(19) **DANMARK**

(10) **DK/EP 3175588 T3**

Patent- og
Varemærkestyrelsen

(12) Oversættelse af
europæisk patentskrift

(51) Int.Cl.: ***H 04 M 3/533 (2006.01)*** ***G 06 Q 10/107 (2023.01)*** ***H 04 L 51/18 (2022.01)***
H 04 W 4/14 (2009.01)

(45) Oversættelsen bekendtgjort den: **2024-01-29**

(80) Dato for Den Europæiske Patentmyndigheds
bekendtgørelse om meddelelse af patentet: **2023-10-25**

(86) Europæisk ansøgning nr.: **15819468.8**

(86) Europæisk indleveringsdag: **2015-07-09**

(87) Den europæiske ansøgnings publiceringsdag: **2017-06-07**

(86) International ansøgning nr.: **US2015039797**

(87) Internationalt publikationsnr.: **WO2016007780**

(30) Prioritet: **2014-07-10 US 201462023059 P**

(84) Designerede stater: **AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR**

(73) Patenthaver: **Fasetto, Inc., 4110 N. Scottsdale Road, Suite 315, Scottsdale, AZ 85251, USA**

(72) Opfinder: **Christmas, Coy, P.O. Box 2036, Superior, WI 54880, USA**
MALPASS, Luke, 26 Sutton Avenue, Stoke-on-Trent, Staffordshire ST56TB, Storbritannien

(74) Fuldmægtig i Danmark: **Marks & Clerk LLP, 44 rue de la Vallée, BP 1775, L-1017 Luxembourg, Luxembourg**

(54) Benævnelse: **SYSTEMER OG FREMGANGSMÅDER TIL BESKEDREDIGERING**

(56) Fremdragne publikationer:
EP-A1- 1 783 971
EP-A2- 0 800 144
WO-A2-00/33545
WO-A2-2007/103908
US-A1- 2003 131 050
US-A1- 2006 149 825
US-A1- 2006 161 631
US-A1- 2014 189 532
US-B1- 7 428 702

DESCRIPTION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Application Serial No. 62/023,059 entitled "SYSTEMS AND METHODS FOR MESSAGE EDITING" and filed on July 10, 2014.

FIELD

[0002] The present disclosure generally relates to transmitting data, and more particularly to systems and methods for editing data transmitted between devices.

BACKGROUND

[0003] Communication between devices is typically performed over a network, such as the internet or a local area network. However, networks may not always be available for communication between devices and additionally may expose communications to security breaches on the network. Devices enabled with Bluetooth® may communicate directly. However, Bluetooth® enabled devices must be within a limited range, and communication speeds may be relatively slow.

[0004] Many devices may include a wireless chip. Manufacturers may create specialized wireless chips which enable communication between devices containing compatible wireless chips. However, such devices may be unable to communicate with other devices containing chips manufactured by other manufacturers.

[0005] Many devices may allow users to transmit data, such as text messages to other users. Typically, once the text message is delivered to a device, the text message is saved locally on the device. It may not be possible for a user to edit or delete a message which was previously sent to another device.

[0006] WO 00/33545 A2 describes a method of a modification of a message in a communication system and a device for execution of the method.

[0007] EP 0 800 144 A2 describes electronic mail systems with the capability to act on previously-sent messages that have passed beyond the scope of control of the sending e-mail system.

[0008] US 2003/131050 describes a system for entering a correction for a previously sent message, determining whether to send a complete corrected message or a message correction specification, and sending the selected message to a messaging server.

SUMMARY

[0009] Aspects of the invention are in accordance with the appended independent claims.

[0010] Systems, methods, and computer-readable media for transmitting data are disclosed. In various embodiments, a method may include implementing a standardized communications protocol ("SCP") on a first device. The method may further include discovering a second device. The method may further include selecting a transmission path. The method may further include transmitting a message to the second device.

[0011] In various embodiments, a method may comprise receiving, by a first device comprising a processor for communicating with a second device, a datagram from a second device. The method may further comprise identifying a standardized communication protocol ("SCP") header in the datagram. The method may further comprise transmitting, by the processor, a list of available transmission paths to the second device. The method may further comprise receiving, by the processor, a message from the second device over at least one transmission path in the list of available transmission paths.

[0012] In various embodiments, a method may comprise receiving, by a messaging server, an original message from a first device; receiving, by the messaging server, a request for new data from a second device; transmitting, by the messaging server, the original message to the second device; receiving, by the messaging server, an edited message; deleting, by the messaging server, the original message from the second device; and transmitting, by the messaging server the edited message to the second device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] A more complete understanding may be derived by referring to the detailed description and claims when considered in connection with the Figures, wherein like reference numbers refer to similar elements throughout the Figures, and:

FIG. 1 illustrates a schematic diagram of a system for transmitting messages according to various embodiments of the disclosure;

FIG. 2 illustrates a process for transmitting data between devices according to various embodiments;

FIG. 3 illustrates a process for a file send protocol according to various embodiments;

FIG. 4 illustrates a discovery protocol according to various embodiments;

FIG. 5 illustrates a definition for a discovery protocol according to various embodiments;

FIG. 6 illustrates a definition for a file transfer request according to various embodiments;

FIG. 7 illustrates a definition for a response to a file transfer request according to various embodiments; and

FIG. 8 illustrates a process for editing messages according to various embodiments.

DETAILED DESCRIPTION

[0014] The detailed description of exemplary embodiments herein makes reference to the accompanying drawings and pictures, which show various embodiments by way of illustration. While these various embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure, it should be understood that other embodiments may be realized and that logical and mechanical changes may be made without departing from the scope of the disclosure. Thus, the detailed description herein is presented for purposes of illustration only and not of limitation. For example, the steps recited in any of the method or process descriptions may be executed in any order and are not limited to the order presented. Moreover, any of the functions or steps may be outsourced to or performed by one or more third parties. Furthermore, any reference to singular includes plural embodiments, and any reference to more than one component may include a singular embodiment.

[0015] Systems, methods and computer program products are provided. In the detailed description herein, references to "various embodiments," "one embodiment," "an embodiment," "an example embodiment," etc., indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to effect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described. After reading the description, it will be apparent to one skilled in the relevant art(s) how to implement the disclosure in alternative embodiments.

[0016] Systems and methods are disclosed herein for peer to peer communication between communication devices. As used herein, a "communication device" may refer to any device capable of communication with another device. For, example and without limitation, a communication device may refer to a smartphone, PDA, laptop, desktop computer, portable phone, GPS device, car navigation system, wireless printers, or any other device.

[0017] The systems and methods disclosed herein may enable communication between devices without connection to the Internet or other networks. A standardized communication system ("SCS") may be installed on a device. The SCS may comprise any combination of hardware and/or software. The SCS may utilize existing physical components of the device, such as 802.11 wireless chips and Bluetooth® systems in order to communicate with other devices. The SCS may be suitable for any communication protocol, such as IP, TCP/UDP, Bluetooth®, raw Manchester encoding, and any other form of wireless communication.

[0018] The SCS may allow communication between devices of varying types and platforms. Additionally, as communication may be directly between devices without transmitting data across a network, communication may be available when networks are unavailable, and communications may be protected from eavesdroppers on a network. Furthermore, direct communication between devices may avoid data charges on cellular data plans.

[0019] Referring to FIG. 1, a system 100 for transmitting messages is illustrated according to various embodiments. A first device 110 comprising an SCS 112 and a second device 120 comprising an SCS 122 are illustrated according to various embodiments. In various embodiments, SCS 112 and SCS 122 may be aftermarket software programs installed on first device 110 and second device 120. However, in various embodiments, SCS 112 and SCS 122 may be embedded into a chip, such as an 802.11 wireless chip, in first device 110 and/or second device 120.

[0020] In various embodiments, the SCS may implement a standardized communication protocol ("SCP") on a device. SCP may attach an SCP header 152 to a packet in order to identify a datagram 150 as an SCP datagram. First device 110 may communicate with second device 120 via SCP. The SCS may recognize the SCP header and may follow the SCP. The SCP may define the ability for devices to discover one another, to request the transfer of raw data, to transmit confirmations on receipt of data, and to perform any other steps involved with transmitting data.

[0021] In various embodiments, the SCS may be implemented at the network layer in the Open Systems Interconnection ("OSI") model (or the Internet layer in the TCP/IP model). Regardless of the protocol being used at the transport layer (e.g. TCP, UDP, SCTP, DCCP), the SCP header may allow devices comprising an SCS to communicate via SCP.

[0022] In various embodiments, at least one of first device 110 and second device 120 may comprise a smartphone. However, in various embodiments, first device 110 and second device 120 may comprise any type of device capable of transmitting and/or receiving data.

[0023] Referring to FIG. 2, a process 200 for transmitting data between devices is illustrated according to various embodiments. In various embodiments, a first user may wish to transmit data from first device 110 to second device 120. The data may comprise any type of data, such as a text message, image, video, text document, or any other type of file.

[0024] First device 110 may discover available devices (step 210). First device 110 may attempt to discover other devices by a variety of methods. In various embodiments, first device 110 may discover other devices via a camera or other optical device. In various embodiments, second device 120 may display a symbol, such as a QR-code, a barcode, or text. The symbol may comprise identifying characteristics about second device 120. For example, in various embodiments the identifying characteristics may comprise at least one of a device name, an IP address of the device, an owner name, an endpoint of the device, and the available transport layers on the device. First device 110 may scan the symbol using a camera. First device 110 may obtain the identifying characteristics from the symbol and use the identifying characteristics in order to transmit data to second device 120.

[0025] In various embodiments, the SCS on first device 110 may search for other devices using a wireless chip in first device 110. Devices comprising an SCS may transmit a broadcast message. The broadcast message may comprise the identifying characteristics of the device. In various embodiments, first device 110 may be within transmission range of second device 120. The transmission range may depend on the specific type of wireless chips in first device 110 and second device 120. However, in various embodiments, the transmission range may be up to about 200 feet - 300 feet. The SCS may open a socket on first device 110 to listen for broadcast messages. The broadcast message may be sent by a variety of hardware. For example, the broadcast message may be transmitted via an 802.11 wireless chip, Bluetooth® chip, or NFC.

