(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 October 2017 (05.10.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/172651 A1

(51) **International Patent Classification**:
G06F 9/46 (2006.01) *G06F 9/50* (2006.01)

(21) **International Application Number**:
PCT/US2017/024381

(22) **International Filing Date**:
28 March 2017 (28.03.2017)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
62/315,498 30 March 2016 (30.03.2016) US
15/394,467 29 December 2016 (29.12.2016) US

(71) **Applicant**: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) **Inventors**: **PLUMB, Graham C.**; MICROSOFT TECHNOLOGY LICENSING, LLC, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US). **SCHLESINGER, Benny**; MICROSOFT TECHNOLOGY LICENSING, LLC, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US). **RINCON, Lilian Dearith**; MICROSOFT TECHNOLOGY LICENSING, LLC, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US). **MOHAMMED, Farookh P.**; MICROSOFT TECHNOLOGY LICENSING, LLC, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) **Agents**: **MINHAS, Sandip** et al.; MICROSOFT CORPORATION, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US).

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

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) **Title**: SELECTING AN AUTONOMOUS SOFTWARE AGENT

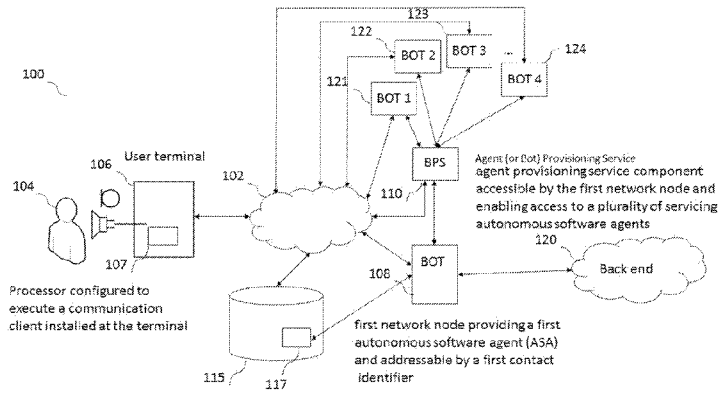

FIG. 1

(57) **Abstract**: A communication system comprising a user terminal having a processor configured to execute a communication client installed at the terminal and a display. The communication client is configured to display contact identifiers on the display, each contact identifier being selectable to initiate a communication event with a network node addressed by the contact identifier. A first network node provides a first autonomous software agent (ASA) and is addressable by a first contact identifier displayed on the user terminal, the ASA configured to receive an intent conveyed by a user at the user terminal; an agent provisioning service component accessible by the first network node and enabling access to a plurality of servicing autonomous software agents (SASA), each capable of implementing an action. The first network node is configured to respond to the received intent and to select one of the SASAs to implement an action corresponding to the user intent.

WO 2017/172651 A1

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

SELECTING AN AUTONOMOUS SOFTWARE AGENT

Background

[0001] Communication systems allow users to communicate with each other over a communication network by conducting a communication event over the network. The network may be, for example, the Internet or public switched telephone network (PSTN). The communication event may be, for example, a voice or video call or an instant message (IM) chat session. During a call, audio and/or video signals can be transmitted between nodes of the network, thereby allowing users to transmit and receive audio data (such as speech) and/or video data (such as webcam video) to each other in a communication session over the communication network.

[0002] Such communication systems include Voice or Video over Internet protocol (VoIP) systems. To use a VoIP system, a user installs and executes client software on a user device. The client software sets up VoIP connections as well as providing other functions such as registration and user authentication. In addition to voice communication, the client may also set up connections for communication events, for instant messaging ("IM"), screen sharing, or whiteboard sessions.

[0003] A communication event may be conducted between a user(s) and an intelligent software agent, sometimes referred to as a "bot". A software agent is an autonomous computer program that carries out tasks on behalf of users in a relationship of agency. The software agent runs continuously some or all of for the duration of the communication event, awaiting inputs which, when detected, trigger automated tasks to be performed on those inputs by the agent. A software agent may exhibit artificial intelligence (AI), whereby it can simulate certain human intelligence processes, for example to generate human-like responses to inputs from the user, thus facilitating a two-way conversation between the user and the software agent via the network.

