 4704 comprise uncompiled code for diagnostic modules. These may be written, for example, by an administrator of the target system that is to be  
5 diagnosed. Because diagnostic framework 4702 can receive and use source code diagnostic modules 4704, diagnostic framework 4702 can use newly provided diagnostic modules without requiring diagnostic framework 4702 to be rewritten.

After receiving compiled diagnostic modules 4703 and source code diagnostic modules 4704, diagnostic framework 4702 compiles source code diagnostic modules 4704 resulting in additional compiled diagnostic  
10 modules. These compiled diagnostic modules along with compiled diagnostic modules 4703 comprise a set of diagnostic modules that may be run by diagnostic framework 4702. Diagnostic module 4702 runs diagnostic modules from among the set of diagnostic modules that can be run. According to an embodiment of the invention, diagnostic framework 4702 runs a selected set of diagnostic modules based on instructions from front end 4701. After running the respective diagnostic modules, diagnostic framework 4702 gathers the results of the diagnostics.  
15 These results may be stored. Additionally or alternatively, these results may be automatically mailed to a service organization or the manufacturer of the target system on which the diagnostics were run.

Figure 48 is a block diagram of a target system with a diagnostic tool, according to an embodiment of the invention. Shown in Figure 48 are target system 4801 which includes diagnostic framework 4802, compiler 4807, compiled diagnostic modules 203 and source code diagnostic modules 4804. Diagnostic framework 4802 includes  
20 provide diagnostics from compiled diagnostic modules block 4805 and produce compiled code from source code diagnostic modules 4806. Produce compiled code from source code diagnostics modules block 4806 is in communication with source code diagnostic modules 4804, compiler 4807 and provide diagnostics from compiled diagnostic modules 4805. Provide diagnostics from compiled diagnostics module 4805 is also in communication with compiled diagnostic modules 4803.

Diagnostic framework 4802 runs diagnostics on target system 4801 that diagnose target system 4801. The diagnostics that diagnostic framework 4802 runs include diagnostics from compiled diagnostic modules 4803 and from source code diagnostic modules 4804. Produce compiled code from source code diagnostic modules block 4806 causes compiler 4807 to compile source code diagnostic modules 4804. The resulting compiled diagnostic modules are provided to provide diagnostics from compiled diagnostic modules block 4805. Diagnostic framework  
30 4802 then runs diagnostics that are provided from provide diagnostics from compiled diagnostic modules block 4805. These diagnostics are run on target system 4801 and may diagnose various aspects of target system 4801.

Note that, according to an embodiment on the invention, compiled diagnostic modules 4803 may be provided from outside of target system 4801. For example, compiled diagnostic modules 4803 may be provided over a computer network, such as over the internet or internal network. These compiled diagnostic modules 4803  
35 may be ultimately provided from a website or other source of updated diagnostic modules, for example, provided by the manufacturer of target system 4801. Alternatively, compiled diagnostic modules 4803 are created and/or stored on target system 4801. Although not shown, additional functionality, such as a front end may be provided to cause diagnostic framework 4802 to load the respective diagnostic modules and run them. The diagnostic modules may be found on target system 4801 based on their respective extensions and/or based on being stored in a particular  
40 directory associated with diagnostic framework 4802. For example, particular file extensions may be associated with source code diagnostic modules 4804. In such a configuration, front end software finds diagnostic modules

4804 based on these extensions and causes diagnostic framework 4802 to compile and run the respective diagnostic modules.

5 Compiler 4807 may be included within diagnostic framework 4802, according to an embodiment of the invention. According to one embodiment of the invention, diagnostic framework 4802 comprises assembly code in a .NET framework. Compiled diagnostic modules 4803 may comprise assemblies in a .NET framework. Source code/diagnostic modules 4804 may be source code that is compiled into assemblies for a .NET framework. Compiler 4807 may comprise a compiler available in a class library, such as a code dom (Document Object Model) compiler available in FCL (Framework Class Library) layer in a .NET framework configuration. The dom compiler is built into the framework of the FCL class library. Thus, the compiler 4807 is provided in the FCL in a .NET  
10 framework in one implementation. According to an embodiment of the invention, compiler 4807 is included within diagnostic framework 4802. Compiler 4807 may be linked with diagnostic framework 4802 at runtime. According to an embodiment of the invention, the compiler that is included in diagnostic framework 4802 is a code dom compiler provided from the FCL layer in a .NET framework.

15 Figure 49 is a flow diagram of a diagnostic tool, according to an embodiment of the invention. The diagnostic tool uses diagnostic source code in order to provide diagnostics. The diagnostic source code allows an administrator or other user to extend the suite of diagnostics that may otherwise be available for troubleshooting and resolution. First, information regarding the diagnostic precompiled code and diagnostic source code which is to be run is received (block 4950). For example, instructions may be received regarding which particular modules are to be run. According to an embodiment of the invention, the modules may include only precompiled diagnostic  
20 precompiled code, only diagnostic source code, or a combination of both diagnostic precompiled code and diagnostic source code.

The diagnostic source code is compiled (block 4951). By compiling source code, the diagnostics that would otherwise be available to the diagnostic tool are extended. An administrator or other user is able to write source code and then this source code is available to be used as part of a collection of diagnostics.

25 Information is gathered regarding the diagnostic methods (block 4952). According to an embodiment of the invention, this information is gathered about the diagnostic methods that are to be run. This gathering may take place through a reflection process. In reflection, information regarding all the diagnostic methods is gathered, such as, the names of the methods and the names of the respective parameters, according to an embodiment of the invention. This information gathered in the process of reflection is information stored in the metadata associated  
30 with the respective methods and parameters.

The diagnostics are run (block 4953). Running the diagnostics may occur in response to instructions from a front end software or logic. The instructions may include particular instructions as to which diagnostics to run. According to an embodiment of the invention, if the appropriate information is not available to run certain diagnostics, such diagnostics are automatically not run. Results of the diagnostics are gathered (block 4954). These  
35 results may then be stored, according to an embodiment of the invention, and may be made available to various recipients. For example, the information may be provided through a network to the manufacturer of the target system that is being diagnosed. According to another embodiment of the invention, the information is stored in a diagnostic run file. An administrator may then be able to review the diagnostic run file or otherwise analyze it in order to determine the results of the diagnostics that ran.

40 Figure 50 is a relationship diagram of diagnostic assemblies, according to an embodiment of the invention. Shown in Figure 50 are diagnostic assemblies 5001, diagnostic assembly 5002, diagnostic type 5003, diagnostic method 5004, diagnostic parameter 5005 and source compiler 5006. Diagnostic assemblies 5001 symbolizes a

collection of various diagnostic assemblies that are available, for example, diagnostic assemblies that are available to a framework that runs diagnostic assemblies. Thus, diagnostic assembly 5001 includes multiple instances of diagnostic assembly 5002. Diagnostic assembly 5002 is a collection of diagnostic type 5003. Diagnostic type 5003 is a collection of diagnostic method 5004, and diagnostic method 5004 is a collection of diagnostic parameter 5005.

5 During a process of reflection, which may be carried out by a framework that runs diagnostics, information is gathered in a descending fashion from the various items shown starting with diagnostic assemblies 5001. For example, diagnostic assemblies 5001 is reflected upon to determine the associated diagnostic assembly such as diagnostic assembly 5002. Diagnostic assembly 5002 is reflected upon to determine the corresponding members of diagnostic type such as diagnostic type 5003. Diagnostic type 5003 is reflected upon to determine the corresponding  
10 members of diagnostic method such as diagnostic method 5004, and diagnostic method 5004 is reflected upon to determine the corresponding diagnostic parameters such as diagnostic parameter 5005 that diagnostic method has associated with it.

Source code of diagnostic modules is provided as input to source compiler 5006. The resulting compiled modules are also provided to the collection of diagnostic assemblies 5001. As shown here, output of source  
15 compiler 5006 is provided to diagnostic assembly 5002. Thus, the collection of diagnostic assemblies 5001 may include diagnostic assemblies that were provided as source code. Such source code may be provided to the diagnostic tool during run time according to an embodiment of the invention.

When there is a request to load specific diagnostic assemblies, such as when a framework responsible for loading the assemblies is required to do so, the collection of diagnostic assemblies is formed in a manner as shown  
20 in Figure 50. A resulting hierarchy of diagnostics is then formed at starting time.

Figure 51 is a block diagram of a diagnostic tool that uses methods with parameters, according to an embodiment of the invention. In an embodiment of the invention, the tool supports run time configurable parameters. Over time, new diagnostic modules may be provided that require new parameters. The diagnostic framework software is able to handle these new diagnostic modules that have new parameters, even though  
25 particular parameters were not known to the diagnostic framework in advance. Thus, new diagnostic modules may be provided that receive values for new parameters, even though the existence of these parameters was not contemplated at the time of the creation of the diagnostic framework which loads and runs the respective diagnostic modules.

Included in Figure 51 are front end 5101, diagnostic framework 5102 and diagnostic modules 5103. Front  
30 end 5101 is in communication with diagnostic framework 5102. Diagnostic modules 5103 are also in communication with diagnostic framework 5102. Diagnostic framework 5102 loads and runs diagnostic modules, which perform diagnostic functions. For example, diagnostic framework 5102 may load diagnostic module 5104 and diagnostic module 5105. Front end 5101 may cause diagnostic framework 5102 to load and run respective diagnostic modules 5103.

35 Diagnostic framework 5102 gathers information about diagnostic modules 5104 and 5105. This information may include the names of the modules and their respective parameters. This gathering of information may take place through a process of reflection. Diagnostic framework may then make the existence and/or name of the parameters in the respective diagnostic modules available so that runtime values of these parameters may be provided. For example, diagnostic framework may gather information about diagnostic module 5104 and diagnostic  
40 module 5105 determining that these modules have parameter 5106 and parameter 5107, respectively. Then, diagnostic framework 5102 may make information regarding the existence of parameters 5106 and 5107 available so the runtime values of parameter 5106 and parameter 5107 may be provided.

According to the embodiment of the invention, front end 5101 queries diagnostic framework 5102 for the names of the parameters. In this example, diagnostic framework 5102 then provides the names of parameter 5106 and parameter 5107 to front end 5101. A module such as obtain parameter value block 5108 of front end 5101 may obtain the values for these parameters. In one embodiment of the invention, front end 5101 includes a user interface. The user interface may obtain the parameter value by requesting that a user, such as the system administrator, provide runtime values for these parameters. In alternative embodiments of the invention, front end 5101 obtains the runtime values for the parameters in other ways. For example, front end 5101 may receive the values of the parameters from a batch file. Front end 5101, according to an embodiment of the invention, may be software that periodically runs automatically and causes diagnostic framework 5102 to run. Obtain parameter value block 5108 may also obtain parameter values automatically, according to an embodiment of the invention.

The parameter runtime values are obtained and provided to diagnostic framework 5102, and diagnostic framework 5102 receives these values and passes these runtime parameters values to the diagnostic modules. For example, a runtime value may be obtained for parameter 5106 of diagnostic module 5104 and for parameter 5107 of diagnostic module 5105. After running the respective diagnostic modules, for example diagnostic module 5104 and diagnostic module 5105, diagnostic framework 5102 gathers the results of the diagnostics. These results may be stored, or provided in some other manner.

Figure 52 is a flow diagram of a diagnostic tool that uses methods with parameters, according to an embodiment of the invention. Diagnostic methods that require values for parameters are run automatically. Even though the existence of the parameters may not be known in advance, the parameters are automatically recognized and provided with values so that the diagnostics requiring the values of these parameters may be run.

Accordingly, information is received regarding diagnostic code which will be run (block 5250). For example, a list of diagnostic modules that will be run may be received. Information is being gathered regarding these diagnostic methods (block 5251). This information may include the names of the methods, their respective types, and so on. These run time values may be received through a user interface, according to an embodiment of the invention. This information may be gathered through reflection, according to an embodiment of the invention. In reflection the metadata associated with the respective assemblies, types, methods and parameters is reflected upon in order to find the respective information. The information also includes the values for the parameters thus, the parameters used by the methods are identified (block 5255) and the runtime values for the parameters are received (block 5256). Using these parameters and their values, the diagnostics are run (block 5257). Finally, the results of the diagnostics are gathered (block 5258).

Figure 53 is a block diagram of a diagnostic tool, according to an embodiment of the invention. The diagnostic tool shown in the target system may be included in a network environment with other technology. Shown are target system 5301, appliance 5302 and Internet 5303. Target system 5301 is coupled to appliance 5302 and Internet 5303. Target system 5301 includes diagnostic framework 5305, diagnostic front end 5304, compiled diagnostic modules 5306, source code diagnostic modules 5307 and target system function 5308. Diagnostic framework 5305 is coupled with diagnostic front end 5304. Diagnostic framework 5305 is also in communication with compiled diagnostic modules 5306 and source code diagnostic modules 5307. Diagnostic framework 5305 operates on target system function 5308.

Diagnostic framework 5305 receives diagnostic code that includes compiled diagnostic modules 5306 and source code diagnostic modules 5307. Diagnostic front end 5304 causes diagnostic framework 5305 to load respective diagnostics. These diagnostics include diagnostics from among compiled diagnostic modules 5306 and source code diagnostics modules 5307. Diagnostic framework 5305 is able to compile or cause to be compiled

source code diagnostic modules 5307, such that these diagnostic modules are used to diagnose aspects of target system 5301, such as target system function 5308.

In an embodiment of the invention, target system 5301 is a component of a communications system. For example, target system 5301 may comprise a messaging and communications server such as a server that stores, forwards and otherwise processes email and other messages. Appliance 5302 may also comprise an aspect of the communications system. For example, appliance 5302 may comprise a voicemail appliance. In one example embodiment of the invention, appliance 5302 comprises an appliance associated with voicemail system, and target system 5301 comprises a communications server which provides storage and forwarding of voicemail messages and other functions, such as sorting and forwarding of email messages. Diagnostics may help to diagnose such a communications system. Internet 5303 may provide the communication for various aspects of a target system. For example, in an example where target system 5301 represents a part of a voicemail system, Internet 5303 may be used to forward emails and/or voicemails to other systems.

In an example in which target system 5301 represents a communication system, such as a voicemail system, diagnostic modules provided in compiled diagnostic modules 5306 and/or source code diagnostic modules 5307 may include one or various combinations of the following diagnostics:

- shared assembly versions. This diagnostic gathers version information about the installed target system assemblies (code).
- service account. This diagnostic checks to make sure the necessary target system service account information is present in the registry. An embodiment of this diagnostic also checks ability to impersonate this account, which tests the password for the account.
- registry paths. This diagnostic checks to ensure the path information for the target system software is correct in the registry.
- web sites. This diagnostic checks the target system web software installation for correctness.
- user extensions. This diagnostic checks to ensure the source configurable rules (for user extensions) are present and will compile.
- web service. This diagnostic checks to make sure the target system service account has proper permissions for the target system web service software to run.
- web application. This diagnostic checks to make sure the target system service account has proper permissions for the target system web application software to run.
- outlook extensions. This diagnostic checks to make sure the target system service account has proper permissions for the target system email system (e.g. outlook) extension software to run.
- identity key. This diagnostic checks whether the target system service account has proper permissions to access its identity key in the registry, which is used for proper operation of the software.
- greeting check. An embodiment of this diagnostic checks the ability to get and set a test user's recorded greetings. An embodiment of this diagnostic takes a test user email address as a parameter.
- pin code. This diagnostic checks to see if a pin code for a test user is correct. An embodiment of this diagnostic takes a test user email address and a pin code for parameters.

The following is a non-exhaustive description of various embodiments of systems in which embodiments of the diagnostic methods and systems described herein may be employed. For example, diagnostic methods and

systems described herein may be employed in an interface module (IM) of an integrated multi-media communication system.

As mentioned above, the configuration files of an MCS may be saved for backup purposes and uploaded on demand to the MCS. A configuration file may contain any information (data, code, or otherwise) that is used to enable the MCS to operate in its environment. Configuration information may be very detailed and large, and is not limited to any particular implementation. For example, in various embodiments, the configuration information may comprise network addresses of the IM, type of telephony integration, network address of the MCS, auto-attendant telephone numbers, voicemail main numbers, time-out values, specific configuration values for the telephony integration, and/or other information. Configuration includes but is not limited to any configuration including any provisioning or the like.

Figure 54 is a block diagram of an embodiment including configuration files and code on an MCS. Shown are MCS1, 5410A, and MCS2 5410B 5420. MCS1 includes memory 5470 and request block 5471. Memory 5470 includes configuration 5473 and code 5472. IM 5420 includes configuration information 5475 and code 5476. MCS1 5410A and MCS2 5410B are coupled to IM 5420, and network component 5474 is coupled to IM 5420 and/or MCS1 and/or MCS2. Thus, configuration information 5473 maybe saved for back-up purposes on or through interface module 5420. Alternatively, configuration information 5473 may be pushed automatically from IM 5420 or other network component such as network component 5474 in order to allow MCS1 to reconfigure itself. In another embodiment, configuration information 5473 is pulled automatically from IM 5420 or other network component such as network component 5474 in order to allow MCS1 to reconfigure itself. Providing configuration information 5473 (from IM 5420 or other source) to allow MCS1 to configure itself may be used to provide and/or replicate configurations to multiple MCSs, for example, allowing configuration of a set of MCSs in an embodiment. In an embodiment, Memory 5470 may be implemented in any form of storage or other memory such a disk, volatile memory, for example, a Cache as described herein or in other memory according to various embodiments of the invention.

Additionally, as discussed further below, code 5472 MCS1 5410A maybe upgraded. As shown here, code 5476 from interface module 5420 is provided to MCS1 5410A to replace code 5472. Reconfiguration of MCS1 5410A for replacement of configuration code 5473, or replacement of code 5472 may take place at various circumstances. An MCS, such as MCS1 5410A may receive information from interface module 5420 or other device having memory, such as network component 5474, for example, through a push of the information. The information may also or alternatively be pulled. For example, code or logic in MCS1 5410A, such as request block 5471 may request or initiate a pull of information from interface module 5420 or other device having memory, such as network component 5474. MCS1 5410A may pull or otherwise initiate receipt of information on its own initiative or under other circumstances discuss below. Other MCSs such as MCS2 5410B may also receive configuration information and/or code to reconfigure themselves or replace and/or update code. For example, a network may initially include MCS1 but not MCS2. Later, when MCS2 is added, MCS2 automatically receives configuration information and/or code and automatically configures itself and starts operating.

Figure 55 is a flow diagram for providing configuration information to a component such as an MCS according to an embodiment. In an embodiment, the MCS pulls configuration files automatically from the IM or other network components (such as servers, storage-attached disks, PCs, devices, etc) and subsequently automatically reconfigures itself and starts operating. Thus, the MCS requests configuration information (block 5584), reconfigures itself (block 5585), and starts operating (block 5586). According to various embodiments, the MCS may pull this configuration information at any time, at its own initiative (block 5581), according to a time

schedule (block 5582), or otherwise on its own initiative when certain events occur (block 5583). Such events may include, for example, receiving data from the environment such as, but not limited to, network packets, protocol requests, network broadcast messages. The MCS may also periodically or otherwise poll its environment to determine any changes in the environment including, but not limited to, change of data availability, or new MCSs or  
5 IMs available on the network. Responsive to one or more such changes in the environment, according to an embodiment, the MCS pulls the appropriate configuration information from the IM or other network components and subsequently automatically reconfigures itself and starts operating.

In one embodiment, when the MCS starts and initializes, the MCS sends a request and in response receives configuration information from the IM or other source. The information request to the IM may contain information  
10 uniquely identifying the MCS, such as network IP address, the unique number assigned the physical network card, a serial number assigned to the MCS and stored in memory, or other unique identifiers. It may also contain information about hardware specific information associated with the MCS, such as telephony hardware associated with the MCS, number of CPUs and their processing capacity, amount of memory, etc. Additionally or alternatively, this information is stored in a component in the private or public communications network, and is  
15 looked up based on the unique identifier. In one embodiment, the hardware specific information is made available over a public or other network. For example, the hardware specific information may be made available on a server or other network component on the public network. The manufacturer of the MCS or other organization may make this information available on the server or other network component.

The request for configuration information may be preceded by a network broadcast, requesting a network  
20 IP address. In one embodiment, a network component will push back information to the MCS which includes its network IP address. This technique may be implemented using protocols such as BOOTP and DHCP.

The MCS may also initially (typically, but not limited to, before fetching configuration information) fetch the code that it is executing to operate in its environment. The code may be fetched in a manner and/or under the circumstances described herein for fetching of configuration file. Upon request, the code may be pushed from any  
25 network component (e.g., IM, PC, server, device, and/or MCS) to the MCS. The code may be in encrypted or raw form, and is typically, but not necessarily, compressed for efficient transfer. The code may also or alternatively be cryptographically signed (for example, with a digital signature using public key cryptography). After receiving the code, the MCS decrypts and/or decompresses the code, if applicable, and starts executing. The code may be store in any place and in any form of memory in the MCS. The code is typically the computer readable software code  
30 instructions used by the MCS in its operation, but can include any form of code, such as object code (e.g., binary code), source code, interpreted code, on-the-fly interpreted code or other software or code. The code may include the entire operation system (OS), system software and application software of the MCS and or any parts or combinations of the foregoing. According to an embodiment, the code is pushed to the MCS rather than being pulled to the MCS. The code may be pushed to the MCS using an HTTP "POST" protocol according to an  
35 embodiment. The code may be provided to multiple MCSs as described herein to provide code to a set of MCSs according to an embodiment.

Figure 56 is a flow diagram for upgrading code on a component such as an MCS according to an embodiment. In an embodiment, updates of code are provided in the MCS using a "one-click" upgrade system. In this case, the operator retrieves code from a public or private communications network, uses the MCS administration  
40 interface to point to the code, and pushes an "upgrade" button. Thus, a selection of which code is to be uploaded is received, for example, from an operator (block 5687) and, a request is received to upgrade (block 5688). In one embodiment, the MCS will subsequently fetch the code (block 5689), store it in a separate area of memory, restart

into a special upgrade system code image, delete the old code, copy over the new code (block 5690), and restart, using the new code. The upgrade system may create a separate copy of the old configuration files and restore them into the new upgrade system, once the code upgrade is complete. Under one embodiment, the updated code includes only code that has changed as compared to code already provided on the MCS. Under another

5 embodiment of the invention, an upgrade of the MCS software includes replacement of the entire software image of the software which provides the MCS function including the operating system. During such a replacement of the entire image, installation code may remain on the MCS according to an embodiment.

Figure 57 is a block diagram of an MCS with upgradeable code according to an embodiment of the invention. According to one embodiment of the invention MCS 5710E includes memory partitioned into at least

10 three partitions including main system code 5792 and an install system code 5793. Memory on the MCS also has a partition for an upgrade package code 5794. According to one embodiment of an upgrade process, install system code 5793 causes code from the current main system code 5792 to be replaced with upgrade package 5794. Upgrade package 5794 is provided by an IM, other network component or other source. Main system code 5792 may be structured to run in a mode in which main system code 5792 cannot access other partitions such as install

15 system code.

The upgrade process can include verifying the integrity of the upgrade package such as by a checksum and/or other verification. The verification may also or alternatively involve verification of the digital signature of the code that has been cryptographically signed. In various embodiments, the verification checks for error, malicious code and/or unauthorized code changes. According to an embodiment, the upgrade process includes

20 restarting the upgrade from the beginning or where the process left off in the event of an error. Such an error may be an error in the transfer, unpacking, installation or other error. The system may track where it is within the upgrade process and appropriately restart the upgrade at this point in the event of an error.

With the combination of network address fetching, code fetching, one-click upgrades, configuration information fetching, contact caching, contact list caching, (voice) message caching, and other forms of caching

25 mentioned above, an embodiment of the invention includes a "stateless" MCS where no information is deliberately persisted in the MCS and all code and/or data can be restored from other network components. One advantage of such an embodiment is relative ease in deploying a new MCS when more capacity is needed. Another advantage relative ease in replacing original MCS hardware with new MCS hardware, in case the original MCS hardware has faulty components and/or is not able to continue operation. A further advantage is ease in updating an MCS with

30 new code, when newer versions are available. Caching of data may be implemented in accordance with the description of caching of user information, voicemails and other data as described herein and the other patent applications referenced herein. Thus, an embodiment of the invention may include an MCS in which configuration information, user information, voicemail and other messages, and/or other data is loaded from storage outside of the MCS.

The MCS can be enabled to allow secure access from remote sites according to an embodiment. When

35 enabled, the MCS will establish a coupling through the private and/or public communications networks to a remote network component. The network component network address and network port is specified by the administrator when enabling this secure access. Following this, a user is able to login into and/or access the MCS from the remote node via the coupling. This access may be used by technical personnel to configure, diagnose, and repair the

40 software in the MCS from a remote configuration. In one embodiment, this coupling is established using a Secure Shell ("SSH") tunnel and a user is then able to login via an SSH client to the MCS and access a webserver on the



MCS. In other embodiments, the coupling is not limited to using SSH and the subsequent user access is not limited to SSH clients or webservice access.

5 The coupling allows remote access through the network and its firewalls to the MCS, access which otherwise may not be possible because of the security restrictions (firewalls) typically put in place to limit remote access to the private communications network. The coupling also avoids providing general VPN access to the private communications network. Since this feature does give remote access, it is typically only enabled on demand (when remote access to the MCS is required) and is disabled on demand or automatically after a time-out.

10 An ICS of an embodiment includes an integrated messaging system. The integrated messaging system includes: a communication server that couples among networks of different types; an interface module that couples to the communication server, wherein the interface module pulls a plurality of user information from a messaging server of a network, wherein the user information includes information relevant to at least the network; and a cache store that couples to the communication server and to the interface module to hold at least one of information from the communication server and the user information pulled from messaging server, the interface module directing a message from at least one of the messaging server and the cache to at least one device on the networks using the user information.

15 In an embodiment, a type of the message includes at least one of a voice mail type, an email type, a multimedia messaging system type, an instant messaging type, and a short messaging system type.

20 The communication server may further include a plurality of applications including at least one voice application, wherein at least one voice browser couples the applications to a communications exchange, wherein the plurality of applications further includes at least one mobile application.

25 In an embodiment, the system further includes a form-based interface that is transferable via an electronic message from the messaging server, the form-based interface enabling user actions on information of the messaging server using information of the cache via a coupling with at least one of the communication server and the interface module. the actions on information of the messaging server include directing and playing a voice mail message to the device, generating a reply voice mail message for use in replying to the voice mail message, generating a forwarding voice mail message for use in forwarding the voice mail message, and generating an audio call to a caller leaving the voice mail message.

30 In an embodiment, the system further includes at least one form, wherein the messaging server transfers the form to a client device via a network coupling with the client device, wherein an embedded browser control of the form establishes a coupling between at least one client device and at least one of the interface module and the communication server.

In an embodiment, the system further includes a web server that couples the communication server to the interface module.

In an embodiment, the interface module couples to a groupware application of the messaging server.

35 An embodiment further includes a detector that detects at least one state of the messaging server, wherein the communication server operates in accordance with a plurality of operating modes, wherein the communication server selects an operating mode in response to the state of the messaging server.

40 An embodiment further includes a first message of a first type, wherein the communication server generates the first message in response to receiving a data stream at an input, and may include a second message of a second type, wherein the messaging server generates the second message in response to determining the first type of the first message, wherein the second message includes data of the first message.

The first type may be a voice mail type and the second type may be an electronic mail (email) type.

The at least one device on the networks includes a first set of client devices coupled to the network and a second set of client devices coupled to a public network of the networks, wherein client devices of the first set and the second set include at least one of portable computers, portable telephones, cellular telephones, personal digital assistants, and multi-modal devices.

5 An embodiment further includes: a second messaging server that couples to the messaging server via the networks; a second interface module that couples to at least one of the messaging server and the second messaging server; and a second communication server that couples among the networks via at least one of the interface module and the second interface module.

10 The user information includes information of at least one user, the information including at least one of network mail box number, name, telephone extension, at least one telephone number, email address, greeting, site identification.

In an embodiment, the interface module is hosted on the messaging server.

15 An ICS as described herein further includes a device comprising: at least one server that couples messaging applications among a communication network and a messaging network; at least one interface module that couples to the messaging applications and the messaging network, the interface module transferring message information between the messaging network and the messaging application and retrieving a plurality of user information from the messaging server, wherein the user information includes information of users of the messaging network; and a cache store that couples to the server and to the interface module to hold at least one of the message information and the user information, the server manipulating the message information using the user information.

20 The messaging information may include information of messages that include at least one of a voice mail type, an email type, a multimedia messaging system type, an instant messaging type, and a short messaging system type.

25 In an embodiment, the server further comprises: at least one voice browser; and a plurality of applications that couple to the voice browser, the applications including at least one voice application and at least one mobile application, wherein the voice browser couples the applications to a communications exchange.

30 An embodiment further includes a web server that couples at least one of the server and the interface module to a client device of the user via a form-based interface of the client device, wherein the server transfers the form-based interface via an electronic message to the client device via at least one of the interface module and the messaging network, and wherein the form-based interface allows the user to control actions on information of the messaging network using information of the cache store.

The actions may include directing and playing a voice mail message to the client device, generating a reply voice mail message for use in replying to the voice mail message, generating a forwarding voice mail message for use in forwarding the voice mail message, and generating an audio call to a caller leaving the voice mail message.

The server couples to a groupware application of the messaging network in an embodiment.

35 An embodiment further includes a detector that detects at least one state of the messaging network, wherein the server selects an operating mode in response to a state of the messaging network.

40 An embodiment further includes a first message of a first type, wherein the server generates the first message in response to receiving a data stream from the communication network; and a second message of a second type, wherein the messaging network generates the second message in response to determining the first type of the first message, wherein the second message includes data of the first message.

In an embodiment, the user information includes information of at least one user, the information including at least one of network mail box number, name, telephone extension, at least one telephone number, email address, greeting, site identification.

5 An ICS as described herein further includes a method comprising: receiving data streams from networks of different types; generating messages at a communication server using information of the data streams; transferring the messages, wherein transferring includes at least one of caching information of the messages and forwarding the messages to a messaging server; retrieving user information from the messaging server, wherein the user information includes information relevant to at least the network; caching the user information from the pulling; and directing the message from at least one of the messaging server and a cache to at least one device on the networks  
10 using the cached user information.

In an embodiment, a type of the message includes at least one of a voice mail type, an email type, a multimedia messaging system type, an instant messaging type, and a short messaging system type.

An embodiment further includes: transferring information of a form-based interface to the device via an electronic message; forming a coupling between the device and the communication server using information of the form-based interface; and directing the message by controlling actions on the messages using the cached user  
15 information, wherein a user directs actions on the messages from the device.

In an embodiment, the actions include directing and playing the message to the device, generating a reply message for use in replying to the message, generating a forwarding message for use in forwarding the message, and generating an audio call to a caller leaving the message.

20 The method can include detecting at least one state of the messaging server, and controlling an operating mode of the communication server in response to the detected state.

The messages may include a first type message, wherein generating the messages includes generating messages of the first type in response to the received data streams, wherein the messages include a second type message, wherein transferring further includes transferring messages of the second type to the device in response to  
25 detecting the first type message.

An embodiment further includes: assigning a site identification to a user; filtering the user information of users of a plurality of sites using the site identification; forming sets of users in accordance with the filtered user information; associating each set of users with at least one of the networks.

An ICS as described herein further includes a device comprising: means for receiving data streams from  
30 networks of different types; means for generating messages at a communication server using information of the data streams; means for transferring the messages, wherein transferring includes at least one of caching information of the messages and forwarding the messages to a messaging server; means for retrieving user information from the messaging server, wherein the user information includes information relevant to at least the network; means for caching the user information from the pulling; and means for directing the message from at least one of the  
35 messaging server and the cache to at least one device on the networks using the user information of the cache.

An ICS of an embodiment includes a device comprising a server including a plurality of messaging applications, wherein the server is coupled to a cache and a network of a plurality of networks, wherein an address of the network is assigned to the server and the server is configured, wherein the server retrieves a plurality of user information from the network and caches the retrieved user information, the user information including information  
40 of users of the network, wherein the server transfers message information between the plurality of networks and the messaging applications using the cached user information.

5 " An ICS of an embodiment includes a system comprising a plurality of servers each of which includes a plurality of messaging applications, wherein each server is coupled in succession to a cache and a network of a plurality of networks, wherein an address of the network is assigned to each server and each server is configured, wherein each server retrieves a plurality of user information from the network and caches the retrieved user information, the user information including information of users of the network, wherein each server transfers message information between the plurality of networks and the messaging applications using the cached user information.

10 An ICS of an embodiment includes a method comprising at least one of providing a communication server and coupling the communication server to at least one network, assigning a network address to the communication server, configuring the communication server, retrieving and caching user information from a messaging server of the network, wherein the user information includes information relevant to at least the network, and transferring messages received at the communication server using the cached user information, wherein transferring includes at least one of caching information of the messages and forwarding the messages to the network.

15 The method of an embodiment further comprises installing in succession a plurality of additional communication servers to the network. The installing of an embodiment comprises at least one of coupling each additional communication server to the network, assigning a network address to each additional communication server, configuring each additional communication server, retrieving and caching user information at each additional communication server from a messaging server of the network, wherein the user information includes information relevant to at least the network, and transferring messages received at each additional communication server using the cached user information, wherein transferring includes at least one of caching information of the messages and forwarding the messages to the network.

20 The components of the ICS described above include any collection of computing components and devices operating together. The components of the ICS can also be components or subsystems within a larger computer system or network. The ICS components can also be coupled among any number of components (not shown), for example other buses, controllers, memory devices, and data input/output (I/O) devices, in any number of combinations. Further, components of the ICS can be distributed among any number/combination of other processor-based components.

25 Aspects of the ICS described herein may be implemented as functionality programmed into any of a variety of circuitry, including programmable logic devices (PLDs), such as field programmable gate arrays (FPGAs), programmable array logic (PAL) devices, electrically programmable logic and memory devices and standard cell-based devices, as well as application specific integrated circuits (ASICs). Some other possibilities for implementing aspects of the ICS include: microcontrollers with memory (such as electronically erasable programmable read only memory (EEPROM)), embedded microprocessors, firmware, software, etc. Furthermore, aspects of the ICS may be embodied in microprocessors having software-based circuit emulation, discrete logic (sequential and combinatorial), custom devices, fuzzy (neural) logic, quantum devices, and hybrids of any of the above device types. Of course the underlying device technologies may be provided in a variety of component types, e.g., metal-oxide semiconductor field-effect transistor (MOSFET) technologies like complementary metal-oxide semiconductor (CMOS), bipolar technologies like emitter-coupled logic (ECL), polymer technologies (e.g., silicon-conjugated polymer and metal-conjugated polymer-metal structures), mixed analog and digital, etc.

40 It should be noted that the various functions or processes disclosed herein may be described as data and/or instructions embodied in various computer-readable media, in terms of their behavioral, register transfer, logic component, transistor, layout geometries, and/or other characteristics. Computer-readable media in which such

formatted data and/or instructions may be embodied include, but are not limited to, non-volatile storage media in various forms (e.g., optical, magnetic or semiconductor storage media) and carrier waves that may be used to transfer such formatted data and/or instructions through wireless, optical, or wired signaling media or any combination thereof. Examples of transfers of such formatted data and/or instructions by carrier waves include, but  
5 are not limited to, transfers (uploads, downloads, e-mail, etc.) over the Internet and/or other computer networks via one or more data transfer protocols (e.g., HTTP, FTP, SMTP, etc.). When received within a computer system via one or more computer-readable media, such data and/or instruction-based expressions of components and/or processes under the ICS may be processed by a processing entity (e.g., one or more processors) within the computer system in conjunction with execution of one or more other computer programs.

10 Unless the context clearly requires otherwise, throughout the description and the claims, the words “comprise,” “comprising,” and the like are to be construed in an inclusive sense as opposed to an exclusive or exhaustive sense; that is to say, in a sense of “including, but not limited to.” Words using the singular or plural number also include the plural or singular number respectively. Additionally, the words “herein,” “hereunder,” “above,” “below,” and words of similar import refer to this application as a whole and not to any particular portions  
15 of this application. When the word “or” is used in reference to a list of two or more items, that word covers all of the following interpretations of the word: any of the items in the list, all of the items in the list and any combination of the items in the list.

The above description of illustrated embodiments of the ICS is not intended to be exhaustive or to limit the ICS to the precise form disclosed. While specific embodiments of, and examples for, the ICS are described herein  
20 for illustrative purposes, various equivalent modifications are possible within the scope of the ICS, as those skilled in the relevant art will recognize. The teachings of the ICS provided herein can be applied to other processing systems and methods, not only for the systems and methods described above.

The elements and acts of the various embodiments described above can be combined to provide further embodiments. These and other changes can be made to the ICS in light of the above detailed description.

25 In general, in the following claims, the terms used should not be construed to limit the ICS to the specific embodiments disclosed in the specification and the claims, but should be construed to include all processing systems that operate under the claims. Accordingly, the ICS is not limited by the disclosure, but instead the scope of the ICS is to be determined entirely by the claims.

30 While certain aspects of the ICS are presented below in certain claim forms, the inventors contemplate the various aspects of the ICS in any number of claim forms. For example, while only one aspect of the ICS is recited as embodied in machine-readable medium, other aspects may likewise be embodied in machine-readable medium. Accordingly, the inventors reserve the right to add additional claims after filing the application to pursue such additional claim forms for other aspects of the ICS.