[0026] In various embodiments, first device 110 and second device 120 may not be within transmission range of each other. However, an intermediary device, such as a smartphone equipped with hotspot technology, may be within transmission range of first device 110. First device 110 may search for available devices by transmitting a message to intermediary device, instructing intermediary device to look for available devices. Intermediary device may receive a broadcast message from second device 120, and intermediary device may transmit the broadcast message to first device 110. Thus, first device 110 may discover second device 120 without connecting to the internet or a cellular network even though first device 110 may not be within transmission range of second device 120. In various embodiments, any number of intermediary devices may be daisy-chained, such that first device 110 may discover second device 120 from miles apart by transmitting data via a series of intermediary devices.

[0027] First device 110 may display a list of all discovered devices to the user. The user may select second device 120 in order to transmit data to second device 120. The user may select a file or message to be transmitted to second device 120.

[0028] The SCS 112 on first device 110 may determine the transmission hardware to utilize for the transmission (step 220). In various embodiments, first device 110 and second device 120 may each have only one type of transmission hardware, such as an 802.11 wireless chip, and the SCS 112 may thus select the 802.11 wireless chip to transmit the data. However, in various embodiments, multiple transmission paths may be available between first device 110 and second device 120. For example, first device 110 and second device 120 may each comprise an 802.11 wireless chip and a Bluetooth® chip. In various embodiments, the SCS 112 may determine the fastest transmission path, and may select the fastest transmission path to transmit the data. In various embodiments, the transmission path may be selected by default settings. For example, SCS 112 may always select an 802.11 wireless path for transmission when available, and if the 802.11 wireless path is not available, SCS 112 may select a Bluetooth® path. However, in various embodiments, the SCS 112 on first device 110 may transmit a speed test message to second device 120 via each available transmission path, and the SCS 112 may select the fastest transmission path based on the speed test results.

[0029] In various embodiments, the SCS 112 may instruct first device 110 to send the data to second device 120 via multiple transmission paths. A message may be divided into multiple packets. SCS 112 may analyze the available transmissions paths, and send the message over multiple transmission paths in order to expedite transmission of the entire message. For example, SCS 112 may determine that the fastest method of transmitting the message may be to transmit 90% of the packets via an 802.11 wireless path, and 10% of the packets over a Bluetooth® path. SCS 112 may attach an SCP header to each packet being transmitted to second device 120, whether via 802.11 wireless or Bluetooth®. Thus, SCS 122 on second device 120 may recognize the packets as being received by SCP, and SCS 122 may reassemble the packets in order to recreate the entire message. In various embodiments, SCS 112 may analyze all transmission paths available, including but not limited to multiple 802.11 wireless chips, Bluetooth® chips, NFC, PDQ, or any other transmission paths in order to select the fastest transmission method. The SCS on first device 110 may initiate a file send protocol and transmit the data to second device 120 (step 230).

[0030] In various embodiments, first device 110 and second device 120 may be connected to the same local network. First device 110 may transmit a link, such as a QR-code, over a cellular network or the local network to second device 120. In various embodiments, the link may comprise 10kb or less of data. Second device 120 may use the link to request or accept a file transfer. First device 110 may transmit a file over the local network. In various embodiments, the file may be transferred using TCP/IP directly over the local network.

[0031] In various embodiments, second device 120 may have access to an internet connection. First device 110 may transmit a link over a cellular transmission path to second device 120, and second device 120 may use the link to download a file stored on the cloud and/or on a server over the internet. In various embodiments, second device 120 may download the file using TCP/IP.

[0032] In various embodiments, first device 110 may sync its contents with a cloud database. In various embodiments, first device 110 may comprise an SCS folder, and only files stored in the SCS folder may be synced with the database. First device 110 may transmit a link over a cellular transmission path to second device 120 identifying a file stored on the database. In various embodiments, second device 120 may not have access to an 802.11 wireless network at the time second device 120 receives the link. Second device 120 may use the link to access the file whenever second device 120 gains access to an 802.11 wireless network in order to prevent cellular data charges. In various embodiments, second device 120 may use the link to access the file over the cellular network. In various embodiments, second device 120 may stream all or part of the file over either the cellular network or an 802.11 wireless network.

[0033] In various embodiments, first device 110 may share an online folder with second device 120. First device 110 may indicate that second device 120 may have access to an online folder. First device 110 may sync with the online folder to upload files stored on first device 110 to the online folder. Second device 120 may sync with the online folder to download files stored in the online folder to second device 120.

[0034] Referring to FIG. 3, a process 300 for a file send protocol is illustrated according to various embodiments. First device 110 may transmit a request to establish a connection with second device 120 (step 310). In various embodiments the connection may comprise a TCP connection. However, in various embodiments, the connection may comprise any type of connection for transmitting data between devices. Second device 120 may accept the connection request (step 320). In various embodiments, the connection may be between secure sockets on first device 110 and second device 120.

[0035] In various embodiments, first device 110 may transmit a message comprising a cypher book to second device 120 (step 330). The cypher book may comprise a list of one-time cyphers, and may allow second device 120 to decrypt data sent to second device 120 over the secure socket connection using one time cyphers. In various embodiments, first device 110 may encrypt the message comprising the cypher book using known encryption methods, such as Advanced Encryption Standard ("AES") or RSA encryption. However, subsequent messages during the transfer session may be encrypted using the one-time cyphers contained in the cypher book. The messages encrypted using the one-time cyphers may be encrypted and decrypted using significantly less processing power and time than messages encrypted with AES or RSA. Additionally, the messages sent using the one-time cyphers may be indecipherable to parties not containing the cypher book.

[0036] First device 110 may send a file transfer request (step 340). For an example of a file transfer request, refer to FIG. 5. Second device 120 may accept the file transfer request (step 350). In response to second device 120 accepting the file transfer request, first device 110 may break the file into segments, and begin transmitting the segments to second device 120 (step 360). After first device 110 has transmitted all segments of the file, first device 110 may wait for confirmation that second device 120 has received all segments. Second device 120 may transmit a confirmation message to first device 110 indicating that all segments have been received (step 370). Second device 120 may decrypt and reassemble the segments according to SCP in order to recreate the file (step 380).

[0037] Referring to FIG. 4, an example of a discovery protocol 400 is illustrated according to various embodiments. Discovery protocol 400 may be implemented on the transport layer using TCP/UDP. However, in various embodiments, discovery protocols may be implemented using a Bluetooth® serial port, RS-232, or may be sent entirely over datagrams or a Windows® Socket API ("WSA"). The LocalClient in the illustrated embodiment may be a new instance of an IDiscoveredClient (defined in FIG. 5) class filled in with the device's identifying characteristics, such as device name, user name, preview image, and endpoint (in this case an IP address and port). First device 110 may open a new socket for a broadcast message (410). First device 110 may transmit the IP address that first device 110 is listening on for a response to the broadcast message (420). First device 110 may open a new datagram socket to listen for a response message (430). After receiving a response message, first device 110 may decipher the response message into the original IDiscoveredClient message that first device 110 sent (440).

[0038] Referring to FIG. 5, a definition for an example discovery protocol 500 is illustrated according to various embodiments. The definition may be a single common class called IDiscoveredClient that may be implemented by a transmitting device and a receiving device. In various embodiments, the definition may be expanded to include custom fields and any other information that users may desire. In various embodiments, the definition may comprise a name of the device (510), an IP address of the device (520), an owner of the device (530), and endpoint of the device (540), and a transport layer on which the device was discovered (550). However, one skilled in the art will appreciate that the particular fields used may be altered to any desired fields.

[0039] Referring to FIG. 6, a definition for an example file transfer request protocol 600 is illustrated according to various embodiments. The definition may be called "IFileTransferRequest." In various embodiments, the definition may comprise the name of the device transmitting a file (610), the filename to be sent (620), the size of the file (630), the device receiving the file (640), a unique identification for the file transfer (650), and the transport layer associated with the file transfer (660).

[0040] Referring to FIG. 7, a definition for a response to a file transfer request 700 is illustrated according to various embodiments. The receiving device may respond with a definition called IFileTransferResponse to indicate that the receiving device is willing to accept the file transfer. In various embodiments, IFileTransferResponse may comprise the response from the user (710) and the unique identification for the file transfer (720). The transmitting device may receive the response from the receiving device, and the transmitting device may proceed to transmit the file to the receiving device. Once a complete file transfer has occurred, the receiving device may transmit a confirmation to the transmitting device (730).

[0041] Referring to FIG. 8, a process 800 for editing a message is illustrated according to various embodiments. A first user may wish to send a text message to a second user. The first user may transmit the text message from a first device to a message server (step 810). The message server may store the text message in a message database. The message server may transmit a notification to a second user on a second device. The notification may indicate that the second user has received a new text message. The second device may request new data from the message server (step 820). The message server may transmit all new data pertaining to the second user stored after a last fetch time (step 830). For example, the second device may have fetched data at 11:00 AM. The second device may request new data at 1:00 PM, and the server may transmit all new data stored after 11:00 AM. The new data may include the text message from the first user.

[0042] The second device may store the text message in a cache on the second device. Thus, the text message may be stored locally on the second device. The first user may decide to edit the text message. The first user may use the first device, or any other device, to connect to the message server. The first user may indicate that the first user wishes to edit the text message. For example, the user may click on a "delete" button adjacent to the text message to delete the text message, or the user may click on an "edit" button next to the text message, and a cursor may appear to allow the first user to edit the text message.

[0043] The user may edit or delete the text message as desired, and the first device may transmit the edited text message to the message server (step 840). The message server may store the edited text message in the message database. The message server may mark the original text message as invalid or deleted, such that devices may no longer be able to access the original text message. The message server may transmit an instruction to the second device to delete the original text message from the cache of the second device (step 850). In various embodiments, the message server may delete the original text message from any device that has saved the original text message. The message server may transmit the edited text message to the second device (step 860). Thus, regardless of whether the second user has previously viewed the original text message on the second device, the edited text message may be displayed in place of the original text message on the second device. In various embodiments, the edited text message may be displayed with a symbol indicating that the edited text message has been edited. However, in various embodiments, the edited text message may be displayed without any indication that the edited text message was edited.

