'117 Claims	Exemplary Disclosures
	"The intention is that any user messages can be used, but in following mainly short messages will be referred to as an example. In addition to being fast to send, the advantages of a short message service can be utilised, such as not reserving the voice channel. Application related information can either be pre-stored in a terminal memory (permanent memory) or a user may store the application related information in a terminal memory (cache memory) by contacting a server by means of a terminal. Depending on the application, the user may enter user input or modify the information in the applications. In another application the information relating to an application may be sent by a service provider and the information may be such that it is not possible for the user to modify it, only to request the service provider to modify it. The information readily printed in the application can also be transmitted. An application type identifier or header is preferably added to the transmission, so that a receiving terminal identifies the short message as not an ordinary short message, but as a short message containing information relating to and intended for a specific application. The identifier can be a code in an address or a control field of the short message, or it can be a code in the message part of the short message. Because it has been realised that the short message service, already existing in the mobile station system, can be utilised for sending information on applications, the advantages are, e.g., that there is no need to establish an off-line server for sending the application related information, such as, for example, in the system presented in Publication WO 94/23394. Especially advantageous is that one and the same server, i.e. the SMS server (the Short Message Service Centre SM-SC) can be used for sending and forwarding information relating to any application, so there is no need to have separate servers for each application. The SMS server will forward any short message and the terminal will address the information to the correct application according to the header or identifier in the message. And since a short message can be sent simultaneously with a circuit-coupled connection, the sending of the application related information does not engage the terminal's communication line, e.g., in case of a simultaneously incoming call. A network like the GSM network is

'117 Claims	Exemplary Disclosures
	maintained by several operators and usually each operator has at least one SMS server of their own. In this case naturally any SMS server or several servers may be used for the invention." Isomursu at 5:33-51: "Additionally short messages may be sent by a personal computer PC. In this case the mobile services switching centre MSC has a connection to a server GTW (Gateway), which has a connection to e.g. Internet. Thus a computer PC having a connection to Internet (or directly to the server GTW as shown by the broken line) may fetch a WWW (World Wide Web) page from Internet, which physically can be found e.g. from the server GTW. Optionally a service provider or operator may have a separate server GTW having a connection to the mobile services switching centre MSC for sending short messages or other user messages. When using such a WWW page for sending a short message the user inputs the connection information (e.g. phone number) of the receiving terminal MS2 and the message to be sent, whereby the message is transferred via Internet and the server GTW to the mobile services switching centre MSC and further to the short message service centre SM-SC from which the message is directed to the receiving terminal MS2 via the mobile network." Isomursu at 9:4-46: "By means of an application 'Service DTMF Commands', information can be received, e.g., from a network operator in order to utilise services provided by the operator. An application identifier can be 'Service DTMF Commands', and it may have fields for a sender's name, a DTMF command, and an explanation field. The command can be, e.g., a password, a user identifier, or something else related to the services provided by the operator. A user may use the received command, e.g., a password, when utilising the offered services, in which case, the user does not have to input the password through the keyboard, because the password can be obtained

'117 Claims	Exemplary Disclosures
	directly from the application (or the information in the application) in question. Other applications to which information may be received from a network operator are any applications which require some settings in the terminal before an application can be used. An example is an 'Internet Access Point' application which contains information necessary for the terminal to use a WWW browser for example. This information may be provider name, phone number, modem initialisation information, server information, IP address (Internet Protocol). Another example of an application that needs some settings is a Fax Box application. A faxbox is a service implemented e.g. in a server, which receives all faxes for a specific user. The user can then retrieve the faxes from the faxbox. For retrieving faxes the terminal needs to have the contact information to the server. This could be implemented so that when the terminal has received a new fax to the faxbox, the terminal receives a notification in a user message of a received fax, but additionally the user message would include a UID information (Unique Identifier), i.e. contact information to the faxbox server and other necessary information, such as filename of the fax and a password needed. The Fax Box application could function either manually, i.e. by user execution, or automatically so that the terminal contacts the faxbox server automatically for retrieving a received fax according to the identifier information contained in the notifying user message. The contact can be made as fax call to retrieve the fax. Optionally it could be a user message sending the identifier information back to the server and additionally the fax connection number of the terminal so that the server then can contact the terminal for sending the received fax to it via a fax call." Isomursu at 14:47-15:6: "An example of creating a menu for ringing tones is disclosed in FIG. 10 as a sequence of displays to illustrate what the user sees on the display. The command "NEW RINGING TONES" is sent in a user message to a server of a ringing tone service provider in order to request for latest ringing tones. As a response the server sends a user message containing information relating to the Menu applications for

'117 Claims	Exemplary Disclosures
	creating a menu, from which the user can select a new ringing tone. The user selects the desirable ringing tone from the menu (selects ring tone 'Popcorn'). The selection activates a user message to be sent to the server indicating the desired ring tone. After a while the terminal receives the ring tone from the server. A received ringing tone is indicated to the user using the "RINGING TONE RECEIVED" notification. The user can accept or reject the ring tone. Once the user has given acceptance, the selection menu with the options "Save" and "Playback" displays. If the user selects the Save-option the received ringing tone is saved to the phone and it appears to a ringing tone options menu. The saving is indicated to the user by displaying a "SAVED" confirmation note after which the phone goes to the idle state. If the user selects the Playback—option the received ringing tone is played to the user and after that the original selection list displays again. If the user gives a rejection of the new ring tone, the received ringing tone is discarded, the selection list is removed from the screen, and the phone goes to the idle state." Isomursu at 18:60-19:39: "Sending information of a service page in a short message according to the present invention saves considerably the power consumption of the terminal and, thus, prolongs its useful life, which is an important objective in battery-operated devices. In addition, savings are made in phone call expenses, when query information can be sent together with the short message. The whole system has been illustrated in more detail in FIG. 8. A terminal according to the present invention has been presented with reference numeral 1, by means of which a communications link (reference 101) to the Internet INT can be established. The invention could also be implemented by means of a device having means, according to the present invention, for processing information on an application into a short message, which is sent through an ordinary mobile station by coupling the device to the mobile station. Such a device could be, e.g., a computer. To simplify matters, only a communicator will be discussed here as an example of a terminal according to the invention. When the communications link

'117 Claims	Exemplary Disclosures
	is established to the Internet by means of a **communicator** 1, a service page of a provider of services can be downloaded (reference 102) from the Internet into its memory and user interface. By means of solutions known for prior art, the user, after having filled in the page, has sent the service page by means of the **communicator** 1 to a server SERV of the provider of services, i.e., along **route** 101–101′. This means that the communications link to the Internet should be open for transmission. In a system according to the present invention, the information on the service page is sent (reference 103) in a short message to a short message service centre SM-SC, which sends it further (reference 104) to the server SERV of the provider of services. The transmission of the service page through the Internet, according to prior art, lasts considerably longer than the transmission of a short message and, thus, due to the present invention, a shorter transmission time is achieved, thereby, effecting a saving in power, since, in the communicator, transmission and reception, in particular, consume a lot of power compared to other functions. In addition, in the solution according to prior art, a circuit-coupled connection is in use in transmission, in which case the communicator is engaged during transmission. On the other hand, the sending of a short message does not engage the circuit-coupled connection, and an additional advantage is that the short message service centre SM-SC will send the message to a receiver later, if the telephone number of the provider of services happens to be unreachable during the transmission of the message." Isomursu at 20:48-21:20: "In FIG. 9, there is a block diagram of the implementation of a terminal according to the present invention, which in the figure is shown as a device also having means for transmission over the air such as a mobile phone. However a similar implementation can be used for a terminal device of a service provider which usually does not have means for direct radio frequency transmission but is connected to such means (e.g. a base station) over the network as is shown in FIG. 2 (e.g. the personal computer PC or server GTW). The terminal is preferably a mobile phone or communicator, which

'117 Claims	Exemplary Disclosures
	has circuits and a user interface that enable the processing of different applications. The terminal 1 in FIG. 9 comprises, for communication using radio communication, a radio unit RU (the reference has not been marked in the figure), which comprises a transmitter branch 2, known from an ordinary mobile station, (comprising blocks implementing coding, interleaving, ciphering, modulating, and transmitting), a receiving branch 3 (comprising receiving, de-modulating, de-ciphering, de-interleaving, and implementing blocks) and, for transmission using radio communication, a duplex filter 4 that distinguishes between a received and transmitted message, as well as an antenna 5. The terminal has a main control circuit 6 that controls its operation. Furthermore, the main control circuit 6 comprises still a RU controller 7 that carries out control functions of an ordinary mobile station. In addition, the terminal main control circuit 6 comprises blocks 8–12 for carrying out the functions of a data processing unit of the terminal and for sending application related information as a short message according to the present invention. Thus, the blocks 8–12 can be said to form a data processing unit DU of the terminal. The controls of the radio unit RU and the terminal's data processing unit DU do not have to be integrated into the main control circuit but, instead, they could also be implemented apart from each other, so that the RU control circuit 7 is on the radio unit's side, and on the data processing unit's side, there is the DU processor 8, which is in connection with the RU control circuit 7 for establishing communication between the radio unit and the data processing unit." Okonnen at [0011]: "In an embodiment according to the present invention, the network may further comprise at least one of an update server, and a plurality of generators. The generators may be adapted to generate updates able to be processed by at least one provisioned update agent in electronic device. The update server may be adapted to

'117 Claims	Exemplary Disclosures
	store updates accessible by the plurality of servers. The plurality of servers may be communicatively coupled to the update server." Okonnen at [0020]: "In an embodiment according to the present invention, the network may further comprise update agent provisioning information stored in the electronic device. The update agent provisioning information may comprise a device server URL, an index of provisioned update agents, a security key, and electronic device related information. The device server URL may provide references to servers hosting updates to be downloaded. The updates may be compatible with update agents currently available and provisioned in the electronic device." Okonnen at [0047]-[0048]: "FIG. 1 is a block diagram an electronic device network 105 according to an embodiment of the present invention. In the electronic device network illustrated in FIG. 1, an electronic device, for example mobile handset 107, may be communicatively coupled to at least one of a plurality of device servers, such as device server A 135 and device server B 137, for example. "The electronic device, for example, mobile handset 107, may be capable of employing one of several available update agents, for example, update agent A 113 and update agent B 115, resident in the electronic device to update software and/or firmware in the electronic device. The electronic device, for example, mobile handset 107, may also capable of provisioning the available update agents, such as for example, update agent A 113 and update agent B 115, with the necessary parameters and data needed to facilitate the respective update operations." Okonnen at [0066]:

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 154 of 442

'117 Claims	Exemplary Disclosures
	"In an embodiment according to the present invention, a separate key may be employed to authenticate updates by a download agent/download client, (e.g., a browser), and by the update agent. In an embodiment according to the present invention, the security key(s) 213 may also incorporate keys employed for secure communications with external systems, such as keys used for encryption and decryption of data and messages during communications with external systems, such as a device server." *See also* Okonnen at Fig. 1. Hashimoto at [0038]-[0039]: "Fig. 1 shows a multi-party application protection environment to which the present invention is to be applied. "A software vendor 121 produces a program 122, and distributes it to a microprocessor-A 103 provided inside a target system 102 of a system user 112. In order to protect the program to be distributed, the plaintext program 122 is encrypted." Hashimoto at [0040]: "More specifically, the program formed by execution codes and data such as initialization data is encrypted by using a program key selected by the software vendor 121, and the program key used for the encryption is encrypted by using a public key-A of the processor-A of the target system 102 to generate a distribution key." *See also* Hashimoto at Fig. 1. Staack at [0021]:

'117 Claims	Exemplary Disclosures
	"FIG. 1 shows a first embodiment of the invention and illustrates a case where a message is to be sent from a first network element 1 (user A) to a second network element 3 (user B). The network elements 1, 3 are, in the 25 present embodiment, client or user equipments such as terminals. In the present example, the network element 1 (user A) is an equipment trying to send a message (e.g. "MESSAGE user b@sonera.com" addressed to user b@sonera.com) to the receiving element 3 (user B) which is presently out 30 of reach, e.g. switched-off, busy, or located in a nonsupported area, or the like. The connection request of network element 1 is handled by a network element 2 which may be a server (such as a proxy server) which provide e.g. CSCF (Call Server Control Function), and/or is a home location server which contains a database storing information on the present locations of network element 3 and further network elements, reachability thereof, and the like." *See also* Staack at Figs. 1, 2. Fan at 10: The first terminal provided by the embodiment of the present invention further includes an application module, configured to perform corresponding service processing according to the Push message. Based on this situation, the Push message processing module includes: a processing execution module and an analysis control module. Fan at 12: The matching conditions in the set Push message control policy may include the following: user identifier, maximum message length threshold, service control, content control, and control of the trusted PI. The following five Push message control strategies are introduced separately. Fan at 13:

'117 Claims	Exemplary Disclosures
	First, the user identifier The purpose of setting the matching condition is to determine, according to the user identifier, that the terminal to which the Push message is to be sent is the setting terminal of the Push message control policy when the Push sending agent performs the Push message control. Fan at 13: Third, business control. The control of the service received by the terminal is mainly performed by setting a service identifier corresponding to the service type that the terminal allows or disallows or is not currently allowed to receive. The Push sending proxy may send the Push message sent by the PI to the Push receiving proxy in the terminal by using a SIP message, and the Acceptable Contact Head field Accept-Contact header in the SIP message indicates the service type. Fan at 13: If the Push message control policy is stored on the terminal, a list of service types that the terminal does not want or does not want to receive may be defined, or a service list that the terminal wishes to receive may be defined, and the terminal completes control of the service type, and selects a pair. Push messages are discarded or sent to other terminals for reception, or temporarily stored. *See also* Assmann at Fig. 3. White at [0104]: "As described below, transmission rules may be provided to govern the delivery of messages from the server to the client as well as from client and server. Global properties may come into play in the evaluation of these transmission rules. The global properties can be set on a particular mobile device and subsequently moved to the server. This enables the server to make decisions on what messages should

'117 Claims	Exemplary Disclosures
	be sent to particular client device based on settings made at the particular client device (e.g., what types of messages are of interest to client). For example, when a client determines what network it is on, it can store that information in the global properties table and provide the information up to the server during the synchronization process. On the basis of this information, the server may apply server-side transmission rules for selecting messages to be downloaded to the client." Holbrook at cl. 1: "1. A method for the delivery of an incoming message in a messaging system comprising a central server of said messaging system, a plurality of messaging nodes, a plurality of user accounts with distinct messaging address identifiers, and a communications means for establishing a first communications link between each of said messaging nodes and said central server; said method comprising the steps of a) maintaining an association table, associating user accounts with at least one messaging node; b) identifying at least one user account indicated as a recipient of said incoming message, determined from a header of said incoming message; c) determining membership of a messaging node within a primary messaging zone, corresponding to a subset of said messaging nodes associated in said association table with said recipient user account; d) transmitting said incoming message across said first communications link to said messaging node, and

'117 Claims	Exemplary Disclosures
	e) buffering said incoming message at said messaging node, prior to a user request to collect new messages from said messaging system, whereby said incoming message is buffered at messaging nodes from which recipients may subsequently request the collection of said incoming message." Cooley at ¶[0013]: "FIG. 1 shows one embodiment of a system, such as system 10, in which the concepts of the invention can be practiced. In FIG. 1 there is shown a link from a user device, such as cellular telephone 15 to website 11. This link extends from the user device through a carrier network, such as network 14 and via service center 12 to the desired website. The message returned from website 11 contains an HTTP request against service center 12. The service center then queues a Short Message Service (SMS) message back to device 15 through an SMS broker, such as broker 13. Broker 13 then routes an SMS message through any carrier network, such as network 14, which need not be the same network that was used for the originating message from the device." Cooley at ¶[0015]: "When the incoming text message arrives at the user's device it is intercepted, under control of the device processor and an application, such as application 31 (as will be discussed), running on the device. This is possible because application 31 is registered with the device operating system when it was loaded on the device. Application 31 then routes the incoming message to the proper specific application in accordance with the data contained in the text message. Thus, in the situation with the flight reservation to Las Vegas (as discussed above), information pertaining to that flight is routed to a specific application, such as to air travel application 22-1 (shown in FIG. 2B), which application is dedicated to air travel data pertaining to the device user."

'117 Claims	Exemplary Disclosures
	Cooley at ¶[0017]: "When the user touches tile 23-1 all the flights that the user has booked on, for example, AMERICAN™ airlines (AA) are displayed. Touching tile 24-1 then yields the screen shown in FIG. 2E which then displays the information that has been received from an external source. In this case this information would have come typically from the AA server in the manner discussed above. There could be any amount of information and tiles (or layers) to this information, all controlled by application 30 residing on the user's device with the updated information coming, from time to time, via messages from various external locations. Note that in FIG. 2F the gate and status of the flight have been updated. This update could have come as a result of a request from the user or could have come as a text message from the airline as part of an update service for customers. In any event, the message would have gone first to service center 12 and then via SMS broker 13 to the user's device where application 30 would have taken the data and placed it in the proper tile." Cooley at ¶[0024]: "FIG. 5 illustrates an example computer system 500 adapted according to one embodiment of the present invention. That is, computer system 500 comprises an example system on which embodiments of the present invention may be implemented (such as servers 11, 12, broker 13, and handheld device 15 of the example implementation of FIG. 1). Central processing unit (CPU) 501 is coupled to system bus 502. CPU 501 may be any general purpose or specialized purpose CPU. However, the present invention is not restricted by the architecture of CPU 501 as long as CPU 501 supports the inventive operations as described herein. CPU 501 may execute the various logical instructions according to embodiments of the present invention. For example, one or more CPUs, such

'117 Claims	Exemplary Disclosures
	as CPU 501, may execute machine-level instructions according to the exemplary operational flows described above in conjunction with FIG. 4." Cooley at ¶[0028]: While FIG. 5 shows a purpose computer, it should be noted that the exact configuration of a portion of a system according to various embodiments may be slightly different. For example, handheld devices according to one or more embodiments may be any kind of processor-based device, such as a cell phone, a Personal Digital Assistant, a specialized device, and/or the like. Additionally, servers (e.g., servers 11, 12 of FIG. 1) and the SMS broker according to one or more embodiments may be any kind of processor-based device capable of sending messages (e.g., email messages, SMS messages, MMS messages), such as a personal computer, a server-type computer, and the like. Moreover, embodiments of the present invention may be implemented on application specific integrated circuits (ASICs) or very large scale integrated (VLSI) circuits. In fact, persons of ordinary skill in the art may utilize any number of suitable structures capable of executing logical operations according to the embodiments of the present invention." Bobo at 7:39-50: "After the user gains access to the mailbox at step 72, the user can request information stored within the MSDS 10. The MSDS 10 receives the request at step 76 and, at step 78, determines whether the information exists. As is common practice, the MSDS 10 also determines the validity of the request at step 78. The request from the user will include the mailbox number for the user, the message identifier, display preferences, and, if the message is a facsimile message, a page identifier. If for any reason the request is invalid, such as when a hacker is

'117 Claims	Exemplary Disclosures
	attempting to gain access to privileged information, the request for the information will be terminated." Satake at ¶[0136]: "The system shown in FIG. 1 is formed by a stock price information server 1, an SMS server 6 that transmits an SMS message from the stock price information server 1 to a user, a user terminal 7 serving as a mobile communication terminal belonging to the user, and a mobile exchange system 8 that relays communication between the stock price information server 1, the SMS server 6, and the user terminal 7. The mobile exchange system 8 and the user terminal 7 are connected by two transmission paths, namely, a traffic channel that is used for data communication and a paging channel that is used for SMS (i.e., PUSH messages) communication." *See also* Satake at Fig. 1.
[1d] the network message server to generate corresponding Internet data messages based on the requests, each such message containing at least one application identifier for an indicated application and application data corresponding to one of the requests, and	The references below disclose and/or render obvious this limitation. For example, see the following passages and/or figures, as well as related disclosures: Huskins at 4:10-19: "The data management session control server 21 transmits messages that include one or more distinct service identifiers (e.g., service__id=0x03) . The server 21 creates a DMSC message data structure and passes information and data to the unidirectional messaging server 22 such as application-private data and data size, unique client identifier, a group flag which is a Boolean variable specifying if the client identifier is a group mask or individual client identifier, a service identifier value for the DMSC service and application-private message identifier. " Huskins at 4:20-33:

'117 Claims	Exemplary Disclosures
	"In accordance with one embodiment of the present invention, the server 10 may implement a unidirectional messaging system. In a unidirectional messaging system, the server 10 may transmit messages to a plurality of clients that are unable to respond in any way. One example of such a network is a direct-to-home (DTH) broadcast network that may be compliant with the DVB protocol. The network may use a connection oriented communication protocol or a real time connectionless communication protocol as two examples. There are many applications of unidirectional messaging from server to client such as instant messaging, command and control and notification and signaling, as examples. In other cases, the network may be a bidirectional network, for example with an Internet Protocol (IP) multicast backbone." Huskins at 6:11-21: "In one embodiment of the present invention, the network management command and control server 16 may provide a specific message identifier that causes its message to be received by a network management session management agent 48 tuned to a particular service identifier. Similarly, messages from the instant messaging server 18 may include a service identifier that cause those messages to be forwarded to the agent 48 in the client 12. Likewise, messages from the data management session control server 21 may have appropriate identifiers that cause them to be shunted to the data management session control agent 45 on the client 12." Huskins at 6:28-7:6: "Turning next to Figure 2, the software 50 on the client 12 initially receives the unidirectional messaging server address and port from the server 10. The client 12 may also be assigned a client identifier as indicated in block 52. Thus, an Internet Protocol multicast system may be established wherein each client has a UMS address and port as well as a unique client identifier, assigned by the server 10. In some embodiments, the server 10 may dynamically adjust addresses and ports as well as client identifiers to enable communication of particular messages, message

'117 Claims	Exemplary Disclosures
	groups or types of messages to particular clients in a dynamic and reconfigurable fashion." Huskins at 7:19-26: "However, if the particular client 12 is the intended recipient, the server 38 checks the message's service identifier and passes the message to the correct agent 44, 46 or 48, as indicated in block 64. The message is then delivered to the appropriate agent 44 as indicated in block 66. In the agent 44, the information is parsed and passed to an appropriate process for handling as indicated in block 68." Huskins at 8:16-30: "Thereafter, the disk management server 20 may create a data structure and pass this data structure to the server 22 as indicated in block 76. The server 22 creates a unidirectional message and assigns a client value, sets a group flag, and copies private data in the private bytes of the message as indicated in block 78. More particularly, a unique client identifier may be assigned. The client identifier may either be a particular preassigned client identifier or, as one example, may be zero when multiple clients are targeted. A group flag may be a Boolean value specifying whether the client identifier is a group mask or a particular identifier. A group mask is an identifier that identifies a subset of the clients 12 on the network. This subset may include a plurality of clients but less than the total number of addressable clients." Huskins at 11:3-10: "Referring to Figure 4, the software 82 on the client 12 for implementing a network management session, initially receives the unidirectional messaging server address and port from the server 10. The client 12 may also be assigned a client identifier as indicated in block 84. Thus, an Internet Protocol multicast system may be established wherein each client has a UMS address and port as well as a unique client identifier, assigned by the server 10. Having receives its address, port and client identifier, the client 12 receiver joins a multicast group and listens for

'117 Claims	Exemplary Disclosures
	messages addressed specifically to it or to any groups that the client 12 belongs to, as indicated in block 86." Huskins at 12:27-33: "The server 21 may provide messages with different message identifiers including a session create message, a session delete message and session update message, as examples. The session delete message simply deletes a previously created session and the session update message causes additional information to be provided for a scheduled session." Chou at 2:3-13: "In accordance with one aspect of the invention, push content is generated in a wireless secure server or other type of server, responsive to information received in the wireless secure server from the enterprise application. The push content is deliverable from the wireless secure server to the mobile client device over a wireless network. The wireless secure server receives from the mobile client device, responsive to the push content, a request for additional information identifiable at least in part by the push content. The additional information is deliverable from the server to the mobile client device over the wireless network." Chou at 2:44-53: "In the above-noted illustrative embodiment, the wireless secure server communicates with the mobile client device utilizing wireless application protocol (WAP). The push content is deliverable from the wireless secure server to the mobile client device via a series connection of a push initiator and a push proxy gateway. The request for additional information is deliverable from the mobile client device to the wireless secure server via a WAP gateway, and the additional information is deliverable from the wireless secure server to the mobile client device via a WAP gateway." Chou at 4:33-40:

'117 Claims	Exemplary Disclosures
	"In the illustrative embodiment, the WSS 106 preferably interacts with the enterprise application 104 through hypertext transfer protocol (HTTP) connections, and utilizes WAP to deliver multimedia service content to mobile client devices. More specifically, with regard to the HTTP connections, the enterprise application initiates HTTP-based communications with the register connector 122, and sends HTTP POST communications to the application connector 124." Rakic at [0061]: "Secure push message client 404 may validate secure push messages that are received from administrator device 104. When secure push message client 404 validates a secure push message, secure push message client 404 may perform a set of tasks that are associated with the secure push message (e.g., set security parameters in client device 102) and send a response to administrator device 106, to indicate that client device 102 has validated the secure push message." Santamaria at [0036]: "In addition, in some embodiments of the invention, the mobile device 160 is required to periodically refresh the connection with the network data access element(s) 150 in order to maintain network data connectivity (e.g. such as NAT routers in the carrier's network, firewalls, or other network elements in the network). For example, since the data context consumes network resources, the network data access element(s) 150 may tear down the data context if it is not actively being used and/or has not been used in a certain amount of time (e.g., the data context is removed and the network resources of that data context are reallocated). It should be understood that if the data context is torn down, the mobile device 160 is not connected with any notification services. Of course, it should also be understood that the data context may be torn down for other reasons besides period of inactivity (e.g., if the location of the mobile device is out of range of the network, if the mobile device 160 is turned off, etc.). In other embodiments of the invention, the data context does not have to be refreshed by the mobile device 160. Typically, the data context timeout interval is less than the notification

'117 Claims	Exemplary Disclosures
	service timeout interval. If the data context is up, a notification service connection message transmitted by the mobile device 160 to a notification service inherently refreshes the connection with the network data access element(s) 150." Santamaria at [0061]: "At time 47, the mobile device 160 is awakened from the sleep mode 516 from an independent event (represented by independent wake 566). There may be many causes of an independent wake, including user interaction (e.g., the user disables sleep mode), receiving an item of interest (e.g., receiving an email message from the push notification service 110), receiving a phone call and/or text message, etc. Once awake, the mobile device determines whether the time is within any notification service connection message transmission window. For example, time 47 is within the window 552 for the push notification service 110 (which starts at time 45 and ends at time 55) and within the window 550 for the pull notification service 130 (which starts at time 42 and ends at time 50). Leveraging the fact that the mobile device is awake and within notification service connection message transmission windows, the mobile device transmits a notification service connection message to the push notification service 110 (represented by the beat 582) to refresh the push notification service connection 110, and transmits a notification service connection message to the pull notification service 130 (represented by the check 580) to check for items of interest. In addition, while not illustrated in FIG. 5 for simplicity purposes, the mobile device 160 resets the notification service connection message transmission timer and reschedules the notification service connection message transmission time and wake for the push notification service 110 and the pull notification service 130. At time 49, the mobile device enters sleep mode 518." Santamaria at [0064]: "Since multiple mobile devices may be accessing the same notification services through the same network (e.g., with reference to FIG. 1, according to one embodiment of the invention the mobile device 160 and the mobile device(s) 170

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 167 of 442

'117 Claims	Exemplary Disclosures
	are at least partially coupled with the same network access data element(s) 150), the global notification service connection message timer is configured so that all of the mobile devices do not transmit a notification service connection message at the same time. According to one embodiment of the invention, these notification service connection message transmission times are cryptographically randomized (and thus spread throughout the day). For example, a unique identifier of each mobile device (e.g., a universally unique identifier (UUID), a Media Access Control (MAC) address, etc.) modded over a given time interval to determine these global notification service connection message transmission times." *See also* Santamaria at Fig. 4. Isomursu at 2:56-3:33: "The intention is that any user messages can be used, but in following mainly short messages will be referred to as an example. In addition to being fast to send, the advantages of a short message service can be utilised, such as not reserving the voice channel. Application related information can either be pre-stored in a terminal memory (permanent memory) or a user may store the application related information in a terminal memory (cache memory) by contacting a server by means of a terminal. Depending on the application, the user may enter user input or modify the information in the applications. In another application the information relating to an application may be sent by a service provider and the information may be such that it is not possible for the user to modify it, only to request the service provider to modify it. The information readily printed in the application can also be transmitted. An application type identifier or header is preferably added to the transmission, so that a receiving terminal identifies the short message as not an ordinary short message, but as a short message containing information relating to and intended for a specific application. The identifier can be a code in an address or a control field of the short message, or it can be a code in the message part of the short message. Because it has been realised that the short message service, already existing in the

'117 Claims	Exemplary Disclosures
	mobile station system, can be utilised for sending information on applications, the advantages are, e.g., that there is no need to establish an off-line server for sending the application related information, such as, for example, in the system presented in Publication WO 94/23394. Especially advantageous is that one and the same server, i.e. the SMS server (the Short Message Service Centre SM-SC) can be used for sending and forwarding information relating to any application, so there is no need to have separate servers for each application. The SMS server will forward any short message and the terminal will address the information to the correct application according to the header or identifier in the message. And since a short message can be sent simultaneously with a circuit-coupled connection, the sending of the application related information does not engage the terminal's communication line, e.g., in case of a simultaneously incoming call. A network like the GSM network is maintained by several operators and usually each operator has at least one SMS server of their own. In this case naturally any SMS server or several servers may be used for the invention." Isomursu at 6:28-59: "In the present invention, using short messages as an example, the transmission of each application related information will be identified by means of a specific code, i.e., an identifier, which enables the receiving terminal to process the received message directly into an application, as specified, containing the received data. The identifier is preferably implemented by using ASCII characters in an information field of the short message transmission frame, i.e., in a field INFO (FIG. 5), which contains the actual short message in characters. Alternatively the identifier may be in the form of some other character or string code, such as bits, since for sending a short message the data is anyway converted into bits. Because the information relating to the applications is transmitted in a short message, it can also be received by means of an ordinary mobile station, which does not support the application service, but is capable of transmitting and/or receiving short messages. By placing

'117 Claims	Exemplary Disclosures
	the application identifier in the field INFO, there is also the advantage that in an ordinary mobile station, which does not support this type of application service, but is capable of transmitting and/or receiving short messages, the application identifier is displayed to a user of the terminal, e.g. a communicator and, hence, the user notices not having received an ordinary short message, but the information relating to a specific application. In addition, a user of this type of ordinary mobile station can also transmit a message, such as mentioned above, by first writing, on the message, the identifier of the application in question in characters, and the rest of the information correctly divided. The reception of such a transmission by means of a terminal, according to the present invention, will produce a fully received application related information record." Okonnen at [0020]: "In an embodiment according to the present invention, the network may further comprise update agent provisioning information stored in the electronic device. The update agent provisioning information may comprise a device server URL, an index of provisioned update agents, a security key, and electronic device related information. The device server URL may provide references to servers hosting updates to be downloaded. The updates may be compatible with update agents currently available and provisioned in the electronic device." Okonnen at [0021]: "In an embodiment according to the present invention, the index of provisioned update agents may provide an index value used to compute an address location of a provisioned update agent. The index of provisioned update agents may be an index to a table containing an address for an update agent in non-volatile memory the electronic device." Okonnen at [0047]-[0048]:

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 170 of 442

'117 Claims	Exemplary Disclosures
	"FIG. 1 is a block diagram an electronic device network 105 according to an embodiment of the present invention. In the electronic device network illustrated in FIG. 1, an electronic device, for example mobile handset 107, may be communicatively coupled to at least one of a plurality of device servers, such as device server A 135 and device server B 137, for example. "The electronic device, for example, mobile handset 107, may be capable of employing one of several available update agents, for example, update agent A 113 and update agent B 115, resident in the electronic device to update software and/or firmware in the electronic device. The electronic device, for example, mobile handset 107, may also capable of provisioning the available update agents, such as for example, update agent A 113 and update agent B 115, with the necessary parameters and data needed to facilitate the respective update operations." Okonnen at [0069]: "FIG. 2B is a block diagram 225 illustrating an update agent table 231 stored in non-volatile memory 111 of an electronic device, for example, mobile handset 107, according to an embodiment of the present invention. The update agent table 231 in the mobile handset 107 may provide references, such as location addresses and other related information, including information regarding update agent provisioning, to the update agents currently available in the mobile handset 107." Okonnen at [0070]-[0071]: "The update agent table 231 may maintain references to a plurality of available updates and associated update agents. When a new update agent is added, such as via an update of firmware/software in the mobile handset 107, an additional update may also be inserted into the mobile handset 107. The additional update may invoke insertion of a corresponding entry in the update agent table 231 and population thereof with associated relevant information, such as the address location of the new update agent, (i.e., where the new update agent is stored in non-

'117 Claims	Exemplary Disclosures
	volatile memory 111). An update package may be downloaded and installed in the mobile handset 107 and may result in removal of an existing update agent or replacement thereof by another. "In an embodiment according to the present invention, the update agent table 231 may comprise an array of addresses for all available update agents in the mobile handset." Okonnen at [0072]-[0073]: "FIG. 2C is a block diagram 255 illustrating an update agent table 261 located in non-volatile memory 111 of an electronic device, for example, mobile handset 107, according to an embodiment of the present invention. The update agent table in the mobile handset 107 may contain references to a plurality of update agents currently available and provisioned in the mobile handset. "The update agent table 261 may maintain, for example, a mapping between update agent names, the update agent corresponding address locations in non-volatile memory, and the type of updates that the update agents are adapted to process, for all available update agents in the mobile handset 107. The update agent table 261 may also contain optional information regarding provisioning status of the update agents, such as a provisioning flag indicating whether the associated update agent has been provisioned or not." Okonnen at [0075]-[0076]: "The type information may be provided as 'file extensions' in an embodiment according to the present invention. For example, the file extension 'dup' may correspond to updates generated to update generators, such as generator A 139 and generator B 141, as illustrated in FIG. 1. The type field in each entry in the update agent table illustrated in FIG. 2C may be a list of file extensions in an embodiment according to the present invention.

'117 Claims	Exemplary Disclosures
	"Other structures and schemas are also contemplated for the update agent table 261. The update agent table 261 may map the name of an update agent to the location where the agent resides in memory. The update agent table 261 may also map the provisioning status of an update agent and the type(s) of updates the update agent is adapted to process. Management of the update agents may be facilitated by the mobile handset 107. New table entries for update agents may be added and entries no longer in use may be deleted, etc. Management of the update agents may be supported by client software, such as a device management client." *See also* Okonnen at Fig. 1. Hashimoto at [0033]-[0034]: "Also, at a time of the task execution interruption, the physical addresses at which the program and the data are arranged are often changed by the system resource management (re-location), and in order to decrypt and normally execute the program encrypted in units of blocks, there is a need for the information to be arranged in a certain address space. For this reason, a virtual address is used in the case of calculating the relative address that is expressed by a difference between the target address and the program start address (offset address). "In this way, the block interchange can be prevented effectively, and the limited physical memory can be utilized efficiently." Hashimoto at [0038]-[0039]: "Fig. 1 shows a multi-party application protection environment to which the present invention is to be applied. "A software vendor 121 produces a program 122, and distributes it to a microprocessor-A 103 provided inside a target system 102 of a system user 112. In order to protect the program to be distributed, the plaintext program 122 is encrypted."

'117 Claims	Exemplary Disclosures
	Hashimoto at [0047]: "The core 902 has a register array 915 and a task switcher 914. The task switcher 914 executes the task switching, that is the saving and the recovery of the context when the interruption occurs. The context is generated in correspondence to the execution of the task, and managed in correspondence to the task. In order to realize this, the task switcher 914 internally has a task state table 921 and a hash calculation unit 922. The task state table 921 maintains the task state in correspondence to the task ID as will be described below. Also, although not shown in the figure, the core 902 issues a notice for change of the context in the register array, a distribution key setting command, etc." Hashimoto at [0048]: "The instruction cache 903 caches the execution codes of the program. In addition to an ordinary cache tag for identifying the address, a tag for identifying a protected task is also attached thereto, and the independence of the application is guaranteed among the internal data of the microprocessor. Note that, although not shown in Fig. 2, the instruction cache 903 internally has an offset table 1207 (see Fig. 9) for describing start addresses (offsets) of the program (execution codes) and the data, as will be described below." *See also* Hashimoto at Fig. 1. Fahrny at [0008]: "According to the present invention, a controller for managing media stream decryption keys is provided. The controller comprises a media decryption engine, a table, and a content key list. The media decryption engine generally receives an encrypted media stream from a headend and presents a decrypted media stream to a receiving device in response to a decryption key. The decryption key is generally a function of a content key. The table may contain a content key and index and a

'117 Claims	Exemplary Disclosures
	plurality of corresponding content keys. Content keys that correspond to a particular encrypted media stream may be selected from the content key list using an index from the content key table, and that is referenced by an identifier received from a headend in connection with the encrypted media stream. The table may optionally (i.e., alternatively) contain initialization vector (IV) values that may be indexed and selected." Fahrny at [0023]: "Keylist: A list of decoder addresses and respective decoder keys in ordered pairs. Keylists may be used by the Uplink Control System (UCS) for generation of authorization messages that are addressed to the diagnostic circuit that is embedded in decoders that are specific to the encoder system." Fahrny at [0037]-[0038]: "For VOD services, a table of keys can be generated and delivered at session setup time. The keys for VOD service may be delivered with synchronization information related to key change as well as other information for short term working key epochs. A VOD Session ID or, alternatively, a Program ID may be used as an index to reference the keys list with the appropriate record of information for the VOD transport decryption. In alternative embodiments of reduced hierarchy key management of the present invention, one-way hashing may be implemented in the protection, selection and processing of the decryption key. "The reduced hierarchy key management of the present invention generally provides a new, more secure, and elegant system and method to deliver content keys for decrypting the program streams in conditional access systems (e.g., Broadcast and Video On Demand applications). The key management of the present invention may dramatically reduce the complexity required to deliver new content keys when a first Entitlement Message has been sent (i.e., presented, transmitted, provided, broadcast, etc.) to each set top box. The reduced hierarchy

'117 Claims	Exemplary Disclosures
	key management system and method of the present invention may be implemented as a portion of a new CAS system. The new CAS system generally provides for the manufacture and distribution of devices that are compatible with infrastructure, regardless of specific content security mechanisms that are used in that infrastructure. The new CAS system may provide far more efficient manufacturing, distribution and operations, and in fact enable new business models, including the retail availability of extremely low cost customer premises equipment (CPE) when compared to conventional approaches." Fahrny at [0041]: "Referring to FIG. 1 a, a diagram illustrating a media decoder (i.e., controller, processor, apparatus, circuit, device, etc.) 100 of the present invention is shown. The decoder 100 may be implemented in connection with a digital media stream distribution system (described in more detail in connection with FIGS. 2(a-b)). The controller 100 is generally implemented as a security processor (or processing system) that provides at least one security feature (e.g., encryption, decryption, authentication, security key management, copy protection, digital rights management, etc.) to at least one digital media input/output stream. The decoder 100 generally has an input 102 that receives at least one signal (e.g., VIDIN and PID), an input/output 104 that receives/presents a signal (e.g., TFHE) as well as additional data, signals, messages, and the like, an input 106 that receives a working key modifier and application function signal (e.g., WKM), and an output 108 that presents a signal (e.g., VIDOUT)." Fahrny at [0043]-[0044]: "The signal PID may be implemented as a program identifier for the respective program that has been selected by a user (e.g., customer, client, viewer, listener, etc.). The signal TFHE may be implemented as at least one entitlement management message (EMM) that is received from and presented to the headend

'117 Claims	Exemplary Disclosures
	via an out-of-band (OOB) transmission. The working key modifier and application function (e.g., a factor, an operator, or a combination of a factor and an operator that was applied to the content key to enhance encryption) WKM is generally combined with a content key to generate a working key that is used to decrypt an encrypted media stream (e.g., the stream VIDIN) to generate a clear output media stream (e.g., the media stream VIDOUT). "The controller 100 generally comprises a media stream transport decryption engine 110, a table 112, a list 114, and a combiner 116. The decoder 100 is generally implemented via at least one processor (e.g., microprocessor, controller, etc.) and at least one memory (e.g., random access memory (RAM), read only memory (ROM), NVROM, flash, EPROM, etc.) where one or more processes, routines, engines, lists, tables, etc. may be stored. The engine 110, the table 112, the list 114, and the combiner 116 are generally implemented within the processor and memory of the decoder 100." Fahrny at [0046]: "The input/output 104 may provide for interfacing that corresponds to (or is related to) entitlement management message (EMM) downloads that are authenticated between the headend (e.g., headend 202, described in more detail in connection with FIGS. 2(a-b)) and the media decoder 100. The input/output 104 may further provide for interfacing that corresponds to downloads to the decoder related to at least one of entitlement structure, content keys lists, IV lists, content key index tables, and digital signatures. Fahrny at [0047]: "The table 112 generally comprises a content key index table. The contents of the table 112 are generally loaded from the headend (e.g., via the input/output 104). During the downloading from the headend, the content key list table may be decrypted and extracted using the respective unit or device key. The table 112 may

'117 Claims	Exemplary Disclosures
	receive the identifier PID via the input 102. The table 112 may present an index (e.g., IND) to the content key list 114 in response to the identifier PID based on the respective value in the table 112 using a content key index contained therein. In an alternative (i.e., optional) example, the table 112 comprises a content key and IV index table." *See also* Fahrny at Fig. 1a. Staack at [0021]: "FIG. 1 shows a first embodiment of the invention and illustrates a case where a message is to be sent from a first network element 1 (user A) to a second network element 3 (user B). The network elements 1, 3 are, in the 25 present embodiment, client or user equipments such as terminals. In the present example, the network element 1 (user A) is an equipment trying to send a message (e.g. "MESSAGE user b@sonera.com" addressed to user b@sonera.com) to the receiving element 3 (user B) which is presently out 30 of reach, e.g. switched-off, busy, or located in a nonsupported area, or the like. The connection request of network element 1 is handled by a network element 2 which may be a server (such as a proxy server) which provide e.g. CSCF (Call Server Control Function), and/or is a home location server which contains a database storing information on the present locations of network element 3 and further network elements, reachability thereof, and the like." *See also* Staack at Figs. 1, 2. Fan at 12: The matching conditions in the set Push message control policy may include the following: user identifier, maximum message length threshold, service control, content control, and control of the trusted PI. The following five Push message control strategies are introduced separately. Fan at 13:

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 178 of 442

'117 Claims	Exemplary Disclosures
	First, the user identifier The purpose of setting the matching condition is to determine, according to the user identifier, that the terminal to which the Push message is to be sent is the setting terminal of the Push message control policy when the Push sending agent performs the Push message control. White at [0068]: "The present invention also includes features for server-initiated send and receive of messages are also provided. Messages can be pushed to client devices, allowing client applications to implement message-driven logic. The system has the ability to detect when messages are available/staged on the server available for delivery to the client. This includes notifying a client that there are messages (and/or data) at the server available for synchronizing down to the client. This notification may cause the client to initiate a synchronization session with the server. The present invention also includes support for "resumable" downloads providing for transmitting large messages or groups of messages to clients in a piecemeal fashion so as to minimize the retransmission of data during network failures." White at [0134]-[0135]: "At step 2 at FIG. 5, the QAnywhere Notifier 544 sends a push notification to the network address of the mobile client with message store "B". At step 3, the MobiLink Listener 527 running on the mobile client with message store "B" (i.e., Message Store 520 as shown at FIG. 5) receives the push notification and signals the QAnywhere Agent 525. The following is the procedure that runs in a thread in the QAnywhere Agent and receives push notifications from the MobiLink Listener. Both push notifications and network status notifications are received on a listening TCP/IP socket. "At step 4 at FIG. 5, the QAnywhere Agent 525 sends a message to the "system" queue in the Message Store 520 indicating that a push notification was received.

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 179 of 442

'117 Claims	Exemplary Disclosures
	The following is the procedure in the QAnywhere Agent 525 that sends push notification as a QAnywhere message into the "system" queue:" White at [0137]: "At step 5 at FIG. 5, the QAnywhere Agent 525 signals the MobiLink synchronization client 527 to perform a synchronization with the server. The signaling mechanism used is the same as that described above in the case of a message that is sent from the mobile client to the server. Next, at step 6, the MobiLink synchronization client 527 performs a data synchronization of the QAnywhere message schema with the MobiLink server 540. The message that is to be received by the client is included in the download portion of the synchronization (i.e., the portion transmitted from the server to the client)." Cooley at ¶[0013]: "FIG. 1 shows one embodiment of a system, such as system 10, in which the concepts of the invention can be practiced. In FIG. 1 there is shown a link from a user device, such as cellular telephone 15 to website 11. This link extends from the user device through a carrier network, such as network 14 and via service center 12 to the desired website. The message returned from website 11 contains an HTTP request against service center 12. The service center then queues a Short Message Service (SMS) message back to device 15 through an SMS broker, such as broker 13. Broker 13 then routes an SMS message through any carrier network, such as network 14, which need not be the same network that was used for the originating message from the device." Cooley at ¶[0019]: "The SMS message contains envelope information which includes the phone number to which it is destined, and in this case would include additional routing information to route it to application 31 so that the message does not get stored in

'117 Claims	Exemplary Disclosures
	the message box so the message will not be text formatted. Thus the system takes advantage of a feature that allows SMS messages to be addressed to internal applications. The application address is added at the service center 12, or if desired could be added by the sender if the application address is made known to the sender. This could, for example, be accomplished by registering the address with the website in advance." Cooley at ¶[0020]: "Another method for doing this would be to run a Simple Mail Transfer Protocol (SMTP) gateway at service center 12. This would allow application 31 to receive incoming messages in email form. Such email messages could then be parsed to see if they contain certain information, such as the tile ID, etc. This approach would benefit from a fixed format email message. The parsing includes filtering the email messages to see if they contain certain information. The filtered information then would be sent to the device by an SMS message. Second, the formatted SMS message could be sent to the service center for delivery to the particular user device." Cooley at ¶[0021]: "FIG. 4 shows one embodiment of a method, such as method 40, for controlling applications resident on a mobile device from a remote location. Process 401 determines if a message is incoming to a particular device. If there is a message available, then process 402 determines if the message contains any code or data that identifies the message as one that is to be delivered to a specific application resident on the device and/or delivered customized in a particular manner. If not, then the message is stored in its normal location, for example in SMS layer 32 (FIG. 3) and delivered without modification." Cooley at ¶[0022]:

'117 Claims	Exemplary Disclosures
	"If the message does contain a "diversion" code or other data that indicates that the message is to be delivered to another location or delivered in a customized manner, then process 404 delivers the incoming message (or other data) accordingly as discussed above, such that a particular user's tile is customized for a particular application. Customizing could, for example, include one or more of the following: appearance, context, display parameters, including audio, visual, application state change. In some situations, a control command is sent to a user's tile, for example, using an SMS transport protocol such that the command controls the display parameters and/or the display message instead of actual text being sent to the tile." Cooley at ¶[0023]: "When implemented via computer-executable instructions, various elements of embodiments of the present invention are in essence the software code defining the operations of such various elements. The executable instructions or software code may be obtained from a readable medium (e.g., a hard drive media, optical media, RAM, EPROM, EEPROM, tape media, cartridge media, flash memory, ROM, memory stick, and/or the like). In fact, readable media can include any medium that can store information." Cooley at ¶[0024]: "FIG. 5 illustrates an example computer system 500 adapted according to one embodiment of the present invention. That is, computer system 500 comprises an example system on which embodiments of the present invention may be implemented (such as servers 11, 12, broker 13, and handheld device 15 of the example implementation of FIG. 1). Central processing unit (CPU) 501 is coupled to system bus 502. CPU 501 may be any general purpose or specialized purpose CPU. However, the present invention is not restricted by the architecture of CPU 501 as long as CPU 501 supports the inventive operations as described herein. CPU 501 may execute the various logical instructions according to

'117 Claims	Exemplary Disclosures
	embodiments of the present invention. For example, one or more CPUs, such as CPU 501, may execute machine-level instructions according to the exemplary operational flows described above in conjunction with FIG. 4." Cooley at ¶[0028]: While FIG. 5 shows a purpose computer, it should be noted that the exact configuration of a portion of a system according to various embodiments may be slightly different. For example, handheld devices according to one or more embodiments may be any kind of processor-based device, such as a cell phone, a Personal Digital Assistant, a specialized device, and/or the like. Additionally, servers (e.g., servers 11, 12 of FIG. 1) and the SMS broker according to one or more embodiments may be any kind of processor-based device capable of sending messages (e.g., email messages, SMS messages, MMS messages), such as a personal computer, a server-type computer, and the like. Moreover, embodiments of the present invention may be implemented on application specific integrated circuits (ASICs) or very large scale integrated (VLSI) circuits. In fact, persons of ordinary skill in the art may utilize any number of suitable structures capable of executing logical operations according to the embodiments of the present invention." Holbrook at [0010]: "The present invention may be used in a data network to increase the probability of desired data being available at remote network nodes where the nodes do not have continuous access to the network. An association table associates user accounts with messaging nodes. A primary messaging zone is generated, indicating the messaging nodes associated with a recipient user account, at which there is a significant probability that the user may request the collection of messages. During intermittent communications sessions with members of said primary messaging zone, incoming messages are delivered to these messaging nodes, where the