## CLAIMS

WHAT IS CLAIMED IS:

1. A system comprising:  
a messaging server that receives messages;  
5 an interface that couples the messaging server to a network server and a client device of the network;  
a form-based user interface, wherein the messaging server causes a form including form data to be transferred to the client device via the network, wherein execution of the form data generates the form-based user interface on the client device; and  
a communication coupling between the client device and at least one of the interface and the messaging  
10 server, wherein the form data controls communications between the client device and the messaging server in response to at least one user selection via the form-based user interface, wherein the communication coupling is independent of the network.
2. The system of claim 1, wherein a structure of the form is native to the network.
3. The system of claim 1, wherein the form includes an embedded browser control for use in  
15 establishing the communication coupling.
4. The system of claim 1, wherein the messaging server causes the form and form data to be stored on a network server of the network.
5. The system of claim 4, wherein the messaging server transfers the form to the client device via the network.
- 20 6. The system of claim 4, wherein an embedded browser control of the form establishes a coupling between the client device and at least one of the interface and the messaging server.
7. The system of claim 1, wherein the form-based user interface is transferable via an electronic message from the network, the form-based interface enabling actions on at least one of user information and message information by users of the client device.
- 25 8. The system of claim 7, wherein the message information includes at least one of message contents and message description.
9. The system of claim 7, wherein the messaging server caches at least one of the user information and the message information.
10. The system of claim 7, wherein the network stores at least one of the user information and the  
30 message information.
11. The system of claim 7, wherein the actions include routing a message, playing a message, replying to a message, forwarding a message, generating an audio call, modifying the user information.
12. The system of claim 1, wherein the form-based user interface includes at least one of a folder area, a contents area, and a tool bar.

13. The system of claim 12, further comprising action buttons displayed in the tool bar, the action buttons controlling actions via the communication coupling on at least one of user information and message information.

5 14. The system of claim 12, wherein a subfolder of the folder area includes a uniform resource locator, wherein selection of the subfolder establishes the communication coupling with the messaging server.

15. The system of claim 14, wherein the form-based user interface further comprises a browser window that displays information of the messages, wherein the uniform resource locator launches the browser window on a display area of the client device.

10 16. The system of claim 1, wherein the form-based user interface further includes a popup window displayed at the client device via the communication coupling, the popup window displaying at least one of selection information and status information of the messages.

17. The system of claim 1, wherein a type of the message includes at least one of a voice mail type, an email type, a multimedia messaging system type, an instant messaging type, and a short messaging system type.

15 18. The system of claim 1, wherein at least one of the messaging server and the interface further comprise a web server.

19. The system of claim 18, wherein the web server provides the communication coupling with the client device independent of a connection of the network.

20 20. The system of claim 1, wherein the messaging server further comprises a plurality of applications including at least one voice application and at least one mobile application, wherein at least one voice browser couples the applications to a communications exchange.

21. The system of claim 1, wherein the network includes the network server coupled to a database and a communications network.

25 22. A system comprising:  
a network including a messaging system coupled to a database, a client device, and a communications network;  
a messaging communication server (MCS) that couples to the network;  
an interface module that couples to the MCS and the messaging system;  
a form-based user interface comprising form data, wherein the messaging system generates the form data in response to a message received from the MCS, wherein the messaging system transfers the form data and  
30 information of the message to the client device via the network; and  
a communication coupling between the client device and at least one of the interface module and the MCS, wherein the communication coupling is independent of the network, the form data controlling the communication coupling in response to at least one user selection via the form-based user interface.

35 23. A device comprising:  
a messaging server that receives messages;

a form-based user interface (FBUI) that includes buttons, wherein the messaging server causes the FBUI to be transferred to a client device of a network using an electronic message transmitted via a first coupling that is a coupling of the network; and

5 a browser control embedded in the FBUI, wherein the browser control establishes a second coupling that couples the client device and the messaging server, wherein the second coupling is independent of the first coupling, the browser control transferring information of actions selected by a user via the buttons, wherein the messaging server applies the actions to at least one of user information and information of the messages.

24. The device of claim 23, wherein the information of the messages includes at least one of message contents and message description.

10 25. The device of claim 23, wherein the messaging server caches at least one of the user information and the information of the messages.

26. The device of claim 23, wherein the network stores at least one of the user information and the information of the messages.

15 27. The device of claim 23, wherein the actions include routing a message, playing a message, replying to a message, forwarding a message, generating an audio call, modifying the user information.

28. The device of claim 23, wherein the form-based user interface includes at least one of a folder area, a contents area, and a tool bar.

29. The device of claim 28, wherein a subfolder of the folder area includes a uniform resource locator, wherein selection of the subfolder establishes the second coupling.

20 30. The device of claim 29, wherein the form-based user interface further comprises a browser window that displays information of the messages, wherein the uniform resource locator launches the browser window on a display area of the client device.

25 31. The device of claim 23, wherein the form-based user interface further includes a popup window displayed at the client device via the second coupling, the popup window displaying at least one of selection information and status information of the messages.

32. The device of claim 23, wherein a type of the messages includes at least one of a voice mail type, an email type, a multimedia messaging system type, an instant messaging type, and a short messaging system type.

33. The device of claim 23, wherein the messaging server further comprises a web server.

30 34. An integrated messaging system, comprising:  
a communication server that couples among networks of different types;  
an interface module that couples to the communication server, wherein the interface module pulls a plurality of user information from a messaging server of a network, wherein the user information includes information relevant to at least the network; and



a cache store that couples to the communication server and to the interface module to hold at least one of information from the communication server and the user information pulled from messaging server, the interface module directing a message from at least one of the messaging server and the cache to at least one device on the networks using the user information.

5           35.       The system of claim 34, wherein a type of the message includes at least one of a voice mail type, an email type, a multimedia messaging system type, an instant messaging type, and a short messaging system type.

          36.       The system of claim 34, wherein the communication server further comprises a plurality of applications including at least one voice application, wherein at least one voice browser couples the applications to a communications exchange.

10          37.       The system of claim 34, wherein the plurality of applications further includes at least one mobile application.

          38.       The system of claim 34, further comprising a form-based interface that is transferable via an electronic message from the messaging server, the form-based interface enabling user actions on information of the messaging server using information of the cache via a coupling with at least one of the communication server and  
15 the interface module.

          39.       The system of claim 34, wherein the actions on information of the messaging server include directing and playing a voice mail message to the device, generating a reply voice mail message for use in replying to the voice mail message, generating a forwarding voice mail message for use in forwarding the voice mail message, and generating an audio call to a caller leaving the voice mail message.

20          40.       The system of claim 34, further comprising at least one form, wherein the messaging server transfers the form to a client device via a network coupling with the client device, wherein an embedded browser control of the form establishes a coupling between at least one client device and at least one of the interface module and the communication server.

          41.       The system of claim 34, further comprising a web server that couples the communication server to  
25 the interface module.

          42.       The system of claim 34, wherein the interface module couples to a groupware application of the messaging server.

          43.       The system of claim 34, further comprising a detector that detects at least one state of the messaging server.

30          44.       The system of claim 43, wherein the communication server operates in accordance with a plurality of operating modes, wherein the communication server selects an operating mode in response to the state of the messaging server.

          45.       The system of claim 34, further comprising a first message of a first type, wherein the communication server generates the first message in response to receiving a data stream at an input.

46. The system of claim 45, further comprising a second message of a second type, wherein the messaging server generates the second message in response to determining the first type of the first message, wherein the second message includes data of the first message.

5 47. The system of claim 46, wherein the first type is a voice mail type and the second type is an electronic mail (email) type.

48. The system of claim 34, wherein the at least one device on the networks includes a first set of client devices coupled to the network and a second set of client devices coupled to a public network of the networks, wherein client devices of the first set and the second set include at least one of portable computers, portable telephones, cellular telephones, personal digital assistants, and multi-modal devices.

10 49. The system of claim 34, further comprising:  
a second messaging server that couples to the messaging server via the networks;  
a second interface module that couples to at least one of the messaging server and the second messaging server; and  
a second communication server that couples among the networks via at least one of the interface module  
15 and the second interface module.

50. The system of claim 34, wherein the user information includes information of at least one user, the information including at least one of network mail box number, name, telephone extension, at least one telephone number, email address, greeting, site identification.

51. The system of claim 34, wherein the interface module is hosted on the messaging server.

20 52. A device comprising:  
at least one server that couples messaging applications among a communication network and a messaging network;  
at least one interface module that couples to the messaging applications and the messaging network, the interface module transferring message information between the messaging network and the messaging application  
25 and retrieving a plurality of user information from the messaging server, wherein the user information includes information of users of the messaging network; and  
a cache store that couples to the server and to the interface module to hold at least one of the message information and the user information, the server manipulating the message information using the user information.

30 53. The device of claim 52, wherein the messaging information comprises information of messages that include at least one of a voice mail type, an email type, a multimedia messaging system type, an instant messaging type, and a short messaging system type.

54. The device of claim 52, wherein the server further comprises:  
at least one voice browser; and  
a plurality of applications that couple to the voice browser, the applications including at least one voice  
35 application and at least one mobile application, wherein the voice browser couples the applications to a communications exchange.

55. The device of claim 52, further comprising a web server that couples at least one of the server and the interface module to a client device of the user via a form-based interface of the client device.

56. The device of claim 55, wherein the server transfers the form-based interface via an electronic message to the client device via at least one of the interface module and the messaging network.

57. The device of claim 56, wherein the form-based interface allows the user to control actions on information of the messaging network using information of the cache store.

58. The device of claim 57, wherein the actions include directing and playing a voice mail message to the client device, generating a reply voice mail message for use in replying to the voice mail message, generating a forwarding voice mail message for use in forwarding the voice mail message, and generating an audio call to a caller leaving the voice mail message.

59. The device of claim 52, wherein the server couples to a groupware application of the messaging network.

60. The device of claim 52, further comprising a detector that detects at least one state of the messaging network, wherein the server selects an operating mode in response to a state of the messaging network.

61. The device of claim 52, further comprising:  
a first message of a first type, wherein the server generates the first message in response to receiving a data stream from the communication network; and  
a second message of a second type, wherein the messaging network generates the second message in response to determining the first type of the first message, wherein the second message includes data of the first message.

62. The device of claim 52, wherein the user information includes information of at least one user, the information including at least one of network mail box number, name, telephone extension, at least one telephone number, email address, greeting, site identification.

63. A method comprising:  
receiving data streams from networks of different types;  
generating messages at a communication server using information of the data streams;  
transferring the messages, wherein transferring includes at least one of caching information of the messages and forwarding the messages to a messaging server;  
retrieving user information from the messaging server, wherein the user information includes information relevant to at least the network;  
caching the user information from the pulling; and  
directing the message from at least one of the messaging server and a cache to at least one device on the networks using the cached user information.

64. The method of claim 63, wherein a type of the message includes at least one of a voice mail type, an email type, a multimedia messaging system type, an instant messaging type, and a short messaging system type.

65. The method of claim 63, further comprising:

transferring information of a form-based interface to the device via an electronic message;

forming a coupling between the device and the communication server using information of the form-based interface; and

5 directing the message by controlling actions on the messages using the cached user information, wherein a user directs actions on the messages from the device.

66. The method of claim 63, wherein the actions include directing and playing the message to the device, generating a reply message for use in replying to the message, generating a forwarding message for use in forwarding the message, and generating an audio call to a caller leaving the message.

10 67. The method of claim 63, further comprising:

detecting at least one state of the messaging server; and

controlling an operating mode of the communication server in response to the detected state.

68. The method of claim 63, wherein the messages include a first type message, wherein generating the messages includes generating messages of the first type in response to the received data streams.

15 69. The method of claim 68, wherein the messages include a second type message, wherein transferring further includes transferring messages of the second type to the device in response to detecting the first type message.

70. The method of claim 63, further comprising:

assigning a site identification to a user;

20 filtering the user information of users of a plurality of sites using the site identification;

forming sets of users in accordance with the filtered user information;

associating each set of users with at least one of the networks.

71. A device comprising:

means for receiving data streams from networks of different types;

25 means for generating messages at a communication server using information of the data streams;

means for transferring the messages, wherein transferring includes at least one of caching information of the messages and forwarding the messages to a messaging server;

means for retrieving user information from the messaging server, wherein the user information includes information relevant to at least the network;

30 means for caching the user information from the pulling; and

means for directing the message from at least one of the messaging server and the cache to at least one device on the networks using the user information of the cache.

72. A device comprising a server including a plurality of messaging applications, wherein the server is coupled to a cache and a network of a plurality of networks, wherein an address of the network is assigned to the server and the server is configured, wherein the server retrieves a plurality of user information from the network and caches the retrieved user information, the user information including information of users of the network, wherein

35

the server transfers message information between the plurality of networks and the messaging applications using the cached user information.

73. A system comprising a plurality of servers each of which includes a plurality of messaging applications, wherein each server is coupled in succession to a cache and a network of a plurality of networks, wherein an address of the network is assigned to each server and each server is configured, wherein each server retrieves a plurality of user information from the network and caches the retrieved user information, the user information including information of users of the network, wherein each server transfers message information between the plurality of networks and the messaging applications using the cached user information.

74. A method comprising:  
providing a communication server and coupling the communication server to at least one network;  
assigning a network address to the communication server;  
configuring the communication server;  
retrieving and caching user information from a messaging server of the network, wherein the user information includes information relevant to at least the network; and  
transferring messages received at the communication server using the cached user information, wherein transferring includes at least one of caching information of the messages and forwarding the messages to the network.

75. The method of claim 74, further comprising installing in succession a plurality of additional communication servers to the network.

76. The method of claim 75, wherein installing comprises:  
coupling each additional communication server to the network;  
assigning a network address to each additional communication server;  
configuring each additional communication server;  
retrieving and caching user information at each additional communication server from a messaging server of the network, wherein the user information includes information relevant to at least the network; and  
transferring messages received at each additional communication server using the cached user information, wherein transferring includes at least one of caching information of the messages and forwarding the messages to the network.

77. A method comprising:  
transferring a first message to a messaging server from a communication server via a first coupling;  
generating a second message at the messaging server in response to a type of the first message and transferring the second message to a client device via a second coupling, wherein the second message is of a different type and includes data of the first message;  
transferring to the client device form data that corresponds to the first message; and  
establishing a third coupling between the client device and the communication server using the form data.

78. The method of claim 77, wherein the messaging server transfers the form to the client device via the second coupling.

79. The method of claim 77, wherein the form includes an embedded browser control for use in establishing the third coupling.

80. The method of claim 77, further comprising directing actions on the first message from the client device using the form data, the actions directed via the third coupling.

5 81. The method of claim 77, wherein the form data conforms to a message structure of the messaging server.

82. The method of claim 77, wherein the form data includes a form-based user interface form ("FBUI"), and wherein the FBUI is transferred via an electronic mail message from the messaging server, the FBUI enabling actions on at least one of user information and message information by users of the client device.

10 83. The method of claim 82, wherein the messaging server caches at least one of the user information and the message information.

84. The method of claim 82, wherein the network stores at least one of the user information and the message information.

15 85. The method of claim 82, wherein the actions include routing a message, playing a message, replying to a message, forwarding a message, generating an audio call, and modifying the user information.

86. The method of claim 77, further comprising action buttons displayed on the form-based user interface, the action buttons controlling actions via the communication coupling on at least one of user information and message information.

20 87. The method of claim 77, wherein the third coupling occurs through at least one network node other than the communication server.

88. A method comprising:  
generating a voice message to include a property, including a message type;  
transferring the voice message to a mail server from a communication server via a first coupling;  
generating an electronic mail message at the mail server in response to the property and transferring the  
25 mail message to a client device via a second coupling, wherein the mail message includes information of the voice message;

transferring form data to the client device via the second coupling, the form data corresponding to the property;

30 and  
establishing a third coupling between the client device and the communication server using the form data;

directing actions on the voice message from the client device via the third coupling, the third coupling established using the form data.

89. The method of claim 88, wherein the form data is displayed in response to selection of a voice message listed in a message structure format of the mail server.

90. The method of claim 89, wherein the form data includes a plurality of action buttons for allowing the selection of actions to be performed on a selected voice message.

91. The method of claim 90, wherein the form data includes an embedded browser, and when an action button is selected, the browser is launched to establish the third coupling.

5 92. The method of claim 90, wherein a voice message listing in the message structure format includes an embedded browser, and when the listing is selected, the browser is launched to establish the third coupling.

93. The method of claim 90, wherein the plurality of action buttons includes:  
a play voice message on phone button;  
a play voice message on computer button;  
10 a call sender button;  
a reply by voice message button; and  
a forward voice message button.

94. A method comprising:  
transferring form data to a client device from a first server via a first coupling;  
15 receiving a first message at the first server, the first message being of a first type;  
generating a second message at the first server in response to the first message, the second message being  
of a second type;  
transferring the second message to the client device via the first coupling;  
establishing a second coupling between the client device and a second server using the form data; and  
20 controlling the first message at the first server using selections made via the form data and transferred from  
the client device to the first server via the second coupling.

95. The method of claim 94, wherein controlling the first message at the first server comprises controlling via a web browser launched by manipulating the form data.

96. The method of claim 94, wherein the first type of message is a voice message and wherein the  
25 second type of message is an email message according to a message structure of the first server.

97. The method of claim 96, wherein the first server is an enterprise groupware email server.

98. The method of claim 94, wherein the first coupling is a local area network ("LAN") coupling and wherein the second coupling is a web browser coupling.

99. A method comprising:  
30 receiving an audio stream at a first server and generating a voice message using information of the audio  
stream;  
transferring the voice message to a second server via a first path;  
generating an electronic mail message at the second server in response to receiving the voice message, the  
electronic mail message including identification information of the voice message;  
35 transferring the electronic mail message to a client device via a second path;

displaying the electronic mail message using form data received from the second server via the second path;

establishing a third path between the client device and the first server using control data of the form; and controlling actions on the voice message at the second server by communicating actions selected in the

5 form to the first server via the third path.

100. The method of claim 99, wherein the first path is a path within an enterprise network.

101. The method of claim 100, wherein the third path is an Internet path.

102. The method of claim 101, wherein the second path is an enterprise network path.

103. The method of claim 99, wherein displaying the electronic mail message comprises displaying an email application form on a client device display, wherein voice mail messages are listed.

10

104. The method of claim 103, further comprising, in response to a selection of a listed voice mail message, launching a popup window that includes a plurality of action button for performing actions on the displayed message.

105. The method of claim 104, wherein selection of an action button causes communication of the desired action to the first server, including establishing the third path.

15

106. The method of claim 105, wherein establishing the third path includes establishing communication between the client device and a web server.

107. The method of claim 99, wherein the third path occurs through at least one network node other than the first server.

108. The method of claim 99, wherein the actions cause transfer of information from the first sever to the client device through the third path.

20

109. A system comprising:  
a communication server that couples among networks of different types to receive messages and requests for messages, wherein the networks include a messaging server;

25 at least one cache server that couples to the communication server and caches the received messages; and

a detector that couples to detect a state of the messaging server, wherein the communication server transfers the received messages to the messaging server for storage when the state is available, wherein the communication server in response to a request for the message retrieves the message from the cache when the state is unavailable and retrieves the message from the messaging server when the state is available.

30

110. The system of claim 109, wherein the detector polls the messaging server to detect the state.

111. The system of claim 109, wherein the detector receives an exception from the messaging server to detect the state.



112. A method comprising:

receiving a message at a first device;

assigning identification information to the message and caching the message;

transferring the received message to a second device in response to a detected state of the second

5 device;

receiving a request for the message at the first device from a requesting device;

retrieving the message from at least one of the first device and the second device in response to the detected state; and

transmitting the retrieved message in response to the request.

10 113. The method of claim 112, wherein the message includes at least one of a voice mail type, an email type, a multimedia type, an instant message type, and a short messaging system type.

114. The method of claim 112, wherein caching the message includes caching at least one of a description of the message, content of the message, mailbox number of a user to whom the message is directed, and a time stamp.

15 115. The method of claim 112, wherein retrieving includes retrieving information of the requested message from the first device in response to an unavailable state of the second device.

116. The method of claim 112, wherein retrieving includes retrieving information of the requested message from the second device in response to an available state of the second device.

20 117. The method of claim 112, wherein transferring includes:  
detecting an unavailable state of the second device;  
holding the message in the cache; and  
retrieving the message from the cache and transferring the message to the second device in response to a change in the detected state to available.

25 118. The method of claim 112, wherein transferring includes:  
detecting an available state of the second device; and  
transferring the message to the cache and to the second device.

119. The method of claim 112, further comprising tracking a status of each message at the first device and caching the status with the message, wherein the status includes read, unread, retrieved, not retrieved, and expired.

30 120. The method of claim 112, wherein receiving the message further comprises receiving the message in at least one of a plurality of servers of the first device.

121. The method of claim 120, wherein caching the message further comprises caching the message in a first store when the message is received at a first server of the first device.

35 122. The method of claim 120, wherein caching the message further comprises caching the message in a second store when the message is received at a second server of the first device.

123. The method of claim 120, wherein receiving the request further comprises receiving the request in at least one of the plurality of servers of the first device.

124. The method of claim 123, wherein retrieving the message further comprises retrieving the message from a second store when the request is received at the first server and the message is received and cached at the second server of the first device.

125. The method of claim 124, further comprising caching the retrieved message at a first store, wherein the first store is a component of the first server.

126. The method of claim 123, wherein retrieving the message further comprises retrieving the message from a first store when the request is received at the second server and the message is received and cached at the first server of the first device.

127. The method of claim 126, further comprising caching the retrieved message at a second store, wherein the second store is a component of the second server.

128. The method of claim 112, further comprising:  
retrieving user information from the second device; and  
caching the retrieved user information, the user information including at least one of name, a plurality of greetings, class of service, permissions, and personal information.

129. The method of claim 128, further comprising:  
receiving user information at the first device, wherein the user information received at the first device is input by the user to the first device; and  
caching the user information received at the first device.

130. The method of claim 129, wherein caching the user information further comprises:  
caching the user information in a first store when the user information is received at a first server of the first device; and  
caching the user information in a second store when the user information is received at a second server of the first device.

131. The method of claim 130, further comprising:  
receiving a call during which the message is received; and  
retrieving at least one item of the user information during the call.

132. The method of claim 131, wherein retrieving the item of the user information further comprises retrieving the item from the second store when the call is received at the first server and the item is received and cached at the second server of the first device.

133. The method of claim 131, wherein retrieving the item further comprises retrieving the item from a first store when the call is received at the second server and the item is received and cached at the first server of the first device.

134. A method comprising:

receiving a message at a first device;

assigning identification information to the message and caching the message;

detecting a state of a second device;

5 transferring the received message to the second device when the second device is in an available state;

receiving a request for the message at the first device; and

10 transmitting the message in response to the request, wherein the message is retrieved for the transmitting from the first device in response to an unavailable state of the second device, wherein the message is retrieved for the transmitting from the second device in response to the available state of the second device.

135. A device comprising:

means for receiving a message at a first device;

means for assigning identification information to the message and caching the message;

15 means for transferring the received message to a second device in response to a detected state of the second device;

means for receiving a request for the message at the first device from a requesting device;

means for retrieving the message from at least one of the first device and the second device in response to the detected state; and

means for transmitting the retrieved message in response to the request.

20 136. A device comprising:

a protocol system that couples to a local cache system that caches data;

a first communication system that couples the protocol system to a remote cache system, the first communication system transferring a first type of information between the remote cache system and the protocol system;

25 a second communication system that couples the protocol system to the remote cache system, the second communication system transferring a second type of information between the remote cache system and the protocol system; and

30 a third communication system that couples the protocol system to a communication server, the third communication system transferring at least one of the first type and second type of information between the protocol system and the communication server.

137. The device of claim 136, wherein the first communication system couples the protocol system to a remote protocol system of the remote cache.

138. The device of claim 136, wherein the second communication system couples the protocol system to a remote cache system of the remote cache using a web server of the remote cache.

35 139. The device of claim 136, wherein the third communication system couples the protocol system to a message application.

140. The device of claim 136, wherein the protocol system controls protocols for communications with the first, second, and third communication systems.

141. The device of claim 136, wherein the first type of information includes at least one of a list of messages and a request for the list of messages.

142. The device of claim 136, wherein the second type of information includes at least one of message contents and a request for the message contents.

5 143. The device of claim 136, wherein the local cache system includes at least one of a file system and a management system that manages information held in the file system.

144. The device of claim 136, wherein the first communication system includes a local communication server that couples the protocol system to a remote inter-node protocol server of the remote cache system.

10 145. The device of claim 136, wherein the first communication system includes a local inter-node protocol server that couples the local cache system to an inter-node protocol server of a second remote cache system.

146. The device of claim 136, wherein the second communication system includes a fetching server that couples the protocol system to a remote management system of the remote cache system via at least one web server.

15 147. The device of claim 136, wherein the second communication system includes a web server that couples the management system to a remote protocol system of at least one remote cache system.

148. A system comprising:  
a first cache system that couples to transfer data with a first server;  
a second cache system that couples to transfer data with a second server; and  
20 a communication system that couples to transfer data among the first cache system and the second cache system, wherein the communication system includes a first coupling that transfers a first type of information and a second coupling that transfers a second type of information.

149. The system of claim 148, wherein the first coupling couples a first protocol system of the first cache system to a second protocol system of the second cache system, wherein the first and second protocol systems control communication protocols.  
25

150. The system of claim 149, wherein the first coupling includes a first communication server that couples the first protocol system to a remote inter-node protocol server of the second cache system.

151. The system of claim 149, wherein the first coupling includes a second communication server that couples the second protocol system to a first inter-node protocol server of the first cache system.

30 152. The system of claim 149, wherein the second coupling includes at least one web server that couples at least one of the first protocol system to the second cache system and the second protocol system to the first cache system.

153. The system of claim 148, wherein at least one of the first cache system and the second cache system couple to a plurality of applications of a messaging system.

154. The system of claim 148, wherein the first type of information includes at least one of a list of messages and a request for the list of messages.

155. The system of claim 148, wherein the second type of information includes at least one of message contents and a request for the message contents.

5 156. The system of claim 148, wherein at least one of the first cache system and the second cache system includes a local cache store, the local cache store including at least one of a file system and a management system that manages information held in the file system.

157. The system of claim 148, wherein the data is cached as requested by at least one of the first server and the second server.

10 158. The system of claim 148, wherein the data is replicated in at least one of the first cache system and the second cache system in advance of performance of operations that use the data.

159. A system comprising:  
at least one messaging communication server (MCS); and  
a cache system that couples to the MCS, the cache system including at least one of a local cache  
15 store and a distributed cache store, the cache system further including a communication system that couples to transfer data among the MCS and the cache system, the communication system including a first coupling that transfers a first type of information that includes at least one of a list of messages and a request for the list of messages, the communication system including a second coupling that transfers a second type of information that includes at least one of message contents and a request for the message contents.

20 160. A system, comprising:  
a network that includes a database storing a groupware application and a directory service, wherein the directory service stores user information for use in providing messaging of a first type among client devices coupled to the network;  
a messaging communication server (MCS) that couples to the network and to at least one communication  
25 network, wherein the MCS provides messaging of a second type among client devices coupled to the network; and  
an interface module that couples the MCS to the network, the interface module pushing user information from the database to the MCS, wherein the MCS uses the pushed user information to provide the second type of messaging, wherein the MCS and the interface module provide integration of the messaging of the second type into the network.

30 161. The system of claim 160, further comprising a cache that couples to the MCS, the cache holding the user information pushed by the interface module.

162. The system of claim 160, wherein the interface module detects changes to the user information and incrementally pushes the changes to the MCS.

35 163. The system of claim 161, wherein:  
the MCS includes at least one voice application and various application programming interfaces ("APIs");  
and

messaging of the second type" includes voice messaging, and wherein the cache further couples to a state machine framework, wherein communications among the at least one voice application, the cache, groupware application and a directory service take place via the state machine framework and the various APIs as appropriate to a state of the groupware application.

5           164.    The system of claim 163, wherein the state of the groupware application includes an online state in which the groupware application communicates with the MCS, and an offline state in which the groupware application does not communicate with the MCS.

10           165.    The system of claim 163, wherein the voice applications include a voice mail messaging application, and wherein the user information includes user information particular to the voice messaging application.

15           166.    A integrated messaging system, comprising:  
              an enterprise network that includes a database storing a groupware application and a directory service, wherein the directory service stores user information for use in providing messaging of a first type among client devices coupled to the enterprise network;  
              at least one messaging communication server (MCS) that couples to the enterprise network and to at least one communication network, wherein the MCS provides messaging of a second type among client devices coupled to the enterprise network;  
              an interface module ("IM") that couples the MCS to the enterprise network, the interface module pushing user information from the database to the MCS, wherein the MCS uses the pushed user information to provide the second type of messaging; and  
20           at least one cache that couples to the MCS, wherein the cache holds the user information pushed by the interface module.

25           167.    The system of claim 166, wherein the second type of messaging includes voice mail messaging, and the MCS further comprises voice applications, including a voice mail application, and wherein the MCS stores voice mail-specific information in the user information in the directory service via the IM.

              168.    The system of claim 167, the at least one cache includes:  
              a local cache that is local to a certain MCS; and  
              a distributed cache that is communicatively coupled among multiple MCSs.

30           169.    The system of claim 168, wherein the user information pushed by the IM includes:  
              a global address list ("GAL") that includes information pertaining to all members of the enterprise;  
              public folders that include information of the enterprise that is shared with all users, including shared contact information and shared calendar information;  
              personal contacts folders that include corresponding contact information for each member of the enterprise;  
              and  
35           a user list that includes information of users of the integrated messaging system.

              170.    The system of claim 169, wherein at least one of the GAL, the public folders, and the user list are pushed periodically.

171. The system of claim 169, wherein the personal folders are pushed on demand, including when a user logs into the MCS.

172. The system of claim 169, wherein the personal folders are pushed incrementally, including upon detecting an update of information in a personal folder.

5 173. The system of claim 169, wherein the user list is pushed incrementally, including upon detecting an update of information in the user list.

174. The system of claim 169, wherein updated information in the personal folders is pushed incrementally upon detecting an update of information in a personal folder.

10 175. The system of claim 169, wherein the personal folders include a user contacts folder, and wherein the contacts folder is pushed incrementally, including pushing the contacts folder to the MCS upon a user initially logging into the MCS, wherein the contacts folder is pushed with a timestamp.

176. The system of claim 175, wherein the timestamp is applied prior to accessing the contacts folder for pushing.

15 177. The system of claim 175, wherein pushing the contacts folder incrementally includes receiving the contacts folder and a timestamp subsequent to the user logging into the MCS, including receiving detected updates to the contacts folder.

178. The system of claim 177, wherein pushing the contacts folder incrementally further comprises: the MCS receiving the requested contacts folder, timestamp and a total number of contacts listed in the contacts folder;

20 comparing the received total number of contacts with a cached total number of contacts corresponding to a previous version of the contacts folder; and

if there is a difference between the two totals, requesting the contacts folder to be pushed by the IM to the MCS.

25 179. A method for integrated messaging, comprising: interfacing with a network using an interface module ("IM"), the network including a database storing a groupware application and a directory service, wherein the directory service stores user information for use in messaging of a first type among client devices coupled to the network;

pushing the user information with the IM from the database to a messaging communication server (MCS), wherein the MCS couples to at least one communication network and to the network;

30 caching the pushed user information; and

providing messaging of a second type among the client devices, wherein the MCS uses the pushed user information to provide the messaging of a second type.

180. The method of claim 179, wherein the messaging of the second type includes voice mail messaging, and wherein the MCS comprises voice applications.

35 181. The method of claim 180, further comprising:

"detecting changes to the user information using the IM; and  
pushing the detected changes to the MCS.

182. The method of claim 180, wherein the voice applications perform functions including:  
maintaining shared address lists that all voice mail users can view and edit;  
5 scheduling meetings that include people and conference rooms by viewing associated free or busy  
schedules

sending a new voice mail;  
forwarding a received voice mail;  
exchanging voice mails and corresponding information with the groupware applications; and  
10 granting people other than a voice mail user access to user voice mailboxes on behalf of the user.

183. The method of claim 182, the cache includes:  
a local cache that is local to a certain MCS; and  
a distributed cache that is communicatively coupled among multiple MCSs.

184. The method of claim 183, wherein the user information pushed by the IM includes:  
15 a global address list ("GAL") that includes information pertaining to all members of the enterprise;  
public folders that include information of the enterprise that is shared with all users, including shared  
contact information and shared calendar information;  
personal contacts folders that include corresponding contact information for each member of the enterprise;  
and  
20 a user list that includes information of users of the integrated messaging.

185. The method of claim 184, wherein the GAL and the public folders are each pushed periodically.

186. The method of claim 184, wherein the personal folders are pushed on demand, including when a  
user logs into the MCS.

187. The method of claim 184, wherein the personal folders are pushed incrementally, including upon  
25 detecting an update of information in a personal folder.

188. The method of claim 184, wherein updated information in the personal folders is pushed  
incrementally upon detecting an update of information in a personal folder.

189. The method of claim 184, wherein the personal folders include a user contacts folder, and wherein  
the contacts folder is pushed incrementally, including pushing the contacts folder to the MCS upon a user initially  
30 logging into the MCS, wherein the contacts folder is pushed with a timestamp.

190. The method of claim 189, wherein the timestamp is applied prior to accessing the contacts folder  
for pushing.

191. The method of claim 189, wherein pushing the contacts folder incrementally includes receiving  
the contacts folder and a timestamp subsequent to the user logging into the MCS, including receiving detected  
35 updates to the contacts folder.



192. The method of claim 91, wherein pushing the contacts folder incrementally further comprises:  
the MCS receiving the requested contacts folder, timestamp and a total number of contacts listed in the contacts folder;

5 comparing the received total number of contacts with a cached total number of contacts corresponding to a previous version of the contacts folder; and  
if there is a difference between the two totals, requesting the contacts folder to be pushed by the IM to the MCS.

193. An integrated messaging system, comprising:  
a messaging communication server (MCS) coupled to multiple networks of different types; and  
10 an interface module coupling the mobile communication server to a first type of network, wherein the first type of network includes a groupware application and a directory service, wherein the MCS performs messaging of a second type, including storing and accessing information particular to the second type in the directory service and wherein the messaging of the second type is independent of the first type of network.

194. The system of claim 193, wherein the first type of network includes an enterprise email network,  
15 and wherein the second type of messaging includes voice mail messaging.

195. The system of claim 194, wherein the information particular to the second type of messaging includes voice mail user attributes and voice mail message information.

196. The system of claim 195, wherein the voice mail user attributes include levels of telephone and VM service, whether a user is enabled to access the MCS to use a voice mail system, the user's phone extension,  
20 whether the user's VM notification is enabled, whether an auto attendant is enabled for a user's phone number, and whether an active greeting is on.

197. The system of claim 195, wherein the voice mail user attributes include data specific to a user of a voice mail system hosted by the MCS, and wherein the user attributes are relatively small and are changed relatively infrequently.

25 198. The system of claim 197, wherein the voice mail message information includes data of indefinite length, including stored voice messages and one or more recorded greetings.

199. The system of claim 198, wherein the voice mail user attributes are stored in the directory service, and wherein the voice mail message information is stored on an email database of the enterprise email network.

200. The system of claim 197, wherein the voice mail user attributes are stored in the directory service,  
30 and are used to access the voice message information that is stored in a database of the groupware application.

201. The system of claim 198, wherein the voice mail user attributes are stored in a schema of the directory service, and wherein the interface module receives a voice mail-related request from the user and directs the request to the voice mail user attributes in the schema in a manner customary for an email-related request within the enterprise email network.

202. The system of claim 200, wherein the voice mail user attributes are stored in a schema of the directory service by extending the schema.

203. The system of claim 200, wherein all the voice mail user attributes are stored in a custom attribute of the directory service.

5 204. The system of claim 201, wherein the voice mail related request includes a request to access the voice mail message information.

205. The system of claim 204, wherein the voice mail user attributes include data that points to a location of the voice mail message information in the database.

10 206. The system of claim 199, wherein the one or more recorded greetings are stored in the email database as an attachment to a specific hidden email message.

207. A communication method, comprising:  
performing voice applications within an enterprise network, including telephony applications;  
coupling to a plurality of network types to perform the voice applications, including the enterprise network,  
and a conventional telephone network;  
15 using a groupware application of the enterprise network to store and access information specific to the voice applications; and  
accessing the information to perform the voice applications, including using a network external to the enterprise network.

20 208. The communication method of claim 207, wherein the plurality of network types further include wired and wireless and wireless networks and the Internet.

209. The communication method of claim 207, wherein the voice applications include a voice mail (VM) application for voice mail messaging, and wherein storing information specific to the voice applications includes storing VM user data that is particular to VM users, and storing VM message information that includes VM messages.

25 210. The communication method of claim 209, wherein storing includes storing the VM user data in a directory service of the enterprise network.

211. The communication method of claim 209, wherein storing includes storing the VM message information in a database of the groupware application.

30 212. The communication method of claim 209, wherein storing includes storing the VM message information in an email database of the groupware application.

213. The communication method of claim 209, wherein storing includes storing the VM user data in a directory service of the enterprise network that also stores email user data, and storing the VM message information in an email database of the groupware application.

214. The communication method of claim 209, further comprising performing a voice mail application function, including:

receiving a request from a VM user to access the VM message information;  
looking up the VM user data for the user;  
5 determining a location of a user VM mailbox from the VM user data; and  
accessing the VM message information in the VM mailbox.

215. The method of claim 214, wherein the request includes a request to retrieve one or more VM messages.

10 216. The method of claim 214, wherein the request includes a request to record an outgoing voice message.

217. The method of claim 214, wherein the request includes a request to delete a voice message.

218. The communication method of claim 209, further comprising performing a voice mail application function, including:

15 receiving a call from a caller for a VM;  
determining that the user does not answer;  
looking up the VM user data for the user;  
determining a location of a user VM mailbox from the VM user data;  
accessing the VM message information in the VM mailbox;  
retrieving a recorded greeting; and  
20 playing the recorded greeting to the caller.

219. The communication method of claim 210, wherein the storing further includes storing the VM user data in an attribute of a user object in the directory service.

220. The communication method of claim 210, wherein the storing further includes storing the VM user data in multiple attributes in an extended schema of the directory service.

25 221. The communication method of claim 211, wherein the storing further includes storing the VM message data in an email database of the groupware application, including storing recorded greeting in a hidden message in a user mailbox.