Summary

[0004] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

[0005] The present disclosure relates to a communication system comprising: a user terminal, a first network node, and an agent provisioning service component. The user terminal has a processor configured to execute a communication client installed at the

terminal and a display. The communication client is configured to cause contact identifiers to be displayed on the display, each contact identifier being selectable to initiate a communication event with a network node addressed by the contact identifier. The first network node provides a first autonomous software agent (ASA) and addressable by a first contact identifier displayed on the user terminal, the autonomous software agent configured to receive an intent conveyed by a user at the user terminal. The agent provisioning service component is accessible by the first network node and enabling access to a plurality of servicing autonomous software agents (SASA), each capable of implementing an action. The first network node is configured to respond to the received intent and to select one of the servicing autonomous software agents to implement an action corresponding to the user intent.

Brief Description of Figures

[0006] For a better understanding of the present subject matter and to show how the same may be carried into effect, reference is made by way of example to the following figures, in which:

- FIG 1 shows a schematic block diagram of a communication system according to a first embodiment of the invention;
- FIG 2 shows a schematic block diagram of a user terminal;
- FIG 2a shows a schematic block diagram, of a software agent;
- FIG 3 shows a schematic block diagram of a communication system according to a second embodiment of the invention;
- FIG 3a shows a flow chart of a method for autonomously actioning a user intent;
- FIG 4a shows a flow chart of a first authentication method;
- FIG 4 shows a flow chart of a second authentication method;
- FIG 5 shows a flow chart of a third authentication method;
- FIGs 6 and 7 are flow charts of bot selection;
- FIGs 8A, 8B, 8C, and 8D show examples of interactions between users and bots;
- FIGs 9, 10, and 11 show examples of bot provisioning services;
- FIGs 12A, 12B, 12C, 12D, 12E, 12F, 12G, 12H, and 12I illustrate examples of interactions between users and bots on a user interface.

Detailed Description of Embodiments

Selecting Bots

[0007] In the present disclosure, an autonomous software agent can select from multiple servicing entities a suitable entity for actioning an intent of a user. The intent can be derived from a conversation in which the user is engaged, with the autonomous software agent or another user, at a remote user terminal. Alternatively, the intent can be conveyed in a message directly from the user to the agent, for example in a video or audio call or text message. Where the intent is derived from a conversation, the term 'listening bot' is utilised.

[0008] The servicing entity selected by the agent can be another autonomous agent, which can be placed in a communication event such as a chat or call, with the user terminal to enable the user to action the intent.

[0009] An intent may be associated with a context defined by context data, which may be made available to the 'listening bot' or the selected bot, subject to permissions.

[0010] As an example, the 'listening bot' can pick out an intent for 'book a room in Dublin' or 'make a reservation' from a conversation being held at the user terminal or in a direct chat message. It can optionally obtain context data – e.g. 'for a work trip'. The user's actual location may be context data in this case - e.g. if the user's present location is Paris, the bot may also pick up that it should provide transport options from Paris to Dublin, as well as reservation options for hotels in Dublin.

[0011] Another example is described later in more detail with reference to FIGs 8A-8D.

General communication system

[0012] FIG 1 shows a block diagram of a communication system 100 according to a first embodiment of the present invention. The communication system 100 comprises a communications network 102, to which is connected a first user device 106, a user account database 115, a bot 108 (remote from the user device 106), a bot back end 120, and a bot provisioning service 110 (agent provisioning service). The network 102 is a packet-based network, such as the Internet.

[0013] The user device 106, is available to a first user 104. The user device 106 is shown to be executing a respective version of a communication client 107.

[0014] The client 107 is for effecting communication events over a communication service within the communications system via the network, such as audio and/or video calls and/or other communication event(s) such as a whiteboard, instant messaging (IM) or screen

sharing sessions, between the user 104 and another user (not shown) or between the user 104 and the bot 108.