'117 Claims	Exemplary Disclosures
	messages are proactively buffered in preparation for a user request to collect messages while a messaging node cannot collect from external sources." Holbrook at [0118]: "To collect new messages, the user must bring portable messaging unit 40 to the immediate physical proximity of a messaging node 20, preferably a member of primary messaging zone 82. A data exchange 90 between messaging node 20 and portable messaging unit 40, as described above in the case of the upload of a message, also triggers the download of any new messages stored on messaging node 20 for the user(s) of portable messaging unit 40. This is not necessarily a current image of all new messages within messaging system 10, because some messages may remain buffered at central server 12 for future delivery to this messaging node 20, or may be buffered at other locations within messaging system 10, such as another messaging node 20 or another portable messaging unit 40." Holbrook at [0174]: "Portable messaging unit 40 uploads any outgoing messages to second CPU 28, which further conveys said outgoing messages to node computer 22 for temporary storage in non-volatile memory. Once successfully stored on node computer 22, a message is sent to portable messaging unit 40 indicating that this message has been recorded. Depending on user settings, portable messaging unit 40 may then automatically delete said outgoing messages from its memory, to free memory for the storage of other data. Alternatively, if the message is retained, the message may be marked as sent, to inform the user and inhibit further sending attempts." Holbrook at cl. 1: "1. A method for the delivery of an incoming message in a messaging system comprising a central server of said messaging system, a plurality of messaging nodes, a plurality of user accounts with distinct messaging address identifiers, and a

'117 Claims	Exemplary Disclosures
	communications means for establishing a first communications link between each of said messaging nodes and said central server; said method comprising the steps of a) maintaining an association table, associating user accounts with at least one messaging node; b) identifying at least one user account indicated as a recipient of said incoming message, determined from a header of said incoming message; c) determining membership of a messaging node within a primary messaging zone, corresponding to a subset of said messaging nodes associated in said association table with said recipient user account; d) transmitting said incoming message across said first communications link to said messaging node, and e) buffering said incoming message at said messaging node, prior to a user request to collect new messages from said messaging system, whereby said incoming message is buffered at messaging nodes from which recipients may subsequently request the collection of said incoming message." Bobo at 7:39-50: "After the user gains access to the mailbox at step 72, the user can request information stored within the MSDS 10. The MSDS 10 receives the request at step 76 and, at step 78, determines whether the information exists. As is common practice, the MSDS 10 also determines the validity of the request at step 78. The request from the user will include the mailbox number for the user, the message identifier, display preferences, and, if the message is a facsimile message, a page identifier. If for any reason the request is invalid, such as when a hacker is

'117 Claims	Exemplary Disclosures
	attempting to gain access to privileged information, the request for the information will be terminated." Bobo at 8:41-52: "For instance, when a facsimile message is received, the MSDS 10, at step 54, will update the total listing of all messages to indicate the newly received message and may additionally generate the HTML files for the newly received facsimile message according to the user's preferences. When the user later requests information on the message at step 76, the HTML information has already been generated and the MSDS 10 may directly send the requested information to the user at step 80. If, on the other hand, the user desires to view the message according to one of the other options, the MSDS 10 will generate the HTML files at step 82 according to that other option at the time of the request." Wakasa at ¶[0061]: "The transmitted message generation section 202 has a function to generate a transmitted message including parameter information and the identification information of an application to be executed by the portable terminal device 100 on the basis of the parameter information. The transmitted message generation section 202 supplies the generated transmitted message to the transmitted message sender section 204." Wakasa at ¶[0062]: "The transmitted message sender section 204 is a communication interface such as a communication line, a communication circuit or a communication device. The transmitted message sender section 204 sends the transmitted message received from the transmitted message generation section 202 to the portable terminal device 100." XMPP at 3-4:

'117 Claims	Exemplary Disclosures
	"2.1.1. Types of Message The 'type' attribute of a message stanza is RECOMMENDED; if included, it specifies the conversational context of the message, thus providing a hint regarding presentation (e.g., in a GUI). If included, the 'type' attribute MUST have one of the following values: ○ chat -- The message is sent in the context of a one-to-one chat conversation. A compliant client SHOULD present the message in an interface enabling one-to-one chat between the two parties, including an appropriate conversation history. ○ error -- An error has occurred related to a previous message sent by the sender (for details regarding stanza error syntax, refer to [XMPP-CORE]). A compliant client SHOULD present an appropriate interface informing the sender of the nature of the error. ○ groupchat -- The message is sent in the context of a multi-user chat environment (similar to that of [IRC]). A compliant client SHOULD present the message in an interface enabling many-to-many chat between the parties, including a roster of parties in the chatroom and an appropriate conversation history. Full definition of XMPP-based groupchat protocols is out of scope for this memo. ○ headline -- The message is probably generated by an automated service that delivers or broadcasts content (news, sports, market information, RSS feeds, etc.). No reply to the message is expected, and a compliant client SHOULD present the message in an interface that appropriately differentiates the message from standalone messages, chat sessions, or groupchat sessions (e.g., by not providing the recipient with the ability to reply). ○ normal -- The message is a single message that is sent outside the context of a one-to-one conversation or groupchat, and to which it is expected that the recipient will reply. A compliant client SHOULD present the message in an interface enabling the recipient to reply, but without a conversation history." *See also* XMPP, 5, 11.

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 187 of 442

'117 Claims	Exemplary Disclosures
[1e] the network message server to transmit each of the generated Internet data messages to the device messaging agent located on the device indicated in the corresponding request, using the corresponding secure Internet data connection for the device indicated in the corresponding request;	The references below disclose and/or render obvious this limitation. For example, see the following passages and/or figures, as well as related disclosures: Huskins at 4:10-19: "The data management session control server 21 transmits messages that include one or more distinct service identifiers (e.g., service__id=0x03) . The server 21 creates a DMSC message data structure and passes information and data to the unidirectional messaging server 22 such as application-private data and data size, unique client identifier, a group flag which is a Boolean variable specifying if the client identifier is a group mask or individual client identifier, a service identifier value for the DMSC service and application-private message identifier. " Huskins at 4:20-33: "In accordance with one embodiment of the present invention, the server 10 may implement a unidirectional messaging system. In a unidirectional messaging system, the server 10 may transmit messages to a plurality of clients that are unable to respond in any way. One example of such a network is a direct-to-home (DTH) broadcast network that may be compliant with the DVB protocol. The network may use a connection oriented communication protocol or a real time connectionless communication protocol as two examples. There are many applications of unidirectional messaging from server to client such as instant messaging, command and control and notification and signaling, as examples. In other cases, the network may be a bidirectional network, for example with an Internet Protocol (IP) multicast backbone." Huskins at 6:11-21: "In one embodiment of the present invention, the network management command and control server 16 may provide a specific message identifier that causes its message to be received by a network management session management agent 48 tuned to a particular service identifier. Similarly, messages from the instant

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 188 of 442

'117 Claims	Exemplary Disclosures
	messaging server 18 may include a service identifier that cause those messages to be forwarded to the agent 48 in the client 12. Likewise, messages from the data management session control server 21 may have appropriate identifiers that cause them to be shunted to the data management session control agent 45 on the client 12." Huskins at 6:28-7:6: "Turning next to Figure 2, the software 50 on the client 12 initially receives the unidirectional messaging server address and port from the server 10. The client 12 may also be assigned a client identifier as indicated in block 52. Thus, an Internet Protocol multicast system may be established wherein each client has a UMS address and port as well as a unique client identifier, assigned by the server 10. In some embodiments, the server 10 may dynamically adjust addresses and ports as well as client identifiers to enable communication of particular messages, message groups or types of messages to particular clients in a dynamic and reconfigurable fashion." Huskins at 7:19-26: "However, if the particular client 12 is the intended recipient, the server 38 checks the message's service identifier and passes the message to the correct agent 44, 46 or 48, as indicated in block 64. The message is then delivered to the appropriate agent 44 as indicated in block 66. In the agent 44, the information is parsed and passed to an appropriate process for handling as indicated in block 68." Huskins at 8:16-30: "Thereafter, the disk management server 20 may create a data structure and pass this data structure to the server 22 as indicated in block 76. The server 22 creates a unidirectional message and assigns a client value, sets a group flag, and copies private data in the private bytes of the message as indicated in block 78. More particularly, a unique client identifier may be assigned. The client identifier may either be a particular preassigned client identifier or, as one example, may be zero

'117 Claims	Exemplary Disclosures
	when multiple clients are targeted. A group flag may be a Boolean value specifying whether the client identifier is a group mask or a particular identifier. A group mask is an identifier that identifies a subset of the clients 12 on the network. This subset may include a plurality of clients but less than the total number of addressable clients." Huskins at 11:3-10: "Referring to Figure 4, the software 82 on the client 12 for implementing a network management session, initially receives the unidirectional messaging server address and port from the server 10. The client 12 may also be assigned a client identifier as indicated in block 84. Thus, an Internet Protocol multicast system may be established wherein each client has a UMS address and port as well as a unique client identifier, assigned by the server 10. Having receives its address, port and client identifier, the client 12 receiver joins a multicast group and listens for messages addressed specifically to it or to any groups that the client 12 belongs to, as indicated in block 86." Huskins at 12:27-33: "The server 21 may provide messages with different message identifiers including a session create message, a session delete message and session update message, as examples. The session delete message simply deletes a previously created session and the session update message causes additional information to be provided for a scheduled session." Chou at 5:31-36: "In accordance with the invention, the system 100 is configured to provide secure service delivery, service monitoring and/or notification over converged networks for mobile users such as a user associated with mobile client 110. This facilitates the implementation of multimodal and multimedia CRM and other enterprise communication services."

'117 Claims	Exemplary Disclosures
	Chou at 6:51-58: "FIG. 1C shows example communication utilizing a WAP push architecture, which allows information to be transmitted to a device without requiring an explicit request from the device. As shown in the figure, a WAP push operation may be accomplished by allowing the PI 116 to transmit push content and delivery instructions to the PPG 114, which then delivers the push content to the WAP mobile client 110 in the manner specified by the delivery instructions." Chou at 7:42-47: "1. A control entity comprising an XML document containing delivery instructions designed for the PPG 114. The control entity is generally viewed as a mandatory entity in the illustrative embodiment. It identifies the target mobile client device and contains delivery instructions such as time delivery restrictions." Rakic at [0061]: "Secure push message client 404 may validate secure push messages that are received from administrator device 104. When secure push message client 404 validates a secure push message, secure push message client 404 may perform a set of tasks that are associated with the secure push message (e.g., set security parameters in client device 102) and send a response to administrator device 106, to indicate that client device 102 has validated the secure push message." Santamaria at [0036]: "In addition, in some embodiments of the invention, the mobile device 160 is required to periodically refresh the connection with the network data access element(s) 150 in order to maintain network data connectivity (e.g. such as NAT routers in the carrier's network, firewalls, or other network elements in the network). For example, since the data context consumes network resources, the network data access element(s) 150 may tear down the data context if it is not actively being used and/or has not been used in a certain amount of time (e.g., the

'117 Claims	Exemplary Disclosures
	data context is removed and the network resources of that data context are reallocated). It should be understood that if the data context is torn down, the mobile device 160 is not connected with any notification services. Of course, it should also be understood that the data context may be torn down for other reasons besides period of inactivity (e.g., if the location of the mobile device is out of range of the network, if the mobile device 160 is turned off, etc.). In other embodiments of the invention, the data context does not have to be refreshed by the mobile device 160. Typically, the data context timeout interval is less than the notification service timeout interval. If the data context is up, a notification service connection message transmitted by the mobile device 160 to a notification service inherently refreshes the connection with the network data access element(s) 150." Santamaria at [0061]: "At time 47, the mobile device 160 is awakened from the sleep mode 516 from an independent event (represented by independent wake 566). There may be many causes of an independent wake, including user interaction (e.g., the user disables sleep mode), receiving an item of interest (e.g., receiving an email message from the push notification service 110), receiving a phone call and/or text message, etc. Once awake, the mobile device determines whether the time is within any notification service connection message transmission window. For example, time 47 is within the window 552 for the push notification service 110 (which starts at time 45 and ends at time 55) and within the window 550 for the pull notification service 130 (which starts at time 42 and ends at time 50). Leveraging the fact that the mobile device is awake and within notification service connection message transmission windows, the mobile device transmits a notification service connection message to the push notification service 110 (represented by the beat 582) to refresh the push notification service connection 110, and transmits a notification service connection message to the pull notification service 130 (represented by the check 580) to check for items of interest. In addition, while not illustrated in FIG. 5 for simplicity purposes, the mobile device 160 resets the notification service connection message transmission timer and reschedules the

'117 Claims	Exemplary Disclosures
	notification service connection message transmission time and wake for the push notification service 110 and the pull notification service 130. At time 49, the mobile device enters sleep mode 518." Santamaria at [0064]: "Since multiple mobile devices may be accessing the same notification services through the same network (e.g., with reference to FIG. 1, according to one embodiment of the invention the mobile device 160 and the mobile device(s) 170 are at least partially coupled with the same network access data element(s) 150), the global notification service connection message timer is configured so that all of the mobile devices do not transmit a notification service connection message at the same time. According to one embodiment of the invention, these notification service connection message transmission times are cryptographically randomized (and thus spread throughout the day). For example, a unique identifier of each mobile device (e.g., a universally unique identifier (UUID), a Media Access Control (MAC) address, etc.) modded over a given time interval to determine these global notification service connection message transmission times." *See also* Santamaria at Fig. 4. Isomursu at 22:36-67: "When address information has been added at a **SMS transmission controller** 10, the message is transferred into an **inbox** 11, which tries to send the message, and which has a buffer wherein the message is stored in case the transmission fails. If the transmission fails, the **inbox** 11 re-tries to send the message. When the **DU controller** 8 notices that the radio unit RU is ready to send the message, the message is transferred to a message **transfer running circuit** 12, which adds to the message information relating to the mobile communications system in question, such as validity information (which indicates to which direction the message is going, i.e., from a mobile station to a message service centre or vice versa), processes the address information into a form required by the mobile communications system, and adds to

'117 Claims	Exemplary Disclosures
	the message the address of the message service centre, as well as the short message identifier (SAPI), and forms from the information to be transmitted, e.g., a digital signal for a **transmitter** 2, and sends the message to the **radio transmitter branch** 2 of the radio unit RU. In the case where the application identifier is placed in bits in the address field ADD, the running **circuit** 12 adds to the message the identifier in question. The **transmitter branch** 2 codes the signal according to the specifications of the mobile communications system, and forms, on the basis of the signal it receives from the running **circuit** 12, the frames to be transmitted, which the transmitter sends using radio communication to the short message service centre SM-SC, wherefrom they are sent further to the receiver (see **FIG. 1**). In the **transmitter branch** 2, the message is processed according to the mobile communications system, e.g., coding, interleaving, ciphering, burst forming, modulating, and transmission." Okonnen at [0047]-[0048]: "FIG. 1 is a block diagram an electronic device network 105 according to an embodiment of the present invention. In the electronic device network illustrated in FIG. 1, an electronic device, for example mobile handset 107, may be communicatively coupled to at least one of a plurality of device servers, such as device server A 135 and device server B 137, for example. "The electronic device, for example, mobile handset 107, may be capable of employing one of several available update agents, for example, update agent A 113 and update agent B 115, resident in the electronic device to update software and/or firmware in the electronic device. The electronic device, for example, mobile handset 107, may also capable of provisioning the available update agents, such as for example, update agent A 113 and update agent B 115, with the necessary parameters and data needed to facilitate the respective update operations." Okonnen at [0070]-[0071]:

'117 Claims	Exemplary Disclosures
	"The update agent table 231 may maintain references to a plurality of available updates and associated update agents. When a new update agent is added, such as via an update of firmware/software in the mobile handset 107, an additional update may also be inserted into the mobile handset 107. The additional update may invoke insertion of a corresponding entry in the update agent table 231 and population thereof with associated relevant information, such as the address location of the new update agent, (i.e., where the new update agent is stored in non-volatile memory 111). An update package may be downloaded and installed in the mobile handset 107 and may result in removal of an existing update agent or replacement thereof by another. "In an embodiment according to the present invention, the update agent table 231 may comprise an array of addresses for all available update agents in the mobile handset." Okonnen at [0072]-[0073]: "FIG. 2C is a block diagram 255 illustrating an update agent table 261 located in non-volatile memory 111 of an electronic device, for example, mobile handset 107, according to an embodiment of the present invention. The update agent table in the mobile handset 107 may contain references to a plurality of update agents currently available and provisioned in the mobile handset. "The update agent table 261 may maintain, for example, a mapping between update agent names, the update agent corresponding address locations in non-volatile memory, and the type of updates that the update agents are adapted to process, for all available update agents in the mobile handset 107. The update agent table 261 may also contain optional information regarding provisioning status of the update agents, such as a provisioning flag indicating whether the associated update agent has been provisioned or not."

'117 Claims	Exemplary Disclosures
	Okonnen at [0075]-[0076]: "The type information may be provided as 'file extensions' in an embodiment according to the present invention. For example, the file extension 'dup' may correspond to updates generated to update generators, such as generator A 139 and generator B 141, as illustrated in FIG. 1. The type field in each entry in the update agent table illustrated in FIG. 2C may be a list of file extensions in an embodiment according to the present invention. "Other structures and schemas are also contemplated for the update agent table 261. The update agent table 261 may map the name of an update agent to the location where the agent resides in memory. The update agent table 261 may also map the provisioning status of an update agent and the type(s) of updates the update agent is adapted to process. Management of the update agents may be facilitated by the mobile handset 107. New table entries for update agents may be added and entries no longer in use may be deleted, etc. Management of the update agents may be supported by client software, such as a device management client." *See also* Okonnen at Fig. 1. Hashimoto at [0038]-[0039]: "Fig. 1 shows a multi-party application protection environment to which the present invention is to be applied. "A software vendor 121 produces a program 122, and distributes it to a microprocessor-A 103 provided inside a target system 102 of a system user 112. In order to protect the program to be distributed, the plaintext program 122 is encrypted." *See also* Hashimoto at Fig. 1. Fahrny at [0023]:

'117 Claims	Exemplary Disclosures
	"Keylist: A list of decoder addresses and respective decoder keys in ordered pairs. Keylists may be used by the Uplink Control System (UCS) for generation of authorization messages that are addressed to the diagnostic circuit that is embedded in decoders that are specific to the encoder system." Fahrny at [0037]-[0038]: "For VOD services, a table of keys can be generated and delivered at session setup time. The keys for VOD service may be delivered with synchronization information related to key change as well as other information for short term working key epochs. A VOD Session ID or, alternatively, a Program ID may be used as an index to reference the keys list with the appropriate record of information for the VOD transport decryption. In alternative embodiments of reduced hierarchy key management of the present invention, one-way hashing may be implemented in the protection, selection and processing of the decryption key. "The reduced hierarchy key management of the present invention generally provides a new, more secure, and elegant system and method to deliver content keys for decrypting the program streams in conditional access systems (e.g., Broadcast and Video On Demand applications). The key management of the present invention may dramatically reduce the complexity required to deliver new content keys when a first Entitlement Message has been sent (i.e., presented, transmitted, provided, broadcast, etc.) to each set top box. The reduced hierarchy key management system and method of the present invention may be implemented as a portion of a new CAS system. The new CAS system generally provides for the manufacture and distribution of devices that are compatible with infrastructure, regardless of specific content security mechanisms that are used in that infrastructure. The new CAS system may provide far more efficient manufacturing, distribution and operations, and in fact enable new business models, including the

'117 Claims	Exemplary Disclosures
	retail availability of extremely low cost customer premises equipment (CPE) when compared to conventional approaches." Fahrny at [0047]: "The table 112 generally comprises a content key index table. The contents of the table 112 are generally loaded from the headend (e.g., via the input/output 104). During the downloading from the headend, the content key list table may be decrypted and extracted using the respective unit or device key. The table 112 may receive the identifier PID via the input 102. The table 112 may present an index (e.g., IND) to the content key list 114 in response to the identifier PID based on the respective value in the table 112 using a content key index contained therein. In an alternative (i.e., optional) example, the table 112 comprises a content key and IV index table." Fahrny at [0048]: "The list 114 generally contains a list of content keys that may be referenced by respective index values (e.g., the index IND). The contents of the list 114 may be loaded via the input/output 104. The list 114 may be configured to present a content key to the combiner 116 in response to the respective index IND. The content keys (and, alternatively or optionally, IV values) in the list 114 that correspond to a particular encrypted media stream VIDIN are selected from the content key (and, alternatively or optionally, IV) table 112 using an entry in the content key (and, alternatively or optionally, IV) index that is referenced by the identifier PID that is received from the headend in connection with the encrypted media stream VIDIN. Content keys and IVs that correspond to a particular encrypted media stream are selected from the content key and IV list using the index IND from a content key and IV table that is referenced by the identifier PID that is received from the headend in connection with the encrypted media stream when Cipher Block Chaining is used as the mode of a selected algorithm."

'117 Claims	Exemplary Disclosures
	Fan at 13-14: FIG. 9 is a flowchart of processing a terminal according to a Push message control policy. In this process, it is assumed that a service type of a Push message is a multimedia multimedia message, and the process includes: Step 901: After receiving the Push message sent by the PI, the Push sending proxy sends the Push message to the Push receiving proxy by using a NOTIFY or MESSAGE message in the SIP message. Step 902: After receiving the Push message, the Push receiving proxy returns a 200 OK message to the Push sending proxy to notify the Push sending proxy that the Push message has been successfully received. Step 903: The Push receiving agent checks the service type carried in the Accept-Contact header field in the SIP message carrying the Push message, and determines that the Push message is a multimedia multimedia message. If the terminal does not allow to receive the message of the multimedia multimedia message service in the Push message control policy, the Push receiving agent discards the Push message without sending it to the application module, and the user does not know that a message arrives; if the Push message controls the policy, the terminal If the multimedia MMS service is not allowed to be received, the Push receiving agent temporarily stores the Push message; if the Push message control policy, the terminal allows to receive the message of the multimedia MMS service, and then sends the message to the application module for corresponding service processing. Fan at 26: The terminal sets the Push message control policy, and stores the set Push message control policy locally. After the PI sends the Push message to the Push sending proxy, the Push sending proxy sends the Push message to the terminal, and the terminal controls the policy according to the stored Push message. Process the Push

'117 Claims	Exemplary Disclosures
	message. FIG. 13 is a flowchart of processing a received Push message by the terminal, where the process includes: Step 1301: The Push sending proxy sends the Push message received from the PI to the Push receiving proxy in the terminal, and the specific method of sending the Push message is that the Push sending proxy carries the Push message in the SIP message header field and sends the Push message to the Push receiving proxy. Step 1302: The Push receiving agent compares the matching conditions between the received Push message and the Push message control policy, such as whether the content is too large, whether the keyword is not expected to be received, whether it is a message that is not currently received, and whether it is from a Trusted PI and so on. If the matching condition that the Push message satisfies is the Push content that the terminal is allowed to receive, step 1303 is continued; otherwise, if the matching condition that the Push message satisfies is the Push content that the terminal is not allowed to receive, the Push receiving proxy discards the Push message, if the Push message is satisfied. The matching condition is that the terminal is not allowed to receive the Push content. The Push receiving agent temporarily stores the Push message. After the operation of discarding or temporarily storing the Push message is performed, step 1304 is directly executed. When the Push receiving agent discards or temporarily stores the Push message, before the step 1304 is directly executed, the Push receiving agent may return a message that the Push message has been deleted to the Push sending agent. This step is an optional step. Step 1303: The Push receiving proxy returns an acknowledgement message to the Push sending proxy, and sends the Push message to the application module in the terminal to perform corresponding services. Cooley at Abstract:

'117 Claims	Exemplary Disclosures
	"Systems and methods are provided that allow an application residing on a user device to intercept messages directed to that device and to extract control information and/or data from the intercepted messages for use by the application. In one embodiment, the intercepted message can cause the application to change its configuration or to present information to the user that came from a sending website. In another embodiment, a message from a device can trigger a website to send back a desired piece of data. The returned data may contain a flag, which directs the data to a formatting server. The formatting server then converts the data into a format that allows the data to be intercepted by the desired application on the user device. This then allows the desired data to be presented to the user in the context of an application resident on the user's device." Cooley at ¶[0005]: "Systems and methods are provided that allow an application residing on a user device to intercept messages directed to that device and to extract control information and/or data from the intercepted messages for use by the application. In one embodiment, the intercepted message can cause the application to change its configuration or to present information to the user that came from a sending website. In another embodiment, a message from a device can trigger a website (such as an airline website) to send back a desired piece of data. The returned data can, for example, contain a flag, which directs the data to a formatting server. The formatting server then converts the data into a format that allows the data to be intercepted by the desired application on the user device. This then allows the desired data to be presented to the user in the context of an application resident on the user's device." Cooley at ¶[0013]: "FIG. 1 shows one embodiment of a system, such as system 10, in which the concepts of the invention can be practiced. In FIG. 1 there is shown a link from a