222. A voice messaging method, comprising:  
receiving a request to access enterprise voice mail applications in an enterprise;  
30 accessing voice mail user attributes specific to the user in a directory service of an enterprise groupware application;  
accessing user voice mail message information in a database of the groupware application; and  
using at least one of the voice mail user attributes to perform a voice mail function.

223. The method of claim 222, wherein the voice mail functions include:  
35 recording a caller voice message left for the user and storing the caller voice message as voice mail message information on the database; and

playing a recorded greeting for a caller, including fetching the recorded greeting stored as voice mail message information on the database.

224. The method of claim 222, wherein the voice mail user attributes include an email mailbox number.

225. The method of claim 224, further comprising:

5 using the email mailbox number to access a hidden email message stored as voice mail message information on the database;

retrieving the recorded greeting as an attachment to the hidden message.

226. The method of claim 222, wherein the voice mail functions include:

10 receiving a user request for access to a user voice mail mailbox stored as voice mail message information in an email database of the groupware application;

accessing voice mail user attributes specific to the user in a directory service of the enterprise groupware application to determine a location of the user's email mailbox; and

accessing the user's email mailbox to retrieve voice mail information for the user.

227. The method of claim 226, wherein the voice mail user attributes are stored in a custom attribute.

15 228. An integrated messaging system, comprising:

a messaging and communication server (MCS) coupled to multiple networks of different types;

20 an interface module coupling the MCS to a first type of network, wherein the first type of network includes a groupware application and a directory service, wherein the MCS performs messaging of a second type, including storing and accessing information particular to the second type in the directory service and wherein the messaging of the second type is independent of the first type of network; and

an event listener that detects messaging events of the second type and informs the MCS so that a user accessing the MCS through one of the multiple networks has access to a current list of messages of the second type.

229. The system of claim 228, wherein the first type of network is an enterprise network including email messaging, and wherein the second type of messaging is voice mail messaging.

25 230. The system of claim 229, wherein the current list of messages comprises a voice mail message list, and wherein the information particular to messaging of the second type includes voice mail messages.

231. The system of claim 228, wherein informing the MCS includes sending updated information of the list of messages upon detecting messaging events of the second type.

30 232. The system of claim 228, wherein informing the MCS includes sending an updated list of messages upon detecting messaging events of the second type.

233. A communication method, comprising:

performing voice applications within an enterprise network, including telephony applications;

coupling to a plurality of network types to perform the voice applications, including the enterprise network, and a conventional telephone network;

using a groupware application of the enterprise network to store and access information specific to the voice applications, including voice messages and voice message information; and

5 providing a user with access to telephony services, including access to voice messages and voice message information from a plurality of networks of different types external to the enterprise network using the groupware application, wherein providing comprises fetching current voice message information and voice mail user information from the groupware application concurrent with the user logging into the voice application.

234. The method of claim 233, wherein concurrent comprises after receiving a user mailbox number and password and before the completion of password authentication.

10 235. The method of claim 233, wherein concurrent comprises after receiving a user email mailbox number and password and before the completion of password authentication.

236. The method of claim 233, wherein providing comprises receiving a user email mailbox number and using the user email mailbox number to access a user voice mailbox on the groupware application.

15 237. The method of claim 233, wherein current voice message information and voice mail user information comprises:  
a list of user voice mails (VMLIST); and  
transient user information, including calendar appointment information.

238. The method of claim 233, wherein the fetched information is cached on a messaging and communication server (MCS).

20 239. The method of claim 233, wherein providing comprises deriving a user email mailbox number and using the user email mailbox number to access a user voice mailbox on the groupware application.

240. The method of claim 239, wherein deriving comprises deriving from a user ID.

241. The method of claim 239, wherein deriving comprises deriving from a user's spoken name.

25 242. The method of claim 233, wherein providing further comprises using an event listener to detect events that affect the voice message information, and transferring updated voice message information and voice mail user information to a MCS that provides the user access to the voice applications and voice messaging information through the plurality of networks.

243. The method of claim 237, wherein providing further comprises receiving a user selection of a voice mail from the VMLIST and performing an action on the selection; and fetching a next voice mail in the VMLIST concurrent with performing the action.

30 244. The method of claim 233, wherein providing further comprises periodically polling the groupware application for updates to the voice message information and voice mail user information.

245. The method of claim 233, wherein the user accesses the telephony services from a plurality of networks of different types including:  
access via an enterprise email application;

access via an enterprise email application web access application;  
access via a private branch exchange ("PBX");  
access via a public network; and  
access via the Internet.

5           246.    An integrated messaging system, comprising:  
server means coupled to multiple networks of different types;  
interface means coupling the server means to a first type of network, wherein the first type of network  
includes a groupware application and a directory service, wherein the server means performs messaging of a second  
type, including storing and accessing information particular to the second type in the directory service and wherein  
10   the messaging of the second type is independent of the first type of network; and  
an event listener that detects messaging events of the second type and informs the server means so that a  
user accessing the server means through one of the multiple networks has access to a current user information,  
including,  
messaging of the second type, including voice messaging; and  
15           transient user information, including user calendar information.

247.    The system of claim 246, wherein the interface means further includes:  
means for turning off a voice message waiting indicator on a user device in response to user accessing a  
voice message from any type of network;  
means for turning on a voice message waiting indicator on a user device in response to a caller leaving a  
20   voice message for the user from any type of network; and  
means for sending a short message service (SMS) message on a user device in response to detection of an  
event.

248.    A communication method, comprising:  
performing voice applications within an enterprise network, including telephony applications;  
25    coupling to a plurality of network types to perform the voice applications, including the enterprise network,  
and a conventional telephone network;  
using a groupware application of the enterprise network to store and access information specific to the  
voice applications, including voice messages and voice message information; and  
providing a user with access to telephony services, including access to voice messages and voice message  
30    information from a plurality of networks of different types external to the enterprise network using the groupware  
application, wherein providing comprises, concurrent with the user logging into the voice application, preparing  
current voice message information and voice mail user information for fetching from the groupware application.

249.    The method of claim 248, wherein concurrent comprises after receiving a user mailbox number  
and password and before the completion of password authentication.

35           250.    The method of claim 248, wherein providing comprises receiving a user email mailbox number  
and using the user email mailbox number to access a user voice mailbox on the groupware application.

251.    The method of claim 248, further comprising:  
fetching the prepared information; and

~~"caching the fetched information on a"~~ messaging and communication server (MCS).

252. The method of claim 248, wherein providing comprises deriving a user email mailbox number and using the user email mailbox number to access a user voice mailbox on the groupware application.

253. The method of claim 251, wherein deriving comprises deriving from a user ID.

5 254. The method of claim 251, wherein deriving comprises deriving from a user's spoken name.

255. The method of claim 248 wherein providing further comprises using an event listener to detect events that affect the voice message information, and transferring updated voice message information and voice mail user information to a MCS that provides the user access to the voice applications and voice messaging information through the plurality of networks.

10 256. The method of claim 248, wherein providing further comprises periodically polling the groupware application for updates to the voice message information and voice mail user information.

257. A method for processing voice messages, the method comprising:  
receiving a voice message from a call;  
receiving a request from the caller to make the voice message private;  
15 converting the voice message to an e-mail message;  
protecting the e-mail message in a protection scheme recognized by a particular e-mail system;

and

sending the e-mail through the e-mail system.

20 258. The method of claim 257, including:  
receiving a user command;  
if the user command is authorized under the protection scheme, then executing the command; and  
if the user command is not authorized under the protection scheme, then not executing the  
command.

25 259. The method of claim 257, wherein the protection scheme prevents the recipient of the message from forwarding the message.

260. The method of claim 257, wherein the protection scheme prevents forwarding of the message outside of a network associated with the recipient.

261. The method of claim 257, wherein the protection scheme prevents forwarding of the message outside of a network associated with the recipient's enterprise.

30 262. The method of claim 257, wherein the protection scheme prevents actions depending on the class of service (COS) of the recipient.

263. The method of claim 262, wherein the COS is stored in a messaging and collaboration server that stores user information not directly associated with voice mail.

264. The method of claim 257, wherein the protection scheme prevents actions depending on the group policy (GP) associated with the recipient.

265. The method of claim 257, wherein the protection scheme prevents the recipient from accessing the message more than a limited number of times.

5 266. The method of claim 263, wherein number of times is only one time.

267. A method for processing voice messages, the method comprising:  
receiving a voice message from a call;  
receiving a request from the caller to make the voice message private;  
converting the voice message to an e-mail message;  
10 protecting and encrypting the e-mail message in a rights management scheme recognized by a particular e-mail system; and  
sending the e-mail through the e-mail system.

268. The method of claim 267, including packaging access control, rights and the voice message into structures defined by the e-mail system and the rights management scheme.

15 269. The method of claim 267, including  
delivering the e-mail to a recipient; and  
taking an action requested by the recipient only if the action is permitted under the rights management scheme for the particular e-mail.

20 270. A voice mail system comprising:  
logic that processes a voice message from a call;  
logic to processes a request from the caller to make the voice message private;  
logic that converts the voice message to an e-mail message;  
logic that protects the e-mail message in a protection scheme recognized by a particular e-mail  
system; and  
25 logic that sends the e-mail through the e-mail system.

271. A system comprising:  
a voice mail system; and  
an email system;  
wherein the voice mail system includes,  
30 logic that processes a voice message from a call;  
logic to processes a request from the caller to make the voice message private;  
logic that converts the voice message to an e-mail message;  
logic that protects the e-mail message in a protection scheme recognized by a particular  
e-mail system; and  
35 logic that sends the e-mail through the e-mail system.

272. The system of claim 271, wherein the protection scheme comprises a rights management system.



273. The system of claim 271, wherein the protection scheme is included within the email system.

274. The method of claim 257, wherein the protection scheme prevents the recipient of the message from forwarding the message.

275. A system comprising:

5 a network;

a set of devices coupled to the network, the set of devices including personal computers;

a voice mail system; and

an email system coupled to the network;

wherein the voice mail system is coupled to the email system and includes,

10 logic that processes a voice message from a call;

logic to processes a request from the caller to make the voice message private;

logic that converts the voice message to an e-mail message;

logic that protects the e-mail message in a protection scheme recognized by a particular  
e-mail system; and

15 logic that sends the e-mail through the e-mail system.

276. A message comprising:

message headers including a header with an identification of the recipient of the message;

audio information including a voice message to the recipient; and

information corresponding to rights that the recipient will have to act upon the message;

20 wherein the message headers, audio information and information corresponding to the rights are packaged within structures defined by a protection scheme; and

wherein at least the audio information is encrypted.

277. The message of claim 276, wherein the protection scheme comprises a rights management system.

278. The message of claim 276, wherein the protection scheme prevents the recipient of the message  
25 from forwarding the message.

279. The message of claim 276, wherein the protection scheme prevents forwarding of the message outside of a network associated with the recipient.

280. The message of claim 276, wherein the protection scheme prevents actions depending on the class of service (COS) of the recipient.

30 281. The message of claim 276, wherein the COS is stored in a messaging and collaboration server that stores and forwards messages including messages not including voice messages.

282. A method of handling voicemail messages in a distributed environment including a local voicemail system and a remote voicemail system, the method comprising:

in the local voicemail system, storing an email containing formatting and data adapted for

35 particular features of the local voicemail system;

encapsulating the formatting and data such that the formatting and data may be included in an email to be sent over a public network;

packaging encapsulated formatting and data into an email;

5 sending the email with the encapsulated formatting and data to the remote voicemail system, the local and remote voicemail systems separated by the public network and in communication by way of an email communication system, and the local voicemail system having particular features not generally part of the email communication system and the remote voicemail system having corresponding features;

extracting the formatting and data adapted for features of the voicemail system from a received email on the remote voicemail system;

10 providing an email containing the extracted formatting and data;

displaying the email containing the extracted formatting and data;

taking actions on the email based on the particular features of the voicemail system and the formatting and data in the email.

15 283. The method of claim 282, wherein the formatting and data are stored in Messaging Application Programming Interface (MAPI) format.

284. The method of claim 282, wherein the formatting and data are encapsulated in Transport Neutral Encoded Format (TNEF).

285. The method of claim 282, wherein the email with encapsulated formatting and data is sent to the remote voicemail system using Simple Mail Transfer Protocol (SMTP).

20 286. The method of claim 282, wherein the actions include actions selected from among a set including playing a voice message, replying to a voice message, forwarding a voice message, and calling the sender of a voice mail message.

25 287. The method of claim 282, presenting a form-based interface on the device of a recipient of the email, wherein the interface provides options to take the actions on the email based on the particular features of the voicemail system.

288. The method of claim 287, wherein the interface includes inputs that allow the recipient to take actions via a server separate from the recipient's device on which the recipient views the email.

289. The method of claim 287, wherein the interface includes buttons for actions particular to voice mail messages.

30 290. The method of claim 287, wherein the form includes a web browser, wherein the web browser communicates with a corresponding browser control of the server.

291. The method of claim 287, wherein the server comprises a messaging and communication server (MCS) of the remote voicemail system and wherein the email is provided to the recipient via a messaging and communication server (MSERV).

292. The method of claim 287, wherein the form complies with format for communication between Outlook and Exchange.

293. The method of claim 282, wherein the features provide access to integrated messaging functions of the voicemail system.

5 294. The method of claim 282, wherein the features provide access to integrated messaging functions of the voicemail system a requirement to install or run a dedicated client application on a client device of the recipient of the email.

10 295. A method of handling voicemail messages in a distributed environment including a local voicemail system and a remote voicemail system, the method comprising:  
in the local voicemail system, storing an email containing formatting and data adapted for particular features of the local voicemail system;  
encapsulating the formatting and data such that the formatting and data may be included in an email to be sent over a public network, wherein the formatting and data are capable of being extracted on the remote voicemail system to provide an email on the remote voicemail system upon which actions can be taken based on the  
15 particular features of the voicemail system and the formatting and data in the email;  
packaging encapsulated formatting and data into an email; and  
sending the email with the encapsulated formatting and data to the remote voicemail system, the local and remote voicemail systems separated by the public network and in communication by way of an email communication system, and the local voicemail system having particular features not generally part of the email  
20 communication system and the remote voicemail system having corresponding features.

296. The method of claim 295, wherein the formatting and data are stored in Messaging Application Programming Interface (MAPI) format, the formatting and data are encapsulated in Transport Neutral Encoded Format (TNEF), and the email with encapsulated formatting and data is sent to the remote voicemail system using Simple Mail Transfer Protocol (SMTP).

25 297. A method of handling voicemail messages in a distributed environment including a local voicemail system and a remote voicemail system, the local and remote voicemail systems separated by a public network and in communication by way of an email system, the local voicemail system having particular features not generally part of the email communication system, the method comprising:  
in the local voicemail system, storing an email containing formatting and data adapted for the  
30 particular features of the voicemail system;  
determining whether the remote voicemail system possesses the particular features not generally part of the email communication system corresponding to the local voicemail system features;  
if the remote voicemail system possesses the particular features, routing an outgoing email containing formatting and data adapted for the particular features of the voicemail system; and  
35 sending an outgoing email.

298. The method of claim 297, including, if the remote system possesses the particular features:  
encapsulating the formatting and data into an email such that the formatting and data may be included in an email to be sent over the public network;

5 sending the email with the encapsulated formatting and data to the remote voicemail system;  
extracting the formatting and data adapted for features of the voicemail system from a received  
email on the remote voicemail system;  
providing an email containing the formatting and data adapted for features of the voicemail  
5 system; and  
taking actions on the email based on the particular features of the voicemail system and the  
formatting and data in the email.

10 299. The method of claim 297, including if the remote voicemail system does not possess the particular  
features not generally part of the email communication system corresponding to the local voicemail system features:  
sending a simplified form of the email with the included voicemail audio data to the remote  
voicemail system.

300. The method of claim 297, including receiving information from a DNS lookup to determine  
whether the remote voicemail system possesses the particular features not generally part of the email  
communication system corresponding to the local voicemail system features.

15 301. The method of claim 297, including checking information in a cache in the local voicemail system  
to determine whether the remote voicemail system possesses the particular features not generally part of the email  
communication system corresponding to the local voicemail system features.

20 302. The method of claim 297, including receiving information from web services to determine  
whether the remote voicemail system possesses the particular features not generally part of the email  
communication system corresponding to the local voicemail system features.

25 303. A distributed voicemail system including:  
a local voicemail system;  
a remote voicemail system, the local and remote voicemail systems separated by a public network  
and in communication by way of an email system, the local voicemail system having particular features not  
generally part of the email communication system, and the remote voicemail system having corresponding features;  
wherein the local voicemail system includes:  
resources that store an email containing formatting and data adapted for the particular  
features of the local voicemail system;  
resources that encapsulate the formatting and data such that the formatting and data may  
30 be included in an email to be sent over the public network;  
resources that package the encapsulated formatting and data into an email; and  
resources that send the email with the encapsulated formatting and data to the remote  
system; and  
wherein the remote voicemail system includes:  
35 resources that extract the formatting and data adapted for features of the voicemail  
system from a received email;  
resources that provide an email containing the formatting and data adapted for features of  
the voicemail system;

resources that display the email containing the formatting and data adapted for features of the voicemail system; and

resources that take actions on the email based on the particular features of the voicemail system and the formatting and data in the email.

5           304.    A first voicemail system, the first voice mail system being capable of communicating, by way of an email system, with a second voicemail system separated from the first voicemail system by way of a public network, the first voicemail system comprising:

                  particular features not generally part of the email communication system, the features corresponding to features of the second voicemail;

10           resources that store an email containing formatting and data adapted for the particular features of the first voicemail system;

                  resources that encapsulate the formatting and data such that the formatting and data may be included in an email to be sent over the public network;

15           resources that package the encapsulated formatting and data into an email from which the formatting and data can be extracted at the second voicemail system so that the second email system can take actions on the email based on the particular features and the formatting and data in the email; and

                  resources that send the email with the encapsulated formatting and data to the second system.

305.    The first voicemail system of claim 304, wherein the formatting and data are stored in Messaging Application Programming Interface (MAPI) format.

20           306.    The first voicemail system of claim 304, wherein the formatting and data are encapsulated in Transport Neutral Encoded Format (TNEF).

307.    The first voicemail system of claim 304, wherein the email with encapsulated formatting and data is sent to the remote voicemail system using Simple Mail Transfer Protocol (SMTP).

25           308.    The first voicemail system of claim 304, wherein the actions include actions selected from among a set including playing a voice message, replaying to a voice message, forwarding a voice message, and calling the sender of a voice mail message.

309.    The first voicemail system of claim 304, including resources that present a form-based interface on the device of a recipient of emails containing voice messages, wherein the interface provides options to take the actions on the email based on the particular features of the first voicemail system.

30           310.    The first voicemail system of claim 309, wherein the form includes a web browser, wherein the web browser communicates with a corresponding browser control of the server.

311.    A method of handling voicemail messages in a distributed environment, comprising:

                  receiving a rich format voicemail email;

                  encapsulating rich format data from the rich format voicemail email;

35           packaging the encapsulated rich format data into an email;

                  sending the email with the encapsulated rich format data to a remote system;

extracting the rich format data from a received email on the remote system;  
providing a rich format voicemail email from the received email and the extracted rich format  
data; and  
displaying the rich format voicemail email to a user of the remote system.

5           312.    The method of claim 311, wherein the rich format contains Messaging Application Programming Interface (MAPI) formatting, the rich format data is encapsulated in Transport Neutral Encoded Format (TNEF), and the email with the encapsulated rich format data is sent to the remote system using Simple Mail Transfer Protocol (SMTP).

10           313.    The method of claim 311, wherein the email with encapsulated rich format data is sent to the remote system across the Internet.

15           314.    An electronic message comprising:  
              an audio stream including a voice message;  
              a header designating a recipient of the message; and  
              formatting and data adapted for particular features of a local voicemail system, the formatting and data encapsulated so that the formatting and data may be included in an email to be sent over a public network and extracted at a remote voicemail system at which actions can be taken based on the particular features of the voicemail system and the formatting and data in the email.

20           315.    The message of claim 314, wherein the formatting and data are stored in Messaging Application Programming Interface (MAPI) format, and the formatting and data are encapsulated in Transport Neutral Encoded Format (TNEF).

316.    The message of claim 314, wherein the actions include actions selected from among a set including playing a voice message, replaying to a voice message, forwarding a voice message, and calling the sender of a voice mail message.

25           317.    The message of claim 314, including information corresponding to a form for a form-based interface, wherein the interface provides options to take the actions on the email based on the particular features of the voicemail system.

318.    The message of claim 317, wherein the form includes a web browser, wherein the web browser communicates with a corresponding browser control of the server.

30           319.    The message of claim 317, wherein the form complies with a format for communication between Outlook and Exchange

320.    The message of claim 317, wherein the message complies with a format for communication between Outlook and Exchange.

35           321.    A method for providing diagnostics, the method comprising:  
              receiving identification of diagnostic compiled code and diagnostic source code;  
              compiling the diagnostic source code on a target system on which the diagnostics are to be performed;

gathering information about diagnostic methods included within the diagnostic compiled code and about diagnostic methods included within diagnostic compiled code resulting from the compiling;  
based on the gathered information, running the diagnostic methods on the target system; and  
gathering results of the diagnostic methods.

- 5           322.    The method of claim 321, including using reflection to gather information about the diagnostic methods.
323.    The method of claim 321, wherein the compiled code comprises .NET framework assembly code.
324.    The method of claim 321, wherein the target system comprises a voicemail system.
325.    The method of claim 321, wherein the target system comprises a messaging and collaboration  
10 server.
326.    The method of claim 321, wherein the diagnostic methods include a diagnostic of a greeting in a voicemail system.
327.    The method of claim 321, wherein the diagnostic methods include gathering version information about the installed target system assemblies.
- 15           328.    The method of claim 321, wherein the diagnostic methods include verifying that target system service account information is present in a registry.
329.    The method of claim 321, wherein the diagnostic methods include checking ability to impersonate an account, including testing a password for the account.
330.    The method of claim 321, wherein the diagnostic methods include checking correctness of path  
20 information in the registry for software of the target system.
331.    The method of claim 321, wherein the diagnostic methods include checking correctness of installation of web software on the target system.
332.    The method of claim 321, wherein the diagnostic methods include checking for presence of source configurable rules for user extensions on the target system.
- 25           333.    The method of claim 321, wherein the diagnostic methods include verifying permissions of the target system service account for the target system web service software to run.
334.    The method of claim 321, wherein the diagnostic methods include verifying permissions of the target system service account for the target system web application software to run.
335.    The method of claim 321, wherein the diagnostic methods include verifying permissions of the  
30 target system service account for the target system email system extension software to run.
336.    The method of claim 321, wherein the diagnostic methods include verifying permissions of the target system service account to access identity key of service account in a registry.

337. The method of claim 321, wherein the diagnostic methods include verifying a pin code.

338. A method for providing diagnostics, the method comprising:

receiving identification of diagnostic code;

gathering information about diagnostic methods included within the diagnostic code;

5 in software running on a target system on which the diagnostic code is to be run, identifying parameters used by code;

receiving values for parameters;

running diagnostic code on the target system using the parameters; and

gathering results of the diagnostic code.

10 339. The method of claim 338, including using reflection to gather information about the diagnostic methods.

340. The method of claim 338, wherein the compiled code comprises .NET framework assembly code.

341. The method of claim 338, wherein the target system comprises a voicemail system.

15 342. The method of claim 338, wherein the target system comprises a messaging and collaboration server.

343. A method for providing diagnostics for a target system, the method comprising:

providing diagnostic framework computer code on the target system;

providing source code of diagnostic modules on the target system;

providing the source code of the diagnostic modules to the framework computer code;

20 compiling the source code of the diagnostic modules; and

running the compiled diagnostic modules on the target system.

344. A diagnostic tool comprising:

logic that receives identification of diagnostic compiled code and diagnostic source code;

logic that causes the diagnostic source code to be compiled;

25 logic that gathers information about diagnostic methods included within the diagnostic compiled code and about diagnostic methods included within the diagnostic compiled code resulting from the compiling;

logic that, based on the gathered information, runs the diagnostic methods; and

logic that gathers results of the diagnostic methods.

345. The diagnostic tool of claim 344, wherein the logic comprises compiled assembly.

30 346. The diagnostic tool of claim 344, wherein the logic comprises software stored on a computer readable medium.

347. An interface module (IM) comprising:

logic to communicate with a messaging communication server (MCS) that processes voice messages;

35 logic to communicate with a messaging and collaboration server (MSERV) that stores voice messages and user information; and



a diagnostic tool comprising:

- 5 logic that receives identification of diagnostic compiled code and diagnostic source code;  
logic that causes the diagnostic source code to be compiled;  
logic that gathers information about diagnostic methods included within the diagnostic compiled  
code and about diagnostic methods included within the diagnostic compiled code resulting from the compiling;  
logic that, based on the gathered information, runs the diagnostic methods; and  
logic that gathers results of the diagnostic methods.

348. The IM of claim 347, wherein the logic comprises software.

349. The IM of claim 347, wherein the logic comprises software stored on a computer readable  
10 medium.

350. A communication system comprising:

a messaging communication server (MCS) that processes voice messages;

a messaging and collaboration server (MSERV) that stores voice messages and user information; and

a diagnostic tool that diagnoses aspects of the communication system, the diagnostic tool comprising:

- 15 logic that receives identification of diagnostic compiled code and diagnostic source code;  
logic that causes the diagnostic source code to be compiled;  
logic that gathers information about diagnostic methods included within the diagnostic compiled  
code and about diagnostic methods included within the diagnostic compiled code resulting from the compiling;  
logic that, based on the gathered information, runs the diagnostic methods; and  
20 logic that gathers results of the diagnostic methods.

351. The system of claim 350, wherein the diagnostic tool is included within the MSERV.

352. A messaging communication server (MCS), comprising:

logic that provides services related to voice messaging including processing of voicemail  
25 messages;

a first storage for configuration information related to at least some of the services;

an interface for connection to a network that includes at least a second storage separate from the  
MCS; and

logic that causes configuration information received from the second storage to be loaded on the  
MCS and stored in the first storage on the MCS.

30 353. The MCS of claim 352, including logic that causes the MCS to automatically request  
configuration information.

354. The MCS of claim 352, including logic that causes the MCS to request configuration information  
in accordance with a time schedule.

35 355. The MCS of claim 352, including logic that causes the MCS to request configuration information  
after occurrence of an event.

356. The MCS of claim 352, wherein the configuration information includes a network address of an interface module coupled to the MCS.

357. The MCS of claim 352, wherein the configuration information includes a network address of the MCS.

5 358. The MCS of claim 352, wherein the configuration information includes telephone numbers.

359. The MCS of claim 352, wherein the configuration information includes voicemail numbers.

360. The MCS of claim 352, wherein the logic includes code.

361. The MCS of claim 352, wherein the first storage comprises volatile memory.

362. The MCS of claim 352, wherein the first storage includes a magnetic drive.

10 363. A method comprising:  
in a messaging communication server (MCS), requesting configuration information for the MCS from a source outside of the MCS;  
receiving the configuration information on the MCS;  
automatically installing the received configuration information on the MCS; and  
15 using the installed configuration information on the MCS to provide services related to voice messaging including processing of voicemail messages.

364. The method of claim 363, wherein the requesting occurs automatically.

365. The method of claim 363, wherein the requesting occurs automatically upon coupling of the MCS into a new environment.

20 366. The method of claim 363, wherein the requesting occurs in accordance with a time schedule.

367. The method of claim 363, wherein the requesting occurs in response to occurrence of an event.

368. The method of claim 367, including polling the environment of the MCS to determine occurrence of the event.

25 369. The method of claim 367, including automatically receiving notification of occurrence of the event.

370. The method of claim 367, including determining occurrence of the event by receipt of a broadcast message.

371. The method of claim 367, wherein the event comprises change of data availability.

30 372. The method of claim 367, wherein the event comprises addition of another MCS in a network that includes the first MCS.

373. The method of claim 367, wherein the event comprises addition of an interface module (IM) operable with the MCS.

5 374. The method of claim 363, including:  
requesting data other than configuration information for the MCS from a source outside of the  
MCS;  
receiving the data on the MCS;  
automatically installing the data on the MCS; and  
using the installed data on the MCS to provide services related to voice messaging including  
processing of voicemail messages.

10 375. The method of claim 374, wherein the data includes voicemail messages.

376. The method of claim 374, wherein the data includes user data.

377. The method of claim 374, wherein the requesting of the data and the configuration information occurs automatically upon coupling of the MCS into a new environment.

15 378. The method of claim 374, wherein the requesting of the data and the configuration information occurs automatically upon detection of an error.

379. A method for setting up a member of an enterprise to be a user of voice applications, the method comprising:

accessing a data in a user object in a directory service of a groupware application, the data including  
telephony-related and voice mail-related data;  
20 using an administrator user interface (UI) to generate telephony-related and voice mail-related data using  
the user object, wherein generating includes,  
performing reflection on at least one compiled file, wherein the file includes methods to generate  
telephony-related and voice mail-related data according to rules;  
compiling a source file, wherein the file includes methods to generate telephony-related and voice  
25 mail-related data according to rules;  
performing reflection on the compiled source file; and  
invoking the compiled file and the compiled source file.

380. The method of claim 379, further comprising:  
collecting the results of invoking the compiled file and the source file;  
30 assembling the collected results in at least one attribute; and  
storing the at least one attribute in the directory service.

381. The method of claim 379, further comprising:  
querying an administrator for a run-time parameter required for a method;  
receiving the run-time parameter; and  
35 running the method using the received parameter.

382. The method of claim 380, wherein the at least one attribute includes one custom attribute that includes all of the results.

383. The method of claim 380, wherein the at least one attribute includes multiple attributes that include one of the results.

5 384. The method of claim 379, wherein the telephony-related and voice mail-related data includes a class of service ("COS") attribute that indicates a class of telephony-related and voice mail-related service for the user.

385. The method of claim 384, wherein the COS indicates  
 MaxNumber of Greetings;  
 10 ExtendedAbsenceGreetingAllowed;  
 ExtendedAbsenceGreetingLength;  
 MaxGreeting Length;  
 ExtendedAbsenceGreetingBlockMsgRecord;  
 GeneralDialoutPrivileges;  
 15 PasscodeAgingEnabled;  
 AccessToSystemDistLists;  
 AllowToSendBroadcastMessage;  
 ActiveMobileNumber; MobilePhoneNumber;  
 ExtendedAbsenceText;  
 20 MailboxLanguage;  
 CallerFirstLanguage;  
 CallerSecondLanguage;  
 CallerThirdLanguage; and  
 AttendantSchedule.

25 386. The method of claim 384, wherein the COS attribute is generated by a method and stored in the directory service.

387. The method of claim 386, wherein generating the COS attribute comprises calculating an effective group policy for the user according to an enterprise group policy schema.

30 388. The method of claim 387, further comprising initially populating the group policy schema with telephony-related and voice mail-related policies.

389. An integrated method for setting up a member of an enterprise to be a user of a voice mail system and a user of an email system, the method comprising:

35 extending a groupware application schema to include telephony-related and voice mail-related permissions, levels of service, and privileges, wherein a policy includes permissions, levels of service, and privileges assigned to one or more users; and

generating an effective policy for the user, including a set of telephony-related and voice mail-related permissions, levels of service, and privileges applicable to the user, according to the groupware application schema.

390. An apparatus for integrating a user setup process in an enterprise, the apparatus comprising:

a messaging communication server (MCS);

an interface module (IM) coupled among the MCS and an enterprise groupware application, including a directory service;

5 a source configurable rules framework (SCR FW) coupled among the groupware application, the MCS, and the IM, wherein the SCR FW is invocable through an administrator user interface (UI) of the groupware application to,

get a user object from the directory service, wherein the user object contains email-related data pertinent to a user;

10 compile a source file, wherein the file includes methods to generate telephony-related and voice mail-related data pertinent to a user according to rules;

perform reflection on the compiled source file; and

invoke the compiled source file.

391. The apparatus of claim 390, wherein the SCR FW is further invocable through the administrator user interface (UI) of the groupware application to:

collect the results of invoking the compiled file and the source file;

assemble the collected results in at least one attribute; and

store the at least one attribute in the directory service.

392. The apparatus of claim 390, wherein the SCR FW is further invocable through the administrator user interface (UI) of the groupware application to:

query an administrator for a run-time parameter required for a method;

receive the run-time parameter; and

run the method using the received parameter.

393. The apparatus of claim 391, wherein the at least one attribute includes one custom attribute that includes all of the results.

394. The apparatus of claim 391, wherein the at least one attribute includes multiple attributes that include one of the results.

395. The apparatus of claim 390, wherein the telephony-related and voice mail-related data includes a class of service ("COS") attribute that indicates a class of telephony-related and voice mail-related service for the user.

396. The apparatus of claim 395, wherein the COS indicates

MaxNumber of Greetings;

ExtendedAbsenceGreetingAllowed;

ExtendedAbsenceGreetingLength;

35 MaxGreeting Length;

ExtendedAbsenceGreetingBlockMsgRecord;

GeneralDialoutPrivileges;

PasscodeAgingEnabled;

"AccessToSystemDistLists;"

AllowToSendBroadcastMessage;  
ActiveMobileNumber; MobilePhoneNumber;  
ExtendedAbsenceText;  
5 MailboxLanguage;  
CallerFirstLanguage;  
CallerSecondLanguage;  
CallerThirdLanguage; and  
AttendantSchedule.

10 397. The apparatus of claim 395, wherein the COS attribute is generated by a method and stored in the directory service.

398. The apparatus of claim 397, wherein generating the COS attribute comprises calculating an effective group policy for the user according to an enterprise group policy schema.

15 399. The apparatus of claim 398, further comprising initially populating the group policy schema with telephony-related and voice mail-related policies.

400. A messaging communication server (MCS), comprising:  
a first storage for code for services related to voice messaging including processing of voicemail  
messages;  
an interface for connection to a network that includes at least a second storage separate from the  
20 MCS; and  
logic that automatically processes receipt of the code from the second storage, loads the code in  
the first storage on the MCS and causes operation of the MCS to be started using the code.

401. The MCS of claim 400, wherein the first storage includes:  
a first partition for currently running code for services related to voice messaging including  
25 processing of voicemail messages;  
a second partition for newly loaded code for services related to voice messaging including  
processing of voicemail messages; and  
a third partition for code to cause the currently running code to be replaced with the newly loaded  
code.

30 402. The MCS of claim 400, including logic that requests receipt of the code in response to occurrence of an event.

403. The MCS of claim 400, including logic that requests receipt of the code in response to an administrator requesting an upgrade.

35 404. A method comprising:  
in a messaging communication server (MCS), requesting, from a source outside of the MCS, code  
for services related to voice messaging including processing of voicemail messages;

receiving the code on the MCS;  
automatically installing the received code on the MCS; and  
using the installed code on the MCS to provide services related to voice messaging including processing of voicemail messages.