[0015] The communication system 100 may be based on voice or video over internet protocols (VoIP) systems. These systems can be beneficial to the user as they are often of significantly lower cost than conventional fixed line or mobile cellular networks, particularly for long-distance communication. The client software sets up the VoIP connections as well as providing other functions such as registration and user authentication, e.g. based on login credentials such as a username and associated password. To effect a communication event, data is captured from the user at their device. For example, in a call the data comprises audio data captured via a microphone of the respective device and embodying that user's speech (call audio) transmitted as an audio stream via the network 102, and may additionally comprise video data captured via a camera of the respective device and embodying a moving image of that user (call video) transmitted as a video stream via the network 102. The call audio/video is captured and encoded at the transmitting device before transmission and decoded at the receiving terminal. The user 104 can thus communicate via the communications network 102 audibly and (for a video call) visually as well as by instant messaging. Alternatively, the call may be established via a cellular or fixed-line (e.g. PSTN) connection.

[0016] Where the remote terminal is a user terminal, the call or chat data is output to a user at the user terminal. Herein a first party terminal is the current user terminal and a third party terminal is a remote user terminal. Where the remote terminal is a bot, the call or chat data is received by the bot and processed to deduce an intent from the conversation being conducted. Context data may also be passed to the bot from the user terminal in some embodiments.

[0017] A communication event may be real-time in the sense that there is at most a short delay, for instance about 2 seconds or less, between data (e.g. call audio/video) being captured from one the user at their device and the captured data being received by the bot 108.

[0018] Only one user 104 of the communication system 100 is shown in FIG 1, but as will be readily appreciated there may be many more users of the communication system 100, each of whom operates their own device(s) and client(s) to enable them to communicate with other users via the communication network 102. For example, group communication events, such as group calls (e.g. video conferences) may be conducted between three or more users of the communication system 100.

User terminal

[0019] FIG 2 shows a block diagram of the user terminal 106. The user terminal 106 is a computer device which can take a number of forms e.g. that of a desktop or laptop computer device, mobile phone (e.g. smartphone), tablet computing device, wearable computing device (headset, smartwatch etc.), television (e.g. smart TV) or other wall-mounted device (e.g. a video conferencing device), set-top box, gaming console, etc. The user terminal 106 comprises a processor 202 (formed one or more processing units, e.g. CPUs, GPUs, bespoke processing units, etc.) and the following components, which are connected to the processor 202: memory 200 (formed on one or more memory units, e.g. RAM units, direct-access memory units, etc.); a network interface(s) 204; at least one input device, e.g. a keyboard 209, mouse 210, camera 207, and microphone(s) 208 as shown; at least one output device, e.g. a loudspeaker (206) and a display(s) 205. The user terminal 106 connects to the network 102 via its network interface 204, so that the processor 202 can transmit and receive data to/from the network 102. The network interface 204 may be a wired interface (e.g. Ethernet, FireWire, Thunderbolt, USB, etc.) or wireless interface (e.g. Wi-Fi, Bluetooth, NFC, etc.). The memory holds the code of the communication client 107 for execution on the processor 202. The client 107 may be e.g. a stand-alone communication client application, plugin to another application such as a Web browser, etc. that is run on the processor in an execution environment provided by the other application. The client 107 has a user interface (UI) for receiving information from and outputting information to the user 104. For example, the client 107 can output decoded IM messages via the display 205. The display 205 may comprise a touchscreen so that it also functions as an input device. For audio/video calls, the client captures call audio/video via the microphone 208 and camera 207 respectively; which it encodes and transmits to one or more other user devices of other user(s) participating in a call. Any of these components may be integrated in the user device 106, or external components connected to the user device 106 via a suitable external interface.

[0020] Returning to FIG 1, the user account database 115 stores, for each user of the communication system 100, associated user account data in association with a unique user identifier of that user. Thus users are uniquely identified within the communication system 100 by their user identifiers, and rendered 'visible' to one another within the communication system 100 by the database 115, in the sense that they are made aware of each other's existence by virtue of the information held in the database 115. The database 115 can be

implemented in any suitable manner, for example as a distributed system, whereby the data it holds is distributed between multiple data storage locations.