[0044] In various embodiments, the first device may transmit a file to the second device via the message server. In various embodiments, the file may be an attachment to a message. Similarly to editing a text message, the first user may edit, delete, or replace the file after the file has been sent to the second user. The message server may replace the original file on the second device with the edited file. The first user may edit or delete any messages or files previously sent to the second user, regardless of which device was used to send or receive the message or file. For example, the first user may use a first smartphone to transmit a message to the second user. The second user may retrieve the message from a second smartphone, as well as from a desktop computer. The first user may edit the text message from a laptop computer. The message server may delete the original message from the first smartphone, the laptop computer, the second smartphone, and the desktop computer. The message server may replace the original message with the edited message on all devices. In various embodiments, the device used to edit the text message may delete the original message from its cache without additional instruction from the message server.

[0045] In various embodiments, the first device may edit an original message on a second device without using the message server or any network. The first user may edit the original message on the first device. The first device may discover the second device using any available hardware. The first device may transmit an instruction to the second device to delete the original message from the cache, and the first device may transmit the edited message to the second device. In response to the first device connecting to the message server, the message server may store the edited message and transmit the edited message to all devices having the original message stored in their cache. Conversely, in various embodiments a first device may transmit an original message to a second device directly without using the message server, and the first device may later transmit an edited message to the second device utilizing the message server.

[0046] In various embodiments, the methods described herein are implemented using the various particular machines described herein. The methods described herein may be implemented using the below particular machines, and those hereinafter developed, in any suitable combination, as would be appreciated immediately by one skilled in the art. Further, as is unambiguous from this disclosure, the methods described herein may result in various transformations of certain articles.

[0047] For the sake of brevity, conventional data networking, application development and other functional aspects of the systems (and components of the individual operating components of the systems) may not be described in detail herein. Furthermore, the connecting lines shown in the various figures contained herein are intended to represent exemplary functional relationships and/or physical couplings between the various elements. It should be noted that many alternative or additional functional relationships or physical connections may be present in a practical system.

[0048] The various system components discussed herein may include one or more of the following: a host server or other computing systems including a processor for processing digital data; a memory coupled to the processor for storing digital data; an input digitizer coupled to the processor for inputting digital data; an application program stored in the memory and accessible by the processor for directing processing of digital data by the processor; a display device coupled to the processor and memory for displaying information derived from digital data processed by the processor; and a plurality of databases. Various databases used herein may include: client data; merchant data; financial institution data; and/or like data useful in the operation of the system. As those skilled in the art will appreciate, user computer may include an operating system (e.g., Windows NT, Windows 95/98/2000, Windows XP, Windows Vista, Windows 7, OS2, UNIX, Linux, Solaris, MacOS, etc.) as well as various conventional support software and drivers typically associated with computers.

[0049] A network may include any cloud, cloud computing system or electronic communications system or method which incorporates hardware and/or software components. Communication among the parties may be accomplished through any suitable communication channels, such as, for example, a telephone network, an extranet, an intranet, Internet, point of interaction device (point of sale device, personal digital assistant (e.g., iPhone®, Palm Pilot®, Blackberry®, cellular phone, kiosk, etc.), online communications, satellite communications, off-line communications, wireless communications, transponder communications, local area network (LAN), wide area network (WAN), virtual private network (VPN), networked or linked devices, keyboard, mouse and/or any suitable communication or data input modality. Moreover, although the system is frequently described herein as being implemented with TCP/IP communications protocols, the system may also be implemented using IPX, Appletalk, IP-6, NetBIOS, OSI, any tunneling protocol (e.g. IPsec, SSH), or any number of existing or future protocols. If the network is in the nature of a public network, such as the Internet, it may be advantageous to presume the network to be insecure and open to eavesdroppers. Specific information related to the protocols, standards, and application software utilized in connection with the Internet is generally known to those skilled in the art and, as such, need not be detailed herein. See, for example, DILIP NAIK, INTERNET STANDARDS AND PROTOCOLS (1998); JAVA 2 COMPLETE, various authors, (Sybex 1999); DEBORAH RAY AND ERIC RAY, MASTERING HTML 4.0 (1997); and LOSHIN, TCP/IP CLEARLY EXPLAINED (1997) and DAVID GOURLEY AND BRIAN TOTTY, HTTP, THE DEFINITIVE GUIDE (2002).

[0050] The various system components may be independently, separately or collectively suitably coupled to the network via data links which includes, for example, a connection to an Internet Service Provider (ISP) over the local loop as is typically used in connection with standard modem communication, cable modem, dish networks, ISDN, Digital Subscriber Line (DSL), or various wireless communication methods, see, e.g., GILBERT HELD, UNDERSTANDING DATA COMMUNICATIONS (1996). It is noted that the network may be implemented as other types of networks, such as an interactive television (ITV) network. Moreover, the system contemplates the use, sale or distribution of any goods, services or information over any network having similar functionality described herein.

[0051] Any communication, transmission and/or channel discussed herein may include any system or method for delivering content (e.g. data, information, metadata, etc), and/or the content itself. The content may be presented in any form or medium, and in various embodiments, the content may be delivered electronically and/or capable of being presented electronically. For example, a channel may comprise a website, a uniform resource locator ("URL"), a document (e.g., a Microsoft Word document, a Microsoft Excel document, an Adobe .pdf document, etc.), an "ebook," an "emagazine," an application or microapplication (as described below), an SMS or other type of text message, an email, facebook, twitter, MMS and/or other type of communication technology. In various embodiments, a channel may be hosted or provided by a data partner. In various embodiments, the distribution channel and/or the may comprise at least one of a merchant website, a social media website, affiliate or partner websites, an external vendor, a mobile device communication, social media network and/or location based service. Distribution channels may include at least one of a merchant website, a social media site, affiliate or partner websites, an external vendor, and a mobile device communication. Examples of social media sites include Facebook®, foursquare®, Twitter®, MySpace®, LinkedIn®, and the like. Moreover, examples of mobile device communications include texting, email, and mobile applications for smartphones.

[0052] The present system or any part(s) or function(s) thereof may be implemented using hardware, software or a combination thereof and may be implemented in one or more computer systems or other processing systems. However, the manipulations performed by embodiments were often referred to in terms, such as matching or selecting, which are commonly associated with mental operations performed by a human operator. No such capability of a human operator is necessary, or desirable in most cases, in any of the operations described herein. Rather, the operations may be machine operations. Useful machines for performing the various embodiments include general purpose digital computers or similar devices.

[0053] In fact, in various embodiments, the embodiments are directed toward one or more computer systems capable of carrying out the functionality described herein. The computer system includes one or more processors. The processor is connected to a communication infrastructure (e.g., a communications bus, cross over bar, or network). Various software embodiments are described in terms of this exemplary computer system. After reading this description, it will become apparent to a person skilled in the relevant art(s) how to implement various embodiments using other computer systems and/or architectures. Computer system can include a display interface that forwards graphics, text, and other data from the communication infrastructure (or from a frame buffer not shown) for display on a display unit.

[0054] Computer system also includes a main memory, such as for example random access memory (RAM), and may also include a secondary memory. The secondary memory may include, for example, a hard disk drive and/or a removable storage drive, representing a floppy disk drive, a magnetic tape drive, an optical disk drive, etc. The removable storage drive reads from and/or writes to a removable storage unit in a well known manner. Removable storage unit represents a floppy disk, magnetic tape, optical disk, etc. which is read by and written to by removable storage drive. As will be appreciated, the removable storage unit includes a computer usable storage medium having stored therein computer software and/or data.

[0055] In various embodiments, secondary memory may include other similar devices for allowing computer programs or other instructions to be loaded into computer system. Such devices may include, for example, a removable storage unit and an interface. Examples of such may include a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as an erasable programmable read only memory (EPROM), or programmable read only memory (PROM)) and associated socket, and other removable storage units and interfaces, which allow software and data to be transferred from the removable storage unit to computer system.

[0056] Computer system may also include a communications interface. Communications interface allows software and data to be transferred between computer system and external devices. Examples of communications interface may include a modem, a network interface (such as an Ethernet card), a communications port, a Personal Computer Memory Card International Association (PCMCIA) slot and card, etc. Software and data transferred via communications interface are in the form of signals which may be electronic, electromagnetic, optical or other signals capable of being received by communications interface. These signals are provided to communications interface via a communications path (e.g., channel). This channel carries signals and may be implemented using wire, cable, fiber optics, a telephone line, a cellular link, a radio frequency (RF) link, wireless and other communications channels.

[0057] The terms "computer program medium" and "computer usable medium" are used to generally refer to media such as removable storage drive and a hard disk installed in hard disk drive. These computer program products provide software to computer system.

[0058] Computer programs (also referred to as computer control logic) are stored in main memory and/or secondary memory. Computer programs may also be received via communications interface. Such computer programs, when executed, enable the computer system to perform the features as discussed herein. In particular, the computer programs, when executed, enable the processor to perform the features of various embodiments. Accordingly, such computer programs represent controllers of the computer system.

[0059] In various embodiments, software may be stored in a computer program product and loaded into computer system using removable storage drive, hard disk drive or communications interface. The control logic (software), when executed by the processor, causes the processor to perform the functions of various embodiments as described herein. In various embodiments, hardware components such as application specific integrated circuits (ASICs). Implementation of the hardware state machine so as to perform the functions described herein will be apparent to persons skilled in the relevant art(s).

[0060] In various embodiments, the server may include application servers (e.g. WEB SPHERE, WEB LOGIC, JBOSS). In various embodiments, the server may include web servers (e.g. APACHE, IIS, GWS, SUN JAVA SYSTEM WEB SERVER).

[0061] As those skilled in the art will appreciate, a device may include but is not limited to an operating system (e.g., Windows NT, 95/98/2000/CE/Mobile, OS2, UNIX, Linux, Solaris, MacOS, PalmOS, etc.) as well as various conventional support software and drivers typically associated with computers. A device may include but is not limited to any suitable personal computer, network computer, workstation, personal digital assistant, cellular phone, smart phone, minicomputer, mainframe or the like. A device can be in a home or business environment with access to a network. In various embodiments, access is through a network or the Internet through a commercially available web-browser software package. A device may implement security protocols such as Secure Sockets Layer (SSL) and Transport Layer Security (TLS). A device may implement several application layer protocols including http, https, ftp, and sftp.

[0062] In various embodiments, components, modules, and/or engines of system 100 may be implemented as micro-applications or micro-apps. Micro-apps are typically deployed in the context of a mobile operating system, including for example, a Palm mobile operating system, a Windows mobile operating system, an Android Operating System, Apple iOS, a Blackberry operating system and the like. The micro-app may be configured to leverage the resources of the larger operating system and associated hardware via a set of predetermined rules which govern the operations of various operating systems and hardware resources. For example, where a micro-app desires to communicate with a device or network other than the mobile device or mobile operating system, the micro-app may leverage the communication protocol of the operating system and associated device hardware under the predetermined rules of the mobile operating system. Moreover, where the micro-app desires an input from a user, the micro-app may be configured to request a response from the operating system which monitors various hardware components and then communicates a detected input from the hardware to the micro-app.