5           405.   The method of claim 404, including:  
              running, from a first partition, previously installed code for services related to voice messaging including processing of voicemail messages;  
              storing the received code into a second partition; and  
10           replacing the previously installed code in the first partition with the received code

10

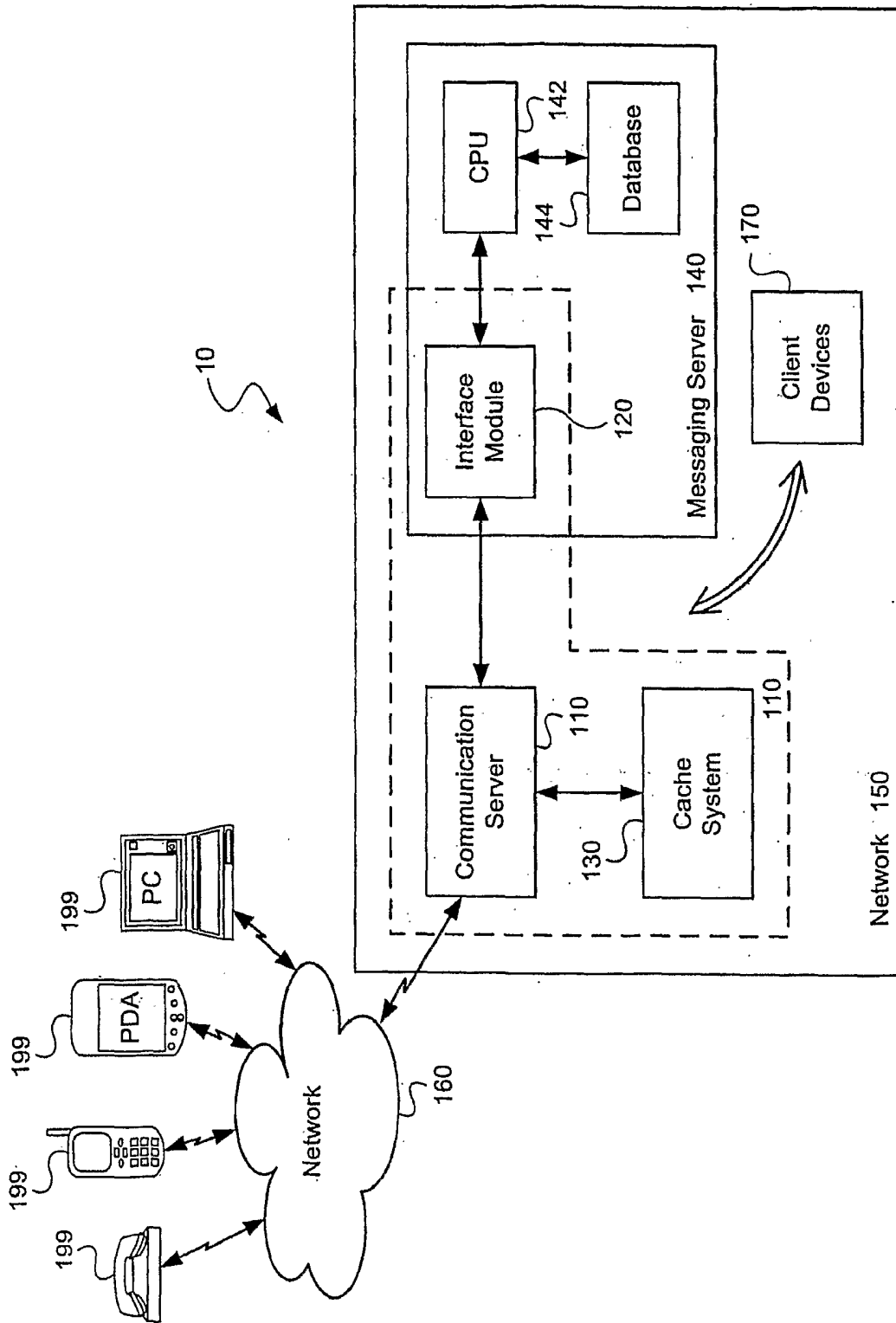


FIG. 1



200

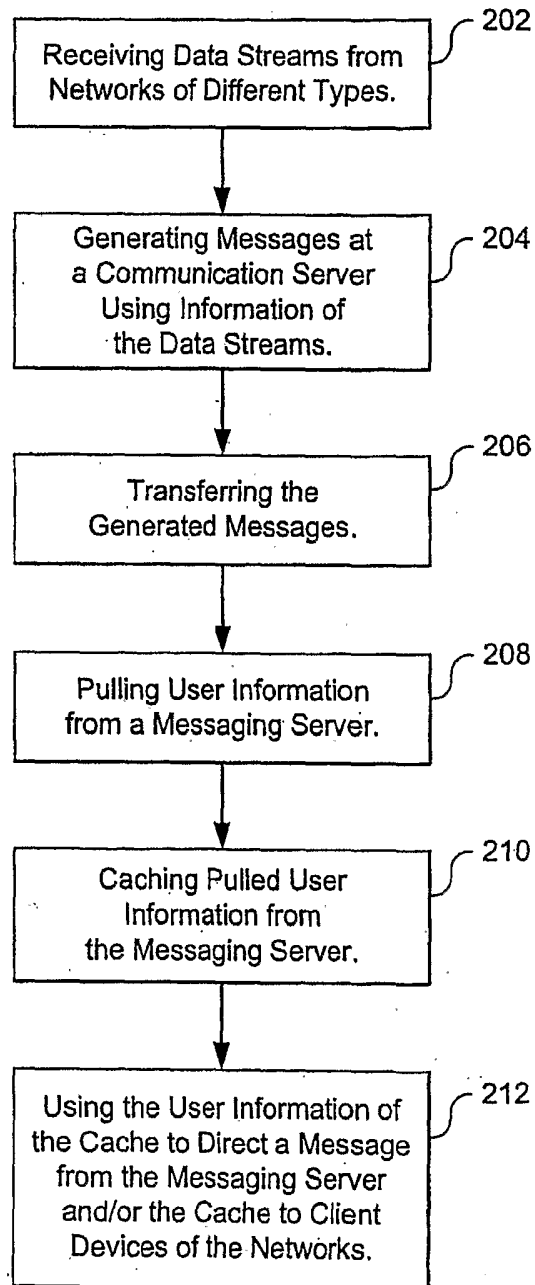


FIG. 2

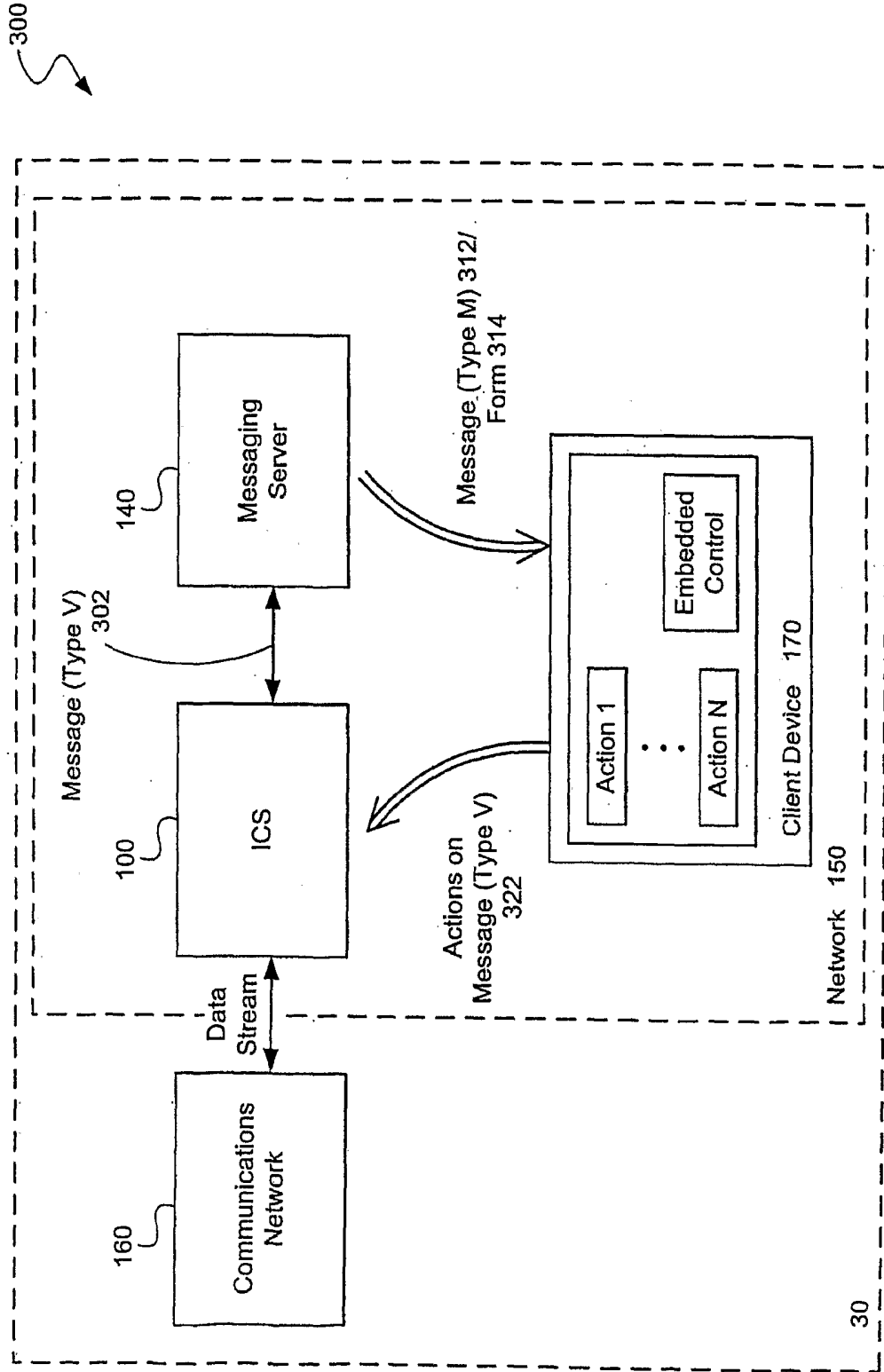


FIG. 3

4/56

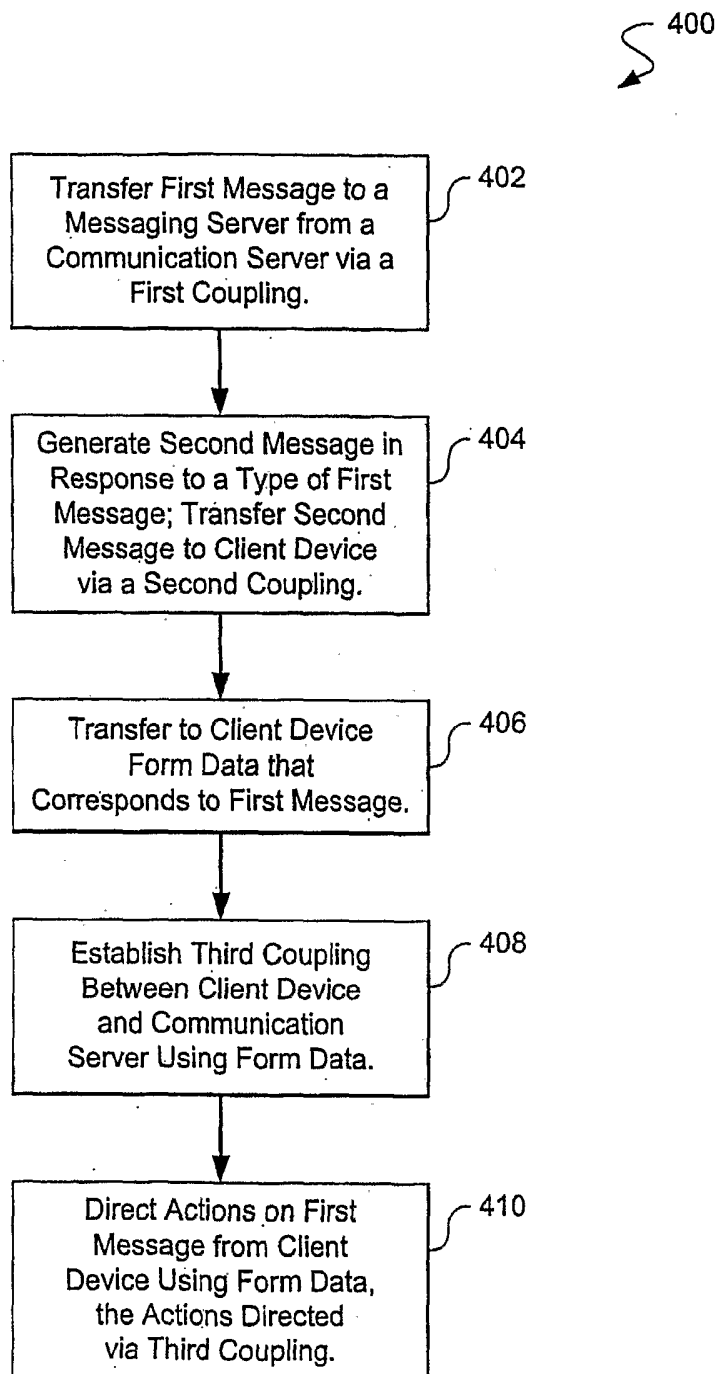


FIG. 4

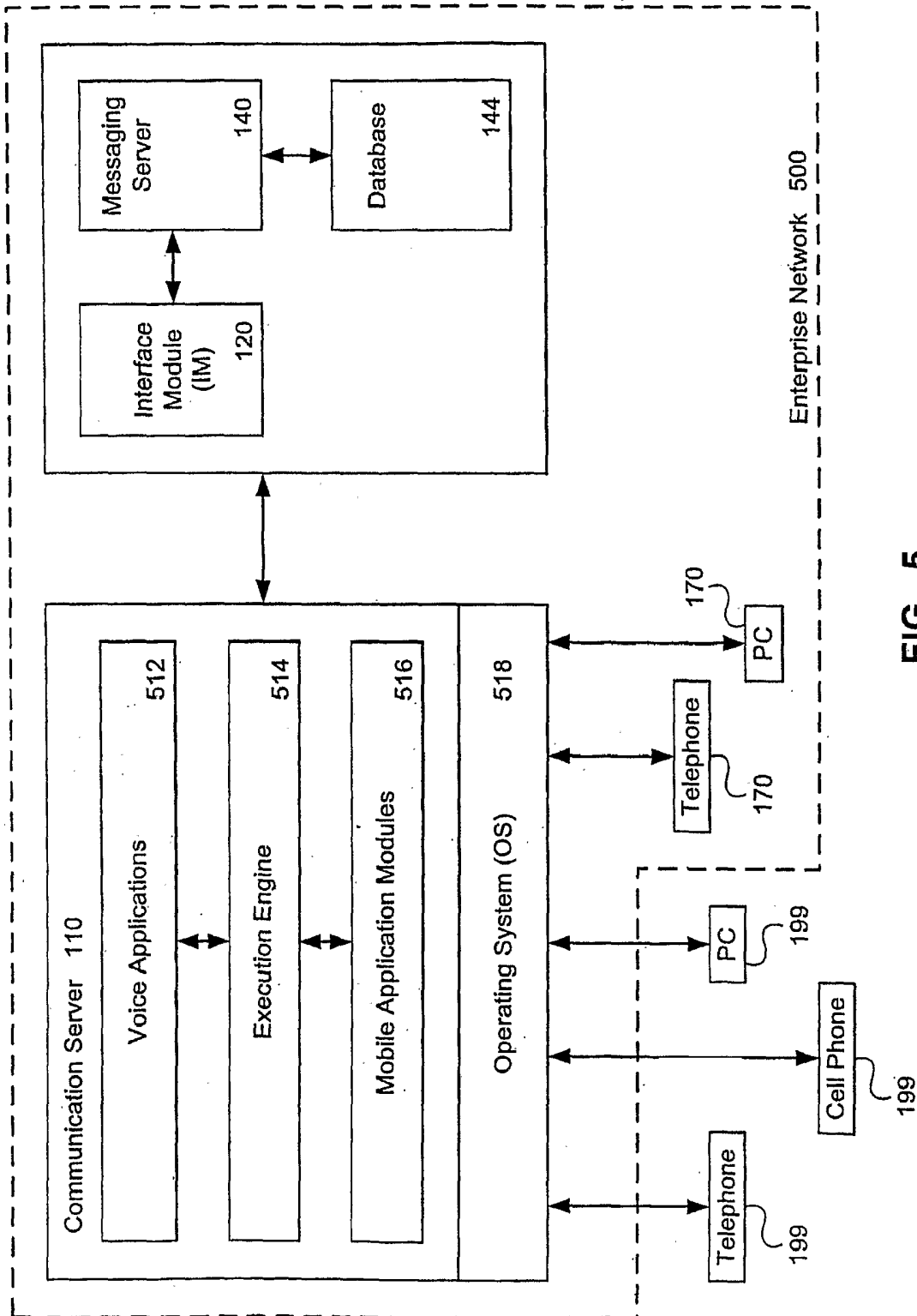


FIG. 5

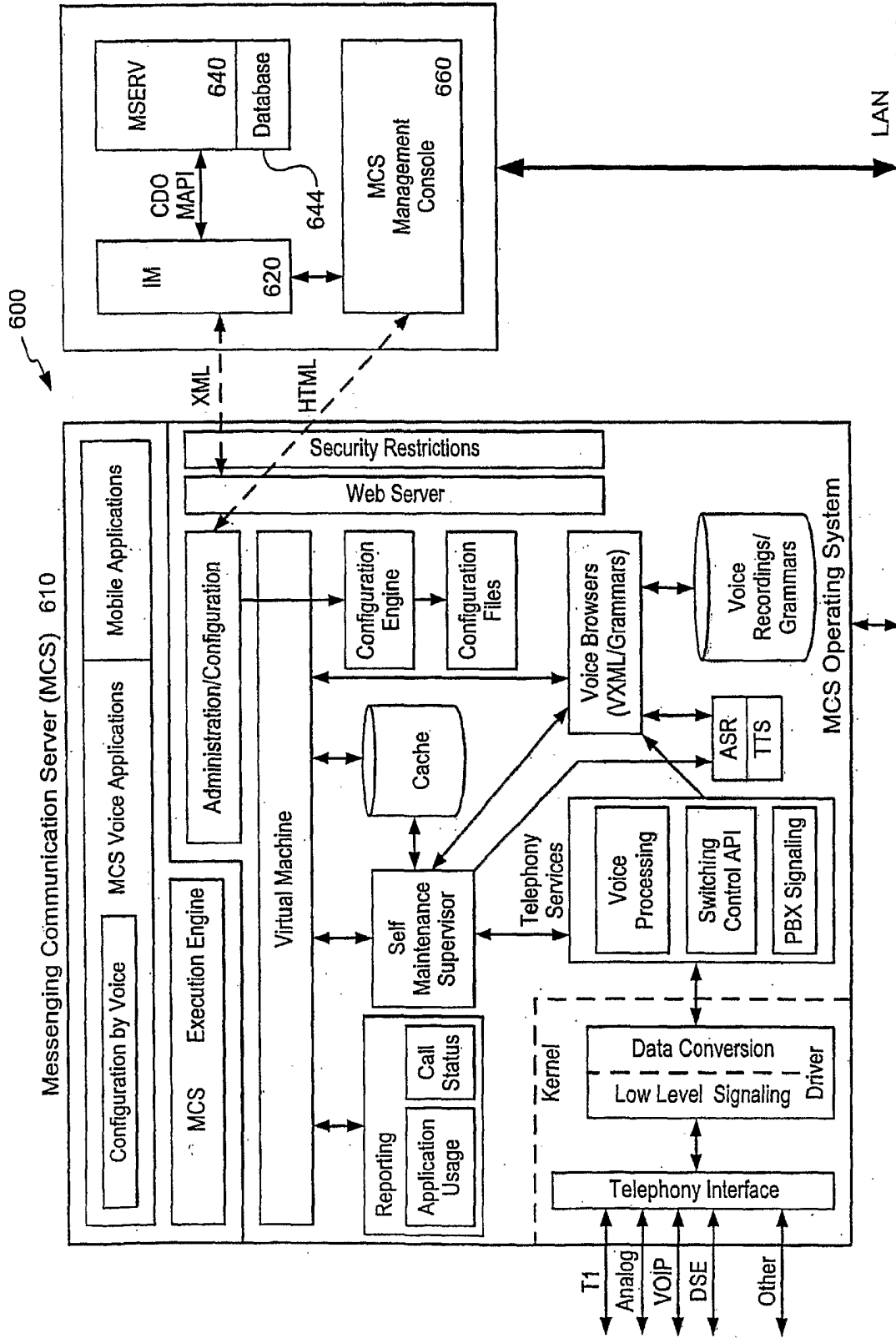


FIG. 6

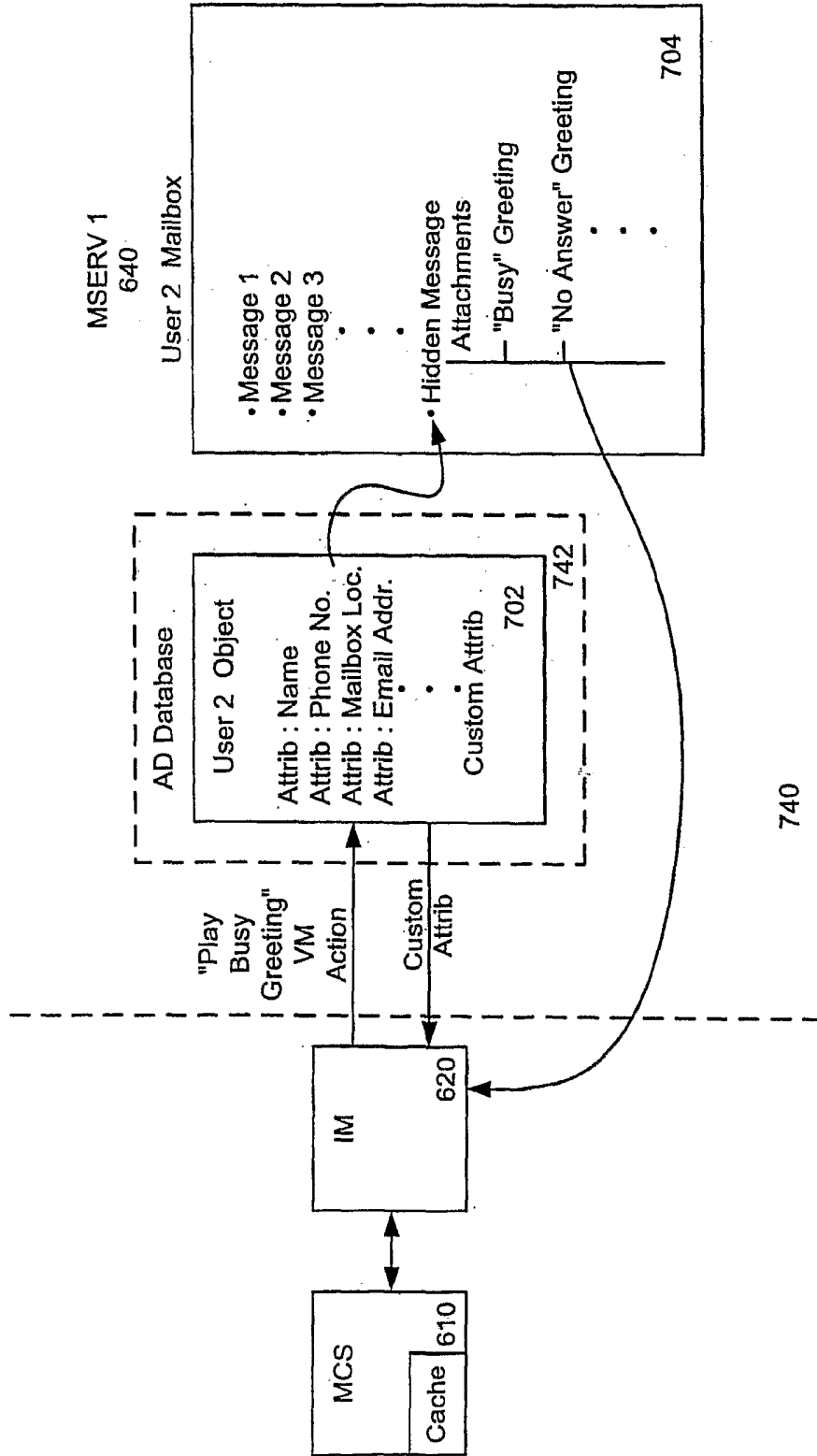