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 201 of 442

'117 Claims	Exemplary Disclosures
	user device, such as cellular telephone 15 to website 11. This link extends from the user device through a carrier network, such as network 14 and via service center 12 to the desired website. The message returned from website 11 contains an HTTP request against service center 12. The service center then queues a Short Message Service (SMS) message back to device 15 through an SMS broker, such as broker 13. Broker 13 then routes an SMS message through any carrier network, such as network 14, which need not be the same network that was used for the originating message from the device." Cooley at ¶[0015]: "When the incoming text message arrives at the user's device it is intercepted, under control of the device processor and an application, such as application 31 (as will be discussed), running on the device. This is possible because application 31 is registered with the device operating system when it was loaded on the device. Application 31 then routes the incoming message to the proper specific application in accordance with the data contained in the text message. Thus, in the situation with the flight reservation to Las Vegas (as discussed above), information pertaining to that flight is routed to a specific application, such as to air travel application 22-1 (shown in FIG. 2B), which application is dedicated to air travel data pertaining to the device user." Cooley at ¶[0017]: "When the user touches tile 23-1 all the flights that the user has booked on, for example, AMERICAN™ airlines (AA) are displayed. Touching tile 24-1 then yields the screen shown in FIG. 2E which then displays the information that has been received from an external source. In this case this information would have come typically from the AA server in the manner discussed above. There could be any amount of information and tiles (or layers) to this information, all controlled by application 30 residing on the user's device with the updated information

'117 Claims	Exemplary Disclosures
	coming, from time to time, via messages from various external locations. Note that in FIG. 2F the gate and status of the flight have been updated. This update could have come as a result of a request from the user or could have come as a text message from the airline as part of an update service for customers. In any event, the message would have gone first to service center 12 and then via SMS broker 13 to the user's device where application 30 would have taken the data and placed it in the proper tile." Cooley at ¶[0018]: "FIG. 3 shows one embodiment of a system for controlling messages on a user device. The incoming SMS message would normally be delivered to level 32 of network stack 30 for subsequent display to device user via, for example, display 16 (FIG. 1). The display would be in text or visual form and would then go into the SMS box in text, graphics or video form as is well-known. However, application 31 intercepts the message because that application is registered with the platform to receive certain SMS messages. This protocol varies from device to device based on operating system. SMS or HTTP transports can be used via the side channel process. SMS has the advantage of being a true "push" notification. Application 31 then routes the incoming message, based on the context and content of the data contained in the message to an individual tile, such as to tile 22-1 (FIG. 2B) or even more specifically to tile 24-1 (FIG. 2D), as discussed above." Cooley at ¶[0019]: "The SMS message contains envelope information which includes the phone number to which it is destined, and in this case would include additional routing information to route it to application 31 so that the message does not get stored in the message box so the message will not be text formatted. Thus the system takes advantage of a feature that allows SMS messages to be addressed to internal

'117 Claims	Exemplary Disclosures
	applications. The application address is added at the service center 12, or if desired could be added by the sender if the application address is made known to the sender. This could, for example, be accomplished by registering the address with the website in advance." Cooley at ¶[0020]: "Another method for doing this would be to run a Simple Mail Transfer Protocol (SMTP) gateway at service center 12. This would allow application 31 to receive incoming messages in email form. Such email messages could then be parsed to see if they contain certain information, such as the tile ID, etc. This approach would benefit from a fixed format email message. The parsing includes filtering the email messages to see if they contain certain information. The filtered information then would be sent to the device by an SMS message. Second, the formatted SMS message could be sent to the service center for delivery to the particular user device." Cooley at ¶[0021]: "FIG. 4 shows one embodiment of a method, such as method 40, for controlling applications resident on a mobile device from a remote location. Process 401 determines if a message is incoming to a particular device. If there is a message available, then process 402 determines if the message contains any code or data that identifies the message as one that is to be delivered to a specific application resident on the device and/or delivered customized in a particular manner. If not, then the message is stored in its normal location, for example in SMS layer 32 (FIG. 3) and delivered without modification." Cooley at ¶[0022]: "If the message does contain a "diversion" code or other data that indicates that the message is to be delivered to another location or delivered in a customized manner,

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 204 of 442

'117 Claims	Exemplary Disclosures
	then process 404 delivers the incoming message (or other data) accordingly as discussed above, such that a particular user's tile is customized for a particular application. Customizing could, for example, include one or more of the following: appearance, context, display parameters, including audio, visual, application state change. In some situations, a control command is sent to a user's tile, for example, using an SMS transport protocol such that the command controls the display parameters and/or the display message instead of actual text being sent to the tile." Wakasa at ¶[0040]: "The server device 200 generates a transmitted message including parameter information and the identification information of an application to be executed by the portable terminal device 100 on the basis of the parameter information. Then, the server device 200 transmits the transmitted message to the portable terminal device 100 via the network 300. The parameter information is typically information used for requesting the application of the portable terminal device 100 to display information desired by the user or carry out predetermined processing. For example, the parameter information is information requesting the application of the portable terminal device 100 to display "Hello" on the display screen. The portable terminal device 100 executes the application corresponding to the identification information of the application included in transmitted message by using the parameter information as input data. For example, by using the parameter information requesting the application of the portable terminal device 100 to display "Hello" on the display screen as input data, the portable terminal device 100 executes the application corresponding to the identification information, displaying "Hello" on the display screen of the portable terminal device 100. As an alternative, the server device 200 holds an application to be provided to the portable terminal device 100 and, in accordance with a request made by the portable terminal device 100, the server device 200 transmits the application to the portable terminal device 100 via the network 300."

'117 Claims	Exemplary Disclosures
	Wakasa at ¶[0053]: "Details of a transmitted message received from the server device 200 are described by referring to FIG. 3 as follows. FIG. 3 is an explanatory diagram showing one example of a transmitted message received from the server device 200. As shown in the figure, the transmitted message includes transmitting parameter 150, a header 152, and a footer 160. The transmitting parameter 150 includes a URL 154, an ID156, parameter information (Param) 158. The Param 158 is the parameter information explained earlier. The URL 154 and the ID 156 are identification information used for identifying an application to be executed on the basis of the Param 158. Applications of the portable terminal device 100 have been stored in the application storage section 114 in a state of being identifiable by using the ID 156 or the URL 154." Wakasa at ¶[0054]: "The information used for identifying an application may be dependent on the specifications of the portable terminal device 100. For example, in the portable terminal device 100 conforming to the specifications set by a mobile communication business company D, the application is identified by using the URL 154. In the portable terminal device 100 conforming to the specifications set by mobile communication business companies K and V, on the other hand, the application may be identified by using the ID 156. In this embodiment, both the URL 154 and the ID 156 are included in a transmitted message so that each of the portable terminal devices 100 manufactured by the mobile communication business companies D, K and V with specifications different from each other is capable of determining whether or not an application exists in the portable terminal device 100 as will be described later." *See also* Wakasa, ¶[0086].

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 206 of 442

'117 Claims	Exemplary Disclosures
	Wakasa at ¶[0061]: "The transmitted message generation section 202 has a function to generate a transmitted message including parameter information and the identification information of an application to be executed by the portable terminal device 100 on the basis of the parameter information. The transmitted message generation section 202 supplies the generated transmitted message to the transmitted message sender section 204." Wakasa at ¶[0062]: "The transmitted message sender section 204 is a communication interface such as a communication line, a communication circuit or a communication device. The transmitted message sender section 204 sends the transmitted message received from the transmitted message generation section 202 to the portable terminal device 100." Satake at ¶[0070]: "The information providing system according to the first aspect of the present invention comprises: a server that is provided with: a message transmitting section that transmits to a mobile communication terminal a message to which a term of validity is attached and that instructs that a predetermined application in the mobile communication terminal be started up; and an information transmitting section that transmits to the mobile communication terminal predetermined information that is requested by the started up application in the mobile communication terminal; and a mobile communication terminal that is provided with: a receiving section that receives a message instructing that an application transmitted by the server be started up; a storage section that stores an application that requests the server for predetermined information and receives the information; a control section that, when the receiving section receives the message, determines whether or not a term of validity is attached to the message, and, based on the term of validity, starts up the

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 207 of 442

'117 Claims	Exemplary Disclosures
	instructed application from among the applications that are stored in the storage section." Satake at ¶[0136]: "The system shown in FIG. 1 is formed by a stock price information server 1, an SMS server 6 that transmits an SMS message from the stock price information server 1 to a user, a user terminal 7 serving as a mobile communication terminal belonging to the user, and a mobile exchange system 8 that relays communication between the stock price information server 1, the SMS server 6, and the user terminal 7. The mobile exchange system 8 and the user terminal 7 are connected by two transmission paths, namely, a traffic channel that is used for data communication and a paging channel that is used for SMS (i.e., PUSH messages) communication." *See also* Satake at Fig. 1. Satake at ¶[0152]: "FIG. 13, information delivered by a contents server 101 toward a mobile telephone 102 is received by an SMS server 104 via a packet network 103. If the mobile telephone 102 is a CDMA type of telephone, the SMS server 104 transmits a PUSH message over the paging channel and the transmitted PUSH message is received by the mobile telephone 102. In this case, even if the mobile telephone 102 is conducting data communication or make a telephone call over the traffic channel in parallel with the transmission of the PUSH message, the message reception processing can be performed. When the mobile telephone 102 receives the PUSH message, it starts up an application that corresponds to this PUSH message, and for data communication over the traffic channel, commences communication with the contents server 101 via a packet serving data node (PSDN) 105. Note that, in the present embodiment, SMS is used

'117 Claims	Exemplary Disclosures
	for the transmission and reception of messages; however, it is also possible to use e-mail instead of SMS." XMPP at 3-4: "2.1.1. Types of Message The 'type' attribute of a message stanza is RECOMMENDED; if included, it specifies the conversational context of the message, thus providing a hint regarding presentation (e.g., in a GUI). If included, the 'type' attribute MUST have one of the following values: ○ chat -- The message is sent in the context of a one-to-one chat conversation. A compliant client SHOULD present the message in an interface enabling one-to-one chat between the two parties, including an appropriate conversation history. ○ error -- An error has occurred related to a previous message sent by the sender (for details regarding stanza error syntax, refer to [XMPP-CORE]). A compliant client SHOULD present an appropriate interface informing the sender of the nature of the error. ○ groupchat -- The message is sent in the context of a multi-user chat environment (similar to that of [IRC]). A compliant client SHOULD present the message in an interface enabling many-to-many chat between the parties, including a roster of parties in the chatroom and an appropriate conversation history. Full definition of XMPP-based groupchat protocols is out of scope for this memo. ○ headline -- The message is probably generated by an automated service that delivers or broadcasts content (news, sports, market information, RSS feeds, etc.). No reply to the message is expected, and a compliant client SHOULD present the message in an interface that appropriately differentiates the message from standalone messages, chat sessions, or groupchat sessions (e.g., by not providing the recipient with the ability to reply). ○ normal -- The message is a single message that is sent outside the context of a one-to-one conversation or groupchat, and to which it is expected that the

'117 Claims	Exemplary Disclosures
	recipient will reply. A compliant client SHOULD present the message in an interface enabling the recipient to reply, but without a conversation history." *See also* XMPP, 5, 11.
[1f] each device messaging agent, when executing, to receive the Internet data messages from the secure Internet data connection corresponding to the device executing the device messaging agent, and	The references below disclose and/or render obvious this limitation. For example, see the following passages and/or figures, as well as related disclosures: Huskins at 4:10-19: "The data management session control server 21 transmits messages that include one or more distinct service identifiers (e.g., service__id=0x03) . The server 21 creates a DMSC message data structure and passes information and data to the unidirectional messaging server 22 such as application-private data and data size, unique client identifier, a group flag which is a Boolean variable specifying if the client identifier is a group mask or individual client identifier, a service identifier value for the DMSC service and application-private message identifier. " Huskins at 4:20-33: "In accordance with one embodiment of the present invention, the server 10 may implement a unidirectional messaging system. In a unidirectional messaging system, the server 10 may transmit messages to a plurality of clients that are unable to respond in any way. One example of such a network is a direct-to-home (DTH) broadcast network that may be compliant with the DVB protocol. The network may use a connection oriented communication protocol or a real time connectionless communication protocol as two examples. There are many applications of unidirectional messaging from server to client such as instant messaging, command and control and notification and signaling, as examples. In other cases, the network may be a bidirectional network, for example with an Internet Protocol (IP) multicast backbone." Huskins at 6:11-21:

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 210 of 442

'117 Claims	Exemplary Disclosures
	"In one embodiment of the present invention, the network management command and control server 16 may provide a specific message identifier that causes its message to be received by a network management session management agent 48 tuned to a particular service identifier. Similarly, messages from the instant messaging server 18 may include a service identifier that cause those messages to be forwarded to the agent 48 in the client 12. Likewise, messages from the data management session control server 21 may have appropriate identifiers that cause them to be shunted to the data management session control agent 45 on the client 12." Huskins at 6:28-7:6: "Turning next to Figure 2, the software 50 on the client 12 initially receives the unidirectional messaging server address and port from the server 10. The client 12 may also be assigned a client identifier as indicated in block 52. Thus, an Internet Protocol multicast system may be established wherein each client has a UMS address and port as well as a unique client identifier, assigned by the server 10. In some embodiments, the server 10 may dynamically adjust addresses and ports as well as client identifiers to enable communication of particular messages, message groups or types of messages to particular clients in a dynamic and reconfigurable fashion." Huskins at 7:19-26: "However, if the particular client 12 is the intended recipient, the server 38 checks the message's service identifier and passes the message to the correct agent 44, 46 or 48, as indicated in block 64. The message is then delivered to the appropriate agent 44 as indicated in block 66. In the agent 44, the information is parsed and passed to an appropriate process for handling as indicated in block 68." Huskins at 8:16-30: "Thereafter, the disk management server 20 may create a data structure and pass this data structure to the server 22 as indicated in block 76. The server 22 creates a

'117 Claims	Exemplary Disclosures
	unidirectional message and assigns a client value, sets a group flag, and copies private data in the private bytes of the message as indicated in block 78. More particularly, a unique client identifier may be assigned. The client identifier may either be a particular preassigned client identifier or, as one example, may be zero when multiple clients are targeted. A group flag may be a Boolean value specifying whether the client identifier is a group mask or a particular identifier. A group mask is an identifier that identifies a subset of the clients 12 on the network. This subset may include a plurality of clients but less than the total number of addressable clients." Huskins at 11:3-10: "Referring to Figure 4, the software 82 on the client 12 for implementing a network management session, initially receives the unidirectional messaging server address and port from the server 10. The client 12 may also be assigned a client identifier as indicated in block 84. Thus, an Internet Protocol multicast system may be established wherein each client has a UMS address and port as well as a unique client identifier, assigned by the server 10. Having receives its address, port and client identifier, the client 12 receiver joins a multicast group and listens for messages addressed specifically to it or to any groups that the client 12 belongs to, as indicated in block 86." Huskins at 12:27-33: "The server 21 may provide messages with different message identifiers including a session create message, a session delete message and session update message, as examples. The session delete message simply deletes a previously created session and the session update message causes additional information to be provided for a scheduled session." Chou at 12:50: "5. Client device receives the push content."

'117 Claims	Exemplary Disclosures
	Isomursu at 22:36-67: "When address information has been added at a **SMS transmission controller** 10, the message is transferred into an **inbox** 11, which tries to send the message, and which has a buffer wherein the message is stored in case the transmission fails. If the transmission fails, the **inbox** 11 re-tries to send the message. When the **DU controller** 8 notices that the radio unit RU is ready to send the message, the message is transferred to a message **transfer running circuit** 12, which adds to the message information relating to the mobile communications system in question, such as validity information (which indicates to which direction the message is going, i.e., from a mobile station to a message service centre or vice versa), processes the address information into a form required by the mobile communications system, and adds to the message the address of the message service centre, as well as the short message identifier (SAPI), and forms from the information to be transmitted, e.g., a digital signal for a **transmitter** 2, and sends the message to the **radio transmitter branch** 2 of the radio unit RU. In the case where the application identifier is placed in bits in the address field ADD, the running **circuit** 12 adds to the message the identifier in question. The **transmitter branch** 2 codes the signal according to the specifications of the mobile communications system, and forms, on the basis of the signal it receives from the running **circuit** 12, the frames to be transmitted, which the transmitter sends using radio communication to the short message service centre SM-SC, wherefrom they are sent further to the receiver (see **FIG. 1**). In the **transmitter branch** 2, the message is processed according to the mobile communications system, e.g., coding, interleaving, ciphering, burst forming, modulating, and transmission." Okonnen at [0021]: "In an embodiment according to the present invention, the index of provisioned update agents may provide an index value used to compute an address location of a provisioned update agent. The index of provisioned update agents may be an index

'117 Claims	Exemplary Disclosures
	to a table containing an address for an update agent in non-volatile memory the electronic device." Okonnen at [0047]-[0048]: "FIG. 1 is a block diagram an electronic device network 105 according to an embodiment of the present invention. In the electronic device network illustrated in FIG. 1, an electronic device, for example mobile handset 107, may be communicatively coupled to at least one of a plurality of device servers, such as device server A 135 and device server B 137, for example. "The electronic device, for example, mobile handset 107, may be capable of employing one of several available update agents, for example, update agent A 113 and update agent B 115, resident in the electronic device to update software and/or firmware in the electronic device. The electronic device, for example, mobile handset 107, may also capable of provisioning the available update agents, such as for example, update agent A 113 and update agent B 115, with the necessary parameters and data needed to facilitate the respective update operations." Okonnen at [0069]: "FIG. 2B is a block diagram 225 illustrating an update agent table 231 stored in non-volatile memory 111 of an electronic device, for example, mobile handset 107, according to an embodiment of the present invention. The update agent table 231 in the mobile handset 107 may provide references, such as location addresses and other related information, including information regarding update agent provisioning, to the update agents currently available in the mobile handset 107." Okonnen at [0070]-[0071]: "The update agent table 231 may maintain references to a plurality of available updates and associated update agents. When a new update agent is added, such as via an update of firmware/software in the mobile handset 107, an additional update may also be inserted into the mobile handset 107. The additional update may invoke insertion of a corresponding entry in the update agent table 231 and

'117 Claims	Exemplary Disclosures
	population thereof with associated relevant information, such as the address location of the new update agent, (i.e., where the new update agent is stored in non-volatile memory 111). An update package may be downloaded and installed in the mobile handset 107 and may result in removal of an existing update agent or replacement thereof by another. "In an embodiment according to the present invention, the update agent table 231 may comprise an array of addresses for all available update agents in the mobile handset." Okonnen at [0072]-[0073]: "FIG. 2C is a block diagram 255 illustrating an update agent table 261 located in non-volatile memory 111 of an electronic device, for example, mobile handset 107, according to an embodiment of the present invention. The update agent table in the mobile handset 107 may contain references to a plurality of update agents currently available and provisioned in the mobile handset. "The update agent table 261 may maintain, for example, a mapping between update agent names, the update agent corresponding address locations in non-volatile memory, and the type of updates that the update agents are adapted to process, for all available update agents in the mobile handset 107. The update agent table 261 may also contain optional information regarding provisioning status of the update agents, such as a provisioning flag indicating whether the associated update agent has been provisioned or not." Okonnen at [0075]-[0076]: "The type information may be provided as 'file extensions' in an embodiment according to the present invention. For example, the file extension 'dup' may correspond to updates generated to update generators, such as generator A 139 and generator B 141, as illustrated in FIG. 1. The type field in each entry in the update agent table illustrated in FIG. 2C may be a list of file extensions in an embodiment according to the present invention.

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 215 of 442

'117 Claims	Exemplary Disclosures
	"Other structures and schemas are also contemplated for the update agent table 261. The update agent table 261 may map the name of an update agent to the location where the agent resides in memory. The update agent table 261 may also map the provisioning status of an update agent and the type(s) of updates the update agent is adapted to process. Management of the update agents may be facilitated by the mobile handset 107. New table entries for update agents may be added and entries no longer in use may be deleted, etc. Management of the update agents may be supported by client software, such as a device management client." *See also* Okonnen at Fig. 1. Fahrny at [0008]: "According to the present invention, a controller for managing media stream decryption keys is provided. The controller comprises a media decryption engine, a table, and a content key list. The media decryption engine generally receives an encrypted media stream from a headend and presents a decrypted media stream to a receiving device in response to a decryption key. The decryption key is generally a function of a content key. The table may contain a content key and index and a plurality of corresponding content keys. Content keys that correspond to a particular encrypted media stream may be selected from the content key list using an index from the content key table, and that is referenced by an identifier received from a headend in connection with the encrypted media stream. The table may optionally (i.e., alternatively) contain initialization vector (IV) values that may be indexed and selected." Fahrny at [0023]: "Keylist: A list of decoder addresses and respective decoder keys in ordered pairs. Keylists may be used by the Uplink Control System (UCS) for generation of authorization messages that are addressed to the diagnostic circuit that is embedded in decoders that are specific to the encoder system." Fahrny at [0037]-[0038]:

'117 Claims	Exemplary Disclosures
	"For VOD services, a table of keys can be generated and delivered at session setup time. The keys for VOD service may be delivered with synchronization information related to key change as well as other information for short term working key epochs. A VOD Session ID or, alternatively, a Program ID may be used as an index to reference the keys list with the appropriate record of information for the VOD transport decryption. In alternative embodiments of reduced hierarchy key management of the present invention, one-way hashing may be implemented in the protection, selection and processing of the decryption key. "The reduced hierarchy key management of the present invention generally provides a new, more secure, and elegant system and method to deliver content keys for decrypting the program streams in conditional access systems (e.g., Broadcast and Video On Demand applications). The key management of the present invention may dramatically reduce the complexity required to deliver new content keys when a first Entitlement Message has been sent (i.e., presented, transmitted, provided, broadcast, etc.) to each set top box. The reduced hierarchy key management system and method of the present invention may be implemented as a portion of a new CAS system. The new CAS system generally provides for the manufacture and distribution of devices that are compatible with infrastructure, regardless of specific content security mechanisms that are used in that infrastructure. The new CAS system may provide far more efficient manufacturing, distribution and operations, and in fact enable new business models, including the retail availability of extremely low cost customer premises equipment (CPE) when compared to conventional approaches." Fahrny at [0047]: "The table 112 generally comprises a content key index table. The contents of the table 112 are generally loaded from the headend (e.g., via the input/output 104). During the downloading from the headend, the content key list table may be decrypted and extracted using the respective unit or device key. The table 112 may receive the identifier PID via the input 102. The table 112 may present an index (e.g., IND) to the content key list 114 in response to the identifier PID based on the

'117 Claims	Exemplary Disclosures
	respective value in the table 112 using a content key index contained therein. In an alternative (i.e., optional) example, the table 112 comprises a content key and IV index table." Fan at 12: The matching conditions in the set Push message control policy may include the following: user identifier, maximum message length threshold, service control, content control, and control of the trusted PI. The following five Push message control strategies are introduced separately. Fan at 13: First, the user identifier The purpose of setting the matching condition is to determine, according to the user identifier, that the terminal to which the Push message is to be sent is the setting terminal of the Push message control policy when the Push sending agent performs the Push message control. Fan at 13-14: FIG. 9 is a flowchart of processing a terminal according to a Push message control policy. In this process, it is assumed that a service type of a Push message is a multimedia multimedia message, and the process includes: Step 901: After receiving the Push message sent by the PI, the Push sending proxy sends the Push message to the Push receiving proxy by using a NOTIFY or MESSAGE message in the SIP message. Step 902: After receiving the Push message, the Push receiving proxy returns a 200 OK message to the Push sending proxy to notify the Push sending proxy that the Push message has been successfully received. Step 903: The Push receiving agent checks the service type carried in the Accept-Contact header field in the SIP message carrying the Push message, and determines that the Push message is a multimedia multimedia message. If the terminal does not allow to receive the message of the multimedia multimedia message service in the

'117 Claims	Exemplary Disclosures
	Push message control policy, the Push receiving agent discards the Push message without sending it to the application module, and the user does not know that a message arrives; if the Push message controls the policy, the terminal If the multimedia MMS service is not allowed to be received, the Push receiving agent temporarily stores the Push message; if the Push message control policy, the terminal allows to receive the message of the multimedia MMS service, and then sends the message to the application module for corresponding service processing. Fan at 26: The terminal sets the Push message control policy, and stores the set Push message control policy locally. After the PI sends the Push message to the Push sending proxy, the Push sending proxy sends the Push message to the terminal, and the terminal controls the policy according to the stored Push message. Process the Push message. FIG. 13 is a flowchart of processing a received Push message by the terminal, where the process includes: Step 1301: The Push sending proxy sends the Push message received from the PI to the Push receiving proxy in the terminal, and the specific method of sending the Push message is that the Push sending proxy carries the Push message in the SIP message header field and sends the Push message to the Push receiving proxy. Step 1302: The Push receiving agent compares the matching conditions between the received Push message and the Push message control policy, such as whether the content is too large, whether the keyword is not expected to be received, whether it is a message that is not currently received, and whether it is from a Trusted PI and so on. If the matching condition that the Push message satisfies is the Push content that the terminal is allowed to receive, step 1303 is continued; otherwise, if the matching condition that the Push message satisfies is the Push content that the terminal is not allowed to receive, the Push receiving proxy discards the Push message, if the Push message is satisfied. The matching condition is that the terminal is not allowed to receive the Push content. The Push receiving agent temporarily stores the Push message. After the operation of discarding or