[0063] "Cloud" or "Cloud computing" includes a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing may include location-independent computing, whereby shared servers provide resources, software, and data to computers and other devices on demand. For more information regarding cloud computing, see the NIST's (National Institute of Standards and Technology) definition of cloud computing at http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc (last visited February 4, 2011).

[0064] As used herein, "transmit" may include sending electronic data from one system component to another. Additionally, as used herein, "data" may include encompassing information such as commands, queries, files, data for storage, and the like in digital or any other form.

[0065] The system contemplates uses in association with web services, utility computing, pervasive and individualized computing, security and identity solutions, autonomic computing, cloud computing, commodity computing, mobility and wireless solutions, open source, biometrics, grid computing and/or mesh computing.

[0066] Any databases discussed herein may include relational, hierarchical, graphical, or object-oriented structure and/or any other database configurations. Common database products that may be used to implement the databases include DB2 by IBM (Armonk, NY), various database products available from Oracle Corporation (Redwood Shores, CA), Microsoft Access or Microsoft SQL Server by Microsoft Corporation (Redmond, Washington), MySQL by MySQL AB (Uppsala, Sweden), or any other suitable database product. Moreover, the databases may be organized in any suitable manner, for example, as data tables or lookup tables. Each record may be a single file, a series of files, a linked series of data fields or any other data structure. Association of certain data may be accomplished through any desired data association technique such as those known or practiced in the art. For example, the association may be accomplished either manually or automatically. Automatic association techniques may include, for example, a database search, a database merge, GREP, AGREP, SQL, using a key field in the tables to speed searches, sequential searches through all the tables and files, sorting records in the file according to a known order to simplify lookup, and/or the like. The association step may be accomplished by a database merge function, for example, using a "key field" in preselected databases or data sectors. Various database tuning steps are contemplated to optimize database performance. For example, frequently used files such as indexes may be placed on separate file systems to reduce In/Out ("I/O") bottlenecks.

[0067] One skilled in the art will also appreciate that, for security reasons, any databases, systems, devices, servers or other components of the system may consist of any combination thereof at a single location or at multiple locations, wherein each database or system includes any of various suitable security features, such as firewalls, access codes, encryption, decryption, compression, decompression, and/or the like.

[0068] Encryption may be performed by way of any of the techniques now available in the art or which may become available-e.g., Twofish, RSA, El Gamal, Schorr signature, DSA, PGP, PKI, GPG (GnuPG), and symmetric and asymmetric cryptosystems.

[0069] The computing unit of the device may be further equipped with an Internet browser connected to the Internet or an intranet using standard dial-up, cable, DSL or any other Internet protocol known in the art. Transactions originating at a device may pass through a firewall in order to prevent unauthorized access from users of other networks. Further, additional firewalls may be deployed between the varying components of the system to further enhance security.

[0070] A firewall may include any hardware and/or software suitably configured to protect ACS components and/or enterprise computing resources from users of other networks. Further, a firewall may be configured to limit or restrict access to various systems and components behind the firewall for devices connecting through a web server. Firewall may reside in varying configurations including Stateful Inspection, Proxy based, access control lists, and Packet Filtering among others. Firewall may be integrated within a web server or any other ACS components or may further reside as a separate entity. A firewall may implement network address translation ("NAT") and/or network address port translation ("NAPT"). A firewall may accommodate various tunneling protocols to facilitate secure communications, such as those used in virtual private networking. A firewall may implement a demilitarized zone ("DMZ") to facilitate communications with a public network such as the Internet. A firewall may be integrated as software within an Internet server, any other application server components or may reside within another computing device or may take the form of a standalone hardware component.

[0071] The computers discussed herein may provide a suitable website or other Internet-based graphical user interface which is accessible by users. In various embodiments, the Microsoft Internet Information Server (IIS), Microsoft Transaction Server (MTS), and Microsoft SQL Server, are used in conjunction with the Microsoft operating system, Microsoft NT web server software, a Microsoft SQL Server database system, and a Microsoft Commerce Server. Additionally, components such as Access or Microsoft SQL Server, Oracle, Sybase, Informix

MySQL, Interbase, etc., may be used to provide an Active Data Object (ADO) compliant database management system. In various embodiments, the Apache web server is used in conjunction with a Linux operating system, a MySQL database, and the Perl, PHP, and/or Python programming languages.

[0072] Any of the communications, inputs, storage, databases or displays discussed herein may be facilitated through a website having web pages. The term "web page" as it is used herein is not meant to limit the type of documents and applications that might be used to interact with the user. For example, a typical website might include, in addition to standard HTML documents, various forms, Java applets, JavaScript, active server pages (ASP), common gateway interface scripts (CGI), extensible markup language (XML), dynamic HTML, cascading style sheets (CSS), AJAX (Asynchronous Javascript And XML), helper applications, plug-ins, and the like. A server may include a web service that receives a request from a web server, the request including a URL (http://yahoo.com/stockquotes/ge) and an IP address (123.56.789.234). The web server retrieves the appropriate web pages and sends the data or applications for the web pages to the IP address. Web services are applications that are capable of interacting with other applications over a communications means, such as the internet. Web services are typically based on standards or protocols such as XML, SOAP, AJAX, WSDL and UDDI. Web services methods are well known in the art, and are covered in many standard texts. See, e.g., ALEX NGHIEM, IT WEB SERVICES: A ROADMAP FOR THE ENTERPRISE (2003).

[0073] Middleware may include any hardware and/or software suitably configured to facilitate communications and/or process transactions between disparate computing systems. Middleware components are commercially available and known in the art. Middleware may be implemented through commercially available hardware and/or software, through custom hardware and/or software components, or through a combination thereof. Middleware may reside in a variety of configurations and may exist as a standalone system or may be a software component residing on the Internet server. Middleware may be configured to process transactions between the various components of an application server and any number of internal or external systems for any of the purposes disclosed herein. WebSphere MQTM (formerly MQSeries) by IBM, Inc. (Armonk, NY) is an example of a commercially available middleware product. An Enterprise Service Bus ("ESB") application is another example of middleware.

[0074] Practitioners will also appreciate that there are a number of methods for displaying data within a browser-based document. Data may be represented as standard text or within a fixed list, scrollable list, drop-down list, editable text field, fixed text field, pop-up window, and the like. Likewise, there are a number of methods available for modifying data in a web page such as, for example, free text entry using a keyboard, selection of menu items, check boxes, option boxes, and the like.

[0075] The system and method may be described herein in terms of functional block components, screen shots, optional selections and various processing steps. It should be appreciated that such functional blocks may be realized by any number of hardware and/or software components configured to perform the specified functions. For example, the system may employ various integrated circuit components, e.g., memory elements, processing elements, logic elements, look-up tables, and the like, which may carry out a variety of functions under the control of one or more microprocessors or other control devices. Similarly, the software elements of the system may be implemented with any programming or scripting language such as C, C++, C#, Java, JavaScript, VBScript, Macromedia Cold Fusion, COBOL, Microsoft Active Server Pages, assembly, PERL, PHP, awk, Python, Visual Basic, SQL Stored Procedures, PL/SQL, any UNIX shell script, and extensible markup language (XML) with the various algorithms being implemented with any combination of data structures, objects, processes, routines or other programming elements. Further, it should be noted that the system may employ any number of conventional techniques for data transmission, signaling, data processing, network control, and the like. Still further, the system could be used to detect or prevent security issues with a client-side scripting language, such as JavaScript, VBScript or the like. For a basic introduction of cryptography and network security, see any of the following references: (1) "Applied Cryptography: Protocols, Algorithms, And Source Code In C," by Bruce Schneier, published by John Wiley & Sons (second edition, 1995); (2) "Java Cryptography" by Jonathan Knudson, published by O'Reilly & Associates (1998); (3) "Cryptography & Network Security: Principles & Practice" by William Stallings, published by Prentice Hall.

[0076] As will be appreciated by one of ordinary skill in the art, the system may be embodied as a customization of an existing system, an add-on product, a processing apparatus executing upgraded software, a stand alone system, a distributed system, a method, a data processing system, a device for data processing, and/or a computer program product. Accordingly, any portion of the system or a module may take the form of a processing apparatus executing code, an internet based embodiment, an entirely hardware embodiment, or an embodiment combining aspects of the internet, software and hardware. Furthermore, the system may take the form of a computer program product on a computer-readable storage medium having computer-readable program code means embodied in the storage medium. Any suitable computer-readable storage medium may be utilized, including hard disks, CD-ROM, optical storage devices, magnetic storage devices, and/or the like.

[0077] The system and method is described herein with reference to screen shots, block diagrams and flowchart illustrations of methods, apparatus (e.g., systems), and computer program products according to various embodiments. It will be understood that each functional block of the block diagrams and the flowchart illustrations, and combinations of functional blocks in the block diagrams and flowchart illustrations, respectively, can be implemented by computer program instructions.

[0078] These computer program instructions may be loaded onto a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions that execute on the computer or other programmable data processing apparatus create means for implementing the functions specified in the flowchart block or blocks. These computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the function specified in the flowchart block or blocks. The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer-implemented process such that the instructions which execute on the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart block or blocks.

[0079] Accordingly, functional blocks of the block diagrams and flowchart illustrations support combinations of means for performing the specified functions, combinations of steps for performing the specified functions, and program instruction means for performing the specified functions. It will also be understood that each functional block of the block diagrams and flowchart illustrations, and combinations of functional blocks in the block diagrams and flowchart illustrations, can be implemented by either special purpose hardware-based computer systems which perform the specified functions or steps, or suitable combinations of special purpose hardware and computer instructions. Further, illustrations of the process flows and the descriptions thereof may make reference to user windows, webpages, websites, web forms, prompts, etc. Practitioners will appreciate that the illustrated steps described herein may comprise in any number of configurations including the use of windows, webpages, web forms, popup windows, prompts and the like. It should be further appreciated that the multiple steps as illustrated and described may be combined into single webpages and/or windows but have been expanded for the sake of simplicity. In other cases, steps illustrated and described as single process steps may be separated into multiple webpages and/or windows but have been combined for simplicity.