FIG. 7

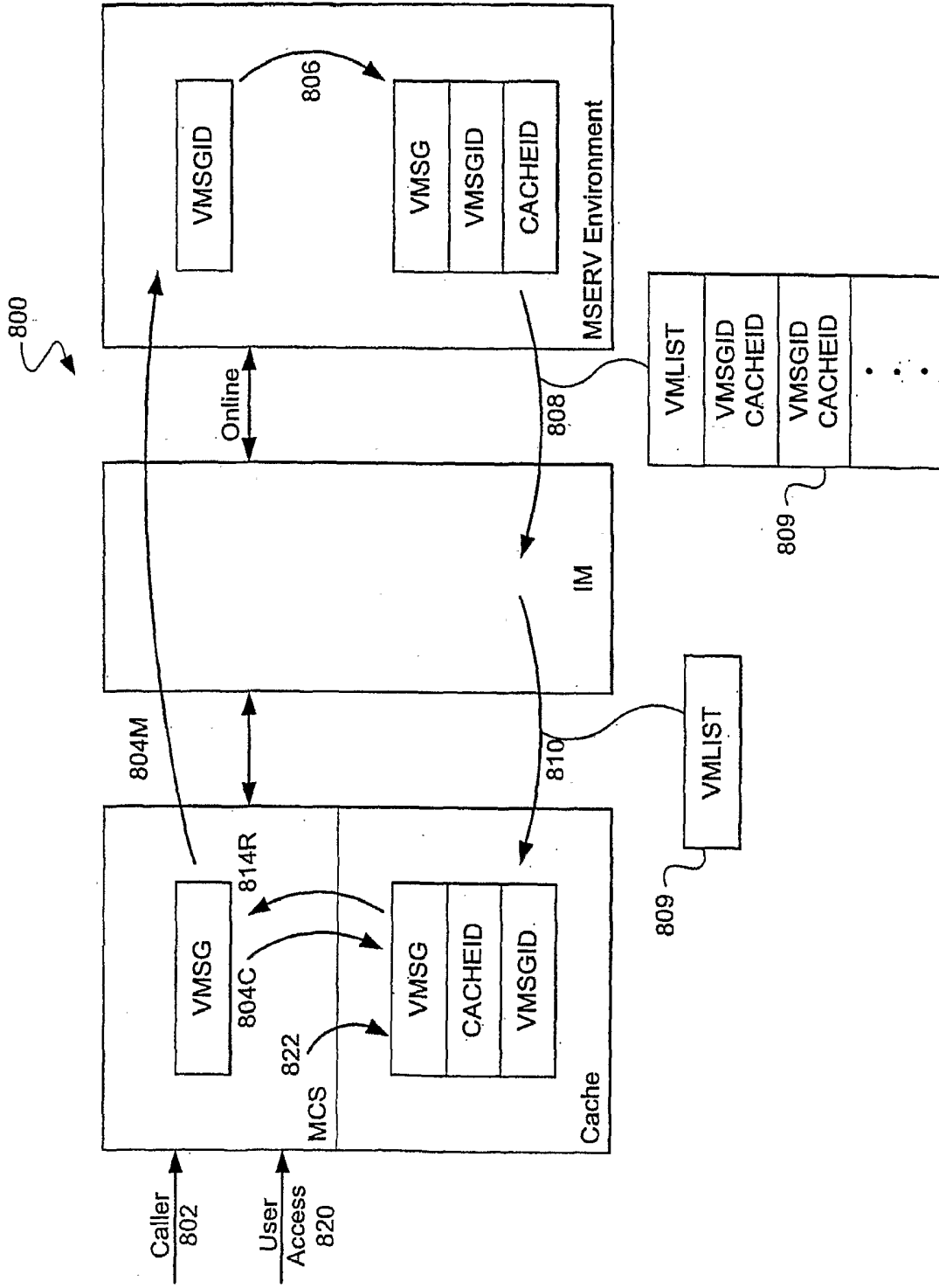


FIG. 8

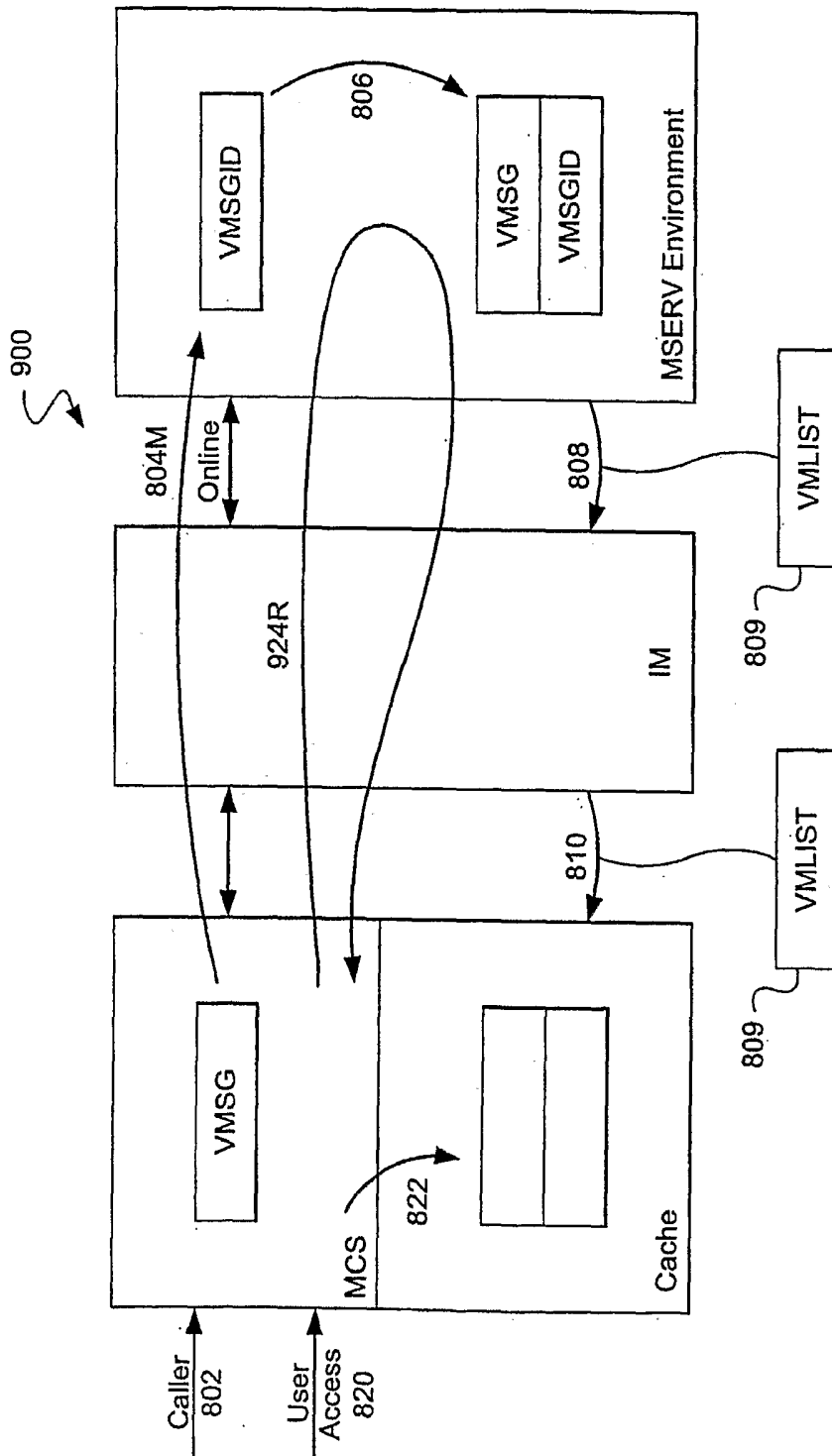


FIG. 9



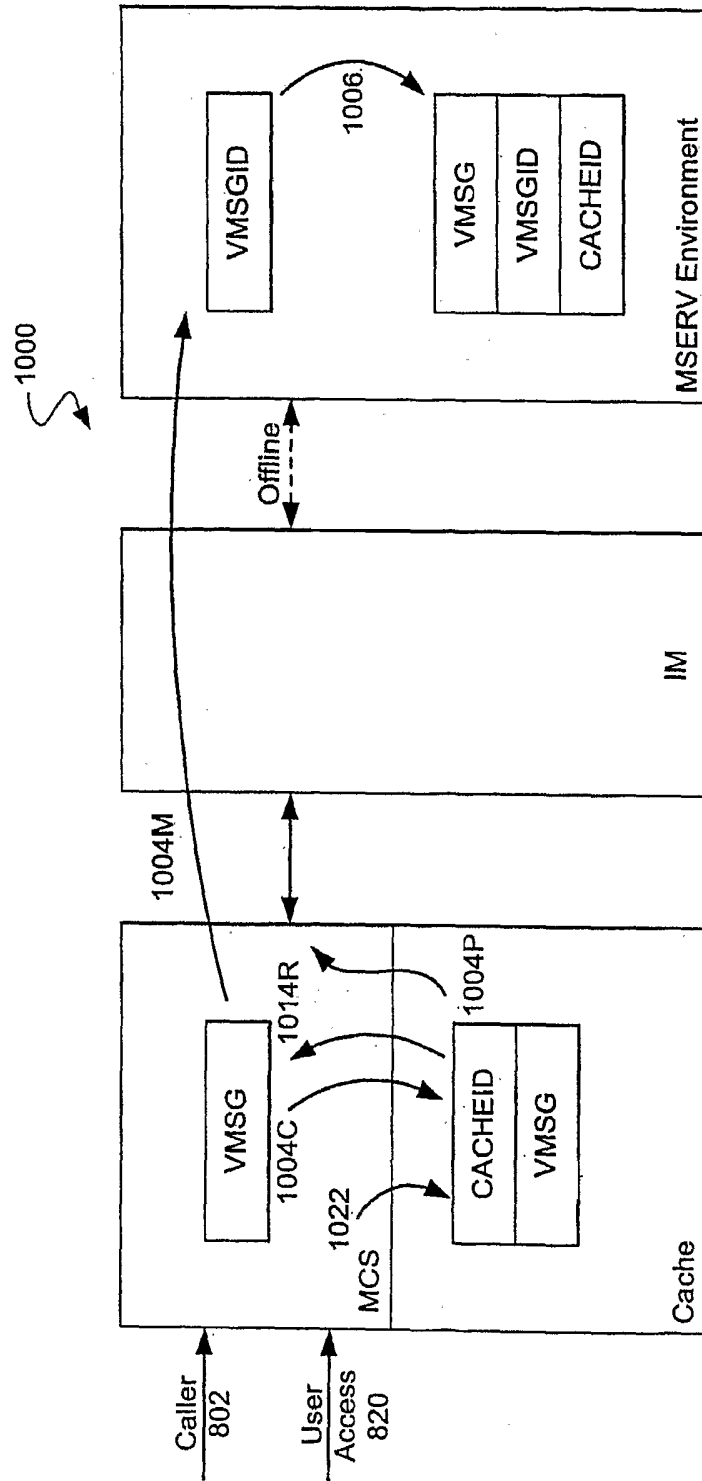


FIG. 10

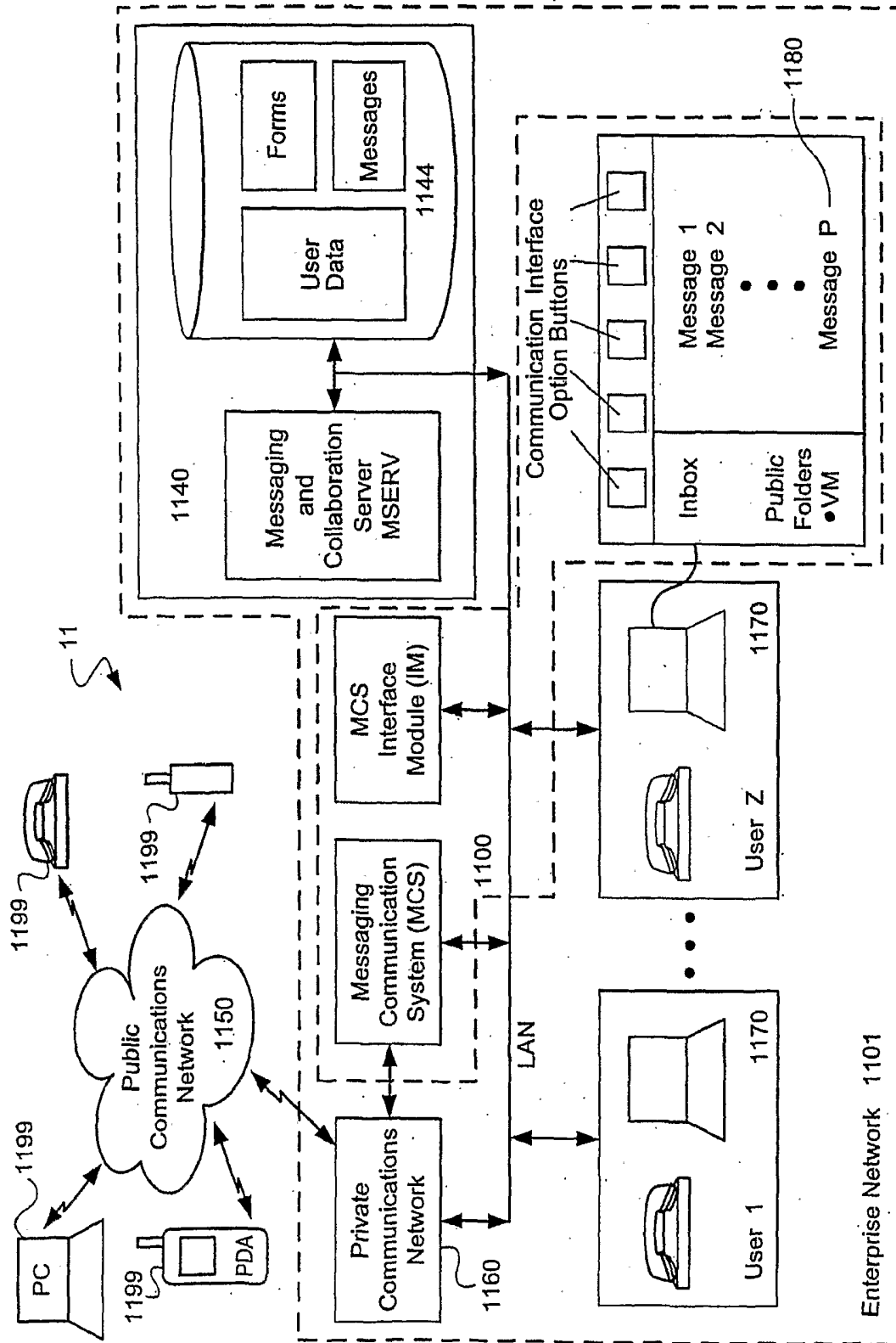


FIG. 11

Enterprise Network 1101

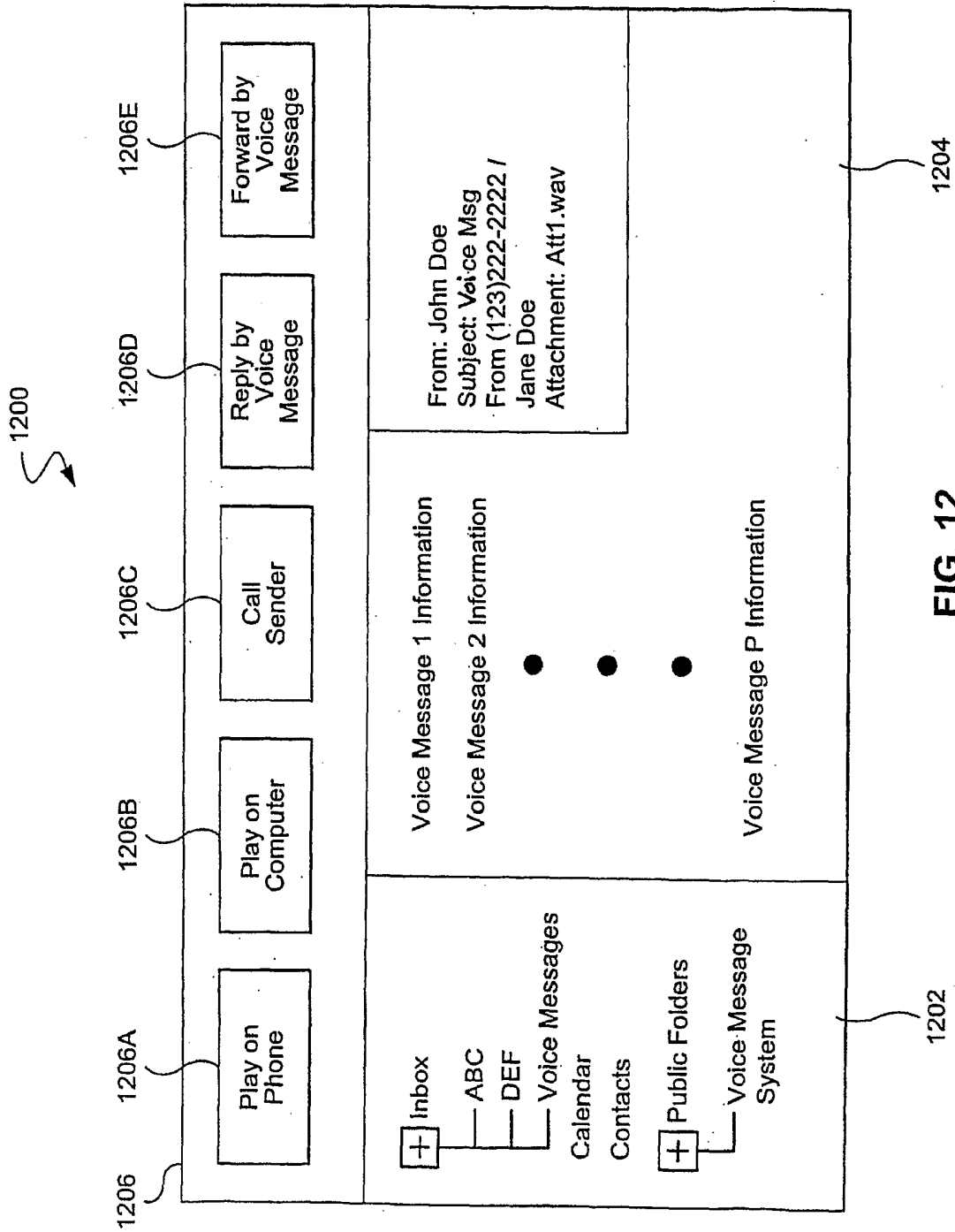


FIG. 12

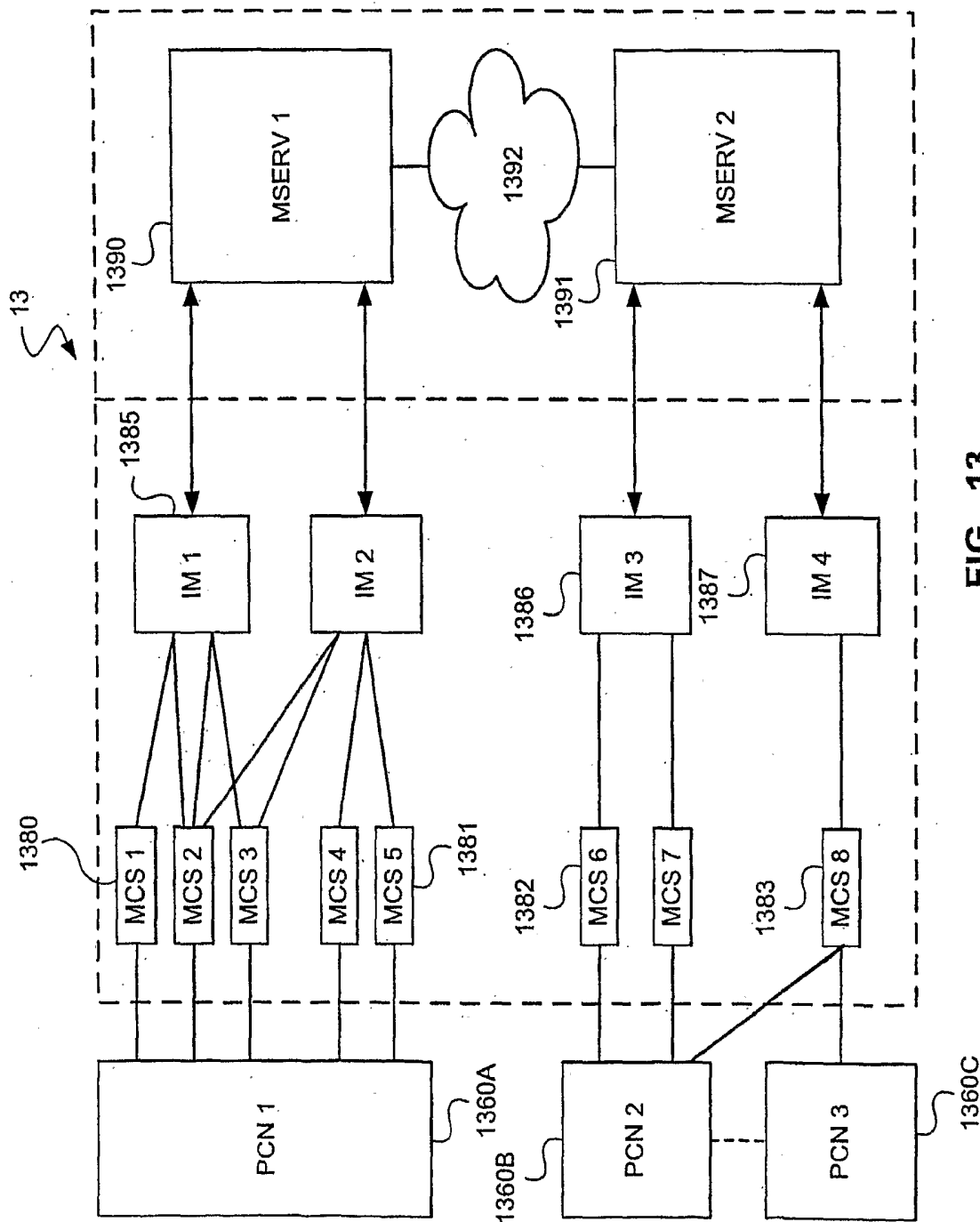


FIG. 13

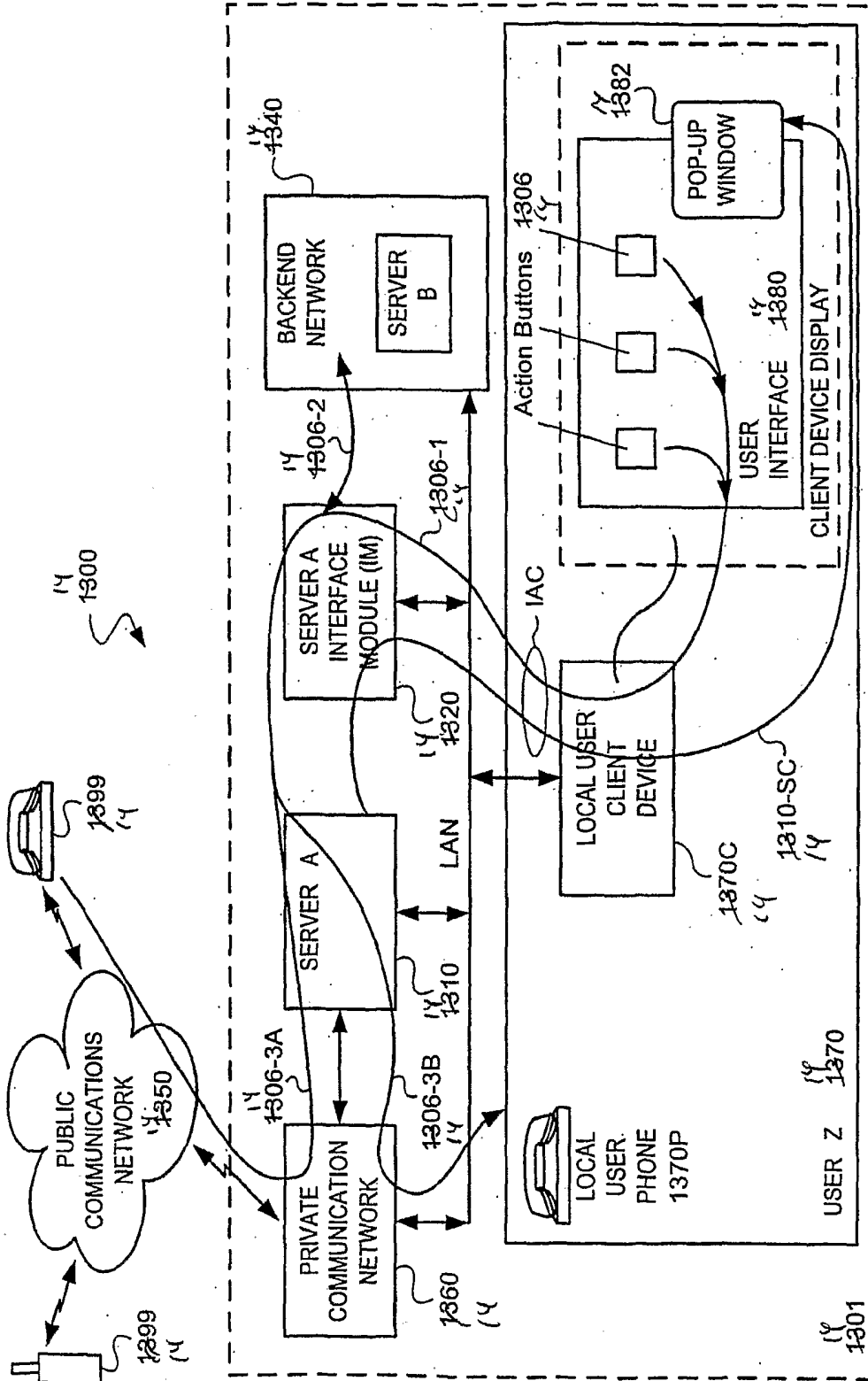


FIG. 13/4

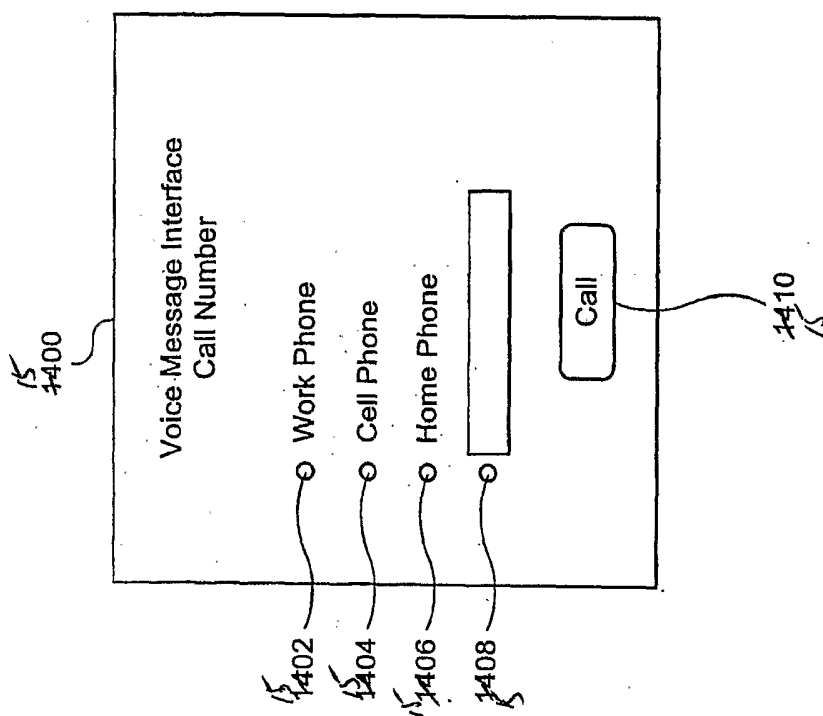


FIG. 14<sup>15</sup>

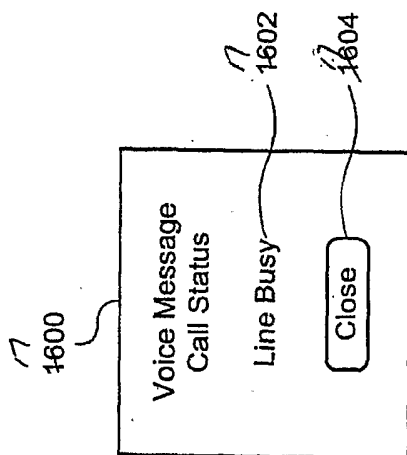


FIG. 16

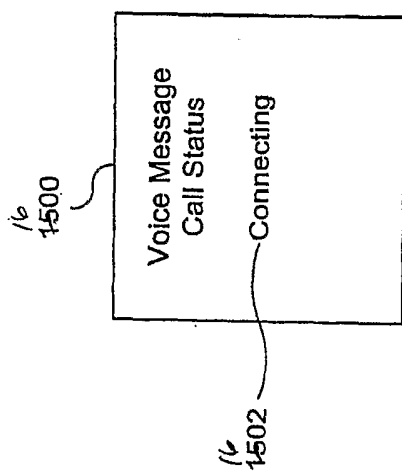


FIG. 15

17/56

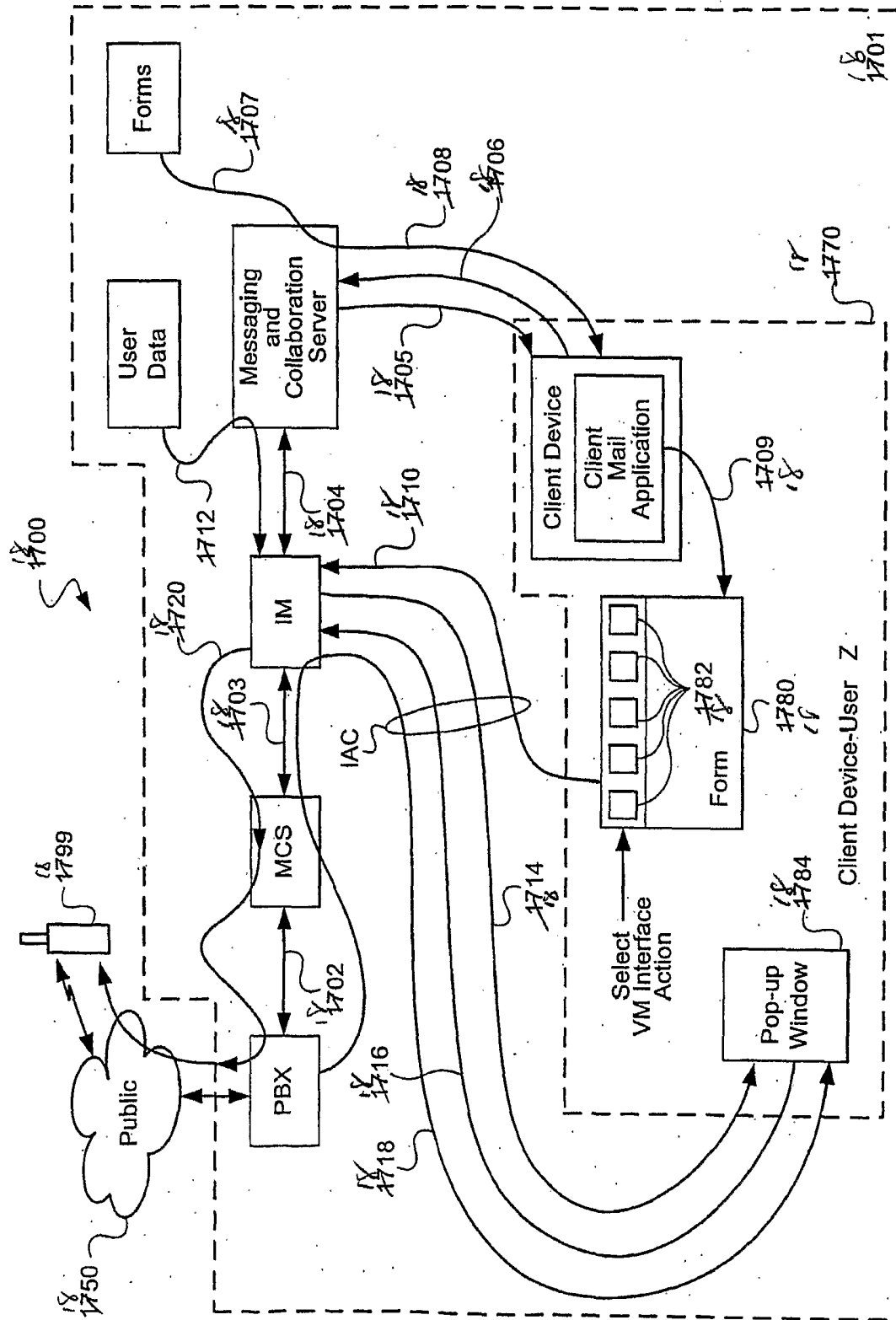


FIG. 17



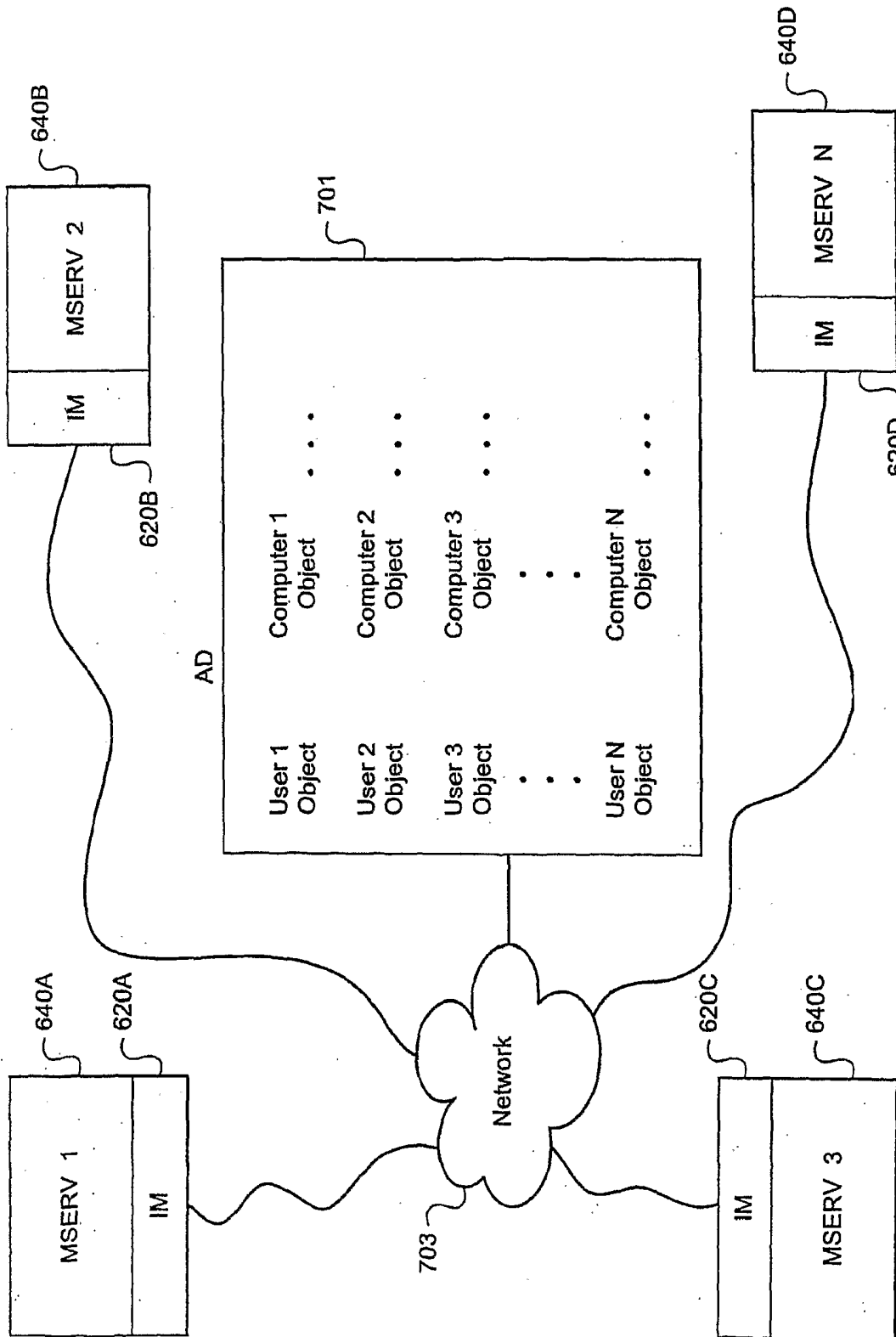


FIG. 7

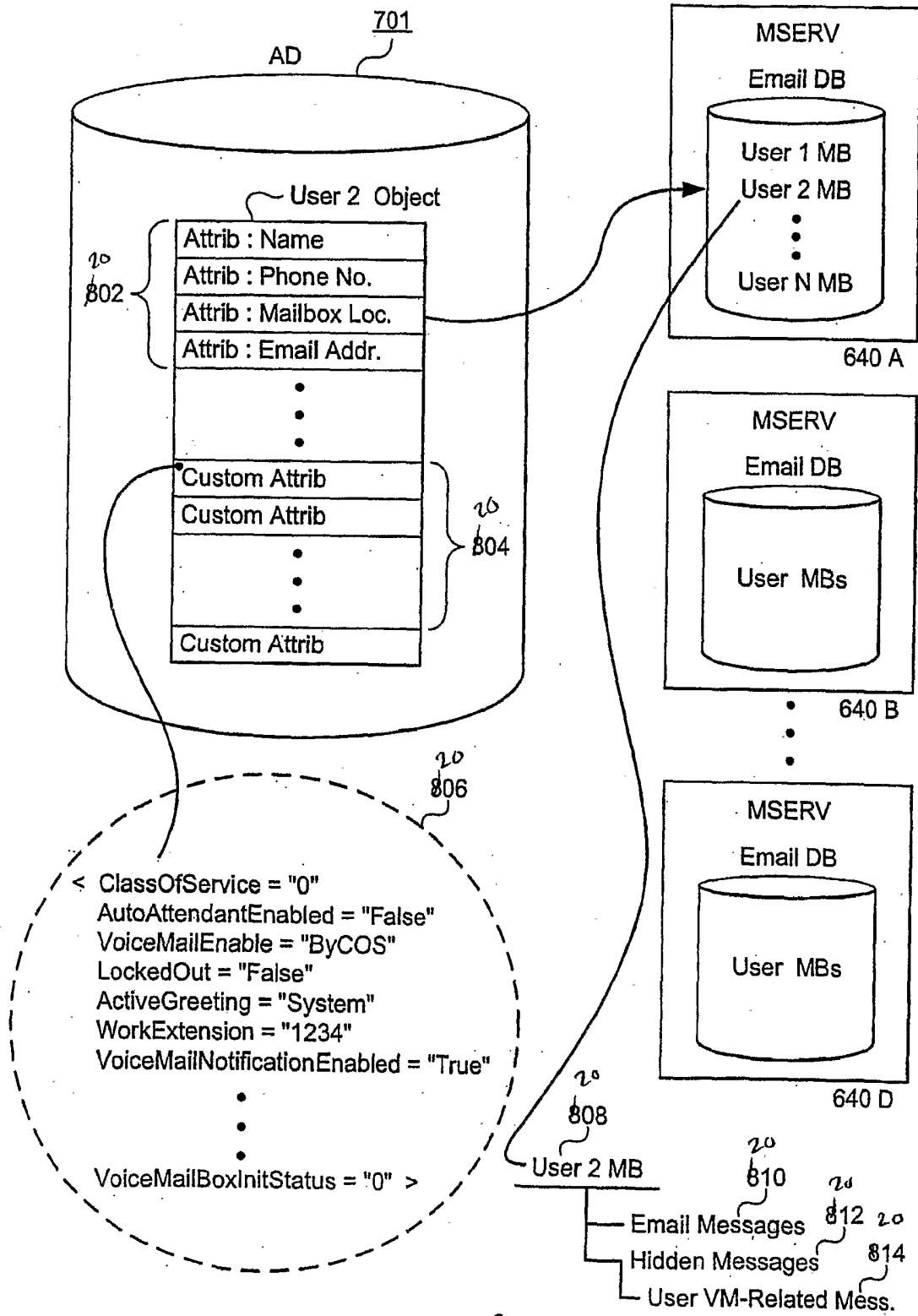


FIG. 8  
20.