Login mechanism

[0021] The communication system 100 provides a login mechanism, whereby users of the communication system can create or register unique user identifiers for themselves for use within the communication system, such as a username created within the communication system or an existing email address that is registered within the communication system as used as a username once registered. The user also creates an associated password, and the user identifier and password constitute credentials of that user. To gain access to the communication system 100 from a particular device, the user inputs their credentials to the client on that device, which is verified against that user's user account data stored within the user account database 115 of the communication system 100. Users are thus uniquely identified by associated user identifiers within the communication system 100. This is exemplary, and the communication system 100 may provide alternative or additional authentication mechanism, for example based on digital certificates.

[0022] At a given time, each username can be associated within the communication system with one or more instances of the client at which the user is logged. Users can have communication client instances running on other devices associated with the same log in/registration details. In the case where the same user, having a particular username, can be simultaneously logged in to multiple instances of the same client application on different devices, a server (or similar device or system) is arranged to map the username (user ID) to all of those multiple instances but also to map a separate sub-identifier (sub-ID) to each particular individual instance. Thus the communication system is capable of distinguishing between the different instances whilst still maintaining a consistent identity for the user within the communication system.

[0023] In addition to authentication, the client 107, can provide additional functionality within the communication system, such as presence and contact-management mechanisms. The former allows users to see each other's presence status (e.g. offline or online, and/or more detailed presence information such as busy, available, inactive, etc.). The latter allows users to add each other as contacts within the communication system. A user's contacts are stored within the communication system 100 in association with their user identifier as part of their user account data in the database 115, so that they are accessible to the user from any device at which the user is logged on. To add another user as a contact, the user uses their client 107 to send a contact request to the other user. If the

other user accepts the contact request using their own client, the users are added to each other's contacts in the database 115. A contact is displayed on the display of the user terminal with a contact identifier which when accessed causes a communication event to be established with a network node associated with that contact.

[0024] All the information relating to presence and contact management mechanism, as well as personalised data for the users accessible by the bot 108 may be stored in a personal data platform 117 comprised in the user account database. In certain embodiments, the information relating to presence and contact management mechanism may alternatively be managed by a separate address book service.

Bot

[0025] FIG 2A shows a block diagram of the bot 108. The bot 108 is implemented on a computer system, which comprises one or more processors 10 (each formed of one or more processing units), memory 12 (formed of one or more memory units, which may be localized or distributed across multiple geographic locations) and a network interface 16 connected to the processor(s) 10. The memory holds code 14 for execution on the processor 10. The code 14 includes the code of the bot functionality. The bot 108 connects to the network 102 via the network interface 16. As will be apparent, the bot 108 may have a more complex hardware architecture than is immediately evident in FIG 2A. For example, as indicated, the bot 108 may have a distributed architecture, whereby different parts of the code 14 are executed on different ones of a set of interconnected computing devices, e.g. of a cloud computing platform.

[0026] In accordance with the present invention, the bot 108 may be configured to provide special functionality, as outlined in more detail below.

[0027] The bot or primary bot or autonomous software agent (ASA) 108 is implemented on a server comprising a server device or a set of multiple inter-connected server devices which cooperate to provide desired functionality. For example, the bot 108 may be based on a cloud-based computer system, which uses hardware virtualization to provide a flexible, scalable execution environment, to which code modules can be uploaded for execution.

[0028] The bot 108 may be an intelligent software agent, the operation of which will be described in due course. The bot 108 is an artificial intelligence software agent configured so that, within the communication system 100, it appears substantially as if it were if another member of the communication system.

[0029] The bot 108 may be connected to a back end database 120, which may be a database or a service.

[0030] The bot 108 is inherently accessible to the user client 107 as one of the user client's contacts, allowing the user terminal to access the bot 108 directly via a chat service.

Bot provisioning

[0031] The bot is also connected to a bot provisioning service 110. The bot provisioning service 110 may also be connected to the network 102. A first bot 108, a so called 'listening bot', is accessed directly by a user at the user terminal. In some embodiments, the listening bot can select other bots from the bot provisioning service. A selected bot may be connected directly to a user terminal or to the listening bot. The bot provisioning service may take part in establishing such connections.