[0080] The term "non-transitory" is to be understood to remove only propagating transitory signals per se from the claim scope and does not relinquish rights to all standard computer-readable media that are not only propagating transitory signals per se.

[0081] Benefits, other advantages, and solutions to problems have been described herein with regard to specific embodiments. However, the benefits, advantages, solutions to problems, and any elements that may cause any benefit, advantage, or solution to occur or become more pronounced are not to be construed as critical, required, or essential features or elements of the disclosure. Reference to an element in the singular is not intended to mean "one and only one" unless explicitly so stated, but rather "one or more." Although the disclosure includes a method, it is contemplated that it may be embodied as computer program instructions on a tangible computer-readable carrier, such as a magnetic or optical memory or a magnetic or optical disk. All structural, chemical, and functional equivalents to the elements of the above-described exemplary embodiments that are known to those of ordinary skill in the art are intended to be encompassed by the present claims. Moreover, it is not necessary for a device or method to address each and every problem sought to be solved by the present disclosure, for it to be encompassed by the present claims. Furthermore, no element, component, or method step in the present disclosure is intended to be dedicated to the public regardless of whether the element, component, or method step is explicitly recited in the claims. As used herein, the terms "comprises", "comprising", or any other variation thereof, are intended to cover a non-exclusive inclusion, such that a process, method, article, or apparatus that comprises a list of elements does not include only those elements but may include other elements not expressly listed or inherent to such process, method, article, or apparatus.

REFERENCES CITED IN THE DESCRIPTION

Cited references

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US62023059 [0001]
- WO0033545A2 [0006]
- EP0800144A2 [0007]
- US2003131050A [0008]

Non-patent literature cited in the description

- **DILIP NAIK**INTERNET STANDARDS AND PROTOCOLS, 1998, [0049]
- **SYBEX**JAVA 2 COMPLETE, 1999, [0049]
- **DEBORAH RAYERIC RAY**MASTERING HTML 4.0, 1997, [0049]
- **LOSHIN**TCP/IP CLEARLY EXPLAINED, 1997, [0049]
- **DAVID GOURLEYBRIAN TOTTY**HTTP, THE DEFINITIVE GUIDE, 2002, [0049]
- **GILBERT HELD**UNDERSTANDING DATA COMMUNICATIONS, 1996, [0050]

SYSTEMER OG FREMGANGSMÅDER TIL BESKEDREDIGERING

PATENTKRAV

1. Fremgangsmåde, der omfatter:

modtagelse, via en beskedserver, af en original besked fra en første enhed (110), der er adskilt fra beskedserveren;

lagring, via beskedserveren, af den originale besked i en beskeddatabase;

overførsel, via beskedserveren, af en notifikation, der indikerer, at den originale besked er tilgængelig for en anden enhed;

modtagelse, via beskedserveren, af en anmodning om nye data fra den anden enhed (120), der er adskilt fra beskedserveren, hvor den anden enhed overfører anmodningen som reaktion på modtagelse af beskednotifikationen;

overførsel, via beskedserveren, af den originale besked fra beskeddatabasen til den anden enhed (120), hvor den anden enhed (120) viser den originale besked, og hvor den anden enhed (120) lagrer den originale besked i den anden enheds lokale cache;

modtagelse, via beskedserveren, af en redigeret besked omfattende en redigeret version af den originale besked;

lagring, via beskedserveren, af den redigerede besked i beskeddatabasen;

markering, via beskedserveren, af den originale besked som ugyldig eller slettet, således at enheder ikke længere kan tilgå den originale besked;

overførsel, via beskedserveren, af en instruktion til den anden enhed (120) om at slette den originale besked fra den anden enheds lokale cache, hvor som reaktion på instruktionen den anden enhed (120) automatisk sletter den originale besked fra den anden enheds lokale cache; og

overførsel, via beskedserveren, af den redigerede besked fra beskeddatabasen til den anden enhed (120), hvor den anden enhed (120) viser den redigerede besked, hvor den anden enhed (120) lagrer den redigerede besked i den anden enheds lokale cache.

2. Fremgangsmåde ifølge krav 1, hvor beskedserveren modtager den redigerede besked fra en tredje, forskellig enhed, der er adskilt fra beskedserveren.

3. Fremgangsmåde ifølge krav 1, og som endvidere omfatter overførsel, via beskedserveren, af samtlige nye data for den anden enhed (120) siden et tidligere hentningstidspunkt til den anden enhed (120).

4. Fremgangsmåde ifølge krav 1, hvor den originale besked omfatter en tekstbesked, en e-mail eller en fil.

5. System, der omfatter:

en processor,

en håndgribelig, ikke-forbigående hukommelse, der er konfigureret til at kommunikere med processoren, hvilken håndgribelig, ikke-forbigående hukommelse har lagret instruktioner derpå, der, som reaktion på udførelse via processoren, bevirker, at systemet udfører handlinger omfattende:

modtagelse, via processoren, af en original besked fra en første enhed (110), der er adskilt fra processoren;

lagring, via processoren, af den originale besked i en beskeddatabase;

overførsel, via processoren, af en notifikation, der indikerer, at den originale besked er tilgængelig for en anden enhed;

modtagelse, via processoren, af en anmodning om nye data fra den anden enhed (120), der er adskilt fra processoren, hvor den anden enhed overfører anmodningen som reaktion på modtagelse af beskednotifikationen;

overførsel, via processoren, af den originale besked fra beskeddatabasen til den anden enhed, hvor den anden enhed viser den originale besked, og hvor den anden enhed (120) lagrer den originale besked i den anden enheds lokale cache (120);

modtagelse, via processoren, af en redigeret besked omfattende en redigeret version af den originale besked;

lagring, via processoren, af den redigerede besked i beskeddatabasen;

markering, via processoren, af den originale besked som ugyldig eller slettet, således at enheder ikke længere kan tilgå den originale besked;

overførsel, via processoren, af en instruktion til den anden enhed (120) om at slette den originale besked fra den anden enheds lokale cache, hvor som reaktion på instruktionen den anden enhed (120) automatisk sletter den originale besked fra den anden enheds lokale cache; og

overførsel, via processoren, af den redigerede besked til den anden enhed (120), hvor den anden enhed (120) viser den redigerede besked, hvor den anden enhed (120) lagrer den redigerede besked i den anden enheds lokale cache.

6. System ifølge krav 5, hvor den redigerede besked modtages fra en tredje, forskellig enhed, der er adskilt fra processoren.

7. System ifølge krav 5, hvor handlingerne endvidere omfatter overførsel, via processoren, af samtlige nye data for den anden enhed (120) siden et tidligere hentningstidspunkt til den anden enhed (120).

8. Fremstillingsgenstand indbefattende et ikke-forbigående, håndgribeligt computerlæsbart lagringsmedium med instruktioner lagret derpå, og der, som reaktion på udførelse via en beskedserver, bevirker, at beskedserveren foretager handlinger, der omfatter:

modtagelse, via beskedserveren, af en original besked fra en første enhed (110), der er adskilt fra beskedserveren;

lagring, via beskedserveren, af den originale besked i en beskeddatabase;

overførsel, via beskedserveren, af en notifikation, der indikerer, at den originale besked er tilgængelig for en anden enhed;

modtagelse, via beskedserveren, af en anmodning om nye data fra den anden enhed (120), der er adskilt fra beskedserveren, hvor den anden enhed overfører anmodningen som reaktion på modtagelse af beskednotifikationen;

overførsel, via beskedserveren, af den originale besked fra beskeddatabasen til den anden enhed (120), hvor den anden enhed (120) viser den originale besked, og hvor den anden enhed (120) lagrer den originale besked i den anden enheds lokale cache;

modtagelse, via beskedserveren, af en redigeret besked omfattende en redigeret version af den originale besked;

lagring, via beskedserveren, af den redigerede besked i beskeddatabasen;

markering, via beskedserveren, af den originale besked som ugyldig eller slettet, således at enheder ikke længere kan tilgå den originale besked;

overførsel, via beskedserveren, af en instruktion til den anden enhed (120) om at slette den originale besked fra den anden enheds lokale cache, hvor som reaktion på instruktionen den anden enhed automatisk sletter den originale besked fra den anden enheds lokale cache; og

overførsel, via beskedserveren, af den redigerede besked til den anden enhed (120), hvor den anden enhed (120) viser den redigerede besked, hvor den anden enhed (120) lagrer den redigerede besked i den anden enheds lokale cache.

9. Fremstillingsgenstand ifølge krav 8, hvor den redigerede besked modtages fra en tredje, forskellig enhed, der er adskilt fra beskedserveren.

10. Fremstillingsgenstand ifølge krav 8, hvor handlingerne endvidere omfatter overførsel, via beskedserveren, af samtlige nye data for den anden enhed (120) siden et tidligere hentningstidspunkt til den anden enhed (120).