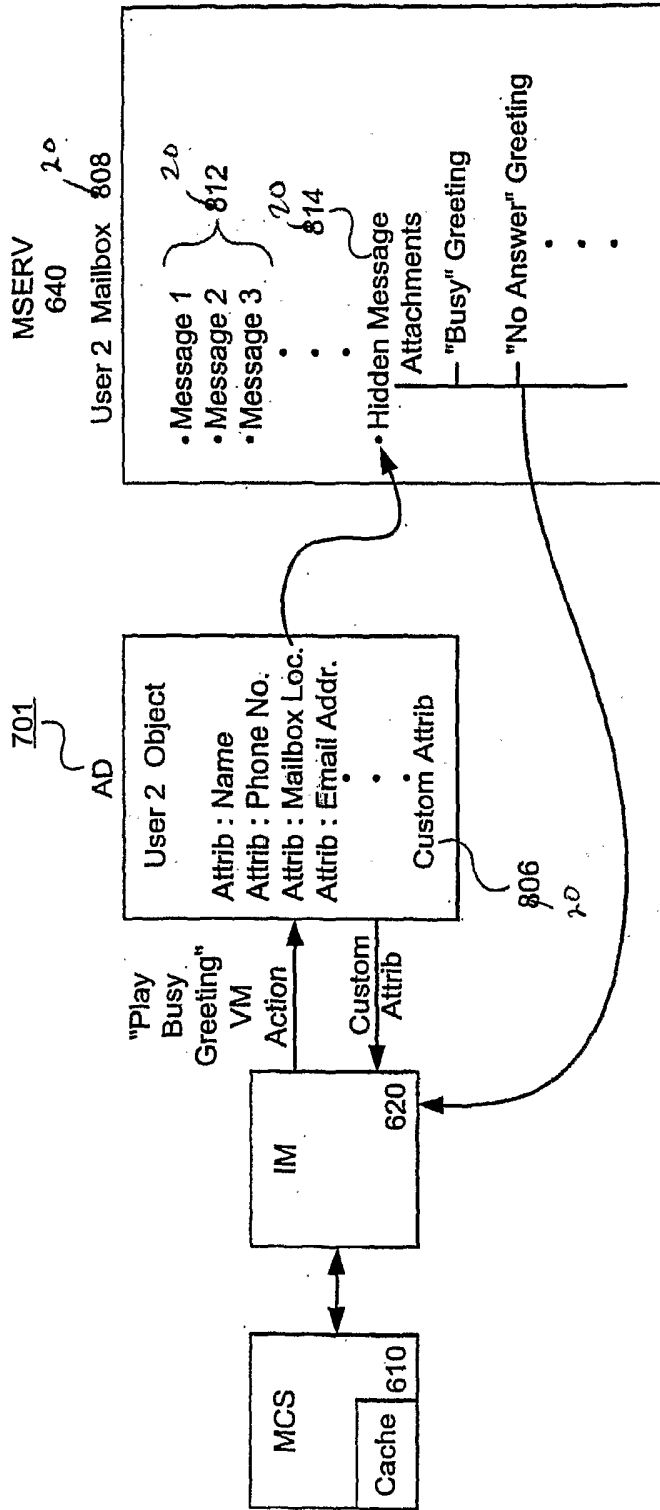


FIG. 9  
21

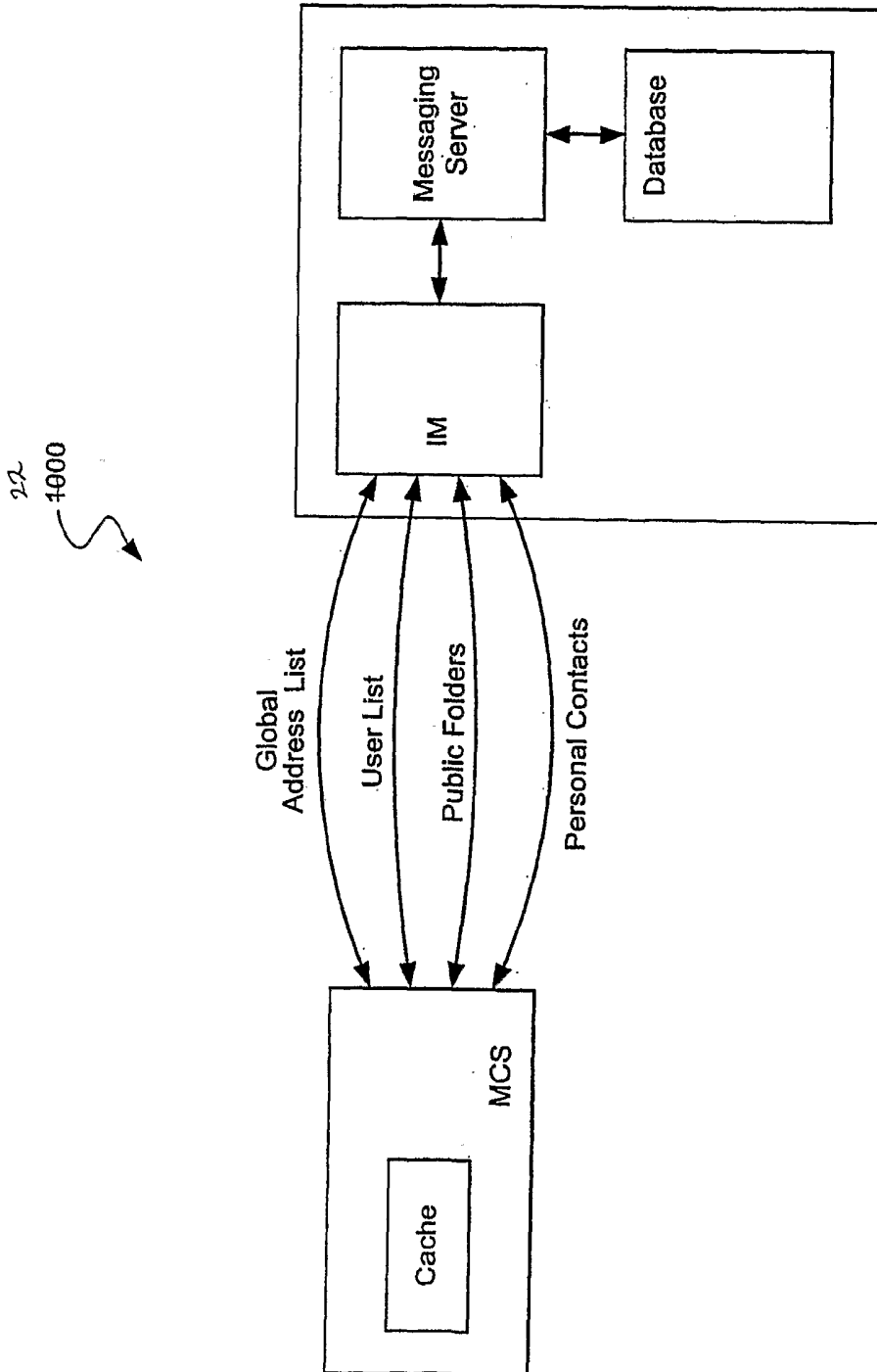


FIG. 10  
22

22/56

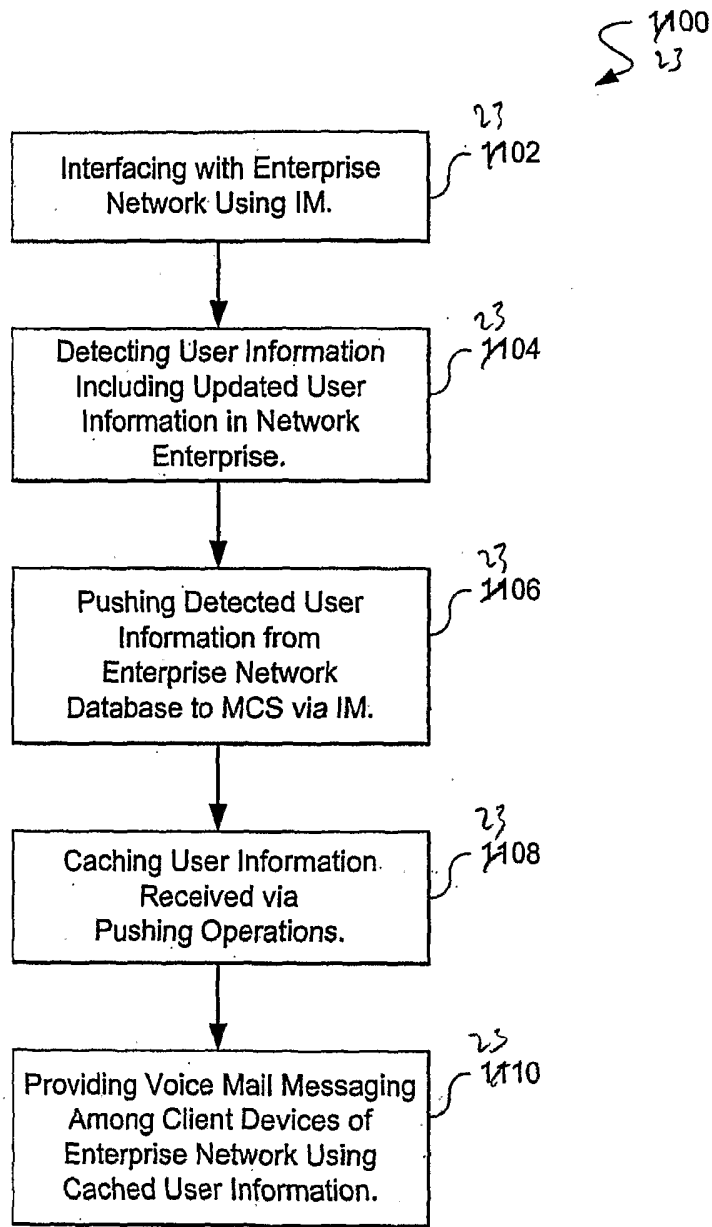


FIG. 11<sup>e</sup>  
23

1200  
24

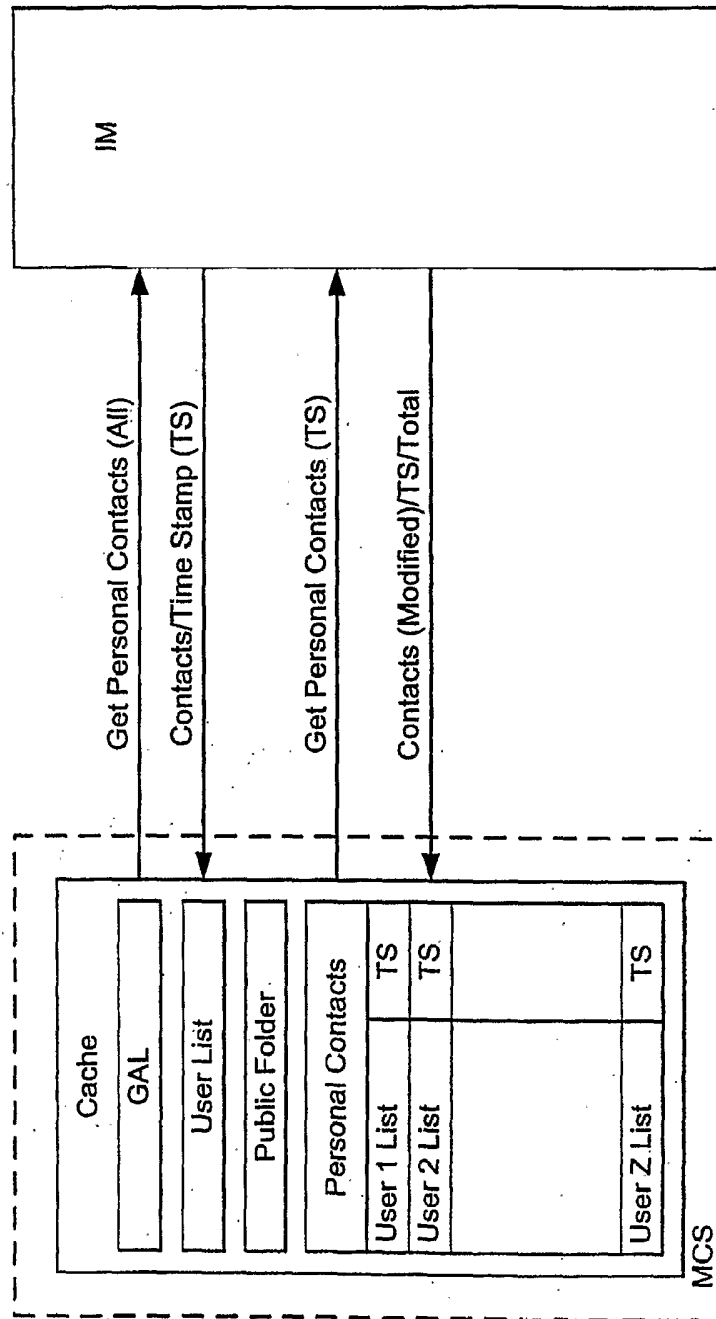


FIG. 12  
24

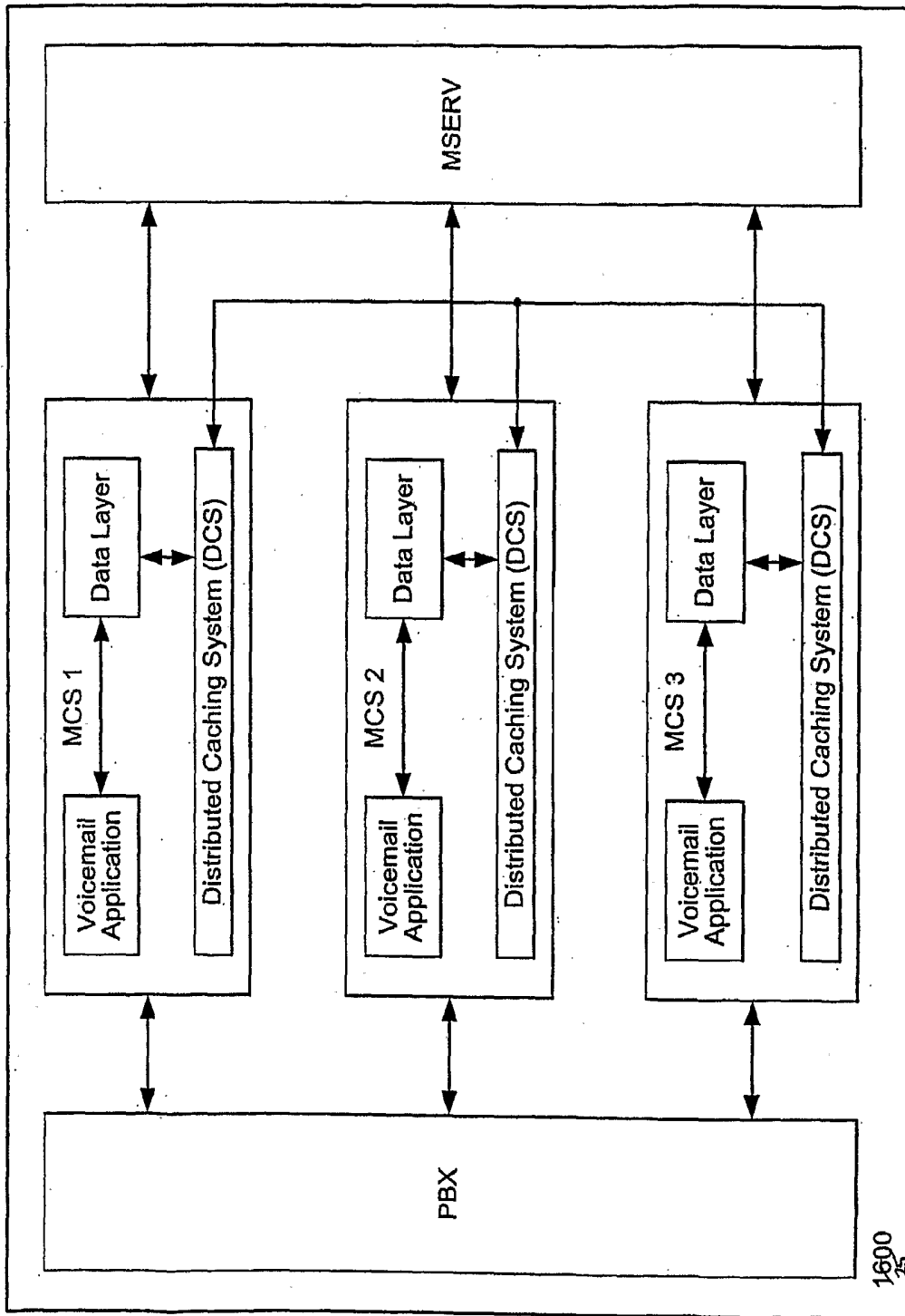


FIG. 16<sub>25</sub>

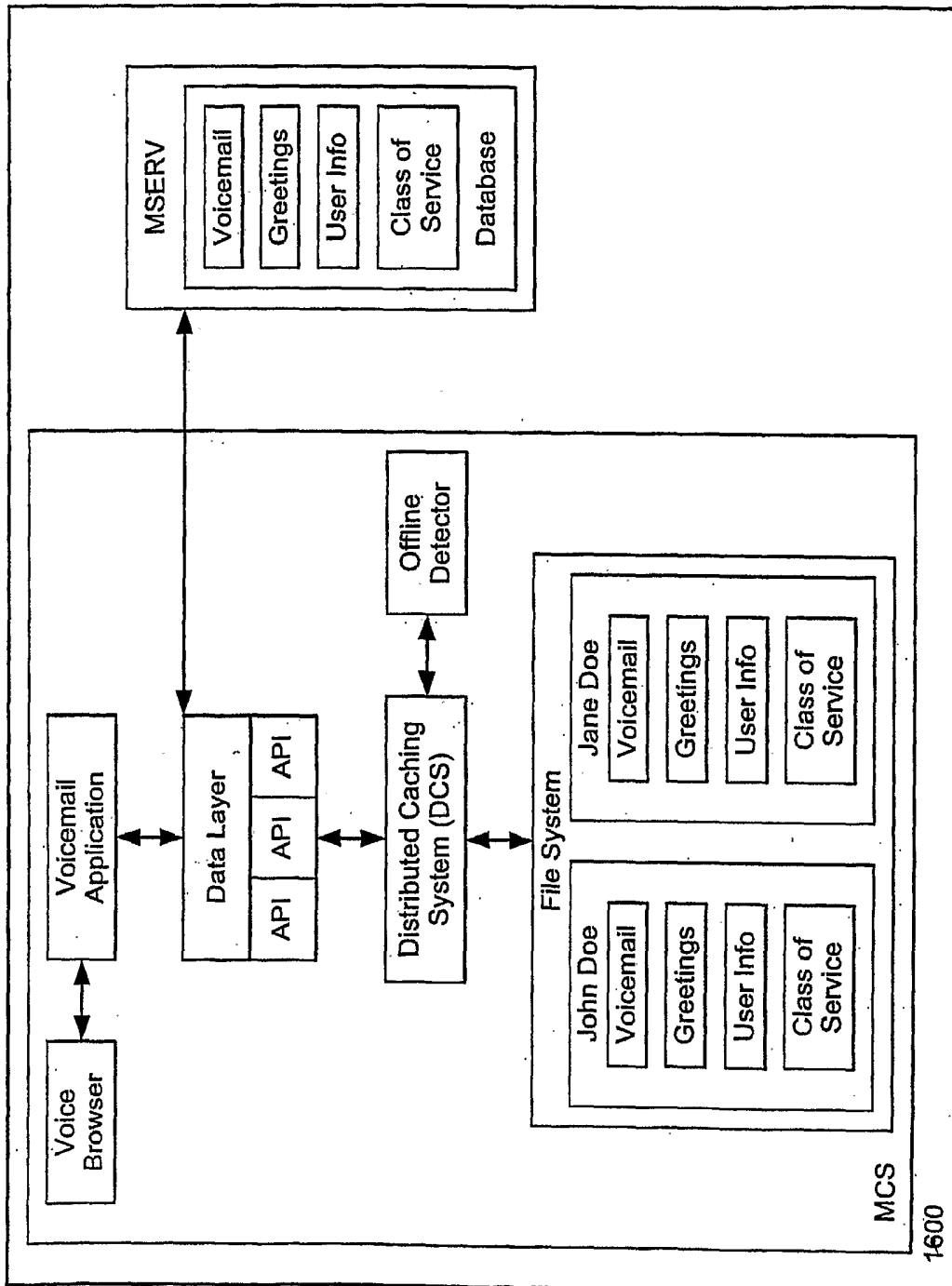


FIG. 17  
26



26/56

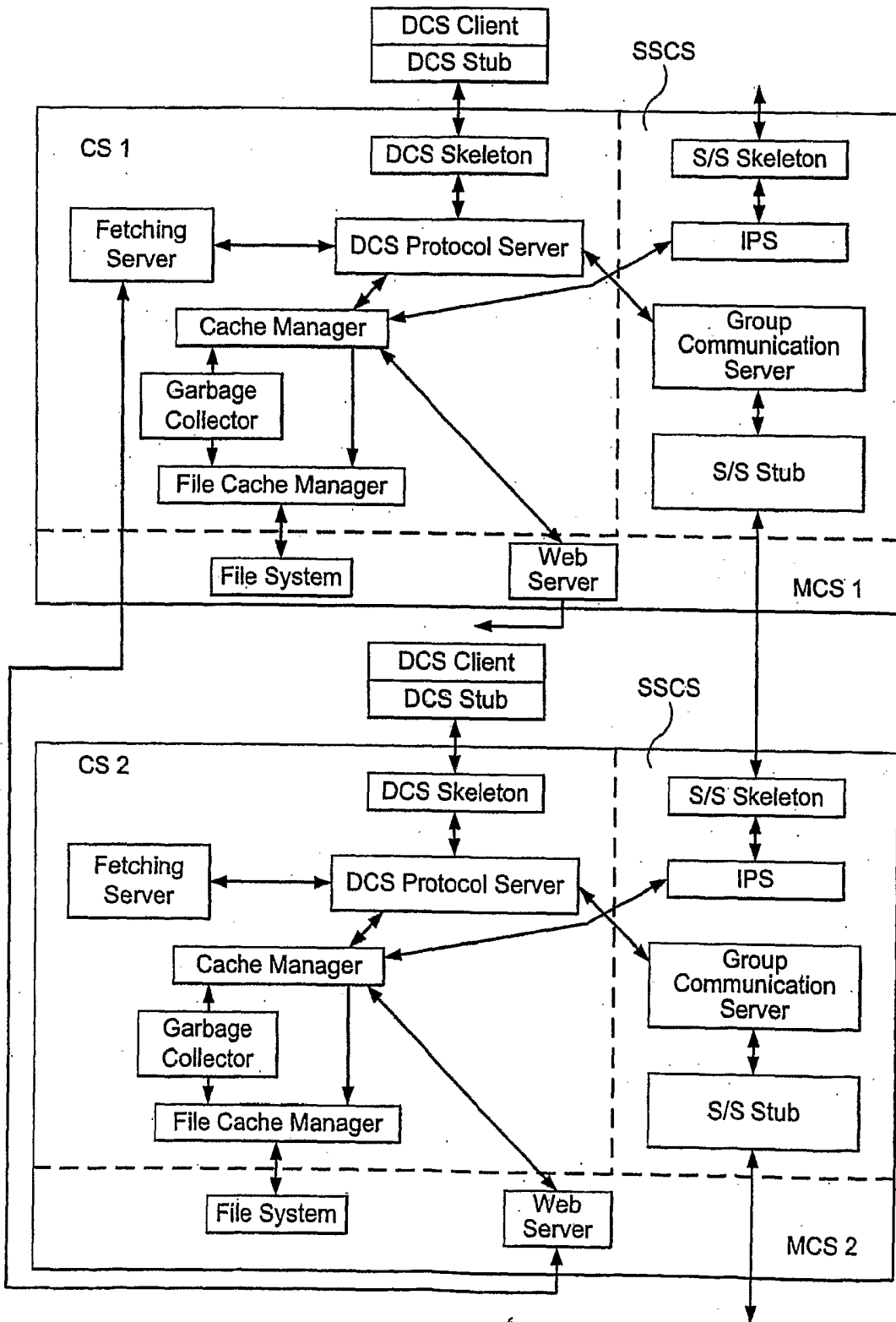


FIG. 18  
27

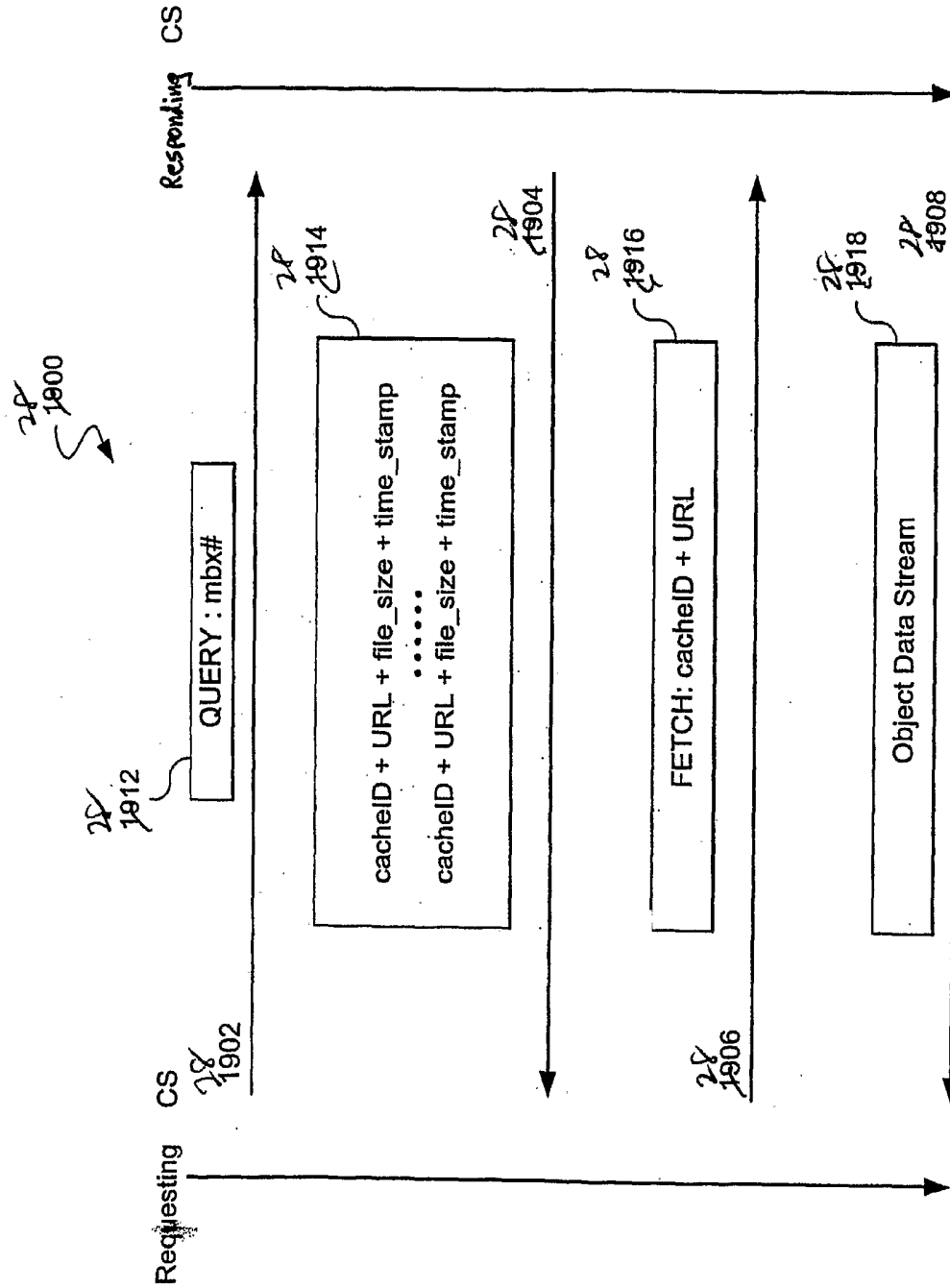


FIG. 19  
28

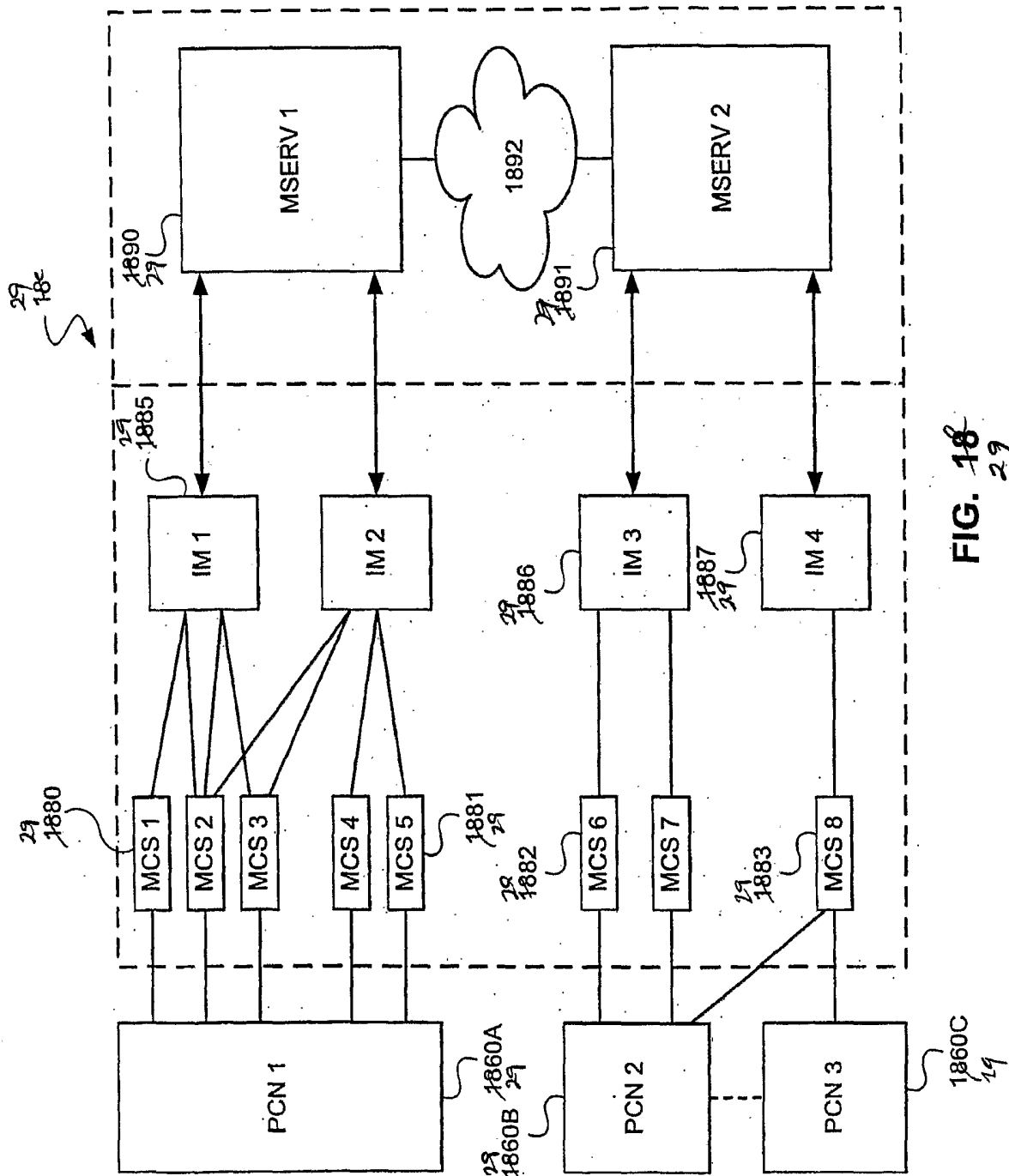


FIG. 18  
29

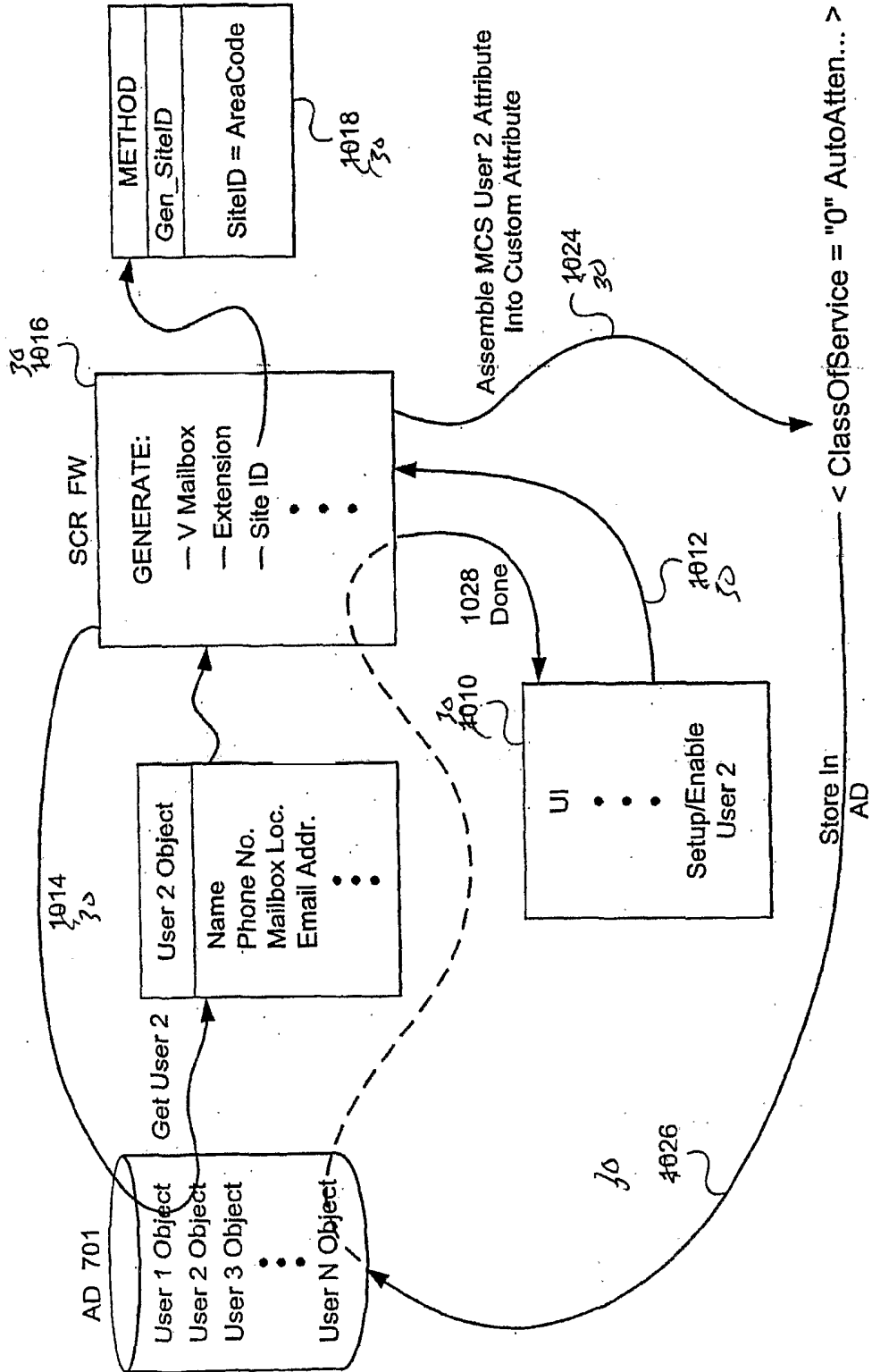


FIG. 10 30

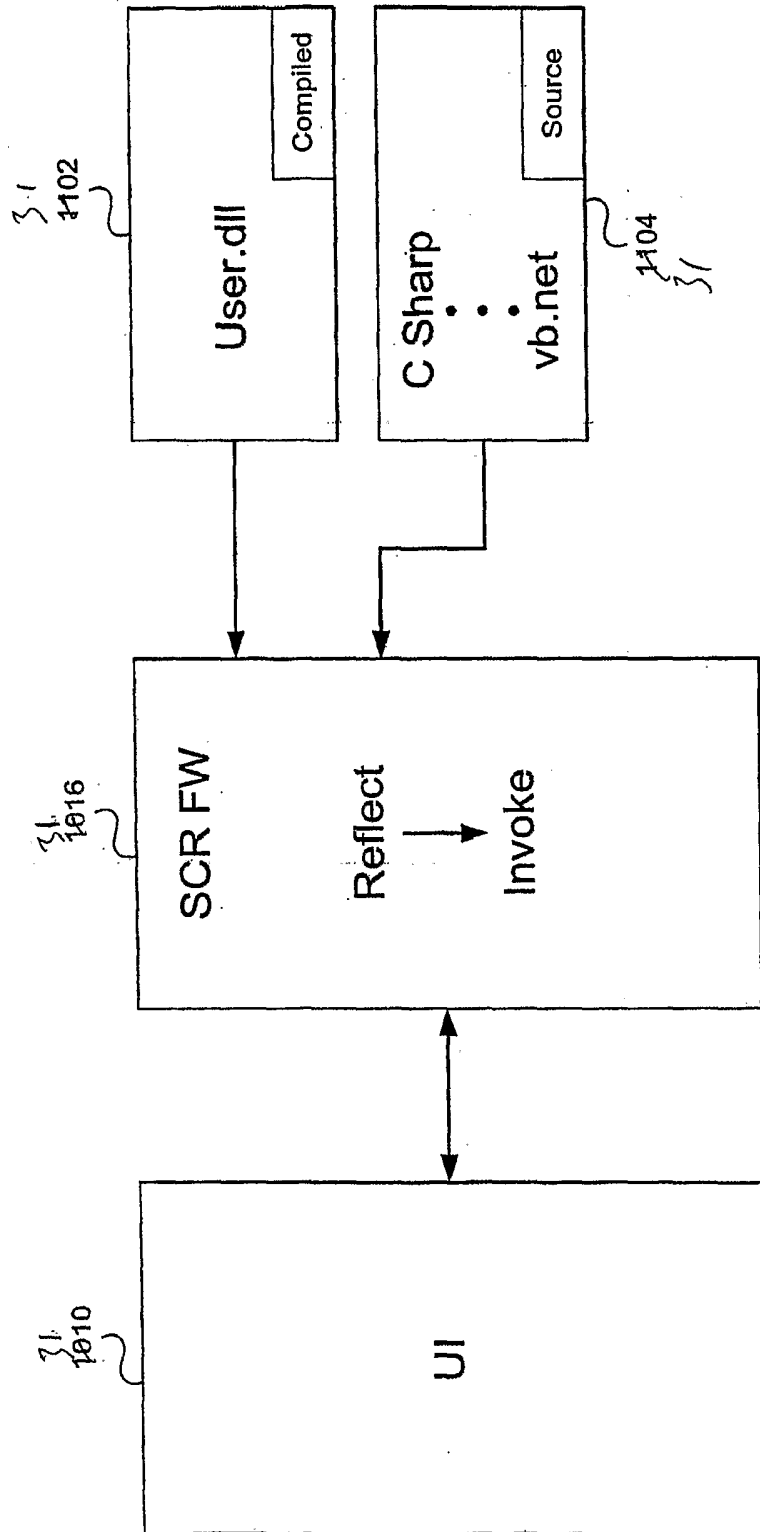


FIG. 11

31/56

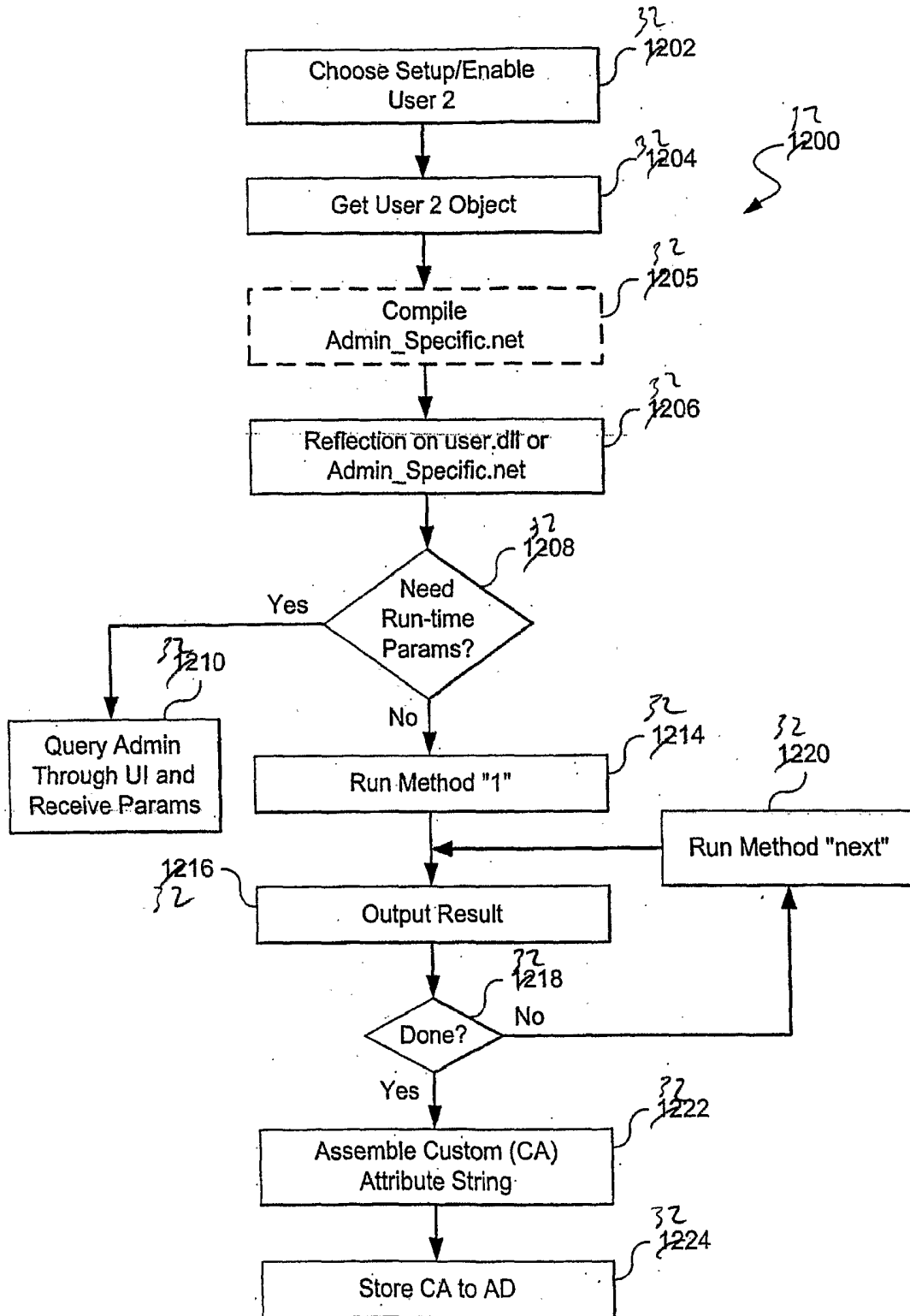