'117 Claims	Exemplary Disclosures
	temporarily storing the Push message is performed, step 1304 is directly executed. When the Push receiving agent discards or temporarily stores the Push message, before the step 1304 is directly executed, the Push receiving agent may return a message that the Push message has been deleted to the Push sending agent. This step is an optional step. Step 1303: The Push receiving proxy returns an acknowledgement message to the Push sending proxy, and sends the Push message to the application module in the terminal to perform corresponding services Cooley at Abstract: "Systems and methods are provided that allow an application residing on a user device to intercept messages directed to that device and to extract control information and/or data from the intercepted messages for use by the application. In one embodiment, the intercepted message can cause the application to change its configuration or to present information to the user that came from a sending website. In another embodiment, a message from a device can trigger a website to send back a desired piece of data. The returned data may contain a flag, which directs the data to a formatting server. The formatting server then converts the data into a format that allows the data to be intercepted by the desired application on the user device. This then allows the desired data to be presented to the user in the context of an application resident on the user's device." Cooley at ¶[0001]: "This disclosure relates to the control and customization of applications resident on a wireless device from locations remote therefrom and more specifically to systems and methods for allowing applications resident on a mobile device to be controlled using the text messaging capability of the wireless network." Cooley at ¶[0005]:

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 220 of 442

'117 Claims	Exemplary Disclosures
	"Systems and methods are provided that allow an application residing on a user device to intercept messages directed to that device and to extract control information and/or data from the intercepted messages for use by the application. In one embodiment, the intercepted message can cause the application to change its configuration or to present information to the user that came from a sending website. In another embodiment, a message from a device can trigger a website (such as an airline website) to send back a desired piece of data. The returned data can, for example, contain a flag, which directs the data to a formatting server. The formatting server then converts the data into a format that allows the data to be intercepted by the desired application on the user device. This then allows the desired data to be presented to the user in the context of an application resident on the user's device." Cooley at ¶[0013]: "FIG. 1 shows one embodiment of a system, such as system 10, in which the concepts of the invention can be practiced. In FIG. 1 there is shown a link from a user device, such as cellular telephone 15 to website 11. This link extends from the user device through a carrier network, such as network 14 and via service center 12 to the desired website. The message returned from website 11 contains an HTTP request against service center 12. The service center then queues a Short Message Service (SMS) message back to device 15 through an SMS broker, such as broker 13. Broker 13 then routes an SMS message through any carrier network, such as network 14, which need not be the same network that was used for the originating message from the device." Cooley at ¶[0015]: "When the incoming text message arrives at the user's device it is intercepted, under control of the device processor and an application, such as application 31 (as will be discussed), running on the device. This is possible because application 31 is

'117 Claims	Exemplary Disclosures
	registered with the device operating system when it was loaded on the device. Application 31 then routes the incoming message to the proper specific application in accordance with the data contained in the text message. Thus, in the situation with the flight reservation to Las Vegas (as discussed above), information pertaining to that flight is routed to a specific application, such as to air travel application 22-1 (shown in FIG. 2B), which application is dedicated to air travel data pertaining to the device user." Cooley at ¶[0017]: "When the user touches tile 23-1 all the flights that the user has booked on, for example, AMERICAN™ airlines (AA) are displayed. Touching tile 24-1 then yields the screen shown in FIG. 2E which then displays the information that has been received from an external source. In this case this information would have come typically from the AA server in the manner discussed above. There could be any amount of information and tiles (or layers) to this information, all controlled by application 30 residing on the user's device with the updated information coming, from time to time, via messages from various external locations. Note that in FIG. 2F the gate and status of the flight have been updated. This update could have come as a result of a request from the user or could have come as a text message from the airline as part of an update service for customers. In any event, the message would have gone first to service center 12 and then via SMS broker 13 to the user's device where application 30 would have taken the data and placed it in the proper tile." Cooley at ¶[0018]: "FIG. 3 shows one embodiment of a system for controlling messages on a user device. The incoming SMS message would normally be delivered to level 32 of network stack 30 for subsequent display to device user via, for example, display 16 (FIG. 1). The display would be in text or visual form and

'117 Claims	Exemplary Disclosures
	would then go into the SMS box in text, graphics or video form as is well-known. However, application 31 intercepts the message because that application is registered with the platform to receive certain SMS messages. This protocol varies from device to device based on operating system. SMS or HTTP transports can be used via the side channel process. SMS has the advantage of being a true "push" notification. Application 31 then routes the incoming message, based on the context and content of the data contained in the message to an individual tile, such as to tile 22-1 (FIG. 2B) or even more specifically to tile 24-1 (FIG. 2D), as discussed above." Cooley at ¶[0019]: "The SMS message contains envelope information which includes the phone number to which it is destined, and in this case would include additional routing information to route it to application 31 so that the message does not get stored in the message box so the message will not be text formatted. Thus the system takes advantage of a feature that allows SMS messages to be addressed to internal applications. The application address is added at the service center 12, or if desired could be added by the sender if the application address is made known to the sender. This could, for example, be accomplished by registering the address with the website in advance." Cooley at ¶[0020]: "Another method for doing this would be to run a Simple Mail Transfer Protocol (SMTP) gateway at service center 12. This would allow application 31 to receive incoming messages in email form. Such email messages could then be parsed to see if they contain certain information, such as the tile ID, etc. This approach would benefit from a fixed format email message. The parsing includes filtering the email messages to see if they contain certain information. The filtered information then would be sent to the device by an SMS message. Second, the formatted SMS

'117 Claims	Exemplary Disclosures
	message could be sent to the service center for delivery to the particular user device." Cooley at ¶[0021]: "FIG. 4 shows one embodiment of a method, such as method 40, for controlling applications resident on a mobile device from a remote location. Process 401 determines if a message is incoming to a particular device. If there is a message available, then process 402 determines if the message contains any code or data that identifies the message as one that is to be delivered to a specific application resident on the device and/or delivered customized in a particular manner. If not, then the message is stored in its normal location, for example in SMS layer 32 (FIG. 3) and delivered without modification." Cooley at ¶[0022]: "If the message does contain a "diversion" code or other data that indicates that the message is to be delivered to another location or delivered in a customized manner, then process 404 delivers the incoming message (or other data) accordingly as discussed above, such that a particular user's tile is customized for a particular application. Customizing could, for example, include one or more of the following: appearance, context, display parameters, including audio, visual, application state change. In some situations, a control command is sent to a user's tile, for example, using an SMS transport protocol such that the command controls the display parameters and/or the display message instead of actual text being sent to the tile." Wakasa at ¶[0040]: "The server device 200 generates a transmitted message including parameter information and the identification information of an application to be executed by the portable terminal device 100 on the basis of the parameter information. Then,

'117 Claims	Exemplary Disclosures
	the server device 200 transmits the transmitted message to the portable terminal device 100 via the network 300. The parameter information is typically information used for requesting the application of the portable terminal device 100 to display information desired by the user or carry out predetermined processing. For example, the parameter information is information requesting the application of the portable terminal device 100 to display "Hello" on the display screen. The portable terminal device 100 executes the application corresponding to the identification information of the application included in transmitted message by using the parameter information as input data. For example, by using the parameter information requesting the application of the portable terminal device 100 to display "Hello" on the display screen as input data, the portable terminal device 100 executes the application corresponding to the identification information, displaying "Hello" on the display screen of the portable terminal device 100. As an alternative, the server device 200 holds an application to be provided to the portable terminal device 100 and, in accordance with a request made by the portable terminal device 100, the server device 200 transmits the application to the portable terminal device 100 via the network 300." Wakasa at ¶[0051]: "FIG. 2 is a block diagram showing the functional configurations of the portable terminal device 100 and the server device 200. As shown in FIG. 2, the portable terminal device 100 includes a transmitted message receiver section 102, an application determination section 104, an application-transmission request sender section 106, an application-transmission request generation section 108, an application receiver section 110, a display section 112 and an application storage section 114." Wakasa at ¶[0052]:

'117 Claims	Exemplary Disclosures
	"The transmitted message receiver section 102 is a communication interface such as a communication line, a communication circuit, a communication device or the like. The transmitted message receiver section 102 has a function for receiving a transmitted message from the server device 200. A transmitted message received by the transmitted message receiver section 102 is passed on to the application determination section 104. As described above, a transmitted message received from the server device 200 includes parameter information and the identification information of an application to be executed by the portable terminal device 100 on the basis of the parameter information." *See also* Wakasa, Fig. 2. Wakasa at ¶[0053]: "Details of a transmitted message received from the server device 200 are described by referring to FIG. 3 as follows. FIG. 3 is an explanatory diagram showing one example of a transmitted message received from the server device 200. As shown in the figure, the transmitted message includes transmitting parameter 150, a header 152, and a footer 160. The transmitting parameter 150 includes a URL 154, an ID156, parameter information (Param) 158. The Param 158 is the parameter information explained earlier. The URL 154 and the ID 156 are identification information used for identifying an application to be executed on the basis of the Param 158. Applications of the portable terminal device 100 have been stored in the application storage section 114 in a state of being identifiable by using the ID 156 or the URL 154." Wakasa at ¶[0054]: "The information used for identifying an application may be dependent on the specifications of the portable terminal device 100. For example, in the portable terminal device 100 conforming to the specifications set by a mobile

'117 Claims	Exemplary Disclosures
	communication business company D, the application is identified by using the URL 154. In the portable terminal device 100 conforming to the specifications set by mobile communication business companies K and V, on the other hand, the application may be identified by using the ID 156. In this embodiment, both the URL 154 and the ID 156 are included in a transmitted message so that each of the portable terminal devices 100 manufactured by the mobile communication business companies D, K and V with specifications different from each other is capable of determining whether or not an application exists in the portable terminal device 100 as will be described later." *See also* Wakasa, ¶[0086]. Wakasa at ¶[0084]: "The controller 124 is a section for receiving a transmitted message transmitted by the non-contact-communication IC chip 120 via the second communication section, executing an application corresponding to identification information included in the transmitted message and displaying a result of the execution on a display screen of the portable terminal device 100. More specifically, when the controller 124 receives a transmitted message from the non-contact-communication IC chip 120, the controller 124 passes on the transmitted message received from the non-contact-communication IC chip 120 to the acceptance application 126. The acceptance application 126 then provides parameter information to the application corresponding to the identification information of the application included in the transmitted message. The identification information, such as ID 156 or URL 154, is used by the acceptance application 126 to determine as to whether or not the application corresponding to the identification information has been stored in the application storage section 114." Satake at ¶[0069]:

'117 Claims	Exemplary Disclosures
	"The mobile communication terminal according to a first aspect of the present invention comprises: a receiving section that receives a message; a storage section that stores an application that requests a server for predetermined information and receives the information; and a determining section that, when the receiving section receives a message that instructs that a predetermined application be started up, determines whether or not a term of validity is attached to the message, and , based on the term of validity, starts up the instructed application that is stored in the storage section." *See also* Satake at ¶¶[0073]-[0080]. Satake at ¶[0070]: "The information providing system according to the first aspect of the present invention comprises: a server that is provided with: a message transmitting section that transmits to a mobile communication terminal a message to which a term of validity is attached and that instructs that a predetermined application in the mobile communication terminal be started up; and an information transmitting section that transmits to the mobile communication terminal predetermined information that is requested by the started up application in the mobile communication terminal; and a mobile communication terminal that is provided with: a receiving section that receives a message instructing that an application transmitted by the server be started up; a storage section that stores an application that requests the server for predetermined information and receives the information; a control section that, when the receiving section receives the message, determines whether or not a term of validity is attached to the message, and, based on the term of validity, starts up the instructed application from among the applications that are stored in the storage section." Satake at ¶[0076]:

'117 Claims	Exemplary Disclosures
	"The mobile communication terminal according to the third aspect of the present invention comprises: a storage section that stores a plurality of applications; a receiving section that receives message that contains a startup instruction for a specific application and information on that application; a storage section that, for each of the stored applications, stores the application information and also stores information that shows whether or not the application is able to be started up by the startup instruction; and a control section that, when the receiving section receives the message instructing that the application be started up, determines whether or not startup of the application is possible by collating the application information contained in the message with the information that shows whether or not the application in the storage section is able to be started up, and controls a startup of the application based on a result of the determination." Satake at ¶[0136]: "The system shown in FIG. 1 is formed by a stock price information server 1, an SMS server 6 that transmits an SMS message from the stock price information server 1 to a user, a user terminal 7 serving as a mobile communication terminal belonging to the user, and a mobile exchange system 8 that relays communication between the stock price information server 1, the SMS server 6, and the user terminal 7. The mobile exchange system 8 and the user terminal 7 are connected by two transmission paths, namely, a traffic channel that is used for data communication and a paging channel that is used for SMS (i.e., PUSH messages) communication." *See also* Satake at Fig. 1. Satake at ¶[0152]: "FIG. 13, information delivered by a contents server 101 toward a mobile telephone 102 is received by an SMS server 104 via a packet network 103. If

'117 Claims	Exemplary Disclosures
	the mobile telephone 102 is a CDMA type of telephone, the SMS server 104 transmits a PUSH message over the paging channel and the transmitted PUSH message is received by the mobile telephone 102. In this case, even if the mobile telephone 102 is conducting data communication or make a telephone call over the traffic channel in parallel with the transmission of the PUSH message, the message reception processing can be performed. When the mobile telephone 102 receives the PUSH message, it starts up an application that corresponds to this PUSH message, and for data communication over the traffic channel, commences communication with the contents server 101 via a packet serving data node (PSDN) 105. Note that, in the present embodiment, SMS is used for the transmission and reception of messages; however, it is also possible to use e-mail instead of SMS." Satake at ¶[0154]: "When a control section 123 receives a PUSH message via an antenna 121 and a radio section 122, it starts up an SMS launcher 124. The control section 123 carries out the integrated control of a speech processing section 125, an operation section 126, a display section 127, the SMS launcher 124, a mailer 128, and an application section 129. A plurality of applications are stored in the application section 129." Satake at ¶[0155]: "When a control section 123 receives a PUSH message via an antenna 121 and a radio section 122, it starts up an SMS launcher 124. The control section 123 carries out the integrated control of a speech processing section 125, an operation section 126, a display section 127, the SMS launcher 124, a mailer 128, and an application section 129. A plurality of applications are stored in the application section 129."

'117 Claims	Exemplary Disclosures
	See also Satake at ¶¶[0157]-[0159].
[1g] to, for each received message, map the application identifier in the message to a software process corresponding to the application identifier, and forward the application data in the message to the software process via a secure interprocess communication service.	The references below disclose and/or render obvious this limitation. For example, see the following passages and/or figures, as well as related disclosures: Huskins at 5:26-33: "The message from the DVB MPE 26 is provided to a DVB MPE receiver 42. The receiver 42 communicates with an IP multicast module 40 and a unidirectional messaging server 38. The server 38 breaks down the message to determine whether a service identifier was included in the data stream. If so, the message is forwarded to an appropriate agent designated to receive messages with particular service identifiers . A disk management agent 44 may be coupled to a disk drive controller 47 in turn coupled to a storage device 43 that may, for example, be a hard disk drive. The disk management agent 44 may be addressed as a message recipient by the client disk management server 20. The disk management agent 44 may cause processes to be undertaken through the controller 47 that in turn control the use of the storage device 43. This control may include determining what information is stored on the storage device 43 and how that information is stored on the storage device 43." Huskins at 7:19-26: "However, if the particular client 12 is the intended recipient, the server 38 checks the message's service identifier and passes the message to the correct agent 44, 46 or 48, as indicated in block 64. The message is then delivered to the appropriate agent 44 as indicated in block 66. In the agent 44, the information is parsed and passed to an appropriate process for handling as indicated in block 68." Rakic at [0050]: "When client device 102 receives the secure push message, client device 102 may validate the secure push message based on the electronic signature. If client device

'117 Claims	Exemplary Disclosures
	102 is able to validate the secure push message, client device 102 may send a response to administrator device 104 to indicate that client device 102 has successfully received and/or validated the secure push message." Rakic at [0051]: "Depending on the implementation, client device 102 may perform a specific task in accordance with contents of the secure push message. For example, client device 102 may set parameters that are related to an application in client device 102, update firmware, reset parameters that are associated with subscription services, set security parameters, etc." Ellison at 6:1-26: "The system memory 140 stores system code and data. The system memory 140 is typically implemented with dynamic random access memory (DRAM) or static random access memory (SRAM). The system memory 140 includes the accessible physical memory 60 (shown in FIG. 11B). The accessible physical memory includes a loaded operating system 142, the isolated area 70 (shown in FIG. 1B), and an isolated control and status space 148. The loaded operating system 142 is the portion of the operating system that is loaded into the system memory 140. The loaded OS 142 is typically loaded from a mass storage device via some boot code in a boot storage such as a boot read only memory (ROM). The isolated area 70, as shown in FIG. 1B, is the memory area that is defined by the processor 110 when operating in the isolated execution mode. Access to the isolated area 70 is restricted and is enforced by the processor 110 and/or the MCH 130 or other chipset that integrates the isolated area functionalities. The isolated control and status space 148 is an input/output (I/O)-like, independent address space defined by the processor 110 and/or the MCH 130. The isolated control and status space 148 contains mainly the isolated execution control and status registers. The isolated control and status space 148 does not overlap any existing address space and is accessed using

'117 Claims	Exemplary Disclosures
	the isolated bus cycles. The system memory 140 may also include other programs or data which are not shown." Ellison at 8:25-32: "FIG. 2 is a diagram illustrating a secure platform 200 according to one embodiment of the invention. The secure platform 200 includes the OS nub 16, the processor nub 18, a key generator 240, a hashing function 220, and a usage protector 250, all operating within the isolated execution environment, as well as a software environment 210 that may exist either inside or outside the isolated execution environment." Ellison at 8:66-9:6: "The processor nub 18 includes a master binding key (BK0) 202. The BK0 202 is generated at random when the processor nub 18 is first invoked, i.e., when it is first executed on the secure platform 200. The key generator 240 generates a key operating system nub key (OSNK) 203 which is provided only to the OS Nub 16. The OS nub 16 may supply the OSNK 203 to trusted agents, such as the usage protector 250." Ellison at 9:28-62: "The software environment 210 may include a plurality of subsets (e.g., subset 230). The usage of the software environment 210 or the usage of the subset 230 is protected by the usage protector 250. The usage protector 250 uses the OSNK 203 to protect the usage of the subset 230. The software environment 210 may include an operating system (e.g., a Windows operating system, a Windows 95 operating system, a Windows 98 operating system, a Windows NT operating system, Windows 2000 operating system) or a data base. The subset 230 may be a registry

'117 Claims	Exemplary Disclosures
	in the Windows operating system or a subset of a database. Elements can be implemented in hardware or software. "The subset 230 is hashed by the hashing function 220 to produce a first hash value 206 and a second hash value 312. One way to detect intrusion or modification of the subset 230 is to compare the state of the subset before and after a time period. The first and second hash values 206 and 312 are typically generated at different times and/or at different places. "The usage protector 250 is coupled to the key generator 240 to protect usage of the software environment 210 or the subset 230, using the OSNK 203. The usage protection includes protection against unauthorized reads, and detection of intrusion, tampering or unauthorized modification. If the two hash values are not the same, then the usage protector 250 knows that there is a change in the subset 230. If it is known that this change is authorized and an updated hash value has been provided, the usage protector 250 would merely report the result. Otherwise, the usage protector 250 may generate an error or a fault function. When a user is notified of the error, fault condition, he or she would know that the subset 230 has been tampered, modified. The user may take appropriate action. Therefore, the usage of the subset 230 is protected." *See also* Ellison at Figs. 1A-1C, 2. Shenfield at [0024]: "Service provider 120 communicates over wireless network 130 with a push client 140 that is located on a mobile device. Push client 140 will be described in more detail below. Push client 140 receives the content that is being delivered from content provider 110 and can communicate the content with a client application 150, which ultimately consumes the content."

'117 Claims	Exemplary Disclosures
	Shenfield at [0205]-[0206]: "Microprocessor 2238, in addition to its operating system functions, preferably enables execution of software applications on the mobile station. A predetermined set of applications that control basic operations, including at least data and voice communication applications for example, will normally be installed on mobile station 2200 during manufacturing. Other applications could be installed subsequently or dynamically. "A preferred software application may be a personal information manager (PIM) application having the ability to organize and manage data items relating to the user of the mobile station such as, but not limited to, e-mail, calendar events, voice mails, appointments, and task items." Wakasa at ¶[0040]: "The server device 200 generates a transmitted message including parameter information and the identification information of an application to be executed by the portable terminal device 100 on the basis of the parameter information. Then, the server device 200 transmits the transmitted message to the portable terminal device 100 via the network 300. The parameter information is typically information used for requesting the application of the portable terminal device 100 to display information desired by the user or carry out predetermined processing. For example, the parameter information is information requesting the application of the portable terminal device 100 to display "Hello" on the display screen. The portable terminal device 100 executes the application corresponding to the identification information of the application included in transmitted message by using the parameter information as input data. For example, by using the parameter information requesting the application of the portable terminal device 100 to display "Hello" on the display screen as input data, the portable terminal device 100 executes the application corresponding to the identification information, displaying "Hello"

'117 Claims	Exemplary Disclosures
	on the display screen of the portable terminal device 100. As an alternative, the server device 200 holds an application to be provided to the portable terminal device 100 and, in accordance with a request made by the portable terminal device 100, the server device 200 transmits the application to the portable terminal device 100 via the network 300." Wakasa at ¶[0051]: "FIG. 2 is a block diagram showing the functional configurations of the portable terminal device 100 and the server device 200. As shown in FIG. 2, the portable terminal device 100 includes a transmitted message receiver section 102, an application determination section 104, an application-transmission request sender section 106, an application-transmission request generation section 108, an application receiver section 110, a display section 112 and an application storage section 114." Wakasa at ¶[0052]: "The transmitted message receiver section 102 is a communication interface such as a communication line, a communication circuit, a communication device or the like. The transmitted message receiver section 102 has a function for receiving a transmitted message from the server device 200. A transmitted message received by the transmitted message receiver section 102 is passed on to the application determination section 104. As described above, a transmitted message received from the server device 200 includes parameter information and the identification information of an application to be executed by the portable terminal device 100 on the basis of the parameter information." *See also* Wakasa, Fig. 2. Wakasa at ¶[0053]:

'117 Claims	Exemplary Disclosures
	"Details of a transmitted message received from the server device 200 are described by referring to FIG. 3 as follows. FIG. 3 is an explanatory diagram showing one example of a transmitted message received from the server device 200. As shown in the figure, the transmitted message includes transmitting parameter 150, a header 152, and a footer 160. The transmitting parameter 150 includes a URL 154, an ID156, parameter information (Param) 158. The Param 158 is the parameter information explained earlier. The URL 154 and the ID 156 are identification information used for identifying an application to be executed on the basis of the Param 158. Applications of the portable terminal device 100 have been stored in the application storage section 114 in a state of being identifiable by using the ID 156 or the URL 154." Wakasa at ¶[0054]: "The information used for identifying an application may be dependent on the specifications of the portable terminal device 100. For example, in the portable terminal device 100 conforming to the specifications set by a mobile communication business company D, the application is identified by using the URL 154. In the portable terminal device 100 conforming to the specifications set by mobile communication business companies K and V, on the other hand, the application may be identified by using the ID 156. In this embodiment, both the URL 154 and the ID 156 are included in a transmitted message so that each of the portable terminal devices 100 manufactured by the mobile communication business companies D, K and V with specifications different from each other is capable of determining whether or not an application exists in the portable terminal device 100 as will be described later." *See also* Wakasa, ¶[0086]. Wakasa at ¶[0084]:

'117 Claims	Exemplary Disclosures
	"The controller 124 is a section for receiving a transmitted message transmitted by the non-contact-communication IC chip 120 via the second communication section, executing an application corresponding to identification information included in the transmitted message and displaying a result of the execution on a display screen of the portable terminal device 100. More specifically, when the controller 124 receives a transmitted message from the non-contact-communication IC chip 120, the controller 124 passes on the transmitted message received from the non-contact-communication IC chip 120 to the acceptance application 126. The acceptance application 126 then provides parameter information to the application corresponding to the identification information of the application included in the transmitted message. The identification information, such as ID 156 or URL 154, is used by the acceptance application 126 to determine as to whether or not the application corresponding to the identification information has been stored in the application storage section 114." Satake at ¶[0069]: "The mobile communication terminal according to a first aspect of the present invention comprises: a receiving section that receives a message; a storage section that stores an application that requests a server for predetermined information and receives the information; and a determining section that, when the receiving section receives a message that instructs that a predetermined application be started up, determines whether or not a term of validity is attached to the message, and , based on the term of validity, starts up the instructed application that is stored in the storage section." *See also* Satake at ¶¶[0073]-[0080]. Satake at ¶[0076]:

'117 Claims	Exemplary Disclosures
	"The mobile communication terminal according to the third aspect of the present invention comprises: a storage section that stores a plurality of applications; a receiving section that receives message that contains a startup instruction for a specific application and information on that application; a storage section that, for each of the stored applications, stores the application information and also stores information that shows whether or not the application is able to be started up by the startup instruction; and a control section that, when the receiving section receives the message instructing that the application be started up, determines whether or not startup of the application is possible by collating the application information contained in the message with the information that shows whether or not the application in the storage section is able to be started up, and controls a startup of the application based on a result of the determination." Satake at ¶[0142]: "When the user terminal 7 receives the aforementioned SMS message, firstly, the first two bytes of the text of the SMS message are recognized, and a determination is made as to whether they are a binary code showing the message is a special message (step S1—the word step is omitted hereinafter). If the message is not a special message, normal SMS message reception processing is performed (S2). If the binary code does show that the message is a special message, the application code to be started up is read, and a determination is made as to whether or not the application code is registered on the application list shown in FIG. 8 that is registered at the time the application is installed (S3). If the application code to be started up is not registered on the application list, the application is not started up and the reason why it is not started up is recorded in the log (S6). Next, a determination is made as to whether or not a term of validity is attached to the startup message, and whether or not the startup message is received within this term of validity (S4). If the startup message is received within this term of validity, an application that is registered on the application list is started up (S5). If the startup message is not received within the

'117 Claims	Exemplary Disclosures
	term of validity, the application is not started up and the reason why it is not started up is recorded in the log (S6)." Satake at ¶[0152]: "FIG. 13, information delivered by a contents server 101 toward a mobile telephone 102 is received by an SMS server 104 via a packet network 103. If the mobile telephone 102 is a CDMA type of telephone, the SMS server 104 transmits a PUSH message over the paging channel and the transmitted PUSH message is received by the mobile telephone 102. In this case, even if the mobile telephone 102 is conducting data communication or make a telephone call over the traffic channel in parallel with the transmission of the PUSH message, the message reception processing can be performed. When the mobile telephone 102 receives the PUSH message, it starts up an application that corresponds to this PUSH message, and for data communication over the traffic channel, commences communication with the contents server 101 via a packet serving data node (PSDN) 105. Note that, in the present embodiment, SMS is used for the transmission and reception of messages; however, it is also possible to use e-mail instead of SMS." Satake at ¶[0154]: "When a control section 123 receives a PUSH message via an antenna 121 and a radio section 122, it starts up an SMS launcher 124. The control section 123 carries out the integrated control of a speech processing section 125, an operation section 126, a display section 127, the SMS launcher 124, a mailer 128, and an application section 129. A plurality of applications are stored in the application section 129." Satake at ¶[0155]:

'117 Claims	Exemplary Disclosures
	"When a control section 123 receives a PUSH message via an antenna 121 and a radio section 122, it starts up an SMS launcher 124. The control section 123 carries out the integrated control of a speech processing section 125, an operation section 126, a display section 127, the SMS launcher 124, a mailer 128, and an application section 129. A plurality of applications are stored in the application section 129." *See also* Satake at ¶¶[0157]-[0159]. Ogawa at 5:59-6:26: "In the exemplary embodiment, if a start instruction is issued by the client starting unit 31 at the client site 3, the shared key encrypted data will be received by the receive unit 32 from the database server 2 via network 4. This received shared key encrypted data will be decrypted using the shared encryption key 52 which was supplied to decryption unit 33 and beforehand stored on the removable storage 5. This decrypted data is then expanded by a decompression (expansion) unit 34 to convert the data back to the original format. Data that was expanded by the decompression unit 34 is re-encrypted by the encryption unit 35, using an inherent encryption key 53 that was stored on the removable storage 5. This inherent encryption key 53 is generated from an inherent identification code that was assigned during manufacturing and stored on the internal memory device of the removable storage 5. Then, data encrypted using this inherent encryption key 53 (hereinafter referred to as inherent key encrypted data) is stored on the storage device 30 via a saving unit 36. After the program 51 has been loaded to the client site 3, if a start instruction is issued by the display start unit 37, the decryption unit 38 will decrypt the inherent key encrypted data stored on the storage device 30, using the inherent encryption

'117 Claims	Exemplary Disclosures
	key 53 stored on the removable storage 5. Then, the decrypted data are supplied to the display unit 39. As a result, the data which are stored in the database server 2 after receiving the data from the data input site 1 via the network 4 will be displayed on the client site 3. The decompression unit 34 can be performed before or after encryption. Data decrypted by the decryption unit 33 are encrypted with the inherent encryption key 53 in a compressed state and stored on the storage device 30. In order to display this encrypted data, the data will be decrypted by the decryption unit 38 with the inherent encryption key 53, then expanded with the decompression unit 34 and displayed by the data display 39." Cohen at [0072]: [0072] The encrypted payload data field 704 is a variable length field that actually contains the encrypted form of the notification, as well as potentially some encrypted header fields. For example, the field 704 may include a content type field as well as a Wireless Application Protocol (also called "WAP") application ID field. The content type field may identify the presence of XML in the multipart message, while the WAP application ID field identifies the application the device should call to process the push message. The padding field 705 ranges from zero to 255 bytes and is often used when cryptographic algorithms require the cipher text to be a certain length or, regardless of the cryptographic algorithm, padding is needed to terminate the cipher text on a 4-byte boundary. The padding length field 706 is an 8-bit value that specifies the length of the padding field 704 in bytes. The payload ID field 707 (also called the "next field" in the ESP standard) identifies the type of data included in the payload data field 703. The payload data field 704, the padding field 705, the padding length field 706, and the payload ID field 707 are all encrypted. The payload data field 704 represents an example of the notification message 502 of the push message 500. ESP object 700 may also include

'117 Claims	Exemplary Disclosures
	authentication data field 708 that includes a digital signature for authenticating the push initiator 601. Novack at Abstract: "Inter-process communications between a first process and a second process are secured. A first process recognizes that inter-process communications must be initiated with a second process. A first computer that supports the first process submits a request to initiate inter-process communications with a second computer that supports the second process. The second computer replies to a request to initiate inter-process communications by replying to the first computer with a request for information of a digital certificate that authenticates the first process." Novack at [0013]: "The present invention enables secure inter-process communications. The implementation of secure inter-process communications will be described in the context of a client-server network and a peer-to-peer network. Of course, the present invention is also applicable to environments other than the networks described herein, including any network environment that includes inter-process communications between different processes running on one or more computers." Novack at [0015]: "According to an aspect of the present invention, a method is provided for securing inter-process communications between a first process and a second process. The method includes receiving, from a device that supports the first process, a request to initiate inter-process communications. The method also includes replying to the device with a request for information of a digital certificate that authenticates the identity of the first process."

'117 Claims	Exemplary Disclosures
	Novack at [0021]: "According to an aspect of the present invention, a computer-readable medium is provides that stores a program that secures inter-process communications between a first process and a second process. The computer-readable medium includes a receiving code segment that receives, from a device that supports the first process, a request to initiate inter-process communications. The computer-readable medium also includes a digital certificate requesting code segment that replies to the device with a request for information of a digital certificate that authenticates the identity of the first process." *See also* Novack at [0027]. Novack at [0033]: "FIG. 1 shows an exemplary method for securing inter-process communications. The process starts at S100, and a first process begins to run on a first computer at S102. A determination is made that the services of a second process are needed at S105, and the first process requests initiation of an inter-process communications session at S110. The request at S110 marks the beginning of inter-process communications. The first computer may send the request to a second computer that runs a second process in the same network. The first computer receives an acknowledgement and a request for authentication at S115. In the example above, the second computer analyzes the initial request and generates an acknowledgement and request for authentication that is received by the first computer at S115." Novack at [0034]: "The analysis on the second computer may be performed by the requested process or another process. As an example, the second computer may analyze all incoming communications requests and demand authentication of all requesters. The second computer may demand authentication of only requestors requesting inter-process

'117 Claims	Exemplary Disclosures
	communications sessions. Of course, the second computer may demand authentication of requestors for only one or more particularly sensitive processes among many. The analysis at the second computer may be performed by the requested process or by a secondary security program that performs security analysis for all communications or all inter-process communications. Accordingly, when the request is initially received, the second computer determines that authentication is necessary before an inter-process communications session can proceed." Novack at [0035]: "In one example, the first process is running on an application server that provides application services over the internet to requesting clients. The application server interacts with a management server that manages multiple databases as a back-end service. The first process may be run when a user contacts an application server and downloads presentation layer elements of an application to interact with the application server. The application includes instructions to invoke additional processes for predetermined user requests. Accordingly, the application will request the services of a second process at S110." Novack at [0037]: "Alternatively, the inter-process communications may occur on a single computer. For example, dynamic data exchange (DDE) is an inter-process communications system built into the Macintosh, Windows and OS/2 operating systems that enables two running applications to share the same data." Novack at [0038]: "At S120, the computer that supports the first process forwards information of a first digital certificate to the computer that supports the second process. The computer that supports the first process forwards a first decryption key to the second process at S125. At S130, the computer that supports the first process receives digital

'117 Claims	Exemplary Disclosures
	certificate information from the computer that supports the second process, and at S135 the computer that supports the first process receives a second decryption key from the computer that supports the second process. Accordingly, the computer that supports the first process and the computer that supports the second process exchange keys at S125 and S135 respectively. As noted above, the logic for the secure inter-process communications may be included in the processes that interact in the inter-process communications. Alternatively, the logic may be provided by secondary authentication processes on the computers." Novack at [0042]: "Digital certificates provide numerous benefits, including authentication of source and security of communications. The authentication of source results from the certification authority certifying that the public key is bound to the source. The communications security results from the strong encryption provided using a private/public key pair. The security can be used to exchange symmetric encryption keys that can be used to more efficiently communicate by the sender and recipient of messages. According to an aspect of the present invention, the source may be a process, as well as the device and/or an owner. The owner may be an individual assignee of the public key and/or an entity such as a corporation or organization. In any case, one or more digital certificates are used to secure the communications channel for inter-process communications, according to an aspect of the present invention. As a result of the present invention, a strong assertion of identity is made from either side of a communications channel which can then be used to support authorization and audit needs. Accordingly, the use of digital certificates for inter-process communications provides the benefits of integrity and confidentiality to inter-process communications." Novack at [0043]:

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 246 of 442

'117 Claims	Exemplary Disclosures
	"Confidentiality of the transmission is accomplished by encrypting it with the message recipient's public key, as only the holder of the associated private key can decrypt the message. The authentication of source is assured by a "digital signature" mechanism, which is generated by encrypting a secure digest of the message with the private key of the sender. A digital certificate may be stored on any type of computer-readable media, including hard drives, smart cards, flash memory, floppy memory and floppy drives. The digital certificate is installed on a computer and made available to provide secure communications. The digital certificate is retrieved and forwarded to a recipient when a particular process is run and inter-process communications are initiated with the recipient. The digital certificate may be used as a generic authentication tool for all communications involving a particular device or user (i.e., including all inter-process communications), or may be used for only one or more predetermined processes involved in inter-process communications. As an example of the use of digital certificates, according to the secure sockets layer protocols, computers exchange digital certificates to secure communications and authenticate sources and destinations." Novack at [0044]: "FIG. 2 shows an exemplary network architecture for secure inter-process communications. As shown, a **computer** 201 communicates through a **network** 220 with an **application server** 230. As an example, the **application server** 230 may be a web server that provides internet applications for users. The user downloads a web page onto the **computer** 201 when the user contacts an internet address corresponding to the desired internet application. Typically, the web page is accessed and downloaded using hyper text transport protocol (HTTP)." Novack at [0048]: "The **management server** 250 may require authentication of the **application server** 230 for each instance of inter-process communications. In this regard, even if

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 247 of 442

'117 Claims	Exemplary Disclosures
	the management server's 250 only function is to process requests for the application server 230, the management server 250 may need to authenticate the application server for each request for inter-process communications when, for example, the application server 230 widely interacts with the public and is therefore subject to unknown risks. The management server 250 may request verification of the application server's 230 address, the particular requesting function, and any other information that will help authenticate the request. Of course, the management server 250 may demand a digital certificate that is bound to the application server 230 or even a particular process that runs on the application server 230." Novack at [0049]: "Digital certificates are provided to the application server 230 and/or the management server 250 by a certification authority upon the request and verification of, e.g., the operator of the application server 230 and the management server 250. However, the digital certificates may also be bound to one or more individual processes on the application server 230 or the management server 250, so that inter-process communications may be authenticated as originating from an authorized process, and not just a predetermined device or user. Alternatively, the digital certificate may be used as appropriate to secure different types of communications, including inter-process communications." *See also* Novack at [0050]-[0053] (further discussing the digital certificates referenced in [0049]). Novack at [0060]: "An exemplary use of secure inter-process communications includes securing so-called secure networks. In this regard, secure networks often focus security efforts on preventing impostors from gaining access to the network. For example, a secure local computer network may route all communications through a firewall server that

'117 Claims	Exemplary Disclosures
	examines all incoming and outgoing communications to ensure they do not present a security risk. However, a lower level of security is provided for communications between addresses within the secure network. In particular, security for communications between addresses in a network is often limited to source and destination authentication. Accordingly, once an impostor gains access to the secure network, the impostor can wreak havoc on the devices in the network. However, using secure inter-process communications according to the present invention, each inter-process communications session may be secured, even between devices in the network." Novack at [0061]: "As another example, inter-process communications between devices on an internal network for a corporation can be secured using the present invention. Digital certificates can be installed for one or more processes on particular devices on the network such that inter-process communications are only enabled for the specified processes and/or specified devices." Novack at [0062]: "Additionally, as noted above, the use of secure inter-process communications according to the present invention is particularly well-suited to peer-to-peer networks, where devices are often connected over insecure public networks. In this regard, if a device in a peer-to-peer network only accepts communications from other devices which have digital certificates installed for specified processes, inter-process communications will be authenticated to ensure that unauthorized processes are not being propagated through the network." Novack at [0063]: "Although the invention has been described with reference to several exemplary embodiments, it is understood that the words that have been used are words of

'117 Claims	Exemplary Disclosures
	description and illustration, rather than words of limitation. Changes may be made within the purview of the appended claims, as presently stated and as amended, without departing from the scope and spirit of the invention in its aspects. Although the invention has been described with reference to particular means, materials and embodiments, the invention is not intended to be limited to the particulars disclosed; rather the invention extends to all functionally equivalent structures, methods, and uses such as are within the scope of the appended claim." Novack at [0064]: "The device in FIG. 4 may include multiple processes that communicate with each other using the secure inter-process communications of the present invention. Additionally, although the terms "client" and "server" are used to refer to devices, the devices may be defined and distinguished by a set of software instructions installed as a pre-packaged program. Further, while examples of the uses of the present invention include secure client/server networks and peer-to-peer networks, secure inter-process communications can be used to enhance security in any inter-process communications environment. Further, while the embodiment shown in FIG. 2 only shows a single computer 201 and a single application server 230, secure inter-process communications can be used to enhance security for multiple application servers 230 belonging to a single entity or multiple users using computers 201. Additionally, while networks 220 and 240 are separately designated in FIG. 2, the networks may be the same, such as an internet protocol (IP) network. Further, while a certification authority is discussed in the context of the present invention, a certification authority is not shown in the figures. In this regard, a certification authority may communicate over the network 220 and/or the network 240 to provide digital certificates to the application server 230 and the management server 250. Of course, the digital certificates may be provided by the certification authority as portable memory modules that are installed on the application server 230 or the management server 250. Furthermore, the entity that provides the application

'117 Claims	Exemplary Disclosures
	server 230 and/or the management server 250 may also act as a certification authority for its customers using the computers 102." *See also* Novack at [0054]-[0059] (further describing the device in [0064]). Okonnen at [0020]: "In an embodiment according to the present invention, the network may further comprise update agent provisioning information stored in the electronic device. The update agent provisioning information may comprise a device server URL, an index of provisioned update agents, a security key, and electronic device related information. The device server URL may provide references to servers hosting updates to be downloaded. The updates may be compatible with update agents currently available and provisioned in the electronic device." Okonnen at [0047]-[0048]: "FIG. 1 is a block diagram an electronic device network 105 according to an embodiment of the present invention. In the electronic device network illustrated in FIG. 1, an electronic device, for example mobile handset 107, may be communicatively coupled to at least one of a plurality of device servers, such as device server A 135 and device server B 137, for example. "The electronic device, for example, mobile handset 107, may be capable of employing one of several available update agents, for example, update agent A 113 and update agent B 115, resident in the electronic device to update software and/or firmware in the electronic device. The electronic device, for example, mobile handset 107, may also capable of provisioning the available update agents, such as for example, update agent A 113 and update agent B 115, with the necessary parameters and data needed to facilitate the respective update operations."

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 251 of 442

'117 Claims	Exemplary Disclosures
	Okonnen at [0052]: "The provisioned data 129 may also be used to store update agent related provisioning information, such as a universal resource locator (URL) of a device server or a management server that may be used to retrieve updates, a security key(s) that may be used to authenticate messages to/from the device server, and selective encryption/decryption of the messages, etc. Each update agent, such as update agent A 113 and update agent B 115, may have corresponding entries in provisioned data 129." Okonnen at [0063]-[0064]: "FIG. 2A is a block diagram 205 illustrating update agent provisioning information 207 associated with an electronic device, for example, mobile handset 107, according to an embodiment of the present invention. The provisioning information may comprising a device server URL 211, an index of provisioned update agents 209, a security key(s) 213, and other related information 215. The device server URL 211 may provide references to remote device servers providing access to updates that may be downloaded. The updates may be compatible with update agents currently available and provisioned in the mobile handset 107. "In an embodiment according to the present invention, the index of provisioned update agent 209 may provide an index value that may be used to compute an address location of a provisioned update agent. For example, if the update agents are located in consecutive blocks of 64 Kbytes each, then an index value of one multiplied by a block size of 64 K and added to an offset value (or preset starting address) provides the address of the provisioned update agent." Okonnen at [0069]: "FIG. 2B is a block diagram 225 illustrating an update agent table 231 stored in non-volatile memory 111 of an electronic device, for example, mobile handset 107,

'117 Claims	Exemplary Disclosures
	according to an embodiment of the present invention. The update agent table 231 in the mobile handset 107 may provide references, such as location addresses and other related information, including information regarding update agent provisioning, to the update agents currently available in the mobile handset 107." Okonnen at [0070]-[0071]: "The update agent table 231 may maintain references to a plurality of available updates and associated update agents. When a new update agent is added, such as via an update of firmware/software in the mobile handset 107, an additional update may also be inserted into the mobile handset 107. The additional update may invoke insertion of a corresponding entry in the update agent table 231 and population thereof with associated relevant information, such as the address location of the new update agent, (i.e., where the new update agent is stored in non-volatile memory 111). An update package may be downloaded and installed in the mobile handset 107 and may result in removal of an existing update agent or replacement thereof by another. "In an embodiment according to the present invention, the update agent table 231 may comprise an array of addresses for all available update agents in the mobile handset." Okonnen at [0072]-[0073]: "FIG. 2C is a block diagram 255 illustrating an update agent table 261 located in non-volatile memory 111 of an electronic device, for example, mobile handset 107, according to an embodiment of the present invention. The update agent table in the mobile handset 107 may contain references to a plurality of update agents currently available and provisioned in the mobile handset. "The update agent table 261 may maintain, for example, a mapping between update agent names, the update agent corresponding address locations in non-volatile

'117 Claims	Exemplary Disclosures
	memory, and the type of updates that the update agents are adapted to process, for all available update agents in the mobile handset 107. The update agent table 261 may also contain optional information regarding provisioning status of the update agents, such as a provisioning flag indicating whether the associated update agent has been provisioned or not." Okonnen at [0075]-[0076]: "The type information may be provided as 'file extensions' in an embodiment according to the present invention. For example, the file extension 'dup' may correspond to updates generated to update generators, such as generator A 139 and generator B 141, as illustrated in FIG. 1. The type field in each entry in the update agent table illustrated in FIG. 2C may be a list of file extensions in an embodiment according to the present invention. "Other structures and schemas are also contemplated for the update agent table 261. The update agent table 261 may map the name of an update agent to the location where the agent resides in memory. The update agent table 261 may also map the provisioning status of an update agent and the type(s) of updates the update agent is adapted to process. Management of the update agents may be facilitated by the mobile handset 107. New table entries for update agents may be added and entries no longer in use may be deleted, etc. Management of the update agents may be supported by client software, such as a device management client." *See also* Okonnen at Figs. 1, 2A. *See also* Okonnen at Figs. 1, 2A. Hashimoto at [0030]: "In order to realize this, the microprocessor has a task state table for assigning an identifier to the task that can take a plurality of states and maintaining a state of the

'117 Claims	Exemplary Disclosures
	task in correspondence to the task identifier, a task register for maintaining the currently executed task identifier, an interface for reading in units of cache lines the program stored in the encrypted state by using a prescribed program key in the external memory when there is a request for the task, an encryption processing unit for generating a decryption key for each cache line according to the program key and decrypting the content read by the interface, a first cache memory formed by a plurality of cache lines having respective tags for maintaining a task identifier corresponding to the decryption key used in decrypting each cache line in the tag, and an access verification unit for comparing the task identifier maintained in the tag of the cache line and the value of the task register and discarding the content of the cache line when these values do not coincide." Hashimoto at [0045]: "Fig. 2 shows a hardware configuration of the microprocessor-A shown in Fig. 1. The processor-A has a core 902, an instruction cache 903, a data cache 904. a fast encryption key processing module (FCM: Fast Ciphering Module) 911 to be shared by these caches, a key value table (KVT) 912, a distribution key management module 913, and a bus interface (BIU) 906, which are arranged within a processor package 200 and interconnected through an internal bus 905." Hashimoto at [0046]: "The core 902, the fast encryption key processing module 911, the key value table 912 and the distribution key management module 913 constitute the protection logic 106 of Fig. 1." Hashimoto at [0047]: "The core 902 has a register array 915 and a task switcher 914. The task switcher 914 executes the task switching, that is the saving and the recovery of the context when the interruption occurs. The context is generated in correspondence to the

'117 Claims	Exemplary Disclosures
	execution of the task, and managed in correspondence to the task. In order to realize this, the task switcher 914 internally has a task state table 921 and a hash calculation unit 922. The task state table 921 maintains the task state in correspondence to the task ID as will be described below. Also, although not shown in the figure, the core 902 issues a notice for change of the context in the register array, a distribution key setting command, etc." Hashimoto at [0048]: "The instruction cache 903 caches the execution codes of the program. In addition to an ordinary cache tag for identifying the address, a tag for identifying a protected task is also attached thereto, and the independence of the application is guaranteed among the internal data of the microprocessor. Note that, although not shown in Fig. 2, the instruction cache 903 internally has an offset table 1207 (see Fig. 9) for describing start addresses (offsets) of the program (execution codes) and the data, as will be described below." Hashimoto at [0104]: "Note that, in the following description, the calculation of the relative address is carried out according to the virtual address, but the judgement of the cache hit/miss is carried out with respect to the physical address. This is because if the virtual address is used for the judgement of the cache hit/miss, the phenomenon called alias in which the same memory block exists in plurality on the cache will occur. This phenomenon becomes particularly problematic when the same physical address region is to be shared by a plurality of tasks. For this reason, the cache hit/miss is judged by the physical address in this embodiment." Hashimoto at [0108]: "Next, the relative address is calculated from the acquired offset value. The relative address is obtained by subtracting the offset value from the memory address

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 256 of 442

'117 Claims	Exemplary Disclosures
	(S503). Note that, when there is an access to the cache memory during the execution of the protected task, the instruction cache 903 carries out the calculation of the relative address in parallel to the access to the cache memory." Hashimoto at [0109]: "The instruction cache 903 sends the obtained relative address value and the task ID to the key value table (KVT) 912 through the internal bus 905. At this point, the request ID is attached. The KVT 912 acquires a value of the key corresponding to the task ID from the task key table 1303, and sends the key value and the offset value to the key generation module 1312 of the fast encryption key processing module (FCM) 911 through the channel 1316. At the same time the KVT 912 sends the offset value, the request ID and the request source ID to the encryption processing module 1313 of the FCM 911 through the channel 1317." Hashimoto at [0110]: "When the key value and the offset value are received from the KVT 912, the key generation module 1312 of the FCM 911 immediately starts the calculation of the key corresponding to that block. Namely, an intermediate value obtained by adding the relative address value to the secret constant Cx of the program is obtained (S504) and the obtained intermediate value is encrypted by using the secret key Kx of the program, to generate the block corresponding key that corresponds to the block of the program (S505)." *See also* Hashimoto at Fig. 2. Fan at 12: The matching conditions in the set Push message control policy may include the following: user identifier, maximum message length threshold, service control,

'117 Claims	Exemplary Disclosures
	content control, and control of the trusted PI. The following five Push message control strategies are introduced separately. Fan at 13: First, the user identifier The purpose of setting the matching condition is to determine, according to the user identifier, that the terminal to which the Push message is to be sent is the setting terminal of the Push message control policy when the Push sending agent performs the Push message control. Fan at 13-14: FIG. 9 is a flowchart of processing a terminal according to a Push message control policy. In this process, it is assumed that a service type of a Push message is a multimedia multimedia message, and the process includes: Step 901: After receiving the Push message sent by the PI, the Push sending proxy sends the Push message to the Push receiving proxy by using a NOTIFY or MESSAGE message in the SIP message. Step 902: After receiving the Push message, the Push receiving proxy returns a 200 OK message to the Push sending proxy to notify the Push sending proxy that the Push message has been successfully received. Step 903: The Push receiving agent checks the service type carried in the Accept-Contact header field in the SIP message carrying the Push message, and determines that the Push message is a multimedia multimedia message. If the terminal does not allow to receive the message of the multimedia multimedia message service in the Push message control policy, the Push receiving agent discards the Push message without sending it to the application module, and the user does not know that a message arrives; if the Push message controls the policy, the terminal If the multimedia MMS service is not allowed to be received, the Push receiving agent

'117 Claims	Exemplary Disclosures
	temporarily stores the Push message; if the Push message control policy, the terminal allows to receive the message of the multimedia MMS service, and then sends the message to the application module for corresponding service processing. Fan at 25: Step 1201: The Push sending proxy receives the Push message from the PI. (Page no. 25) Step 1202: The Push sending proxy compares the matching conditions between the received Push message and the Push message control policy, for example, whether the Push message is too large, whether it contains a keyword that is not desired to be received, whether it is a service that does not want to be currently received, and whether Whether the trusted PI is from the time when the Push message is desired or not desired, whether the set Push message control policy or the like is applied. If the matching condition that is satisfied by the Push message is the Push content that the terminal is allowed to receive, proceed to step 1203; otherwise, if the matching condition that the Push message satisfies is the Push content that the terminal is not allowed to receive, the Push sending proxy discards the Push message, if the Push message satisfies The matching condition is that the terminal does not currently receive the Push content, and the Push sending proxy temporarily stores the Push message. After the operation of discarding or temporarily storing the Push message is performed, step 1206 is directly executed. Step 1203: The Push sending proxy sends the Push message to the SIP/IP core network, and the specific method of sending is that the Push sending proxy carries the Push message in the SIP message and sends the Push message to the SIP/IP core network. Step 1204: The SIP/IP core network sends a Push message to the Push receiving proxy.