DRAWINGS

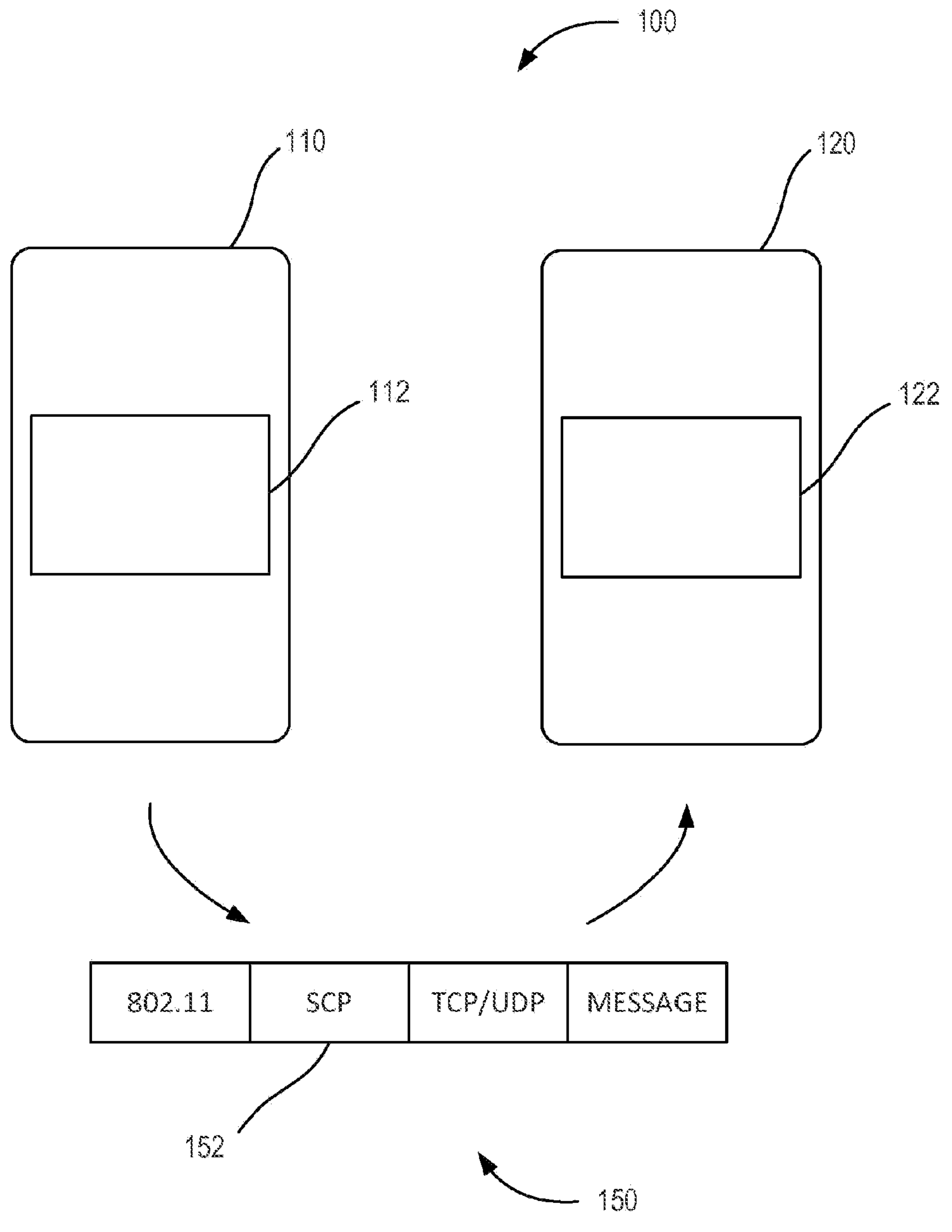


FIG. 1

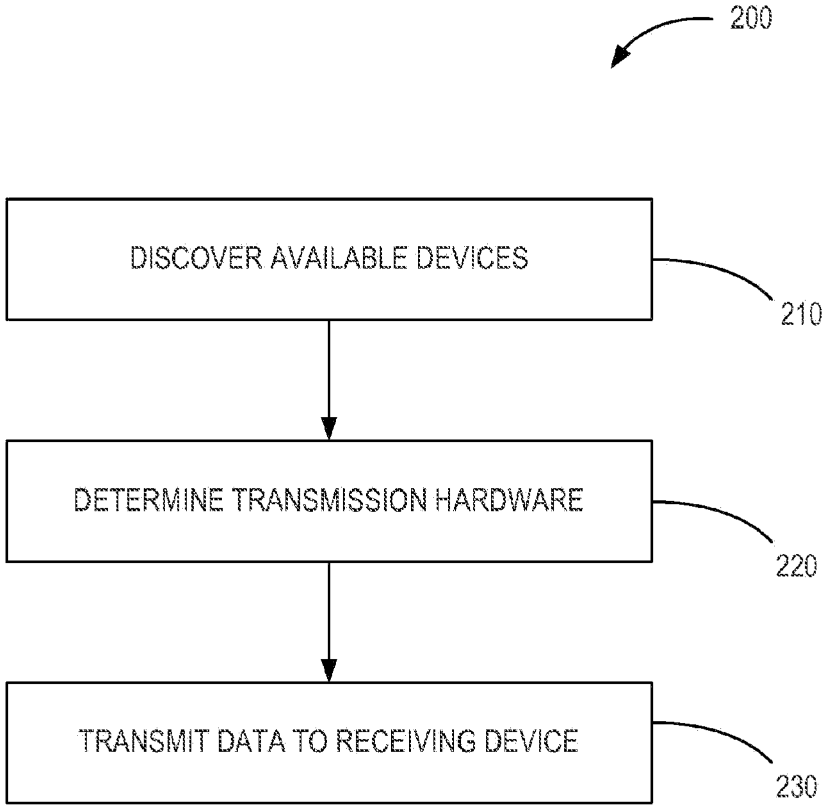

200
DISCOVER AVAILABLE DEVICES
210
DETERMINE TRANSMISSION HARDWARE
220
TRANSMIT DATA TO RECEIVING DEVICE
230

FIG. 2

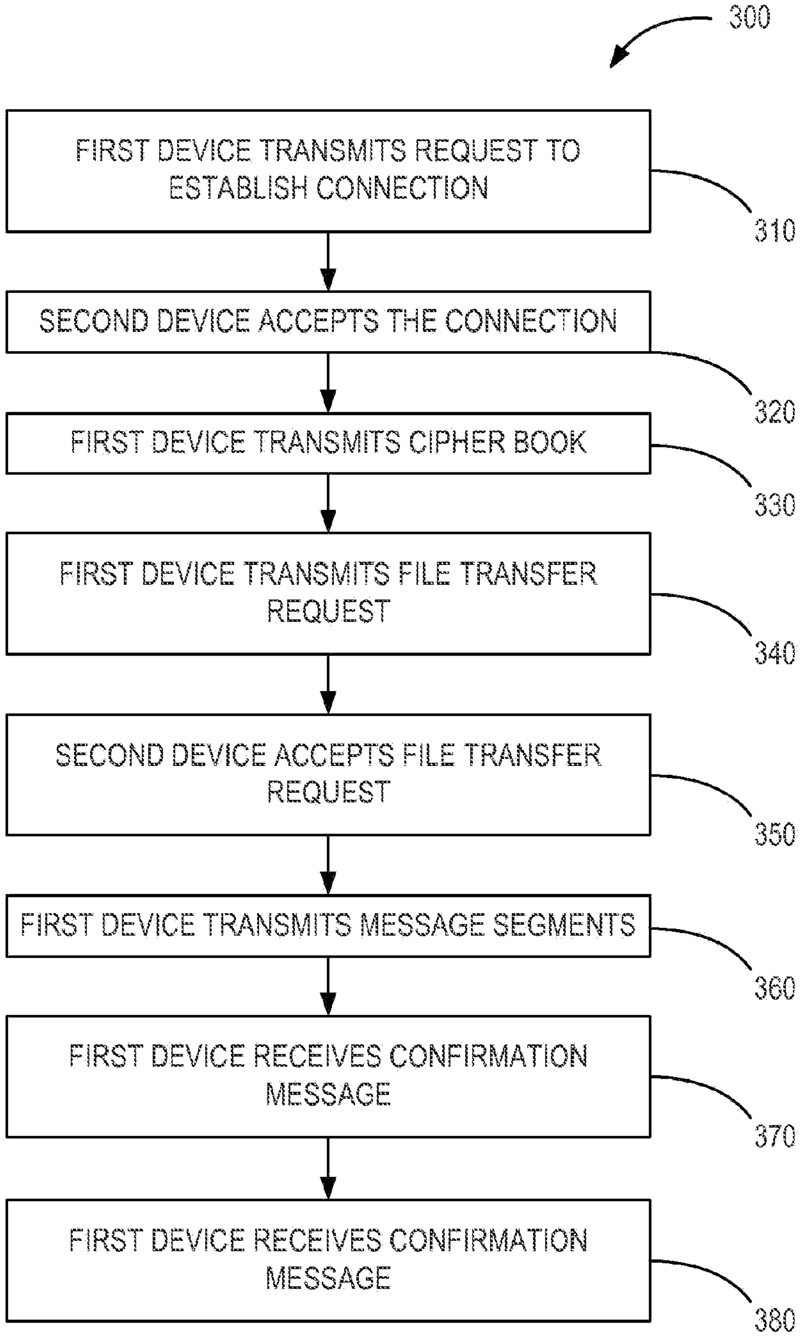
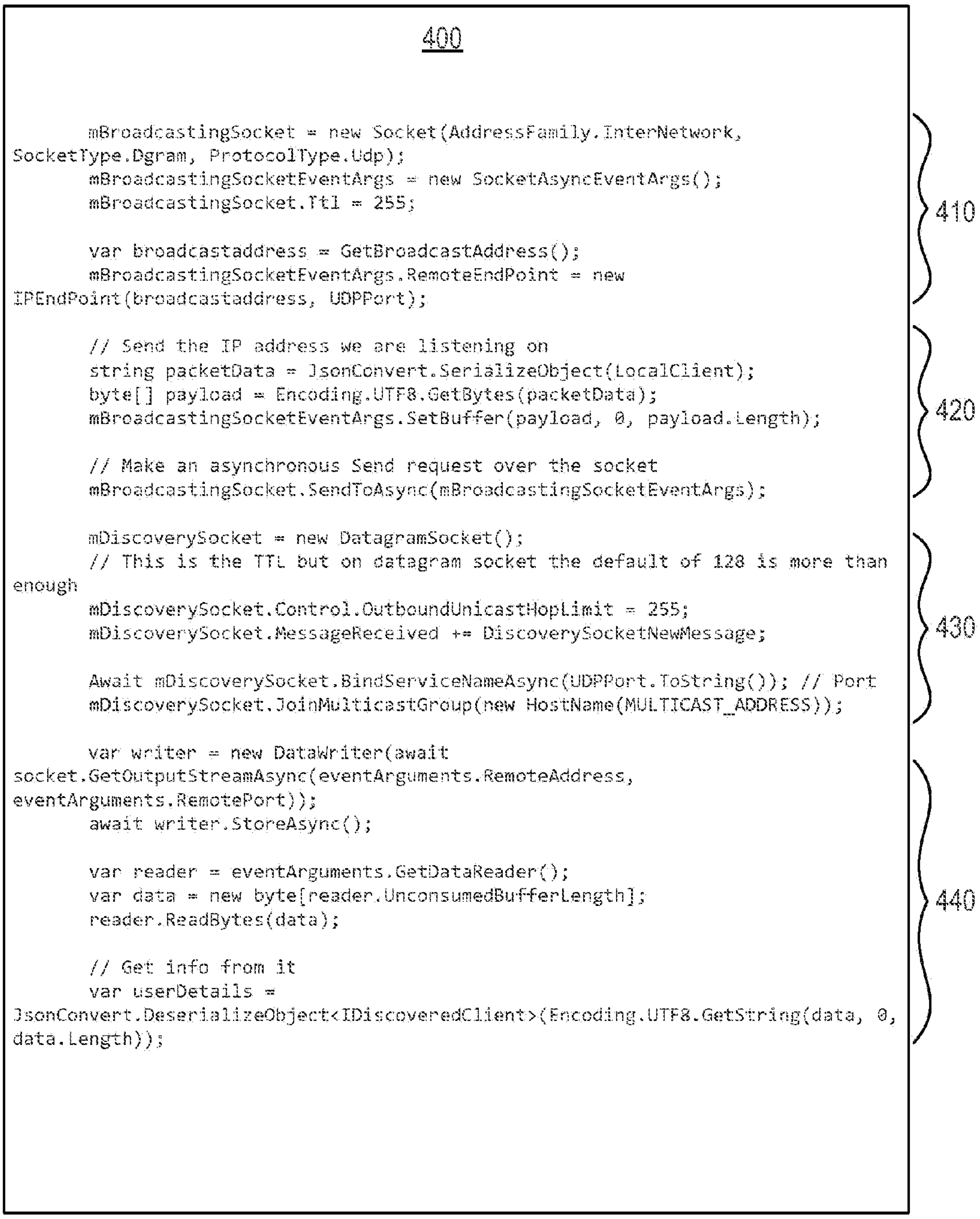