[0032] Once the user terminal 106 is connected to the bot 108, the user 104 can (among other things):

[0033] receive or instigate calls from/to, and/or IM sessions with, the bot 108 using their communication client 107 just as they can receive or instigate calls from/to, and/or IM sessions with other users of the communication system 100;

[0034] add the bot 108 as one of their contacts within the communication system 100. In this case, the communication system 100 may be configured such that any such request is accepted automatically;

[0035] see the bot's presence status. This may for example be "online" all or most of the time, except in exceptional circumstances (such as system failure).

[0036] This allows users of the communication system 100 to communicate with the bot 108 by exploiting the existing, underlying architecture of the communication system 100. No, or minimal, changes to the existing architecture are needed to implement this communication. The bot thus appears in this respect as another user 'visible' within the communication system, just as users are 'visible' to each other by virtue of the database 115, and presence and contact management mechanisms.

[0037] The bot 108 not only appears as another user within the architecture of the communication system 100, it is also programmed to simulate certain human behaviours. In particular, the bot 108 is able to interpret the speech in a user's call audio or in an instant message and respond to it in an intelligent manner. The bot 108 may formulate its responses as synthetic speech that is transmitted back to the user as call audio and played out to them in audible form by their client 107 just as a real user's call audio would be. The bot 108 may also generate synthetic video, in the form of an "avatar", which simulates human visual

actions to accompany the synthetic speech. The bot 108 may also formulate its responses as text that is transmitted back to the user as an instant message. These are transmitted and displayed as call video or instant messages at the user device 106, in the same way that a real user's video would be.

[0038] The bot provisioning service may comprise a database storing the details of the plurality of secondary bots 121, 122, 123, 124. The details stored in the database of the bot provisioning service may include the contact data of each secondary bot, metadata, or category information regarding each secondary bot. In certain embodiments, the category information may include the function of the bot and/or the location which the bot is capable of servicing, such as information regarding a bot's function, a bot's location, and/or or a bot's privacy statements and availability. The bot provisioning service 110 may also allow users to upload bots that they have created for use within the communication system 100.

[0039] In some embodiments the bot provisioning service may be provided by the network node that provides the bot 108 or by another network node in a communication network in which the network nodes are interconnected.

[0040] The bot provisioning service 110 may also be connected to a plurality of other bots, secondary bots, or servicing autonomous software agents (SASA). Each of these secondary bots is provided at a respective network node. In some embodiments, the secondary bots may be provided at a common network node. Only four additional bots 121, 122, 123, 124 are depicted in FIG 1 but as will be readily appreciated there may be many more bots connected to the bot provisioning service 110 of the communication system.

[0041] The user account database 115 may also store metadata relating to each bot, the metadata matching each bot to function, location information, or logical setup.

[0042] By way of example, a bot which has the capability to calculate a transport route in London may be matched to metadata tags such as maps, transport, London, or route calculation.

[0043] The primary bot 108 has the capability to retrieve through the bot provisioning service 110, the contact details of one or more secondary bots 121-124, and communicate those details to the user terminal 106; thus allowing the user terminal 106 to connect to one or more of the plurality of secondary bots 121-124.

[0044] In certain embodiments, the first bot may be configured to return contact data of a secondary bot for presentation on the display of the user terminal and, responsive to the user selecting a contact identifier of the contact data, the user terminal sets up

communication event with SASA addressed by contact data. The contact data may, in some embodiments, be in the form of a swiftcard.

Listening bot

[0045] FIG 3 shows a block diagram of a communication system 300 according to a second embodiment of the present invention. The communication system 300 comprises a communications network 302, to which is connected a first user device 306, a second user device 306', a user account database 315, a bot 308 (remote from the user devices 306 and 306'), a bot back end 320, and a bot provisioning service 310 (agent provisioning service). The network 302 is a packet-based network, such as the Internet.

[0046] The user devices 306 and 306' are available to a first and a second user 304 and 304'. The user devices 306 and 306' are shown to be executing a respective version of a communication client 307 and 307'.

[0047] Only two users 304 and 304' of the communication system 300 are shown in FIG 3, but as will be readily appreciated there may be many more users of the communication system 300, each of whom operates their own device(s) and client(s) to enable them to communicate with other users via the communication network 302. For example, group communication events, such as group calls (e.g. video conferences), may be conducted between three or more users of the communication system 300.