'117 Claims	Exemplary Disclosures
	Fan at 26: The terminal sets the Push message control policy, and stores the set Push message control policy locally. After the PI sends the Push message to the Push sending proxy, the Push sending proxy sends the Push message to the terminal, and the terminal controls the policy according to the stored Push message. Process the Push message. FIG. 13 is a flowchart of processing a received Push message by the terminal, where the process includes: Step 1301: The Push sending proxy sends the Push message received from the PI to the Push receiving proxy in the terminal, and the specific method of sending the Push message is that the Push sending proxy carries the Push message in the SIP message header field and sends the Push message to the Push receiving proxy. Step 1302: The Push receiving agent compares the matching conditions between the received Push message and the Push message control policy, such as whether the content is too large, whether the keyword is not expected to be received, whether it is a message that is not currently received, and whether it is from a Trusted PI and so on. If the matching condition that the Push message satisfies is the Push content that the terminal is allowed to receive, step 1303 is continued; otherwise, if the matching condition that the Push message satisfies is the Push content that the terminal is not allowed to receive, the Push receiving proxy discards the Push message, if the Push message is satisfied. The matching condition is that the terminal is not allowed to receive the Push content. The Push receiving agent temporarily stores the Push message. After the operation of discarding or temporarily storing the Push message is performed, step 1304 is directly executed. When the Push receiving agent discards or temporarily stores the Push message, before the step 1304 is directly executed, the Push receiving agent may return a message that the Push message has been deleted to the Push sending agent. This step is an optional step.

'117 Claims	Exemplary Disclosures
	Step 1303: The Push receiving proxy returns an acknowledgement message to the Push sending proxy, and sends the Push message to the application module in the terminal to perform corresponding services. Holbrook at [0010]: "The present invention may be used in a data network to increase the probability of desired data being available at remote network nodes where the nodes do not have continuous access to the network. An association table associates user accounts with messaging nodes. A primary messaging zone is generated, indicating the messaging nodes associated with a recipient user account, at which there is a significant probability that the user may request the collection of messages. During intermittent communications sessions with members of said primary messaging zone, incoming messages are delivered to these messaging nodes, where the messages are proactively buffered in preparation for a user request to collect messages while a messaging node cannot collect from external sources." D-Bus Specification V0.12 "Introduction D-Bus is a system for low-latency, low-overhead, easy to use interprocess communication (IPC). In more detail: D-Bus is low-latency because it is designed to avoid round trips and allow asynchronous operation, much like the X protocol. D-Bus is low-overhead because it uses a binary protocol, and does not have to convert to and from a text format such as XML. Because D-Bus is intended for

'117 Claims	Exemplary Disclosures
	potentially high-resolution same-machine IPC, not primarily for Internet IPC, this is an interesting optimization. D-Bus is easy to use because it works in terms of messages rather than byte streams, and automatically handles a lot of the hard IPC issues. Also, the D-Bus library is designed to be wrapped in a way that lets developers use their framework's existing object/type system, rather than learning a new one specifically for IPC. The base D-Bus protocol is a one-to-one (peer-to-peer or client-server) protocol, specified in the section called "Message Protocol". That is, it is a system for one application to talk to a single other application. However, the primary intended application of the protocol is the D-Bus message bus, specified in the section called "Message Bus Specification". The message bus is a special application that accepts connections from multiple other applications, and forwards messages among them. Uses of D-Bus include notification of system changes (notification of when a camera is plugged in to a computer, or a new version of some software has been installed), or desktop interoperability, for example a file monitoring service or a configuration service. D-Bus is designed for two specific use cases:

'117 Claims	Exemplary Disclosures
	A "system bus" for notifications from the system to user sessions, and to allow the system to request input from user sessions. A "session bus" used to implement desktop environments such as GNOME and KDE. D-Bus is not intended to be a generic IPC system for any possible application, and intentionally omits many features found in other IPC systems for this reason. At the same time, the bus daemons offer a number of features not found in other IPC systems, such as single-owner "bus names" (similar to X selections), on-demand startup of services, and security policies. In many ways, these features are the primary motivation for developing D-Bus; other systems would have sufficed if IPC were the only goal. D-Bus may turn out to be useful in unanticipated applications, but future versions of this spec and the reference implementation probably will not incorporate features that interfere with the core use cases. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

'117 Claims	Exemplary Disclosures
	However, the document could use a serious audit to be sure it makes sense to do so. Also, they are not capitalized." D-BUS commit 7f0990abc74a3c6aa76211a064b57f081690d340

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 264 of 442

tree: 439413dc2e30e5eb44867eff879b8940208f6d3f

- bus/
- cmake/
- dbus/
- doc/
- test/
- tools/
- .gitignore
- acinclude.m4
- AUTHORS
- autogen.sh
- ChangeLog
- ChangeLog.pre-1-0
- ChangeLog.pre-1-2
- cleanup-man-pages.sh
- configure.in
- COPYING
- dbus-1.pc.in
- dbus-win.patch
- Doxyfile.in
- HACKING
- INSTALL
- Makefile.am
- Makefile.cvs
- NEWS
- NEWS.pre-1-0
- NEWS.pre-1-2
- README
- README.win
- update-dbus-docs.sh

'117 Claims	Exemplary Disclosures

android / platform / external / dbus / 7f0990abc74a3c6aa76211a064b57f081690d340 / . / **README**

blob: ec63a98a9f7d18d6b6edf52cba0c04f7aefaf481 [file] [log] [blame]

```
Sections in this file describe:
 - introduction and overview
 - low-level vs. high-level API
 - version numbers
 - options to the configure script
 - ABI stability policy

Introduction
===

D-Bus is a simple system for interprocess communication and coordination.

The "and coordination" part is important; D-Bus provides a bus daemon that does things like:
 - notify applications when other apps exit
 - start services on demand
 - support single-instance applications

See http://www.freedesktop.org/software/dbus/ for lots of documentation,
mailing lists, etc.
```

Cooley at ¶[0015]:

> "When the incoming text message arrives at the user's device it is intercepted, under control of the device processor and an application, such as application 31 (as will be discussed), running on the device. This is possible because application 31 is registered with the device operating system when it was loaded on the device. Application 31 then routes the incoming message to the proper specific application in accordance with the data contained in the text message. Thus, in the situation with the flight reservation to Las Vegas (as discussed above), information pertaining to that flight is routed to a specific application, such as to air travel

'117 Claims	Exemplary Disclosures
	application 22-1 (shown in FIG. 2B), which application is dedicated to air travel data pertaining to the device user." Cooley at ¶[0017]: "When the user touches tile 23-1 all the flights that the user has booked on, for example, AMERICAN™ airlines (AA) are displayed. Touching tile 24-1 then yields the screen shown in FIG. 2E which then displays the information that has been received from an external source. In this case this information would have come typically from the AA server in the manner discussed above. There could be any amount of information and tiles (or layers) to this information, all controlled by application 30 residing on the user's device with the updated information coming, from time to time, via messages from various external locations. Note that in FIG. 2F the gate and status of the flight have been updated. This update could have come as a result of a request from the user or could have come as a text message from the airline as part of an update service for customers. In any event, the message would have gone first to service center 12 and then via SMS broker 13 to the user's device where application 30 would have taken the data and placed it in the proper tile." Cooley at ¶[0018]: "FIG. 3 shows one embodiment of a system for controlling messages on a user device. The incoming SMS message would normally be delivered to level 32 of network stack 30 for subsequent display to device user via, for example, display 16 (FIG. 1). The display would be in text or visual form and would then go into the SMS box in text, graphics or video form as is well-known. However, application 31 intercepts the message because that application is registered with the platform to receive certain SMS messages. This protocol varies from device to device based on operating system. SMS or HTTP transports can be used via the side channel process. SMS has the advantage of being a true "push"

'117 Claims	Exemplary Disclosures
	notification. Application 31 then routes the incoming message, based on the context and content of the data contained in the message to an individual tile, such as to tile 22-1 (FIG. 2B) or even more specifically to tile 24-1 (FIG. 2D), as discussed above." Cooley at ¶[0021]: "FIG. 4 shows one embodiment of a method, such as method 40, for controlling applications resident on a mobile device from a remote location. Process 401 determines if a message is incoming to a particular device. If there is a message available, then process 402 determines if the message contains any code or data that identifies the message as one that is to be delivered to a specific application resident on the device and/or delivered customized in a particular manner. If not, then the message is stored in its normal location, for example in SMS layer 32 (FIG. 3) and delivered without modification." Cooley at ¶[0022]: "If the message does contain a "diversion" code or other data that indicates that the message is to be delivered to another location or delivered in a customized manner, then process 404 delivers the incoming message (or other data) accordingly as discussed above, such that a particular user's tile is customized for a particular application. Customizing could, for example, include one or more of the following: appearance, context, display parameters, including audio, visual, application state change. In some situations, a control command is sent to a user's tile, for example, using an SMS transport protocol such that the command controls the display parameters and/or the display message instead of actual text being sent to the tile."

'117 Claims	Exemplary Disclosures
2. The network system of claim 1, the network message server further to collect and buffer multiple requests to transmit application data to a particular one of the devices.	The references below disclose and/or render obvious this limitation. For example, see the following passages and/or figures, as well as related disclosures: Rakic at [0059]-[0060]: "FIG. 4 is a functional block diagram of client device 102. As shown, client device 102 may include WAP logic 402 and a secure push message client 404. Although not illustrated in FIG. 4, client device 102 may include additional functional components, such as the components that are shown in FIG. 3, an operating system (e.g., Symbian OS, Palm OS, Windows Mobile OS, Blackberry OS, etc.), an application (e.g., an instant messenger client, an email client, etc.), logic for wireless or wire-line communication protocols other than the WAP, etc. "WAP logic 402 may process messages that are sent or received over wireless communication links in accordance with the WAP. For example, when client device 102 and gateway 106 communicate wirelessly, client device 102 and gateway 106 may exchange messages in accordance with WAP-164 or WAP-247." Rakic at [0064]: "WAP logic 502 may process messages that are sent or received over wireless communication links in accordance with the WAP. For example, when administrator device 104 and gateway 106 communicate through a wire-line network, administrator device 104 and gateway 106 may exchange messages in accordance with WAP-189 or WAP-235." Hämäläinen at [0007]-[0008]: "The service profile prepared by the system administrator on the console using the web-based interface can specify Information Push message delivery formats, such as Short Messaging Service (SMS) messages, Multimedia Messaging Service (MMS) messages, emails, and the like. The service profile can specify scheduled

'117 Claims	Exemplary Disclosures
	delivery or delivery triggered by certain specified events. The service profile can specify delivery to the mobile device using a short-range wireless access point or optionally a GSM, GPRS, or 3 G telephone network. Additionally, the service profile can specify backup delivery channels if the specified primary channel fails to send the user the push message. "In another aspect of the invention, a service manager server manages the delivery of push messages to mobile wireless devices having diverse routing, scheduling, and context or event-triggering requirements. The service manager server includes a service profile server for storing a plurality of service profiles of content delivery characteristics. The profiles each specify a bearer, using a specified delivery mode, for pushing a message to one of a plurality of mobile wireless devices of a specified recipient satisfying one of the plurality of service profiles when the one mobile wireless device is located within a coverage area of one of a plurality of access points. The service manager server also includes a content aggregation logic coupled to the service profile server, for polling web servers identified by a content characteristic in a service profile, accessing content from the web servers, and storing the content in a content queue. The service manager server also includes a service logic coupled to the content aggregation logic, the content queue, and the service profile server, for receiving a notification that content has been accessed and handling the accessed content in accordance with the service profiles in the service profile server. The service manager server also includes a context state logic coupled to the service logic and to the plurality of access points, for maintaining context information about the plurality of mobile wireless devices, including whether a specified mobile device can be reached over a communications link. The service manager server further includes a messaging logic coupled to the context state logic for selecting a specified primary bearer to send the push message to the specified mobile device if the specified mobile device can be reached over the primary bearer."

'117 Claims	Exemplary Disclosures
	Hämäläinen at [0020]-[0024]: "The Service Manager 170 pushes content, e.g. the messages 12A and 12B to the user's wireless device 100 based on the characteristics of the context of the device 100 at the respective access points 140A and 140B. The characteristics of the contexts that trigger the Service Manager 170 to push messages 12A and 12B to device 100 are specified by a service profile prepared by the system administrator on the console 175 using e.g. the web-based interface 400, 400′ , or 400″ shown in FIGS. 4, 5, and 6, respectively. "The service profile prepared by the system administrator on the console 175 using the web-based interface 400, 400′ , or 400″ can specify Information Push content and message delivery formats e.g. SMS messages including text and binary messages (for example WAP Push messages), MMS messages, emails, IP data stream or files such as audio, video and/or data content e.g. via FTP or DVB-H, etc. The service profile can specify scheduled delivery or delivery triggered by certain specified events. The service profile can specify delivery to the mobile device 100 using the short-range wireless access point 140 or optionally the wireless telecommunication, such as GSM, GPRS, or 3G telephone network. Additionally, the service profile can specify backup delivery channels if the specified primary channel fails to send the user the push content or message. Further, the system administrator can itself create ad hoc Information Push content and messages. "The Service Manager 170 in FIG. 1 includes Content Aggregation Logic 172, Context State Logic 174, Service Logic 176, and Messaging Logic 178. A more detailed system diagram of the components of the Service Manager 170 is shown in FIG. 2. "The Content Aggregation Logic 172, also referred to herein as the Content Fetcher 172 in FIG. 2, in one embodiment of the present invention, periodically polls

'117 Claims	Exemplary Disclosures
	content servers 190, such as web servers, and checks if content specified by the administrator's web-based console 175 has changed. The web servers 190 and specification of content are defined by the service profile in the Service Profile Server 204. If the content has changed, it will be stored to the Content Queue 205 and a content changed notification will be sent to the Service Logic 176, also referred to as the InfoPush Service Handler 176." Hämäläinen at [0053]-[0054]: "In the Messaging Logic 178, also referred to herein as the Message Sender 178 in FIG. 3, step 322 then puts the information push message into the message queue 208. Step 322 flows to the decision step 324 to determine if the recipient is reachable. If so, then step 326 sends the message via the selected bearer. If the selected bearer is the Bluetooth or Wireless LAN network in coverage area 150, then the access point 140 is used to send the message to the mobile device 100. If the selected bearer is the GSM telephone network, then the GSM access point 186 is used to send the message to the mobile device 100 in the GSM coverage area 180. "If the decision step 324 of FIG. 3 determines that the recipient is not reachable, then the process flows to step 340 to keep the message in the content queue 205 until the message expires or until the recipient becomes reachable. Step 310′ provides input from the context state handler 174. Step 340 then flows to the decision step 342 to determine if a secondary bearer has been defined in the event that the information push message expires. Step 312′ provides an input from the Service Profile Server 204. If a secondary bearer has been defined, then the process flows to step 350 to resend the message using the secondary bearer. Step 350 flows back to step 322 to put the information push message into the message queue 208." *See also* Hämäläinen at Figs. 1-2.

'117 Claims	Exemplary Disclosures
	Mostafa at 732: "Different MMS standardization bodies have recently started working to enrich MMS with the feature of transporting application data. Figure 3 depicts different layers of communication in transporting application data using MMS, showing the vertical interaction within an MMS-capable terminal and an MMS R/S to achieve communication between two applications residing in two different MMS-capable terminals. The figure assumes a simplified scenario, where the terminals belong to the same MMS R/S. Only HTTP is shown there, as it is the mostly used protocol across the terminal-interface. Lower layer protocols, below MMS in the figure, are MMS-specific, and are beyond the scope of this paper. "MMS architecture has been lately extended to include applications [11]. Application is defined as a system element that may interact with an MMS UA or a VASP to transport its data to its peer application, interacting with another MMS UA or a VASP, using MMS. The extension of MMS architecture is depicted in Figure 4, where unrelated entities and interfaces are not shown. Though an application can interact with either an MMS UA or a VASP to transport its data, MMS UA is mostly mentioned in this regard in this paper to cover both for brevity. Though only one application is shown within a mobile terminal and VASP in Figures 3 and 4, there can be multiple applications, as shown in Figure 5. The application interface between an application and an MMS UA is described in Section 4." *See also* Mostafa at Figs. 1, 3; *see generally* Mostafa at 730-732. Santamaria at [0023]: "A method and apparatus for managing notification service connections is described. In one embodiment of the invention, a mobile device (e.g., laptop, palmtop, portable media player, smartphone, multimedia mobile phone, mobile gaming systems such

'117 Claims	Exemplary Disclosures
	as a "Gameboy", etc.) includes one or more notification service clients (e.g., push notification service clients (e.g., push email clients such as .Mac, Microsoft Exchange, push-IMAP, Yahoo! Push, etc., update/upgrade services, news services, weblog services, podcast services, etc.) and/or pull notification service clients (e.g., pull email clients such as POP3, IMAP, etc.)). For each notification service client, the mobile device independently schedules a notification service connection message transmission interval and sets a notification service connection message transmission timer. In addition, a notification service connection message transmission window is associated with each notification service connection message transmission interval and is based on the value of the notification service connection message transmission interval. Upon a notification service connection message transmission timer expiring for a particular notification service client, the mobile device transmits a notification service connection message for that notification service, and also transmits a notification service connection message for each other notification service whose notification service connection message transmission window overlaps that expiring transmission timer. The mobile device resets the notification service connection message transmission timer for each notification service upon transmitting a notification service connection message." Santamaria at [0061]: "At time 47, the mobile device 160 is awakened from the sleep mode 516 from an independent event (represented by independent wake 566). There may be many causes of an independent wake, including user interaction (e.g., the user disables sleep mode), receiving an item of interest (e.g., receiving an email message from the push notification service 110), receiving a phone call and/or text message, etc. Once awake, the mobile device determines whether the time is within any notification service connection message transmission window. For example, time 47 is within the window 552 for the push notification service 110 (which starts at time 45 and ends at time 55) and within the window 550 for the pull notification service 130

'117 Claims	Exemplary Disclosures
	(which starts at time 42 and ends at time 50). Leveraging the fact that the mobile device is awake and within notification service connection message transmission windows, the mobile device transmits a notification service connection message to the push notification service 110 (represented by the beat 582) to refresh the push notification service connection 110, and transmits a notification service connection message to the pull notification service 130 (represented by the check 580) to check for items of interest. In addition, while not illustrated in FIG. 5 for simplicity purposes, the mobile device 160 resets the notification service connection message transmission timer and reschedules the notification service connection message transmission time and wake for the push notification service 110 and the pull notification service 130. At time 49, the mobile device enters sleep mode 518." Isomursu at 22:36-67: "When address information has been added at a **SMS transmission controller** 10, the message is transferred into an **inbox** 11, which tries to send the message, and which has a buffer wherein the message is stored in case the transmission fails. If the transmission fails, the **inbox** 11 re-tries to send the message. When the **DU controller** 8 notices that the radio unit RU is ready to send the message, the message is transferred to a message **transfer running circuit** 12, which adds to the message information relating to the mobile communications system in question, such as validity information (which indicates to which direction the message is going, i.e., from a mobile station to a message service centre or vice versa), processes the address information into a form required by the mobile communications system, and adds to the message the address of the message service centre, as well as the short message identifier (SAPI), and forms from the information to be transmitted, e.g., a digital signal for a **transmitter** 2, and sends the message to the **radio transmitter branch** 2 of the radio unit RU. In the case where the application identifier is placed in bits in the address field ADD, the running **circuit** 12 adds to the message the identifier in question. The **transmitter branch** 2 codes the signal according to the specifications of

'117 Claims	Exemplary Disclosures
	the mobile communications system, and forms, on the basis of the signal it receives from the running circuit 12, the frames to be transmitted, which the transmitter sends using radio communication to the short message service centre SM-SC, wherefrom they are sent further to the receiver (see FIG. 1). In the transmitter branch 2, the message is processed according to the mobile communications system, e.g., coding, interleaving, ciphering, burst forming, modulating, and transmission." Okonnen at [0047]-[0048]: "FIG. 1 is a block diagram an electronic device network 105 according to an embodiment of the present invention. In the electronic device network illustrated in FIG. 1, an electronic device, for example mobile handset 107, may be communicatively coupled to at least one of a plurality of device servers, such as device server A 135 and device server B 137, for example. "The electronic device, for example, mobile handset 107, may be capable of employing one of several available update agents, for example, update agent A 113 and update agent B 115, resident in the electronic device to update software and/or firmware in the electronic device. The electronic device, for example, mobile handset 107, may also capable of provisioning the available update agents, such as for example, update agent A 113 and update agent B 115, with the necessary parameters and data needed to facilitate the respective update operations." *See also* Okonnen at Fig. 1. Staack at [0009]: "In accordance with one aspect of the invention, the instant messaging service is enhanced by providing a storing capability for messages. When the intended receiver of the message is presently unable to receive the message because he is e.g. not on-line, busy and/or not reachable by the network, e.g. by the proxy server of the receiving user, because of any other reason, the message may be stored. This saving

'117 Claims	Exemplary Disclosures
	of the message enables its later delivery to the receiving user when this user is able to receive the message, e.g. after re-attachment to the network. No connection for bi-directional communication needs to be established." Staack at [0021]: "FIG. 1 shows a first embodiment of the invention and illustrates a case where a message is to be sent from a first network element 1 (user A) to a second network element 3 (user B). The network elements 1, 3 are, in the 25 present embodiment, client or user equipments such as terminals. In the present example, the network element 1 (user A) is an equipment trying to send a message (e.g. "MESSAGE user b@sonera.com" addressed to user b@sonera.com) to the receiving element 3 (user B) which is presently out 30 of reach, e.g. switched-off, busy, or located in a nonsupported area, or the like. The connection request of network element 1 is handled by a network element 2 which may be a server (such as a proxy server) which provide e.g. CSCF (Call Server Control Function), and/or is a home location server which contains a database storing information on the present locations of network element 3 and further network elements, reachability thereof, and the like." Fan at 13-14: FIG. 9 is a flowchart of processing a terminal according to a Push message control policy. In this process, it is assumed that a service type of a Push message is a multimedia multimedia message, and the process includes: Step 901: After receiving the Push message sent by the PI, the Push sending proxy sends the Push message to the Push receiving proxy by using a NOTIFY or MESSAGE message in the SIP message.