300
FIRST DEVICE TRANSMITS REQUEST TO ESTABLISH CONNECTION
310
SECOND DEVICE ACCEPTS THE CONNECTION
320
FIRST DEVICE TRANSMITS CIPHER BOOK
330
FIRST DEVICE TRANSMITS FILE TRANSFER REQUEST
340
SECOND DEVICE ACCEPTS FILE TRANSFER REQUEST
350
FIRST DEVICE TRANSMITS MESSAGE SEGMENTS
360
FIRST DEVICE RECEIVES CONFIRMATION MESSAGE
370
FIRST DEVICE RECEIVES CONFIRMATION MESSAGE
380

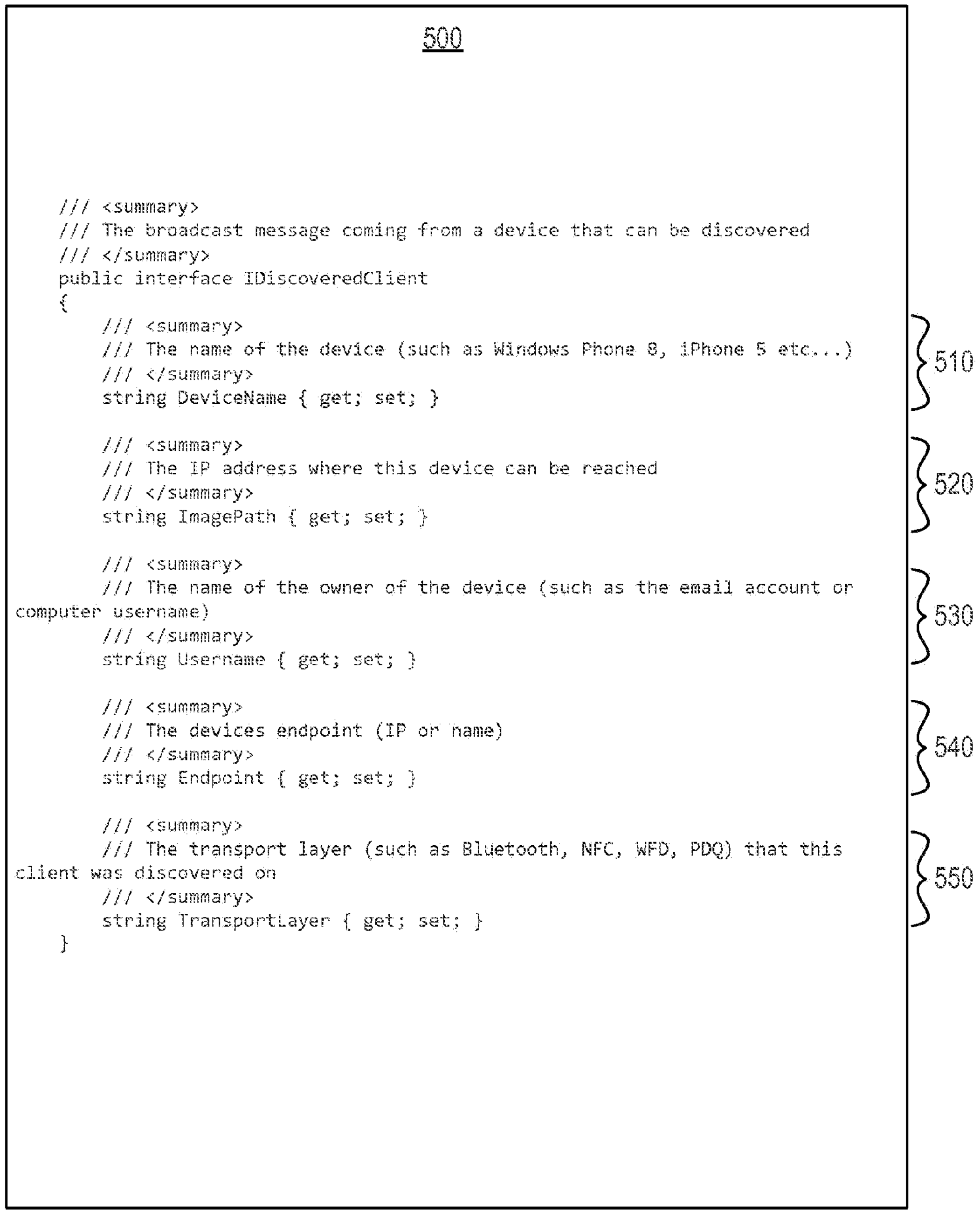
FIG. 3



400
410
420
430
440

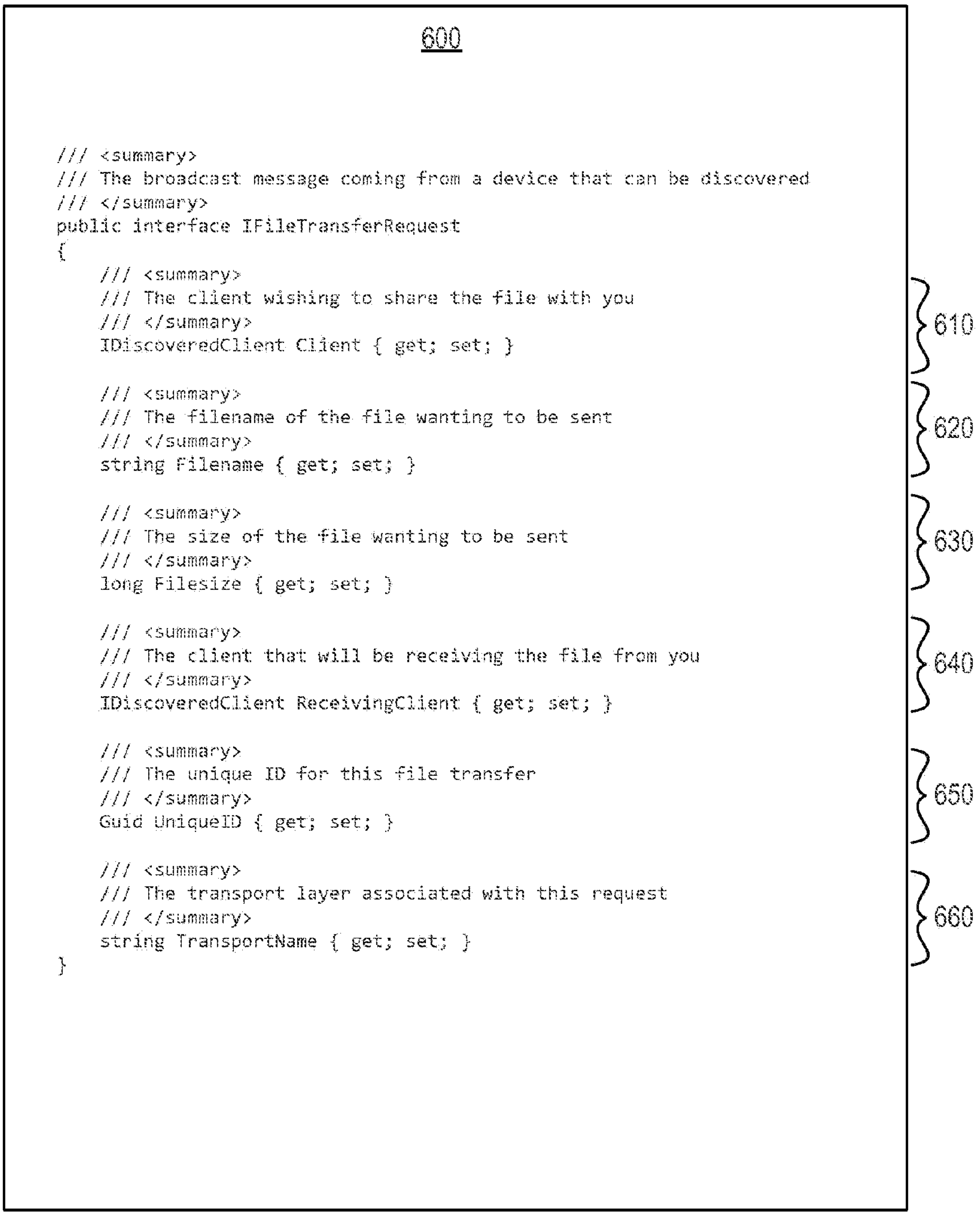
```
        mBroadcastingSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);
        mBroadcastingSocketEventArgs = new SocketAsyncEventArgs();
        mBroadcastingSocket.Ttl = 255;

        var broadcastaddress = GetBroadcastAddress();
        mBroadcastingSocketEventArgs.RemoteEndPoint = new
IPEndPoint(broadcastaddress, UDPPort);

        // Send the IP address we are listening on
        string packetData = JsonConvert.SerializeObject(LocalClient);
        byte[] payload = Encoding.UTF8.GetBytes(packetData);
        mBroadcastingSocketEventArgs.SetBuffer(payload, 0, payload.Length);

        // Make an asynchronous Send request over the socket
        mBroadcastingSocket.SendToAsync(mBroadcastingSocketEventArgs);

        mDiscoverySocket = new DatagramSocket();
        // This is the TTL but on datagram socket the default of 128 is more than
enough
        mDiscoverySocket.Control.OutboundUnicastHopLimit = 255;
        mDiscoverySocket.MessageReceived += DiscoverySocketNewMessage;

        Await mDiscoverySocket.BindServiceNameAsync(UDPPort.ToString()); // Port
        mDiscoverySocket.JoinMulticastGroup(new HostName(MULTICAST_ADDRESS));

        var writer = new DataWriter(await
socket.GetOutputStreamAsync(eventArguments.RemoteAddress,
eventArguments.RemotePort));
        await writer.StoreAsync();

        var reader = eventArguments.GetDataReader();
        var data = new byte[reader.UnconsumedBufferLength];
        reader.ReadBytes(data);

        // Get info from it
        var userDetails =
JsonConvert.DeserializeObject<IDiscoveredClient>(Encoding.UTF8.GetString(data, 0,
data.Length));
```