FIG. 12<sup>l</sup>  
32

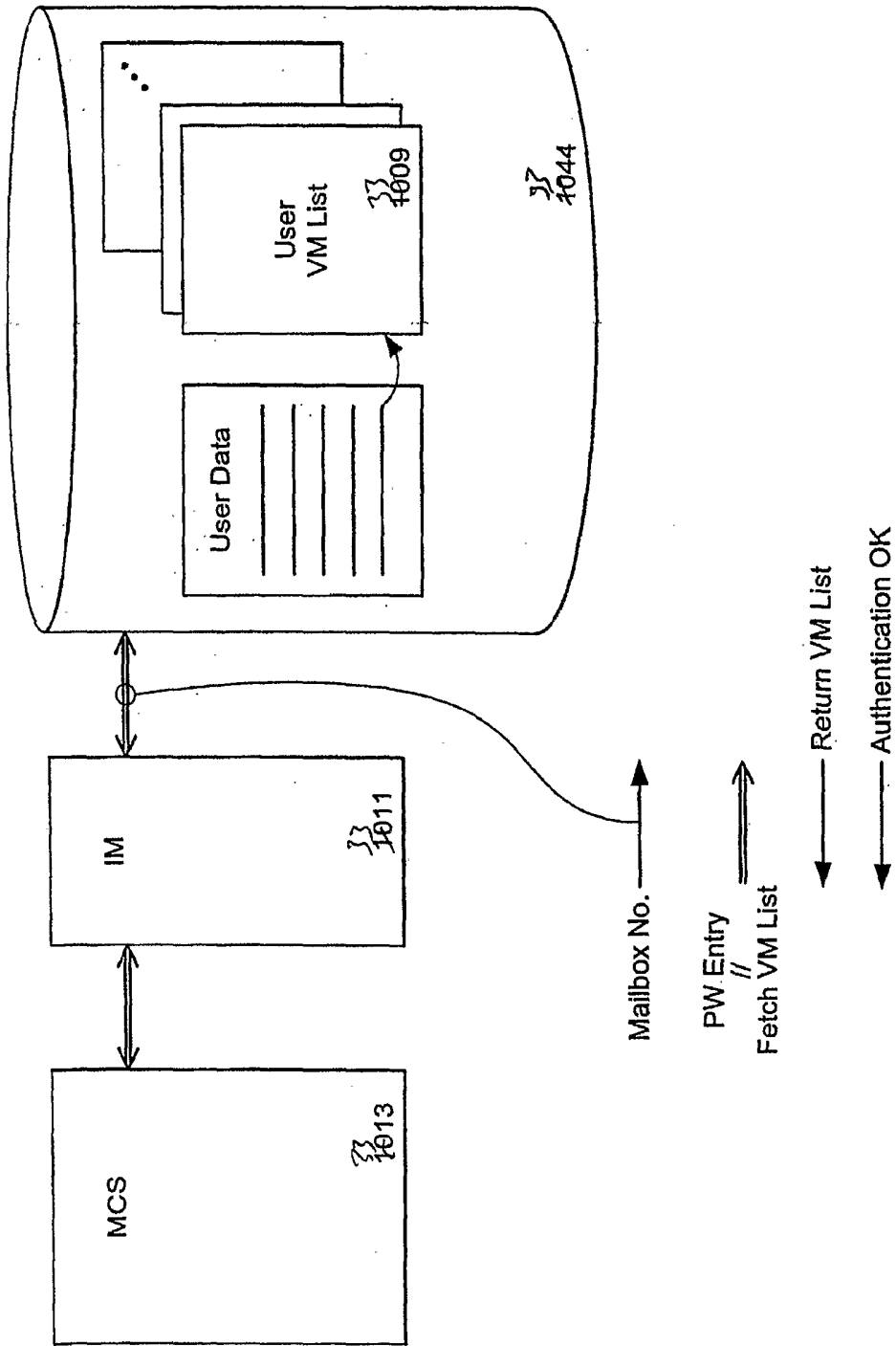


FIG. 10  
33

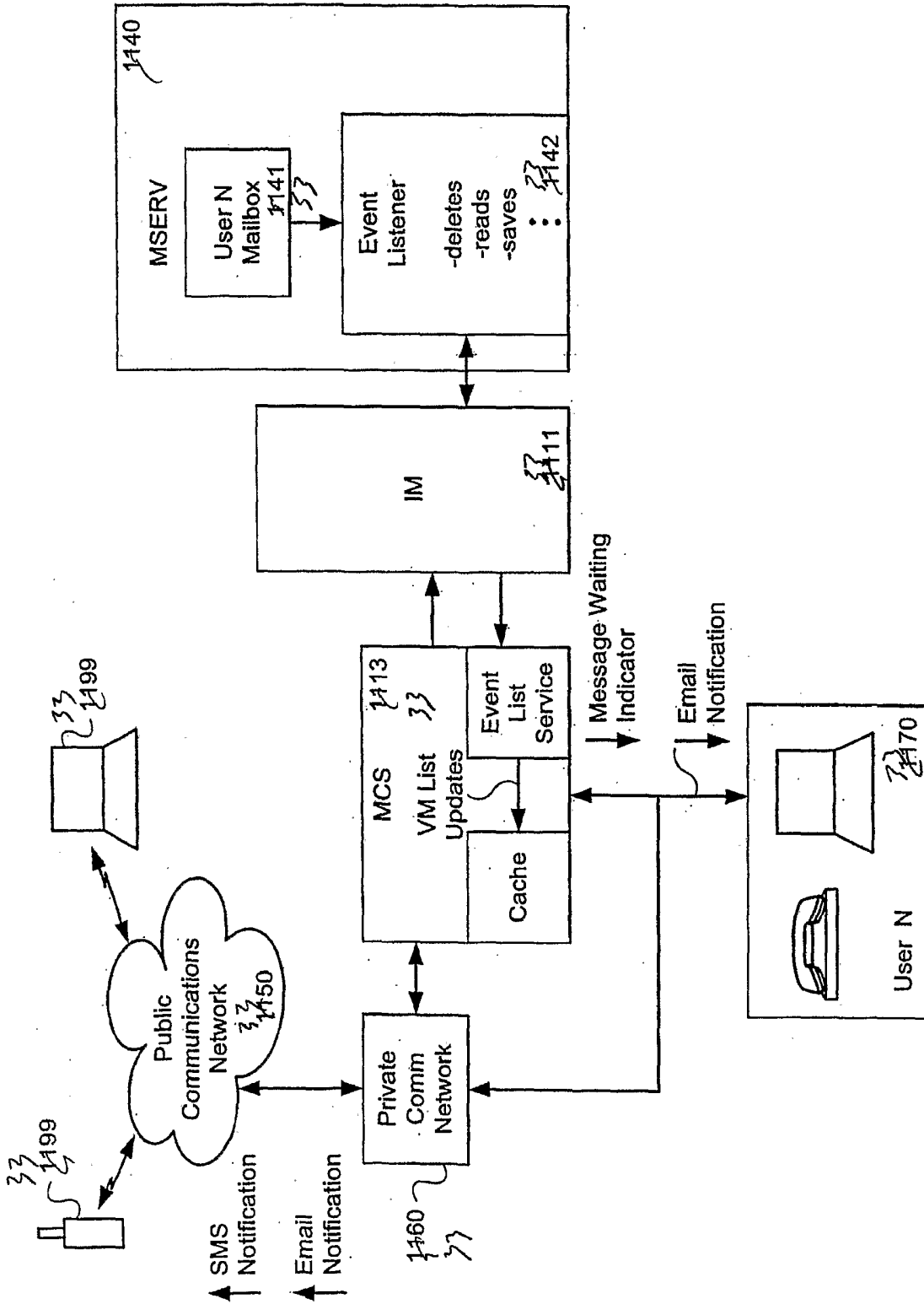


FIG. 1134



34/56

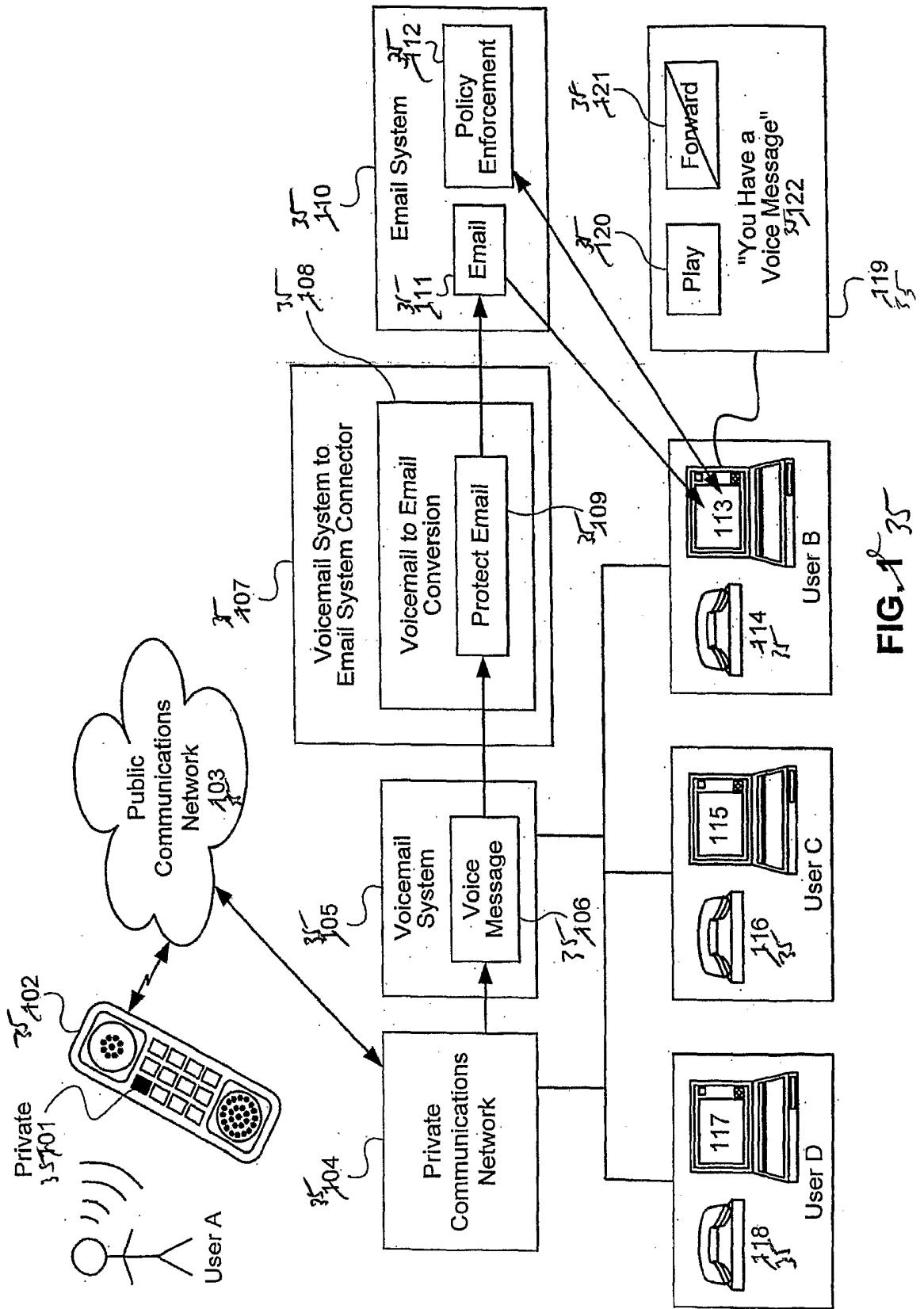


FIG. 1 35

35/56

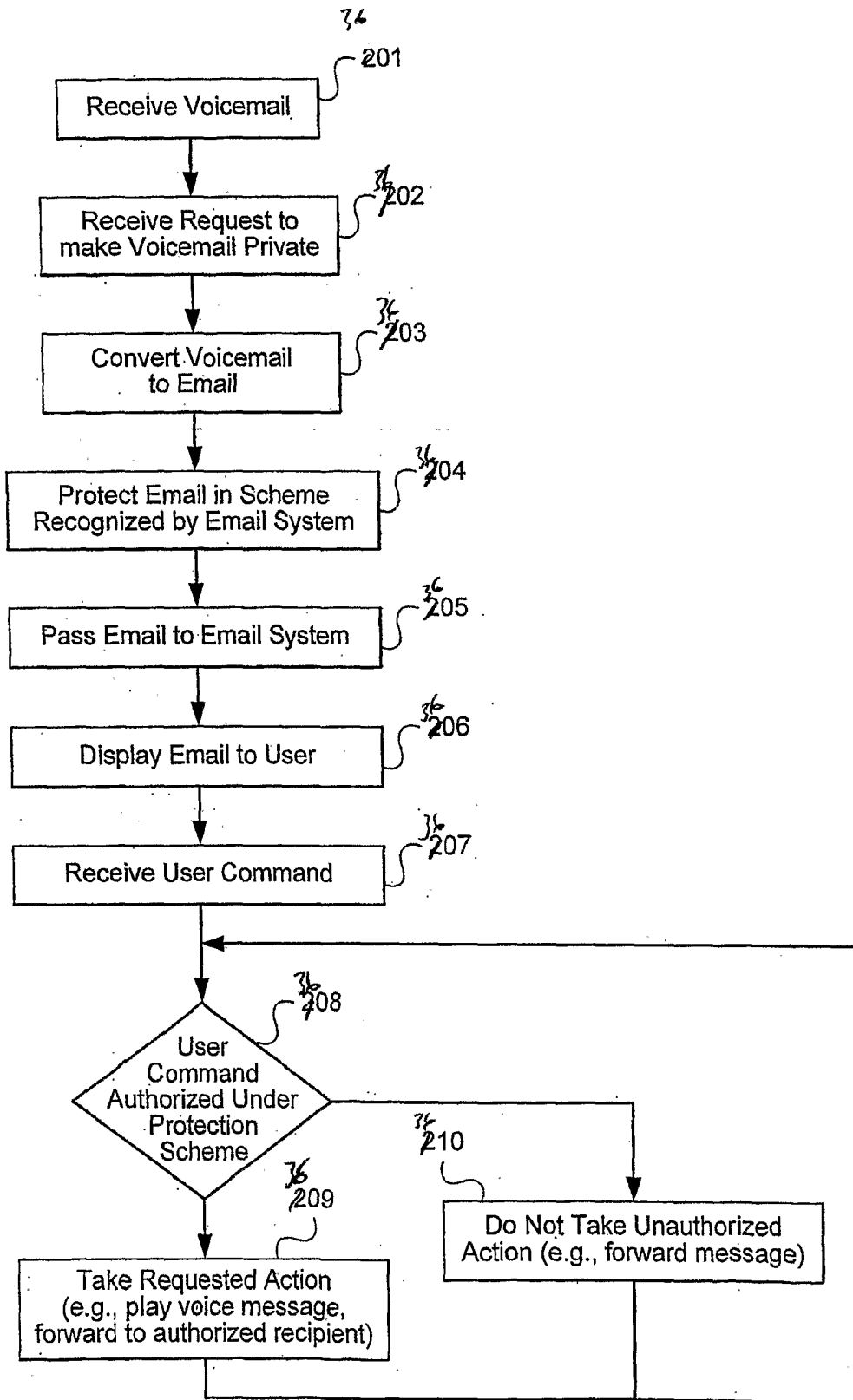


FIG. 2 36

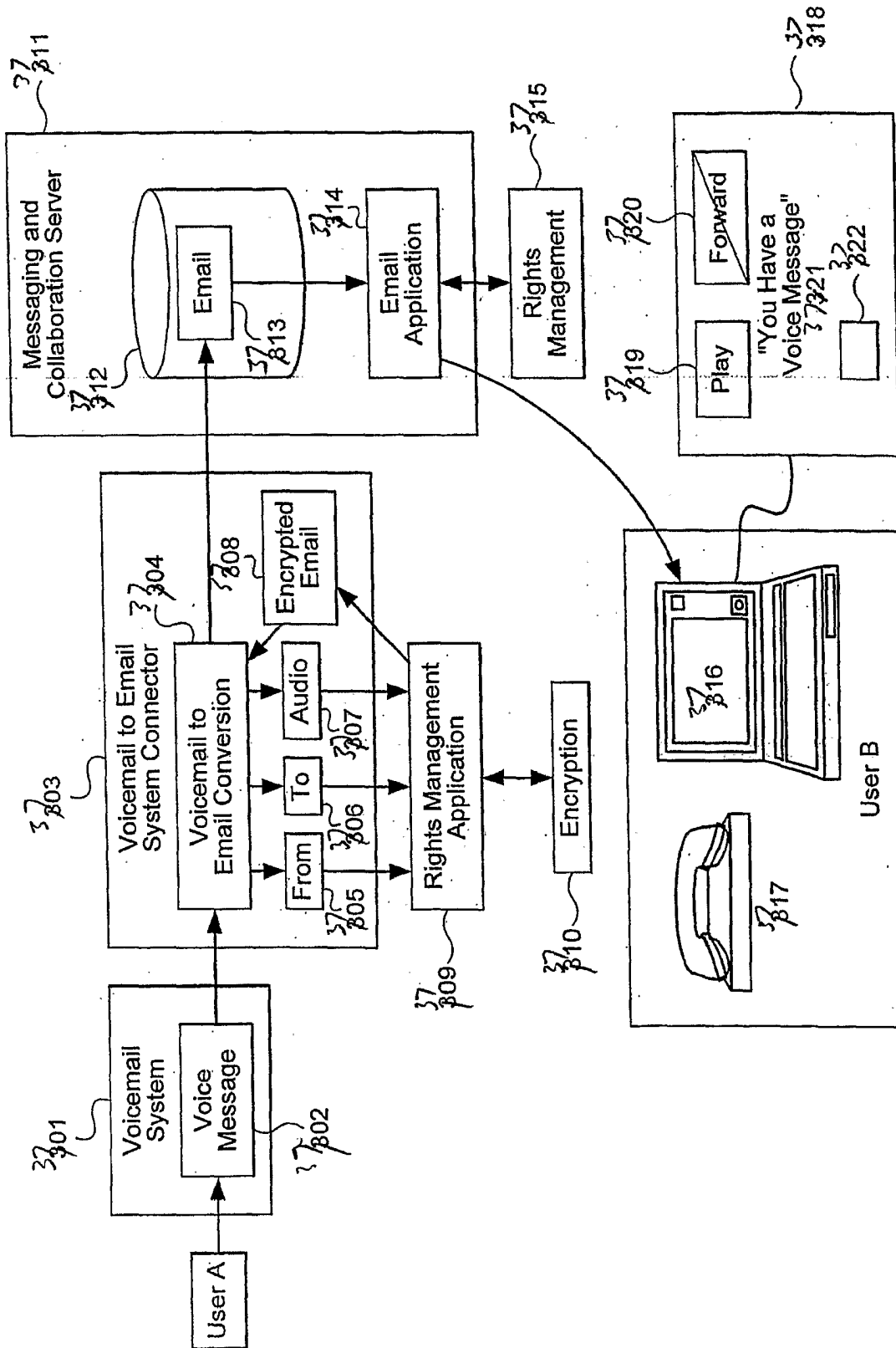


FIG. 3 37

37/56

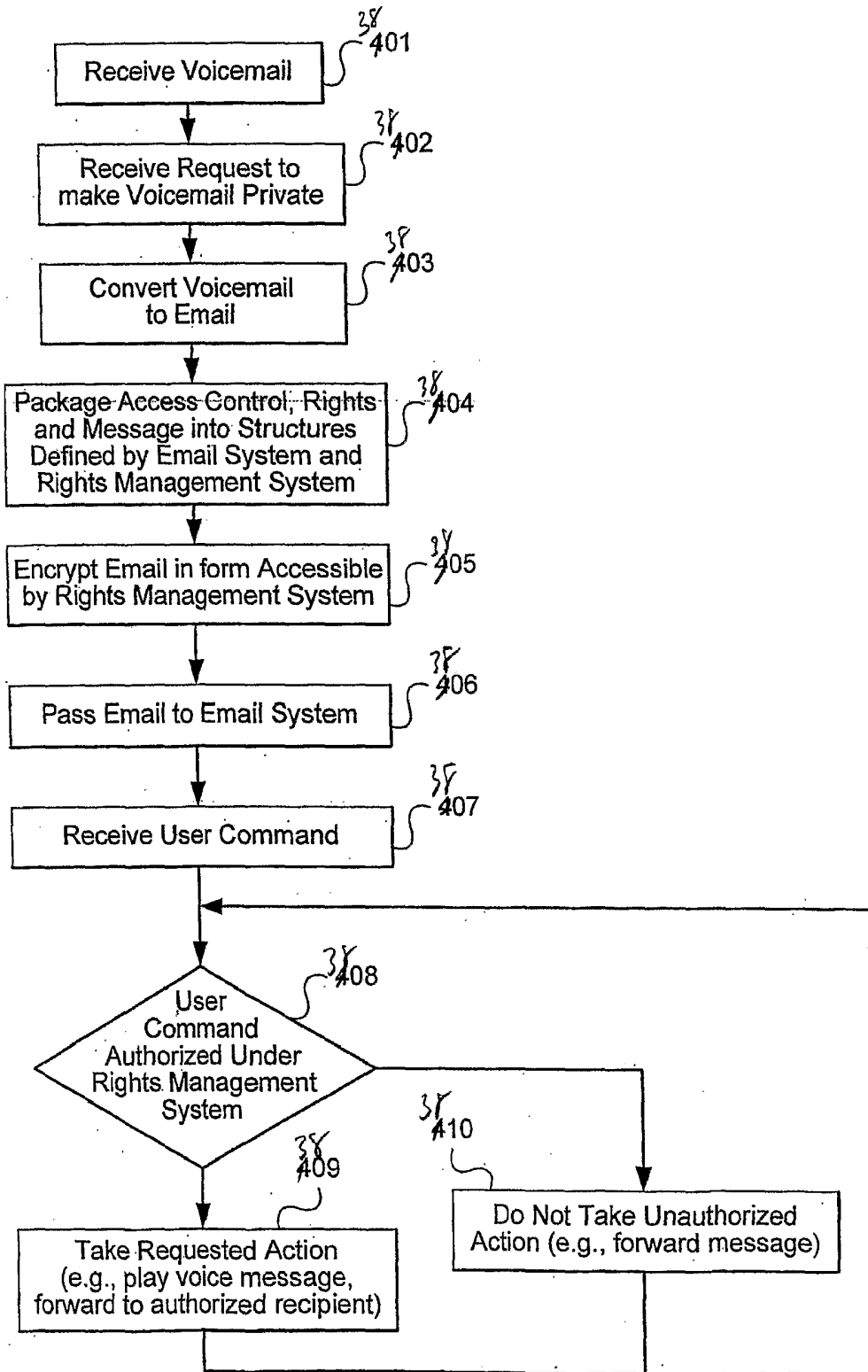


FIG. 4 38

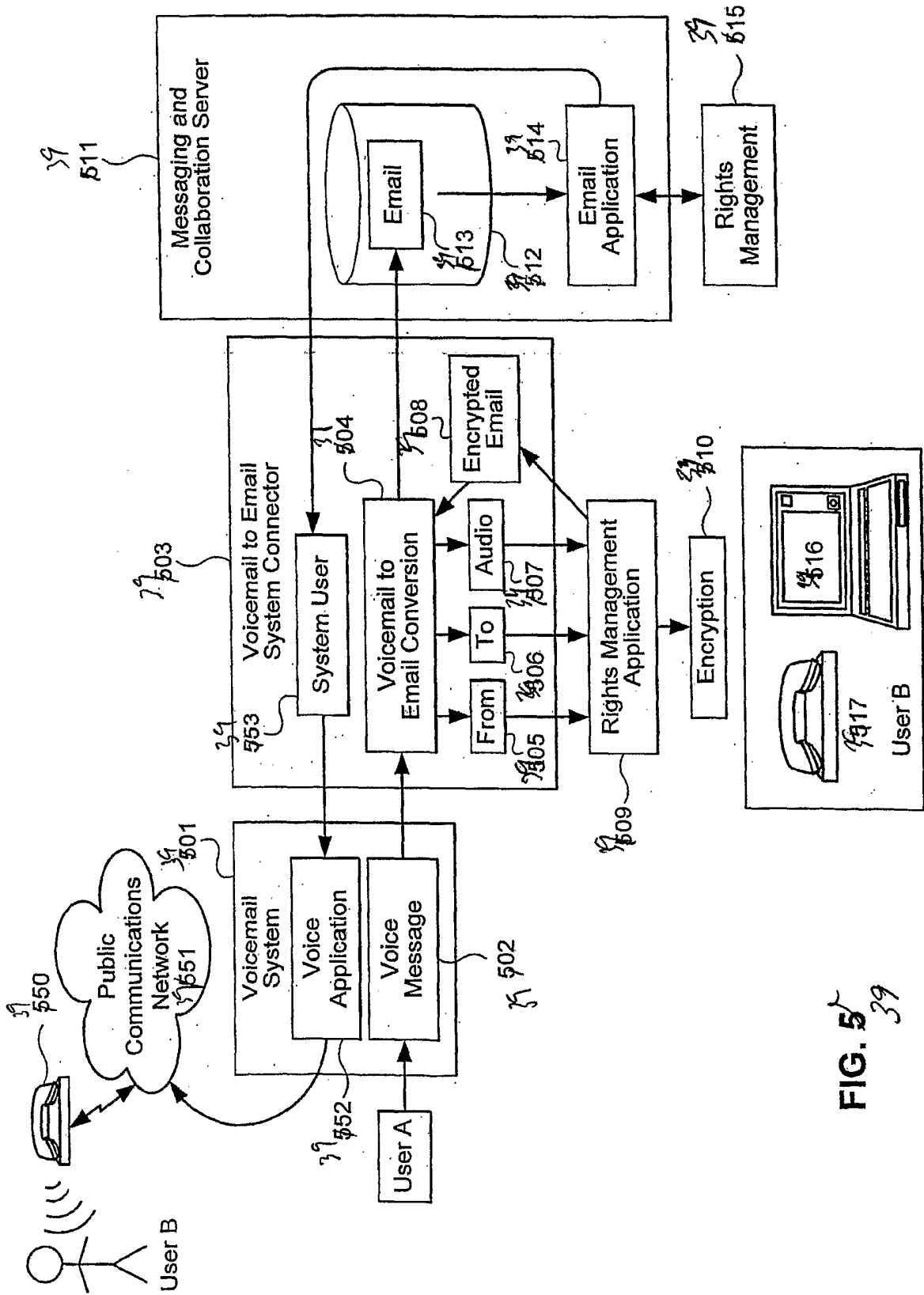


FIG. 5  
39

39/56

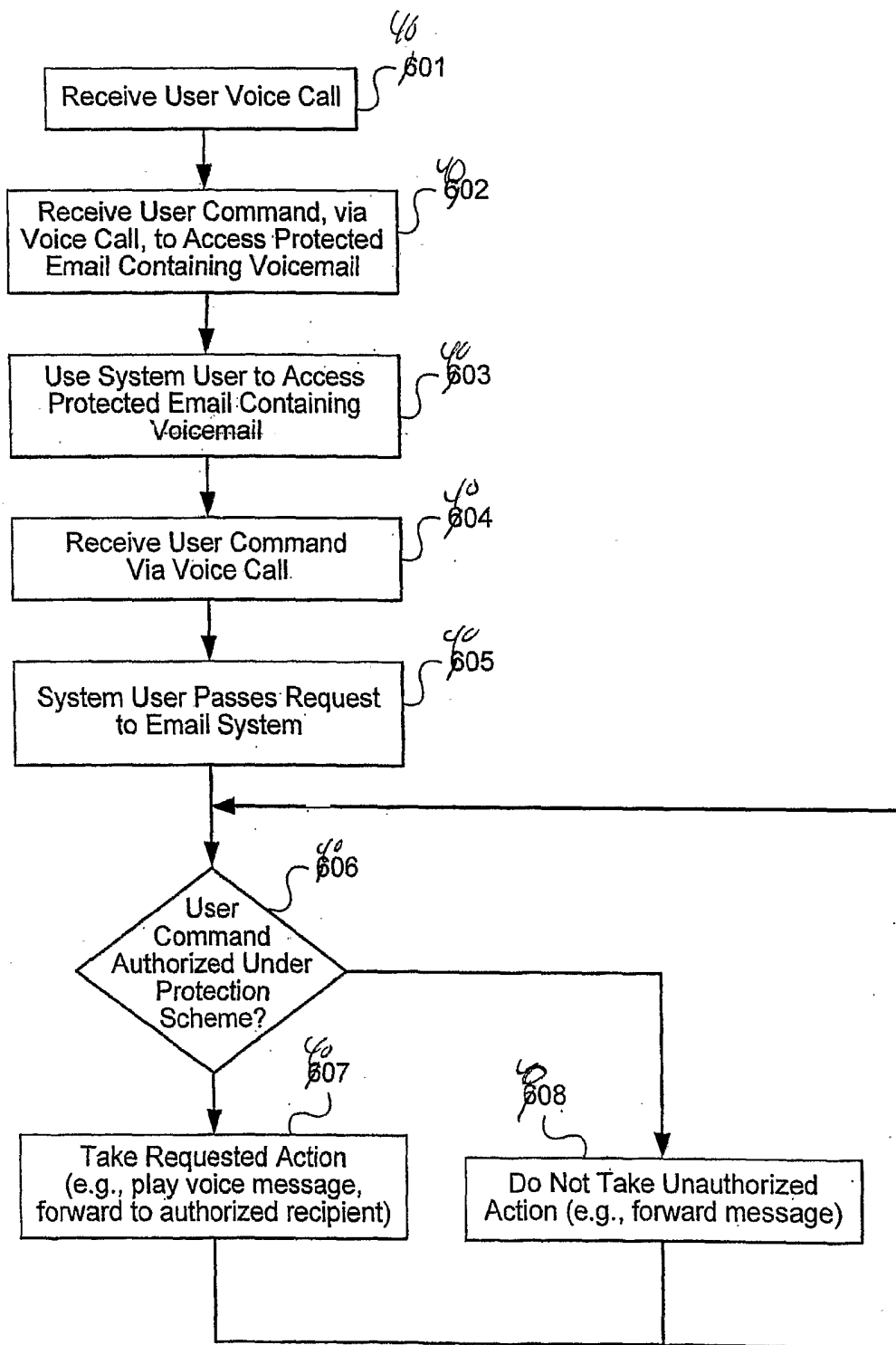


FIG. 6 40

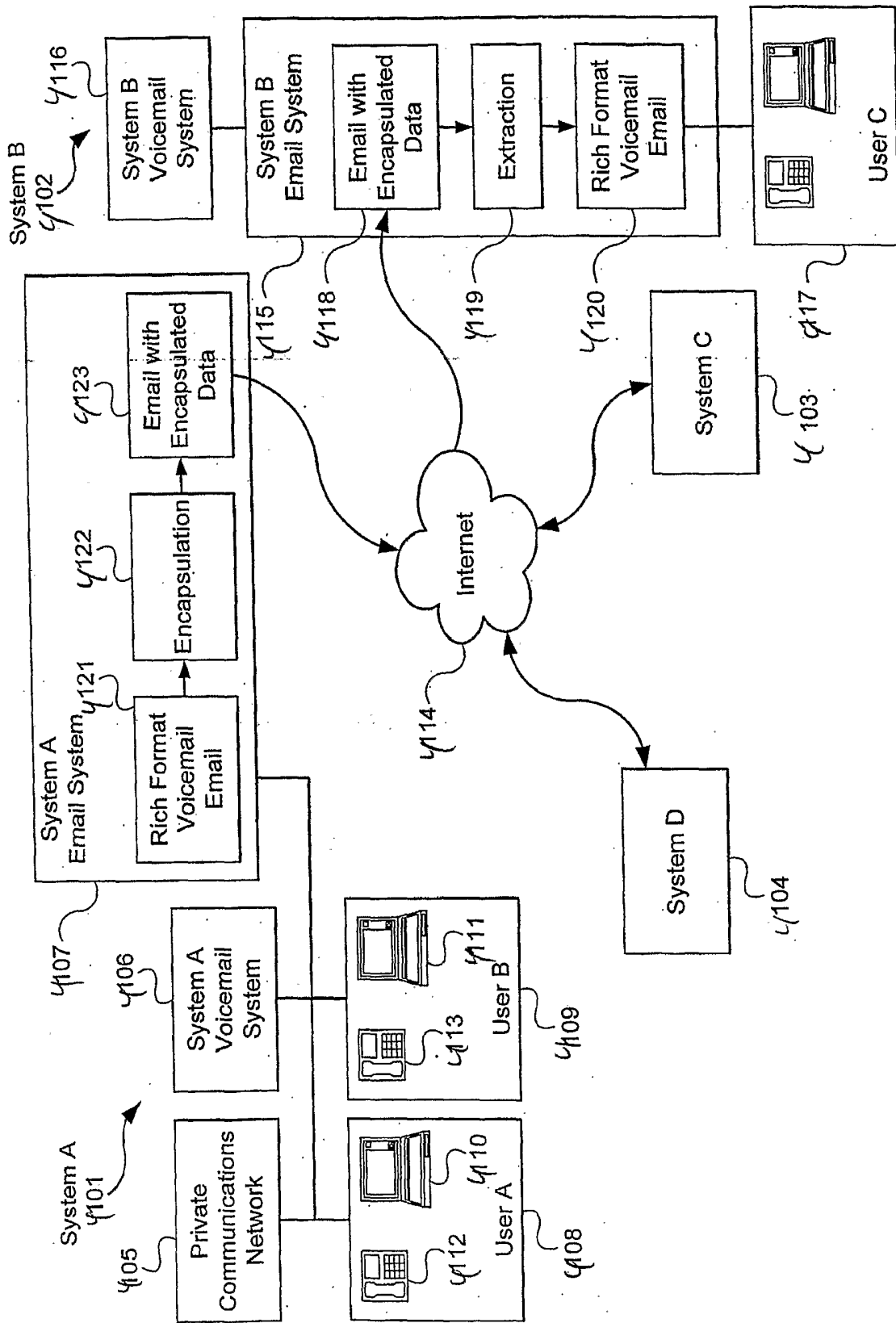


FIG. 1 41

41/56

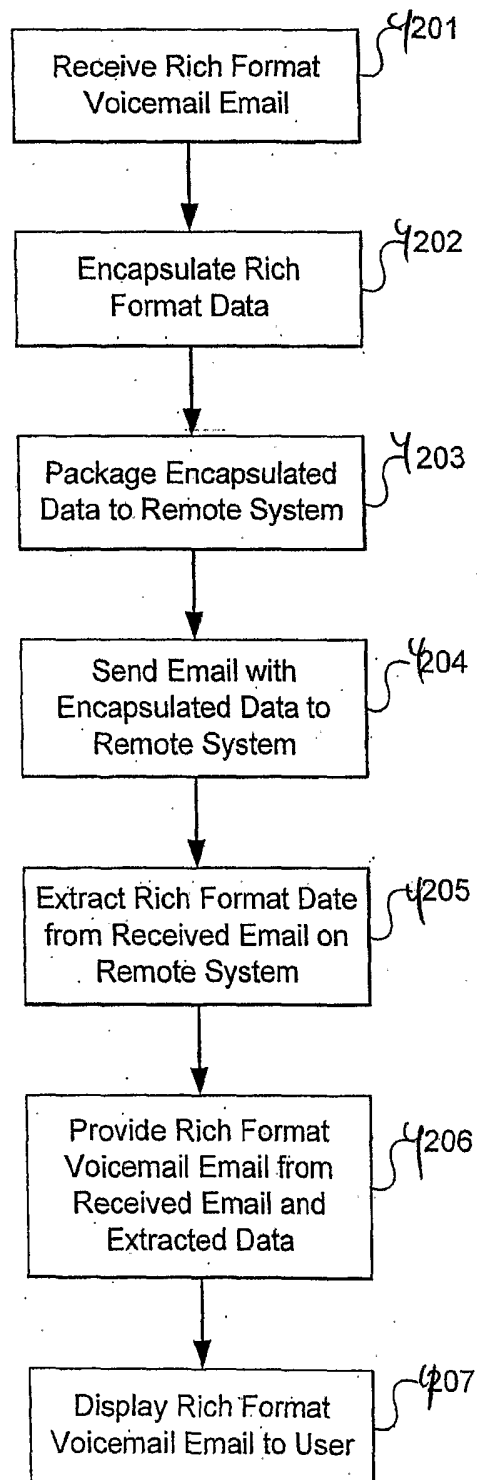


FIG. 2 42



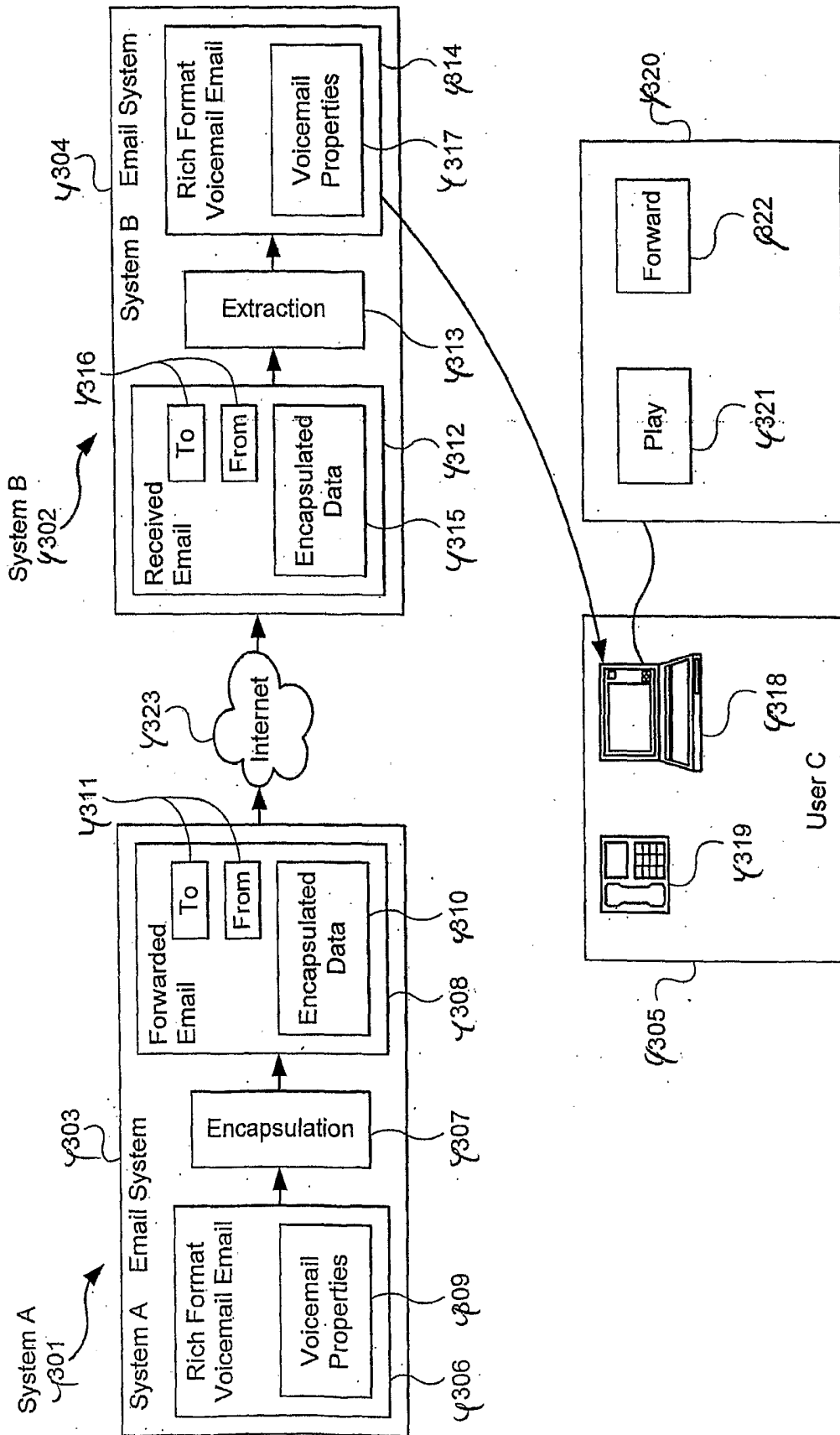


FIG. 3-43

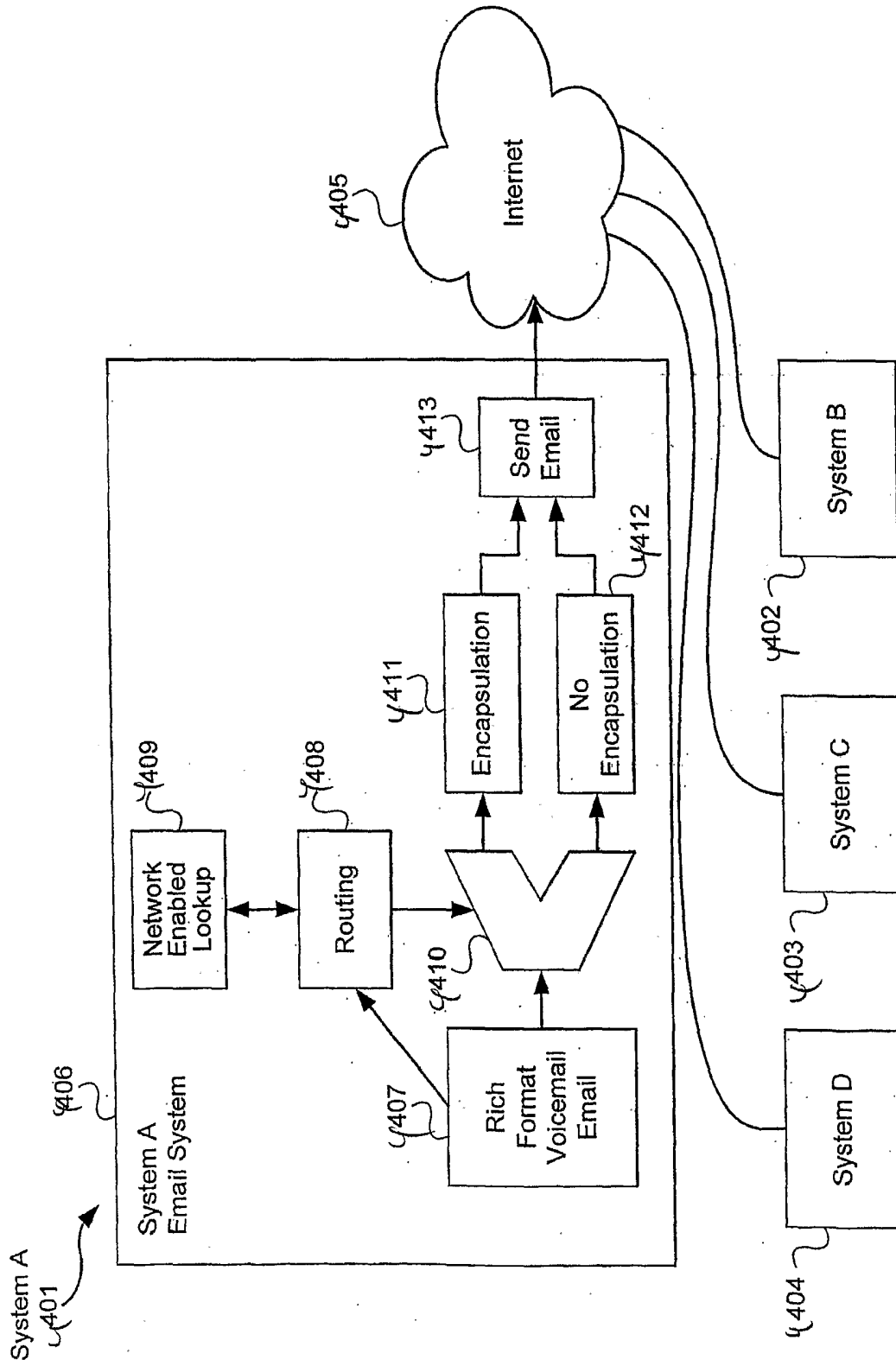


FIG. 4A 44

44/56