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 277 of 442

'117 Claims	Exemplary Disclosures
	Step 902: After receiving the Push message, the Push receiving proxy returns a 200 OK message to the Push sending proxy to notify the Push sending proxy that the Push message has been successfully received. Step 903: The Push receiving agent checks the service type carried in the Accept-Contact header field in the SIP message carrying the Push message, and determines that the Push message is a multimedia multimedia message. If the terminal does not allow to receive the message of the multimedia multimedia message service in the Push message control policy, the Push receiving agent discards the Push message without sending it to the application module, and the user does not know that a message arrives; if the Push message controls the policy, the terminal If the multimedia MMS service is not allowed to be received, the Push receiving agent temporarily stores the Push message; if the Push message control policy, the terminal allows to receive the message of the multimedia MMS service, and then sends the message to the application module for corresponding service processing. Assmann at Abstract: "An electronic mail ("e-mail") system is described that enforces authentication or encryption to/from Mail Transfer Agents and from Mail User Agents. In one embodiment, for example, an electronic mail (e-mail) system is described that comprises: a message transfer agent (MTA) available for a client to connect to; program logic for authenticating the client, through use of a certificate, in order to identify the client as approved for establishing a secured connection with the MTA; program logic for establishing a secured connection between the client and the MTA in instances where the client can be authenticated; program logic for testing encryption strength of the secured connection; and program logic for terminating the secured connection in instances where the secured connection has inadequate encryption strength."

'117 Claims	Exemplary Disclosures
	Assmann at 2:33-45: "The MTA, which is responsible for queuing up messages and arranging for their distribution, is the workhorse component of electronic mail systems. The MTA "listens" for incoming e-mail messages on the SMTP port, which is generally port 25. When an e-mail message is detected, it handles the message according to configuration settings, that is, the settings chosen by the system administrator, in accordance with relevant standards such as Request For Comment documents (RFCs). Typically, the mail server or MTA must temporarily store incoming and outgoing messages in a queue, the "mail queue." Actual queue size is highly dependent on one's system resources and daily volumes." Lee II at 7-9: "Using a personal computer (151), a telephone (152), a fax (153), or a mobile communication terminal (154), a user (150) can send messages (160) from e-mail (161), fax (162), voice mail (163) to a mail server (170) over the general telephone network (110), or Internet network (120). In addition, an e-mail (161) message sent to the mail server (170) can be converted to an e-mail data file (171), a fax (162) message to a picture data file (172), and a voice mail (163) to an audio data file (173) and stored together in a storage unit (200). "A storage unit (200) according to the present invention can be placed on a network and shared by multiple servers. On the other hand, message files stored in the storage unit (200) according to the present invention can be accessed from the storage unit (200) and transmitted to the user to whom the message is to be transmitted by the user (150) at the command or request of the user to whom the message is to be transmitted. "In preferred embodiments according to the present invention, the user (150) can transmit messages (160) to the mail server (170) via the general telephone network

'117 Claims	Exemplary Disclosures
	(110) and the Internet network (120), and the mail server (170) can convert multiple types of messages into storage-capable data files that can be stored together in the storage unit (200)." White at [0141]: "As a result of the foregoing operations, the message sent from a first mobile device with message store ID "A" is delivered to the queue "q" of the message store with the ID "B" at a second mobile device. The message may then be removed from the queue "q" and provided to an application (e.g., application 510 as shown at FIG. 5) based on logic implemented at the second mobile device. Note that the first mobile device and the second mobile device do not need to connect to the central server at the same time. The message sent by the first device is retained in the central message store (consolidated database) until the second device synchronizes with the central server. The methodology of the present invention for synchronization of client mobile devices with a central server and the application of transmission rules for selecting messages to be exchanged during synchronization will next be described." Holbrook at Abstract: "A method and apparatus for proactively buffering data across intermittent network connections, to increase the probability of desired data being available at remote network nodes where the nodes do not have continuous access to the network. An association table (80) associates user accounts with messaging nodes (14). A primary messaging zone (82) is generated, indicating the messaging nodes associated with a recipient user account, at which there is a significant probability that the user may request the collection of messages. During intermittent communications sessions with members of said primary messaging zone (82), incoming messages are delivered to these messaging nodes, where the messages are proactively buffered in preparation for a user request to collect messages while a messaging node (20) cannot collect from external sources. This addresses the problem of a messaging node (20)

'117 Claims	Exemplary Disclosures
	being unable to obtain new messages, on request, across temporarily inactive network connections, as incoming messages will have already crossed the intermittent connection before being requested." Holbrook at [0101]: "Portable messaging unit 40 is a remote and portable user interface for messaging system 10. Central server 12 is a central repository of data, messages and administrative functions within messaging system 10. Messaging node 20 is a local communications interface for exchanging information between central server 12 and portable messaging unit 40. It is expected that communications link 70 between central server 12 and messaging node 20, and additionally communications link 72 between messaging node 20 and portable messaging unit 40, may be an unreliable or intermittent data connection. The operation of messaging system 10 is designed to provide robust performance under such conditions." Holbrook at [0144]: "Messages sent user-to-user within messaging system 10 generally travel twice across instances of first communications link 70—once when the message is sent to central server 12 for distribution, and again when the message is buffered at the messaging node 20 from which the message is subsequently delivered to the recipient. Thus, messages sent simultaneously from several members of messaging nodes 14 might be proactively buffered at messaging node 20 at considerably different times, depending on the phasing and frequency of these nodes' communications with central server 12." Holbrook at cl. 1: "1. A method for the delivery of an incoming message in a messaging system comprising a central server of said messaging system, a plurality of messaging nodes, a plurality of user accounts with distinct messaging address identifiers, and a

'117 Claims	Exemplary Disclosures
	communications means for establishing a first communications link between each of said messaging nodes and said central server; said method comprising the steps of a) maintaining an association table, associating user accounts with at least one messaging node; b) identifying at least one user account indicated as a recipient of said incoming message, determined from a header of said incoming message; c) determining membership of a messaging node within a primary messaging zone, corresponding to a subset of said messaging nodes associated in said association table with said recipient user account; d) transmitting said incoming message across said first communications link to said messaging node, and e) buffering said incoming message at said messaging node, prior to a user request to collect new messages from said messaging system, whereby said incoming message is buffered at messaging nodes from which recipients may subsequently request the collection of said incoming message." Bobo at Abstract: "A Message Storage and Deliver System (MSDS) is connected to a plurality of DID phone lines and receives facsimile messages, voice messages, and data messages. The MSDS assigns a separate telephone number for each user of the system and can simultaneously receive more than one message for a single user. The messages are stored in memory and are also converted into appropriate hyper-text mark-up language (HTML) files. The MSDS is connected to the Internet and notifies the users with an E-mail message each time a message is received. The MSDS can also page the user so that the user receives almost instantaneous notice

Headwater Research LLC
Ex. 2011, IPR2024-00003
Page 282 of 442

'117 Claims	Exemplary Disclosures
	of the message. The users can then connect to the MSDS through the Internet and have the messages downloaded to their computers or the users can simply preview the messages stored in the MSDS. The users of the MSDS therefore have the advantage of being able to receive their messages at any time and at any location at a reasonable cost. The MSDS offers a number of options on how the messages may be sent to the user, such as several preview options available with facsimile messages. The user can also telephone the MSDS to listen to messages or to alter the service provided by the MSDS." Bobo at 6:66-7:12: "When the call reaches the MSDS 10, the call is routed within the MSDS 10 in a manner that will be described in more detail below with reference to FIG. 13. At step 46, the MSDS 10 answers the telephone call and receives the address signal from the DID trunk 15. Next, at step 48, the call is established between the MSDS 10 and the third party and, at step 50, the MSDS 10 receives the message transmitted over the telephone line. The message is stored at step 52, a database within the MSDS 10 is updated at step 54, and the intended recipient of the message is notified at step 56. The intended recipient of the message uses the services provided by the MSDS 10 and will hereinafter be referred to as a user. At step 58, the message is converted into hyper-text mark-up language (HTML)." Bobo at cl. 31: "31. A network message storage and delivery system, comprising: a central processor for receiving an incoming call, for detecting an address signal on said incoming call, for detecting a message on said incoming call, and for placing said message in a storage area, said address signal being associated with a user of said network message storage and delivery system;

'117 Claims	Exemplary Disclosures
	a network server for receiving said message from said storage area, for converting said message into a mixed media page layout language, and for placing said message in said storage area; wherein when said network server receives a request from said user over said network, said network server transmits at least a portion of said message over said network to said user and wherein said network comprises the Internet and said network server comprises an Internet server." XMPP at 86: "For message stanzas, the server MAY choose to store the stanza on behalf of the user and deliver it when the user next becomes available or forward the message to the user via some other means (e.g., to the user's email account). However, if offline message storage or message forwarding is not enabled, the server MUST return to the sender a stanza error. (Note: Offline message storage and message forwarding are not defined in XMPP, since they are strictly a matter of implementation and service provisioning.)." *See also* [1c].
4. The network system of claim 1, the network message server further to encrypt the secure Internet data messages, the device messaging agents further to decrypt each received message to obtain the corresponding application identifier and application data.	The references below disclose and/or render obvious this limitation. For example, see the following passages and/or figures, as well as related disclosures: Chou at Abstract: "Push content is generated in a wireless secure server [WSS] or other type of server, responsive to information received in the wireless secure server from the enterprise application. The push content is deliverable from the wireless secure server to the mobile client device over a wireless network."

'117 Claims	Exemplary Disclosures
	Chou at 11:22-30: "The processing element 200 as shown in the figure includes a processor 202 coupled to a memory 204 and one or more network interfaces 206. The techniques of the present invention may be implemented at least in part in the form of software storable in the memory 204 and executable by the processor 202. The memory 204 may represent random access memory (RAM), read-only memory (ROM), optical or magnetic disk-based storage, or other storage elements, as well as portions or combinations thereof." Chou at 3:33-47: "FIG. 1A shows an example communication system 100 in which the present invention is implemented. The system 100 includes a network 102, an enterprise application 104, and a wireless secure server (WSS) 106. The enterprise application 104, which is assumed to run on a computer, server or other processing element associated with a given enterprise, is separated from the WSS 106 via an enterprise firewall 108, and receives multimodal input such as voice, data, etc. from sources not shown in the figure. The system 100 further includes a wireless application protocol (WAP) mobile client device 110, which communicates with the WSS 106 over network 102. More specifically, the mobile client device 110 communicates with the WSS 106 via WAP gateway (WG) 112, and via a push proxy gateway (PPG) 114 and a push initiator (PI) 116, as shown." Chou at 6:42-45: "5. The gateway converts the HTTP response back into an encrypted, binary format and sends it to the mobile client device. 6. The mobile client device decodes the response and displays the results."

'117 Claims	Exemplary Disclosures
	Chou at 4:14-21: "The WSS 106 further includes a secure connector 120 which illustratively includes a password-protected register connector 122 and an application connector 124. Security can be provided in these elements using conventional encryption or other cryptographic operations of a type known in the art. The application connector 124 communicates alert and/or notification information to the mobile client 110 via PI 116 and PPG 114 as shown." Chou at 10:30-52: "The illustrative embodiment of the invention as described above provides a secure information access technique in which a user can make a service request through multimodal dialogue interaction with the enterprise application but does not have direct access to all the associated information content through the request channel. The enterprise application instead controls the content delivery through an HTTP-based interface to the WSS. Further, the content is transcoded and delivered in such a way that it is sent to a unique registered mobile client device. The location of the content and the associated procedure of pulling the content are protected by encryption, so as not to be apparent to a third party or even an intended user. Therefore, the mobile client device can be safely utilized in a crowded public area, such as in a bus or an airplane. In order to access the content, a potential attacker has to pass the authentication process through multimodal interaction with the enterprise application, and therefore must obtain both a registered mobile client device and the unique access code for that device. Moreover, even if the attacker is able to access the content in this manner, there is no direct pipe provided through the enterprise firewall, so the overall security of the enterprise communication services platform remains uncompromised." Rakic at [0053]:

'117 Claims	Exemplary Disclosures
	"Removable memory 218 may store information that is associated with a user (e.g., a service subscriber). The information (e.g., an International Mobile Subscriber Identity (IMSI) code, a phone number, a personal unblocking key (PUK), etc.) in removable memory 218 may be used to identify the user and/or a service to which the user subscribes. In one implementation, a user may exchange client device 102 with another device without changing the subscription, by transferring removable memory 218 from client device 102 to the other device." Rakic at [0065]-[0068]: "Secure push message server 504 may send secure push messages to client device 102 via a wireless network or gateway 106. Given a message or a payload from an application, a user, or an administrator, secure push message server 504 may generate a key, an electronic signature, and a secure push message. "Secure push message server 504 may generate the key based on codes (e.g., IMEI) that are obtained from client device 102, a component in client device 102 (e.g., removable memory 218), and an administrator. Depending on the implementation, additional codes or information may be used to generate the key. Once the key is generated, secure push message server 504 may generate an electronic signature based on the key and the message. "In possession of the electronic signature and the key, secure push message server 504 may generate and send the secure push message to client device 102. In one implementation, secure push message server 504 may generate the secure push message by appending the electronic signature to the message and/or other information (e.g., a timestamp). "In addition to generating/sending secure push messages to client device 102, secure push message server 504 may obtain and store information that is needed to generate the key. In one implementation, while configuring client device 102,

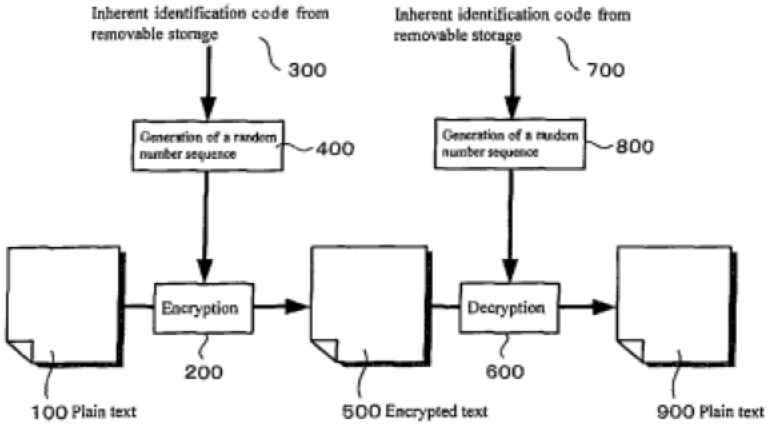
'117 Claims	Exemplary Disclosures
	secure push message server 504 may obtain a client device identifier from client device 102, subscriber information from removable memory 218 that is not yet installed in client device 102, and an administrator code from an administrator. Furthermore, secure push message server 504 may store the client device identifier, the information, and the administrator code in a database." Anderson at 669-670: "Another approach is to do encryption and/or authentication at the IP layer using a protocol suite known as Ipsec. IPsec defines a *security association* as the combination of keys, algorithms and parameters used to protect a particular packet stream. Protected packets are either encrypted or authenticated (or both); in the latter case, an authentication header is added that protects data integrity using HMAC-SHA1, while in the former the packet is encrypted and encapsulated in other packets. (The use of encryption without authentication is discouraged as it's insecure [151].) There's also an *Internet Key Exchange* (IKE) protocol to set up keys and negotiate parameters. IKE has been through a number of versions (some of the bugs that were fixed are discussed in [465]). "IPsec is widely used by firewall vendors who offer a *virtual private network* facility with their products; that is, by installing one of their boxes in each branch between the local LAN and the router, all the internal traffic can pass encrypted over the Internet. Individual PCs, such as workers' laptops and home PCs, can in theory join a VPN given a firewall that supports IPsec, but this is harder than it looks. Compatibility has been a major problem with different manufacturers' offerings just not working with each other; although firewall-to-firewall compatibility has improved recently, getting random PCs to work with a given VPN is still very much a hit-or-miss affair." Anderson at 670-71: "Recall that when discussing public key encryption, I remarked that a server could publish a public key *KS* and any web browser could then send a message *M*

'117 Claims	Exemplary Disclosures
	containing a credit card number to it encrypted using *KS*: $\{M\}_{KS}$. This is in essence what the TLS protocol (formerly known as SSL) does, although in practice it is more complicated. It was developed to support encryption and authentication in both directions, so that both http requests and responses can be protected against both eavesdropping and manipulation. It's the protocol that's activated when you see the padlock on your browser toolbar." "Here is a simplified description of the version as used to protect web pages that solicit credit card numbers: 1. the client sends the server a *client hello* message that contains its name *C*, a transaction serial number *C#*, and a random nonce N_C; 2. the server replies with a *server hello* message that contains its name *S*, a transaction serial number *S#*, a random nonce N_S, and a certificate *CS* containing its public key *KS*. The client now checks the certificate *CS* back to a root certificate issued by a company such as Verisign and stored in the browser; 3. the client sends a *key exchange* message containing a *pre-master-secret* key, K_0, encrypted under the server public key *KS*. It also sends a *finished* message with a message authentication code (MAC)computed on all the messages to date. The key for this MAC is the *master-secret*, K_1. This key is computed by hashing the pre-master-secret key with the nonces sent by the client and server: $K_1 = h(K_{CS},N_C,N_S)$. From this point onward, all the traffic is encrypted; we'll write this as $\{\ldots\}_{KCS}$ in the client-server direction and $\{\ldots\}_{KSC}$ from the server to the client. These keys are generated in turn by hashing the nonces with K_1. 4. The server also sends a finished message with a MAC computed on all the messages to date. It then finally starts sending the data. $C \rightarrow S : C, C\#, N_C$ $S \rightarrow C : S, S\#, N_S, CS$ $C \rightarrow S : [K0]_{KS}$

'117 Claims	Exemplary Disclosures
	$C \rightarrow S$: $[finished, MAC(K_1, everythingtodate)]_{KCS}$ $S \rightarrow C$: $[finished, MAC(K_1, everythingtodate)\}_{KSC}, [data]_{KSC}$ "The design goals included minimizing the load on the browser, and then minimizing the load on the server. Thus the public key encryption operation is done by the client, and the decryption by the server; the standard encryption method (*ciphersuite*) uses RSA for which encryption can be arranged to be very much faster than decryption. (This was a wrong design decision as browsers generally have a lot more compute cycles to spare than servers; it has created a brisk aftermarket for crypto accelerator boards for web servers.) Also, once a client and server have established a pre-master secret, no more public key operations are needed as further master secrets can be obtained by hashing it with new nonces." *See generally* Anderson at Ch. 20 ("Telecom System Security"); § 21.4.5.6 ("TLS"). Ozaki at 3:11-39: "The data-link layer (second layer) is a layer which defines packet format, physical node addresses, a transmitting and receiving method of packets and the like. Layer 2 defines a protocol for interacting packets between two nodes through a physical communication medium. In this second layer, some kind of address or Medium Access Control (MAC) identifier is added with respect to each node and packets are transmitted on the communication medium by specifying a transmission source of the packets. As to communication mediums, there are diversified types of copper wiring, wireless, optical fiber and the like. Also, with respect to the connection mode (topology), there is not only point-to-point connection but also bus, star, and link topologies. The packet transmitted on the communication medium is provides to a node at a reception side where it is processed with respect to the remaining OSI layers. "For example, a NIC (Network Interface Card) functions in conjunction with an associated driver to operate at the physical layer and data-link layer for connecting

'117 Claims	Exemplary Disclosures
	a personal computer, a printer or the like to a premises network (LAN). In many cases, the NIC interfaces to an Ethernet. "By means of the NIC, a node (apparatus) which wants to transmit data awaits a vacant time slot to start a transmission via the Ethernet. At that time, if a plurality of nodes start transmissions simultaneously, data collide in the cable and are broken-down, so that transmitting nodes stop transmissions and will restart transmissions after waiting a random of time. In this manner, it is possible for a plurality of nodes to share a single cable so as to communicate mutually." Ozaki at 10:55-64: "FIG. 3 is a communication layer diagram of a system in a second exemplified embodiment (broadcasting application by UDPsec) of a communication system using the protocol according to the present invention; "FIG. 4A is a diagram showing a packet structure of encryption regions and application regions of completeness authentication with respect to TCPsec/IPsec; "FIG. 4B is a diagram showing a packet structure of encryption regions and application regions of completeness authentication with respect to TCPsec/IP;" *See also* Ozaki at Figs. 3, 4A, 4B. Ogawa at 4:48-57 "The storage device 10 of data input site 1 is designated for storing and transmitting the data. If a start instruction is issued by the transmission initiation until 11, the compression unit 12 will compress the data stored on the storage device 10 in accordance with a compression algorithm as detailed below. Compressed data are supplied to the encryption unit 13 and encrypted using a shared encryption key 14. In addition, data encrypted by the shared encryption key

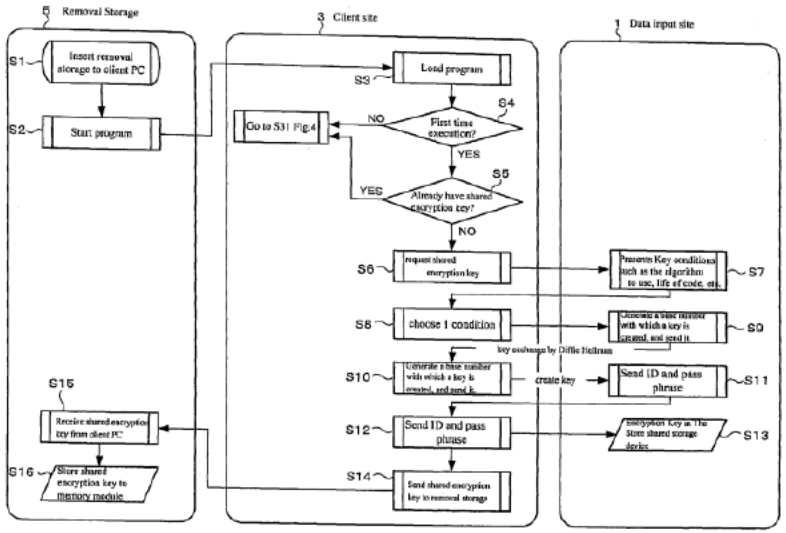
'117 Claims	Exemplary Disclosures
	14 (hereafter referred to as shared key encrypted data) is transmitted to the network 4 via a transmission unit 15." Ogawa at 4:41-6:9 "In the exemplary embodiment, if a start instruction is issued by the client starting unit 31 at the client site 3, the shared key encrypted data will be received by the receive unit 32 from the database server 2 via network 4. This received shared key encrypted data will be decrypted using the shared encryption key 52 which was supplied to decryption unit 33 and beforehand stored on the removable storage 5. This decrypted data is then expanded by a decompression (expansion) unit 34 to convert the data back to the original format. Data that was expanded by the decompression unit 34 is re-encrypted by the encryption unit 35, using an inherent encryption key 53 that was stored on the removable storage 5. This inherent encryption key 53 is generated from an inherent identification code that was assigned during manufacturing and stored on the internal memory device of the removable storage 5. Then, data encrypted using this inherent encryption key 53 (hereinafter referred to as inherent key encrypted data) is stored on the storage device 30 via a saving unit 36. After the program 51 has been loaded to the client site 3, if a start instruction is issued by the display start unit 37, the decryption unit 38 will decrypt the inherent key encrypted data stored on the storage device 30, using the inherent encryption key 53 stored on the removable storage 5. Then, the decrypted data are supplied to the display unit 39. As a result, the data which are stored in the database server 2 after receiving the data from the data input site 1 via the network 4 will be displayed on the client site 3. The decompression unit 34 can be performed before or after encryption. Data decrypted by the decryption unit 33 are encrypted with the inherent encryption key

'117 Claims	Exemplary Disclosures
	53 in a compressed state and stored on the storage device 30. In order to display this encrypted data, the data will be decrypted by the decryption unit 38 with the inherent encryption key 53, then expanded with the decompression unit 34 and displayed by the data display 39." Figure 6  Ogawa at 8:65-9:14: "FIG. 6 shows an alternative exemplary method of encrypting and decrypting data using an inherent identification code of the removal storage. As seen in FIG. 6, a plain text 100 is supplied to an encryption unit 200. Inherent code 300 of the removal storage is supplied to a random number generator unit 400 and the generated random number is supplied to an encryption unit 200 to produce a cryptogram 500. The cryptogram 500 is supplied to a decryption unit 600. The inherent number 700 of the removal storage is supplied to a random number generator unit 800 and the

'117 Claims	Exemplary Disclosures

random number is supplied to a decryption unit 600 and plain text 900 is produced. At this point, if an inherent identification code 300 and 700 of the removal storage and random number generator unit 400 and 800 is the same, then the decrypted plain text 900 is identical to the plain text 100. Of course, the method shown on FIG. 6 is exemplary only and variations in ordering, inputs and the like are embraced by the present disclosure."

Figure 2



Ogawa at 6:42-7:42

"The flow chart in FIG. 2 shows the procedure of the processing operations that distribute the shared encryption keys. FIG. 2 shows the removable storage 5 on the left, the client site 3 in the center, and the data input site 1 on the right.
First, if the removable storage 5 is operably linked to the client site 3 (physically or wirelessly) in step S1, the program 51 will start step S2, and the program 51 will be loaded into the client site 3 in step S3. At the client site 3, it will be determined if