FIG. 4


500
510
520
530
540
550

```
    /// <summary>
    /// The broadcast message coming from a device that can be discovered
    /// </summary>
    public interface IDiscoveredClient
    {
        /// <summary>
        /// The name of the device (such as Windows Phone 8, iPhone 5 etc...)
        /// </summary>
        string DeviceName { get; set; }

        /// <summary>
        /// The IP address where this device can be reached
        /// </summary>
        string ImagePath { get; set; }

        /// <summary>
        /// The name of the owner of the device (such as the email account or
computer username)
        /// </summary>
        string Username { get; set; }

        /// <summary>
        /// The devices endpoint (IP or name)
        /// </summary>
        string Endpoint { get; set; }

        /// <summary>
        /// The transport layer (such as Bluetooth, NFC, WFD, PDQ) that this
client was discovered on
        /// </summary>
        string TransportLayer { get; set; }
    }
```

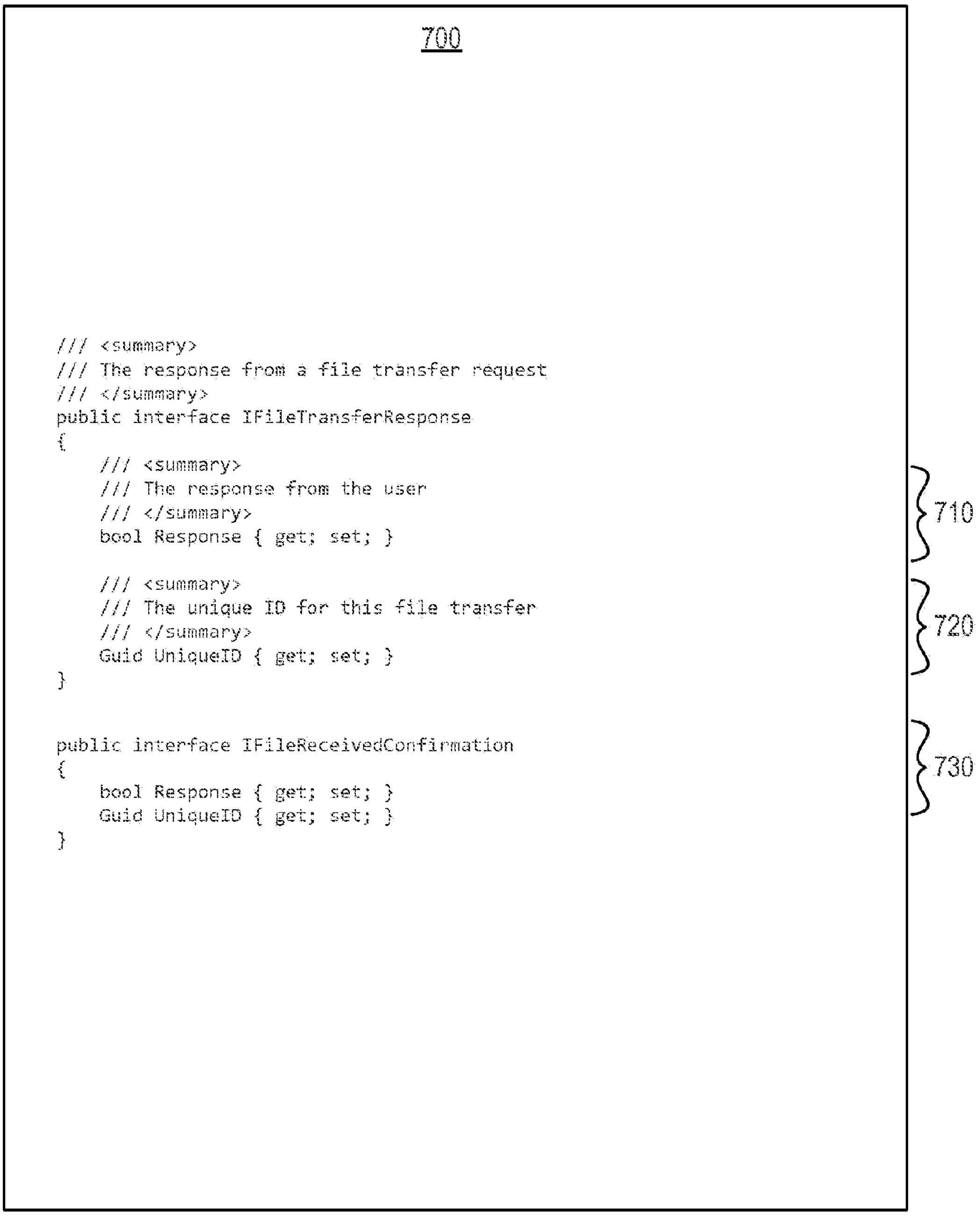
FIG. 5



600
610
620
630
640
650
660

```
/// <summary>
/// The broadcast message coming from a device that can be discovered
/// </summary>
public interface IFileTransferRequest
{
    /// <summary>
    /// The client wishing to share the file with you
    /// </summary>
    IDiscoveredClient Client { get; set; }

    /// <summary>
    /// The filename of the file wanting to be sent
    /// </summary>
    string Filename { get; set; }

    /// <summary>
    /// The size of the file wanting to be sent
    /// </summary>
    long Filesize { get; set; }

    /// <summary>
    /// The client that will be receiving the file from you
    /// </summary>
    IDiscoveredClient ReceivingClient { get; set; }

    /// <summary>
    /// The unique ID for this file transfer
    /// </summary>
    Guid UniqueID { get; set; }

    /// <summary>
    /// The transport layer associated with this request
    /// </summary>
    string TransportName { get; set; }
}
```

FIG. 6



```
/// <summary>
/// The response from a file transfer request
/// </summary>
public interface IFileTransferResponse
{
    /// <summary>
    /// The response from the user
    /// </summary>
    bool Response { get; set; }

    /// <summary>
    /// The unique ID for this file transfer
    /// </summary>
    Guid UniqueID { get; set; }
}


public interface IFileReceivedConfirmation
{
    bool Response { get; set; }
    Guid UniqueID { get; set; }
}
```

FIG. 7

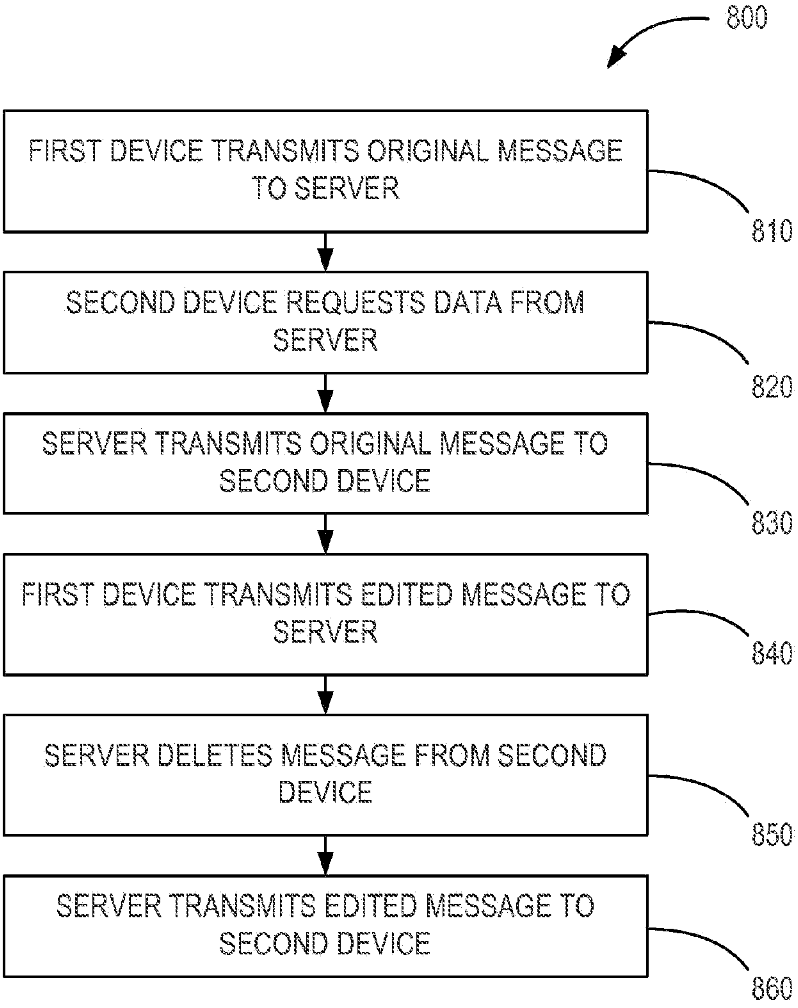

800
FIRST DEVICE TRANSMITS ORIGINAL MESSAGE TO SERVER
810
SECOND DEVICE REQUESTS DATA FROM SERVER
820
SERVER TRANSMITS ORIGINAL MESSAGE TO SECOND DEVICE
830
FIRST DEVICE TRANSMITS EDITED MESSAGE TO SERVER
840
SERVER DELETES MESSAGE FROM SECOND DEVICE
850
SERVER TRANSMITS EDITED MESSAGE TO SECOND DEVICE
860

FIG. 8