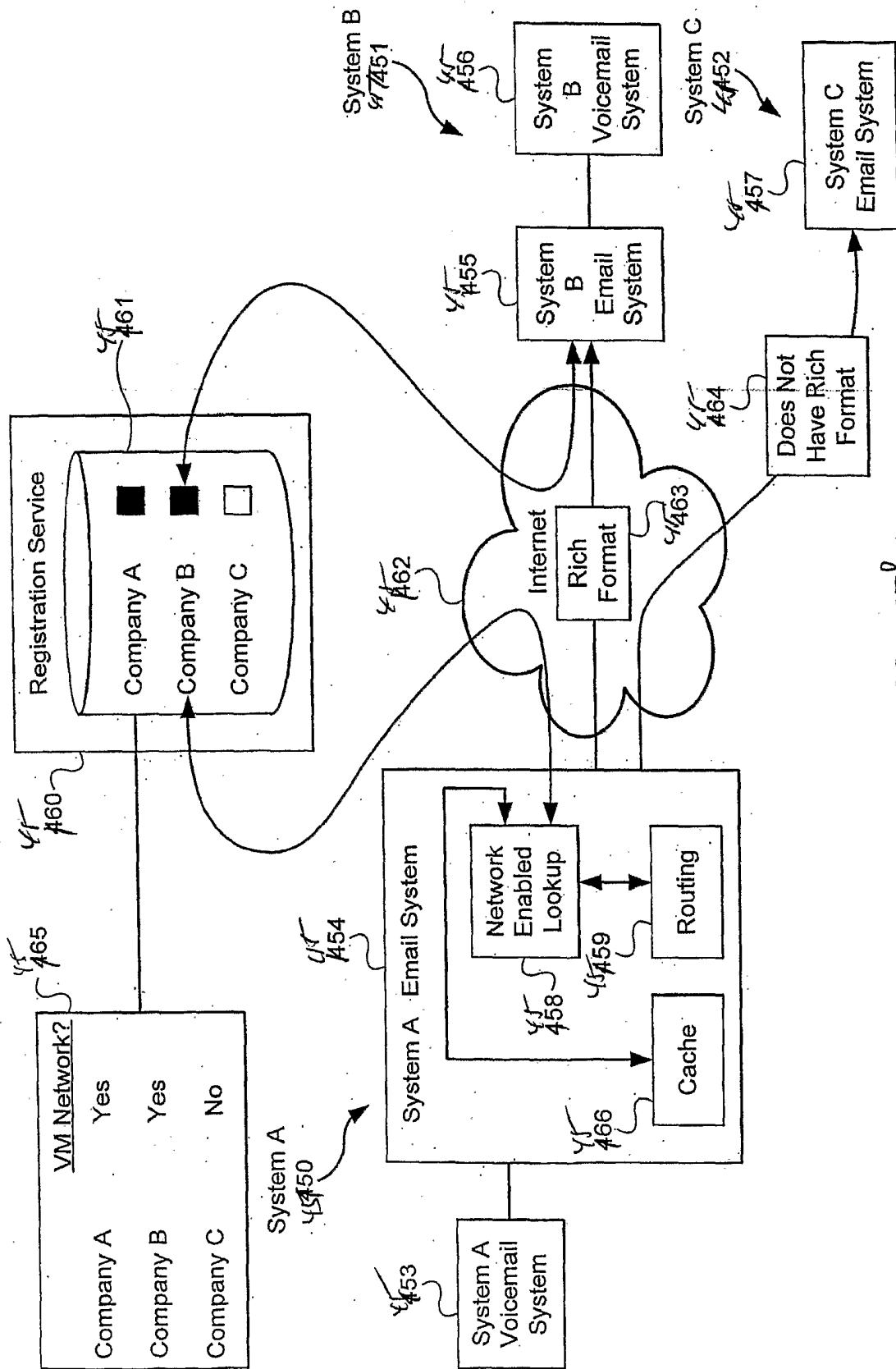


FIG. 4B

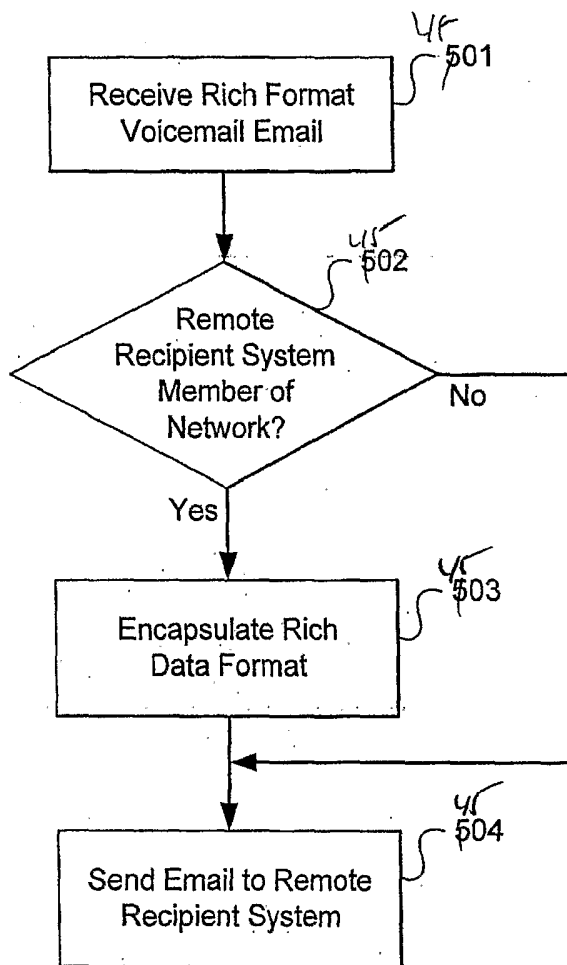


FIG. 5 4/6

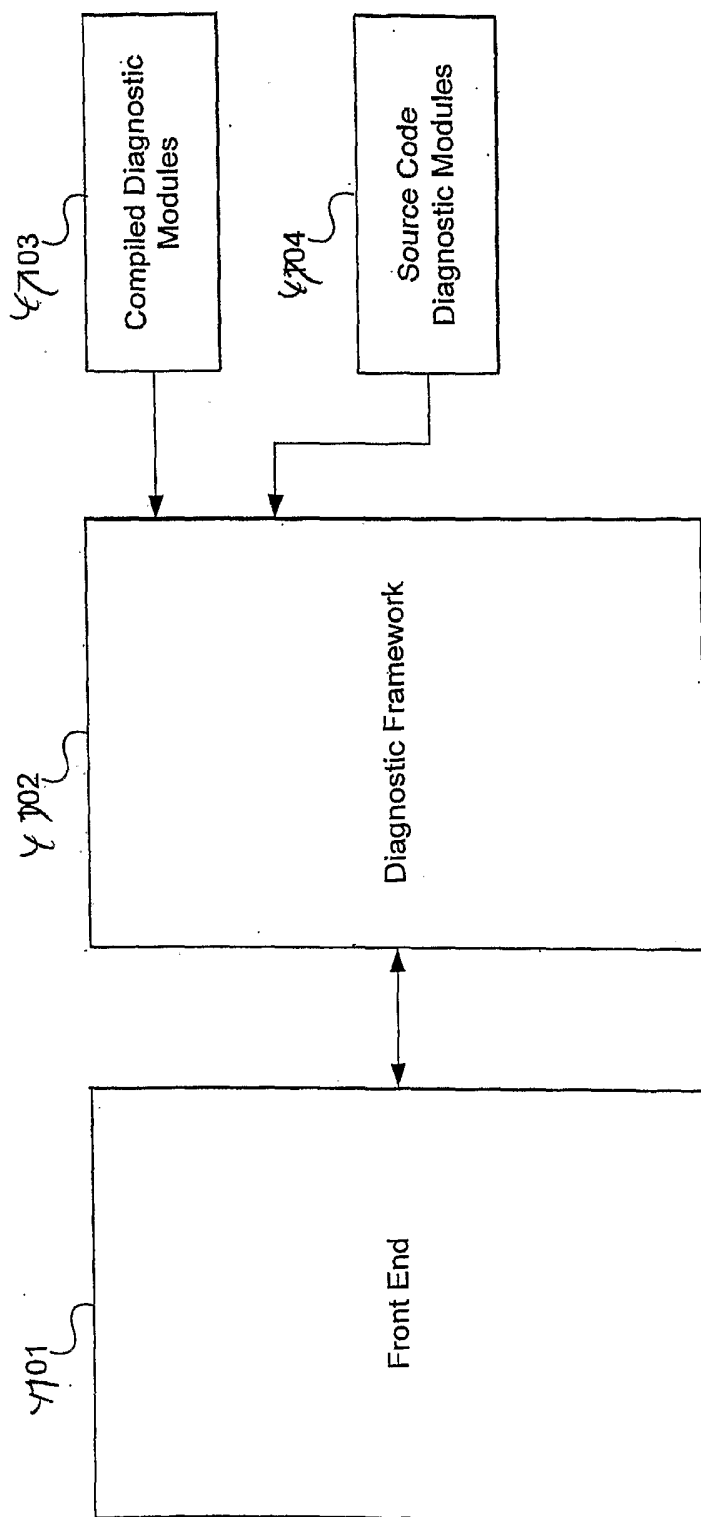


FIG. 147

47/56

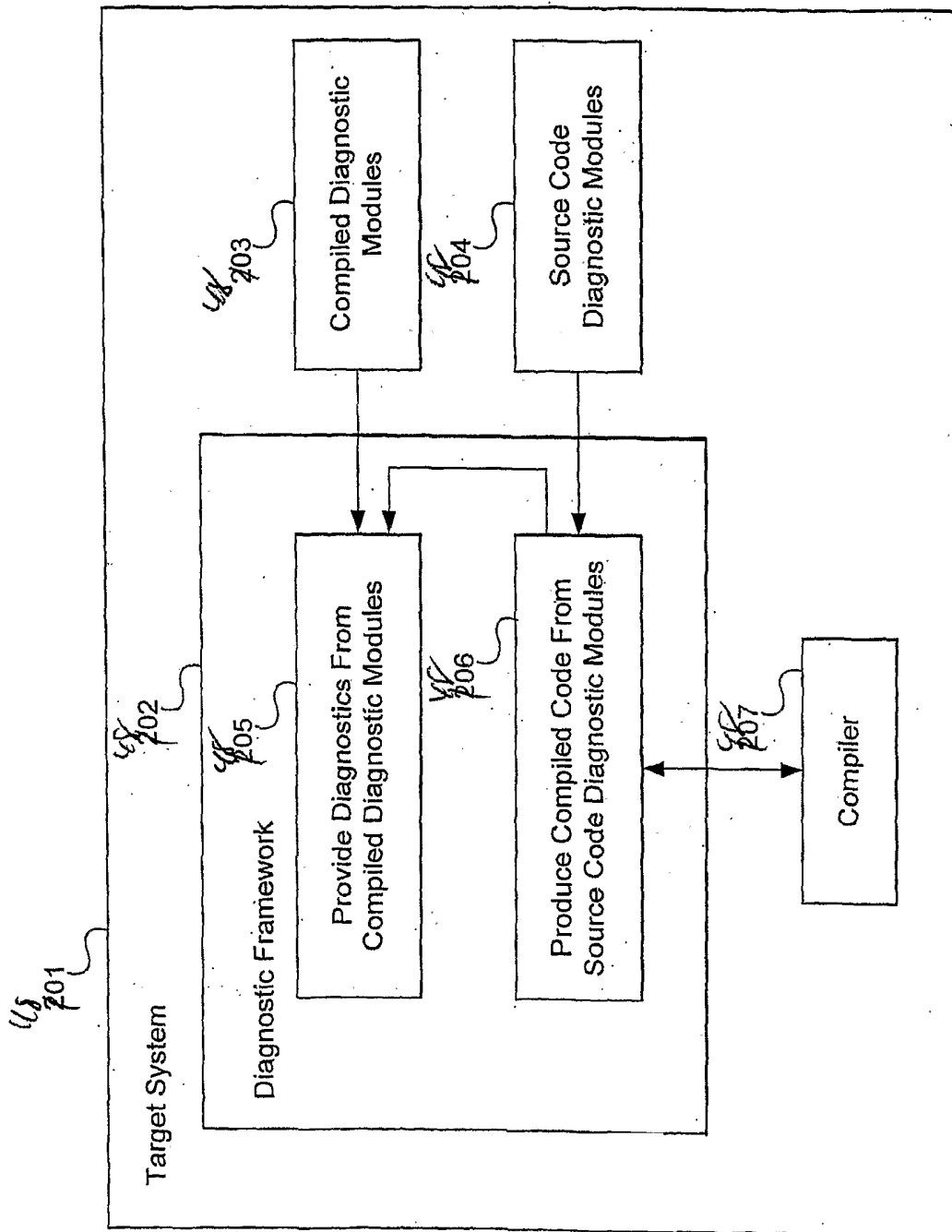


FIG. 2A 48

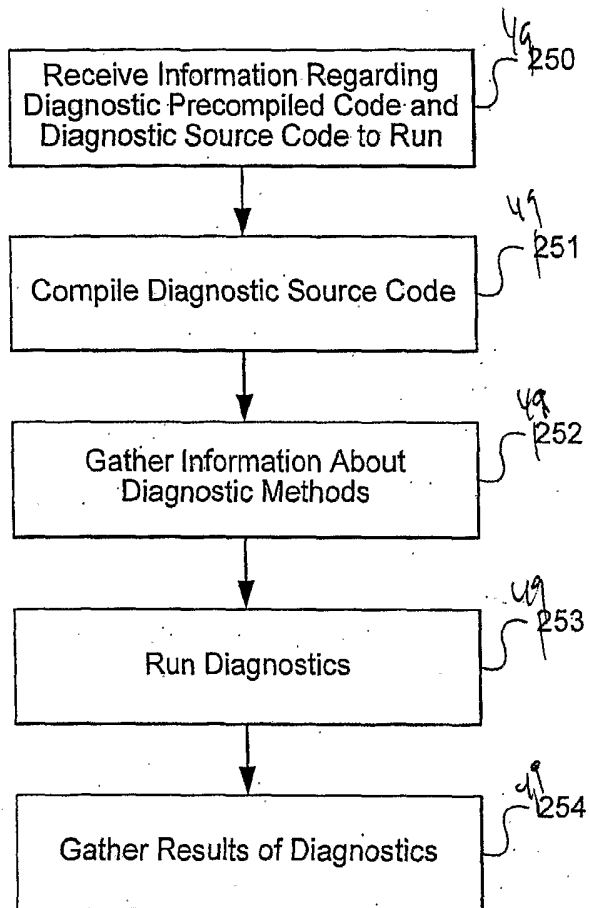


FIG. 2B 49

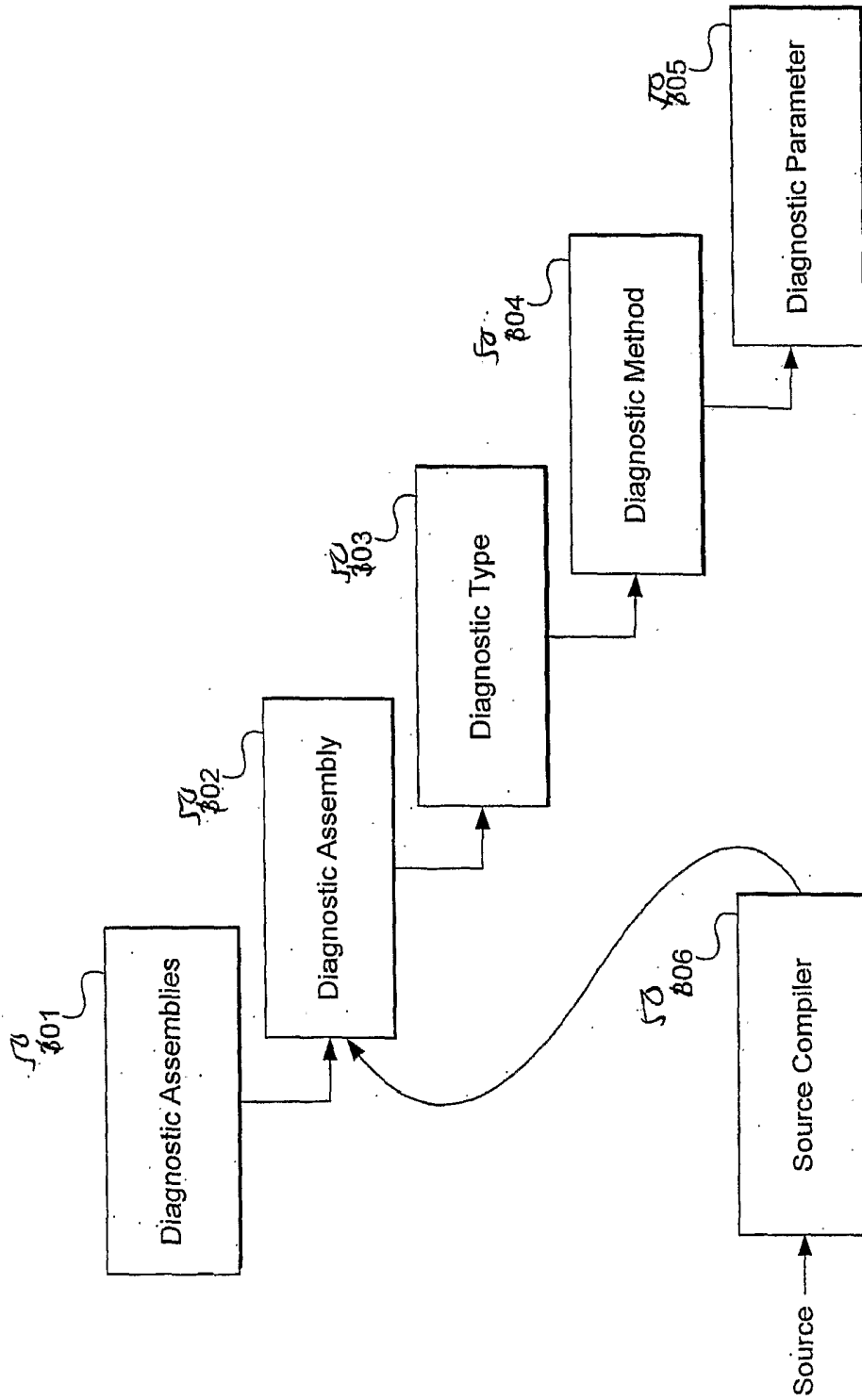


FIG. 3



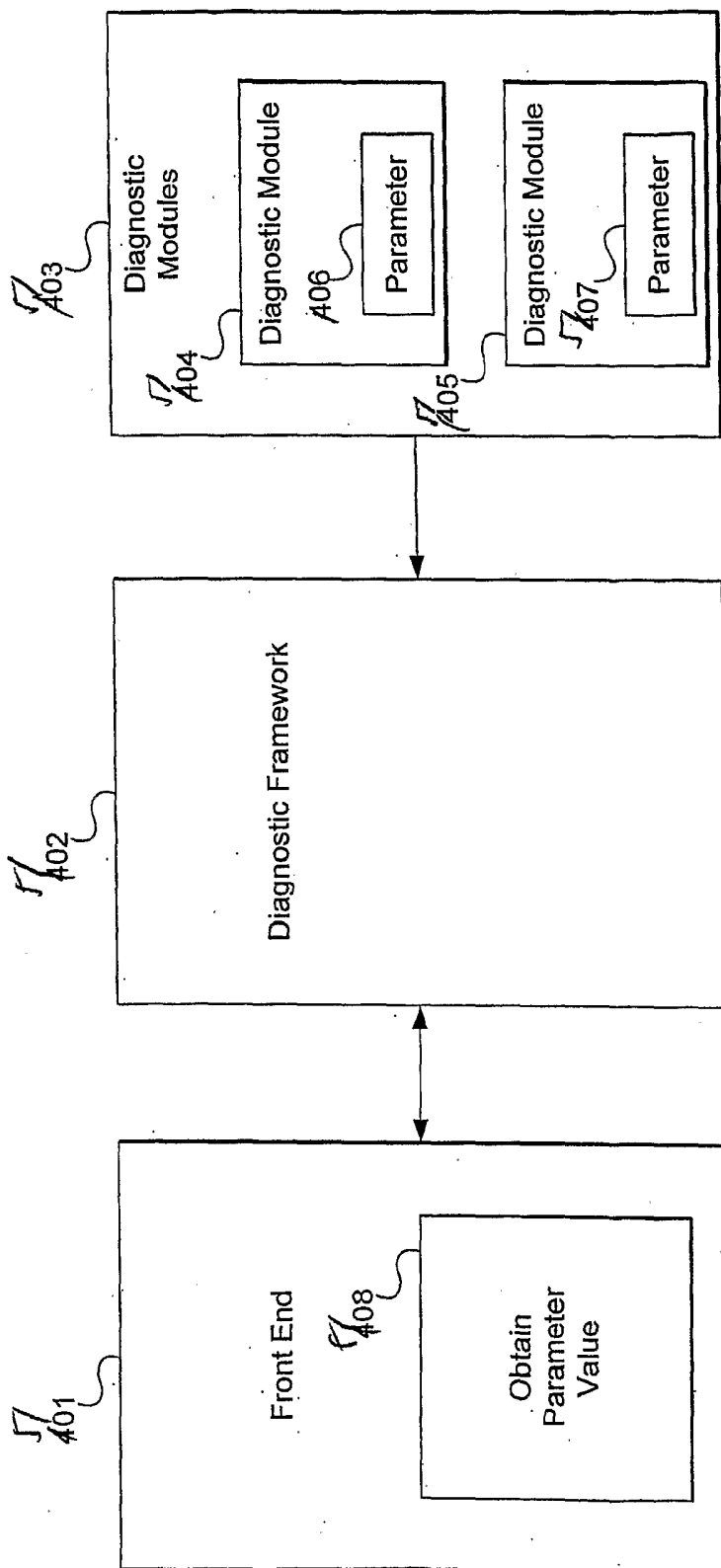


FIG. 4A

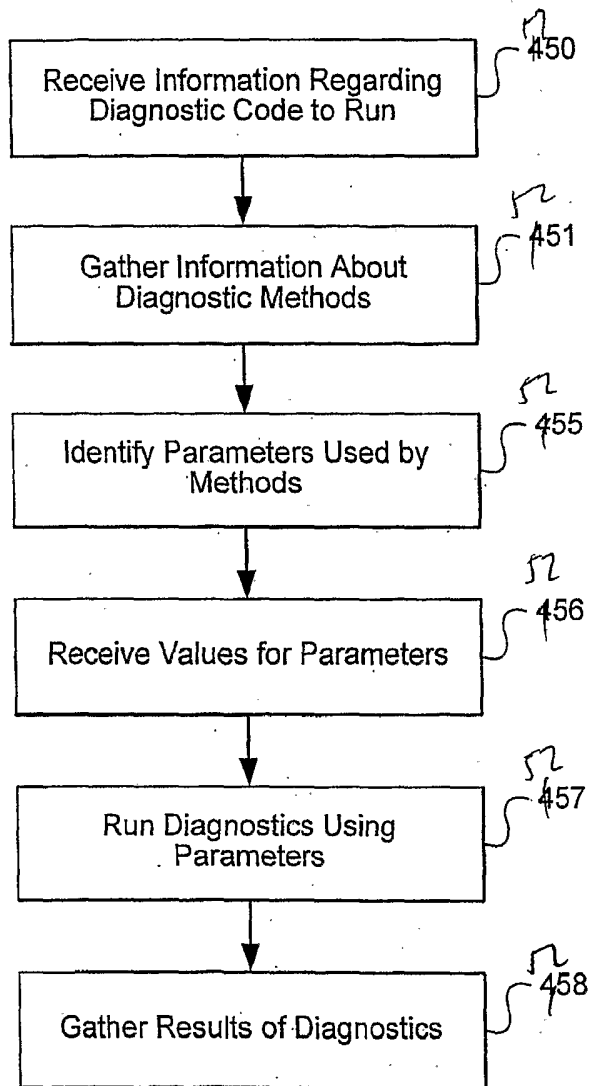


FIG. 4B  
52

52/56

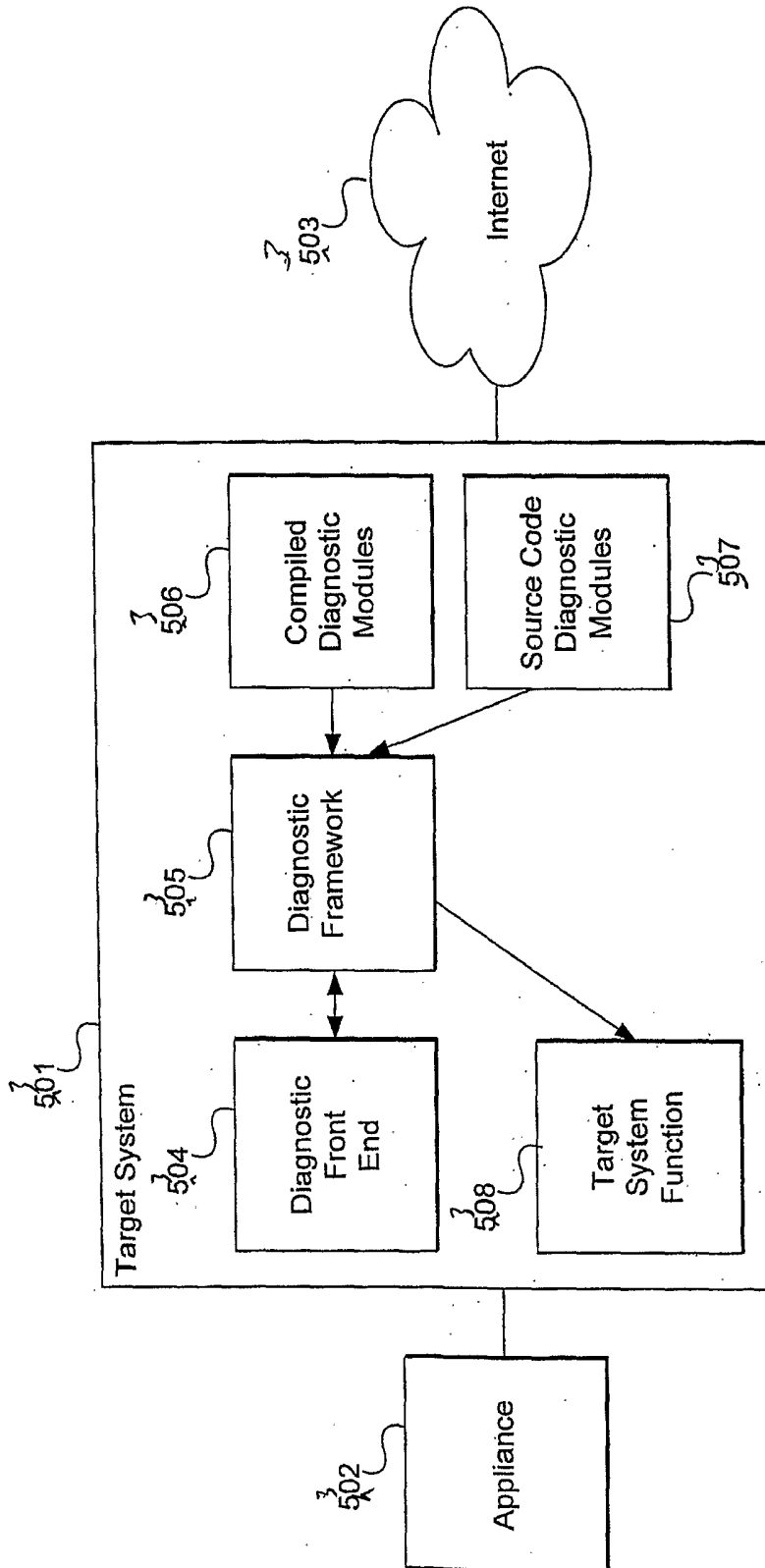


FIG. 5

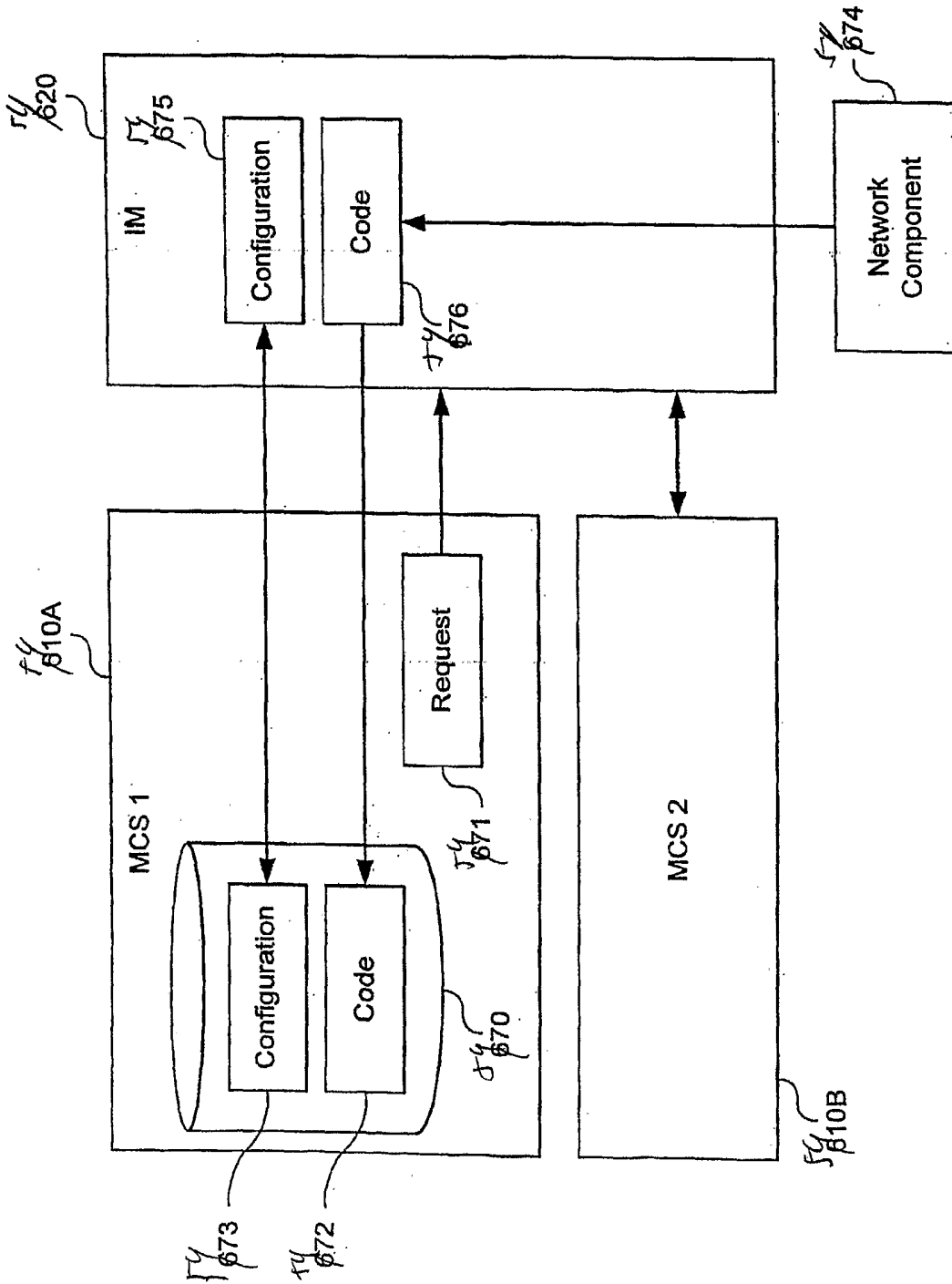


FIG. 6B  
54

54/56

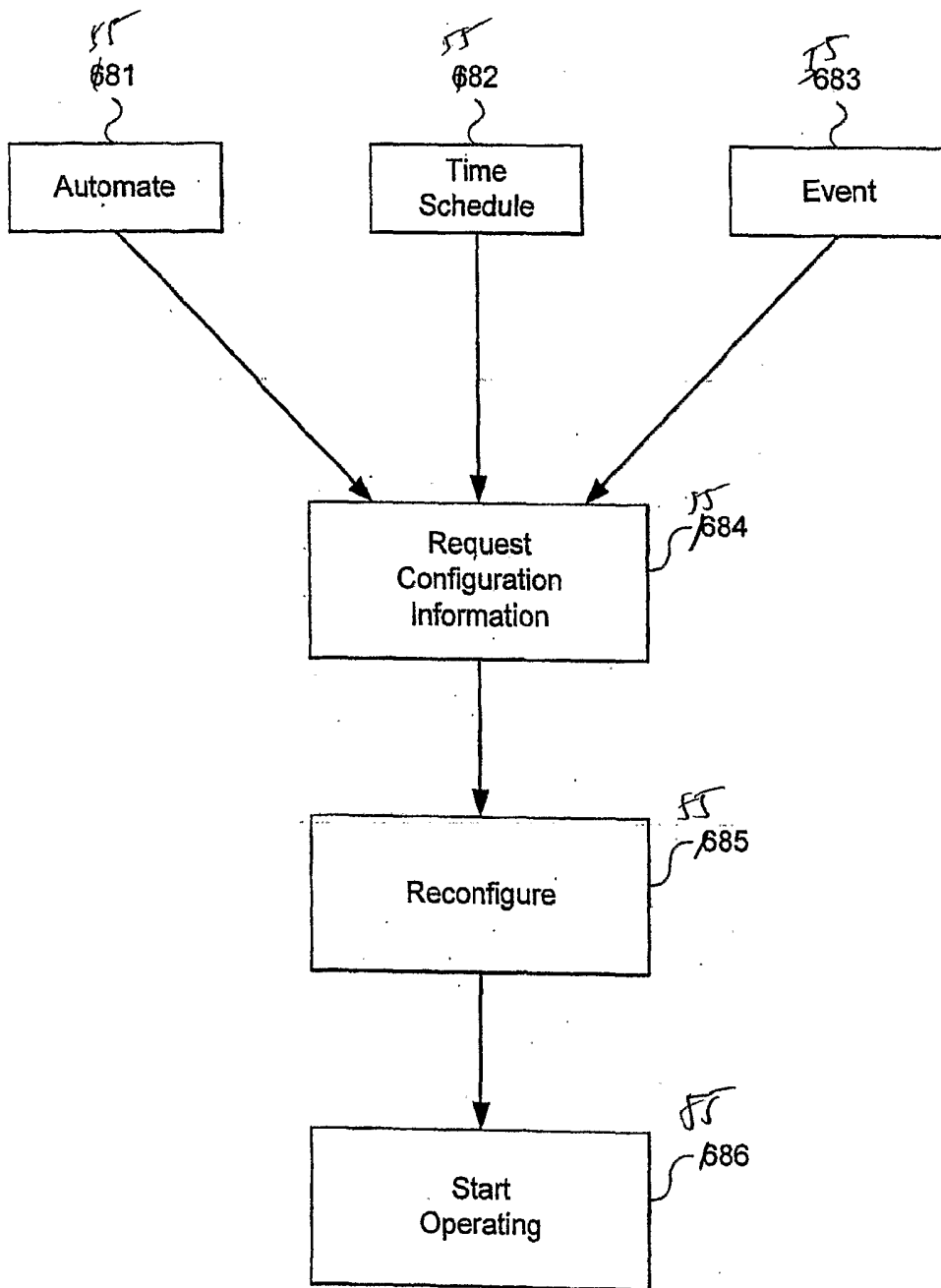


FIG. 6C 55

55/56

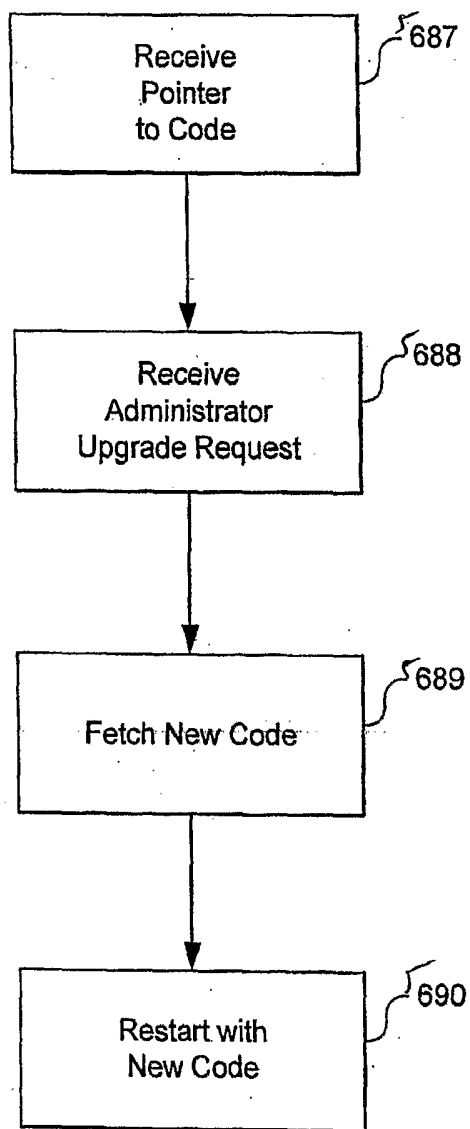


FIG. 6D *56*

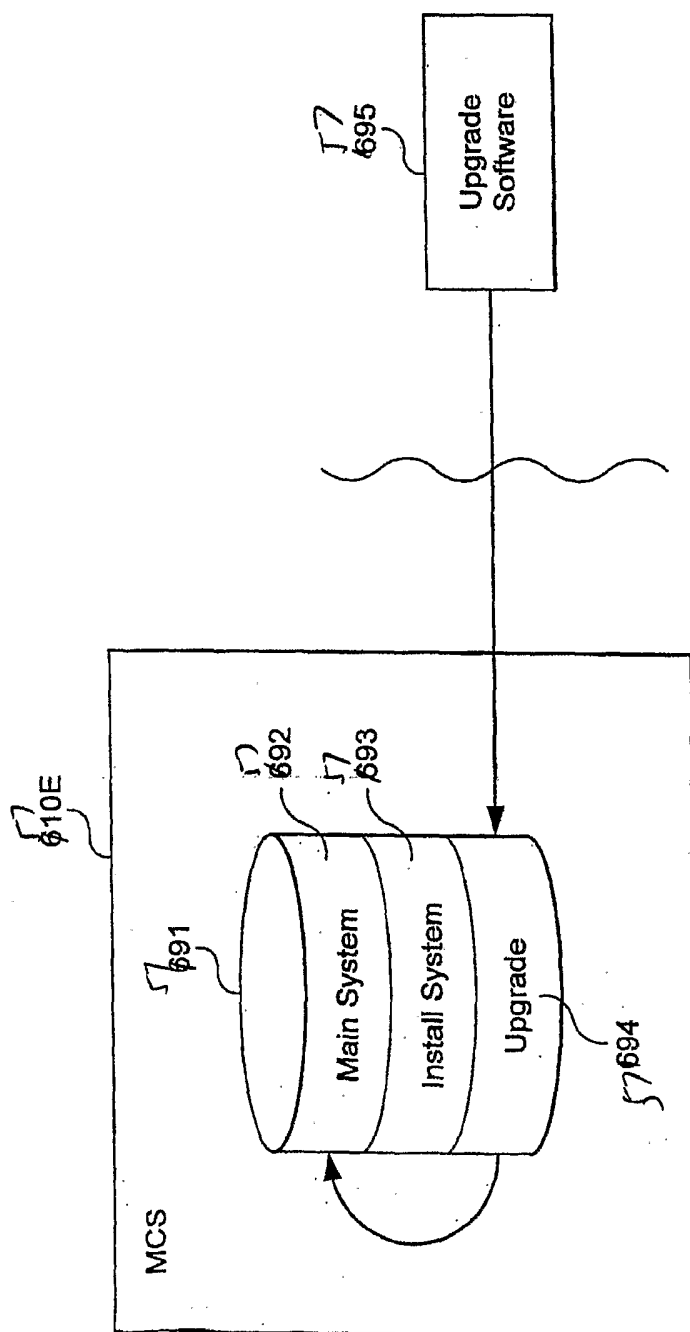


FIG. 6E