[0048] It can be appreciated that the user terminals of FIG 3 are analogous to the user terminals described with reference to FIG 2.

[0049] It can also be appreciated that the architecture of the communication system 300 of FIG 3 is analogous to the architecture of the communication system 100 depicted in FIG 1 and previously described.

[0050] The bot provisioning service 310 may enable the bot 308 to access a communication between the first user 304 and the second user 304'. In some embodiments the bot 308 may also obtain access to the historic communication between the users.

[0051] The bot 308, analogous to the bot 108 described with reference to FIG 1, can generally directly access a user client.

[0052] In embodiments there are one or two conditions for a bot to have access to a communication event (a session such as an IM conversation or VoIP call): authentication and/or permission. Authentication refers to verifying the identity of the bot to check it is not a malicious party masquerading as a legitimate provider. Permission on the other hand refers to whether a user (or the users) have chosen to allow bot to access their communication event.

[0053] In embodiments a user participating in a communication event may be provided with the option to grant the bot with permission to access to the event (for any of one or more of the purposes disclosed herein) by setting a permission setting in a user profile of the user, which may be stored locally on the user's user terminal or more preferably on a server, e.g. in the account database. In alternative or additional embodiments, the user may be provided with the option to grant the bot with the permission to access the communication event by means of a user-actionable prompt output through the UI (e.g. an on-screen prompt such as a op-up window or text-box) which the bot may provide in order to ask the user for the permission in question. In yet another alternative or additional embodiment, the permission may be implicitly granted when the user invites the bot, as a contact from the user's contact list, to become a participant in the communication session. The authentication then provides confidence that the bot is indeed legitimately the bot the user intends to grant permission.

[0054] In a situation where the bot 308 accesses a communication event between two or more users, this connection may typically happen in one of two different ways, as follows.

a) The bot 308 may have automatic access to all the user clients involved in a communication event by virtue of having access to the user client of a single user. In this scenario, the contact between the bot and the users may be direct or via the bot provisioning service, as described with reference to FIG 1.

b) The bot 308 may need to be authenticated by the other users of the communication event and in turn the other users of the communication event need to be authenticated by the bot 308 in order. In this scenario, at least the initial authentication process is mediated by the bot provisioning service. Once the authentication process has been completed, the contact between the bot and the users may be direct or via the bot provisioning service, as described with reference to FIG 1.

[0055] FIG 3a shows a flow diagram illustrating a method by which the system illustrated in FIG 3 may autonomously action a user intent.

[0056] Initially, contact identifiers are displayed, in step 350, on a display of a user terminal associated with the user. Each contact identifier may be selectable to initiate a communication event with a node addressed by the contact identifier. By way of example, the contact identifiers may be displayed as a contact list on a user terminal screen.

[0057] The user may, in step 352, select a contact identifier. In the context of the previous example this might be implemented, by way of example, by selecting a contact form the contact list.

[0058] Upon doing so, a communication event is established, in step 354, between the user terminal and the network node associated with the selected contact identifier. The communication event may, by way of example, be a communication session such as an audio or video call or an IM session.

[0059] Once the communication event has been established, the user terminal may communicate, in step 356, with the user human user associated with the network node associated with the selected contact identifier as well as with another node associated with an autonomous software agent or bot 308. The messages in the communication between the two human users may be available to the bot 308. Therefore, an intent conveyed in a dialogue between the two human users may also be available to the bot 308. In the case where the communication event is an IM session, the text of the messages may be made available to the bot.

[0060] The bot 308 may then action the intent conveyed by the user terminal. The response to the intent received from the bot may then be received and presented, in step 358, to the user.

[0061] In certain embodiments, the communication event from which intent is conveyed to the bot may be a current dialogue in which the user is engaged.

[0062] By way of example, the above method may be applied in a situation where user 304 and user 304' are exchanging IM messages discussing going for dinner near Dublin on Friday, 24 June 2016. The bot may have access to this message and search for restaurants in Dublin with availability on the selected date and present those results to the users in real time, as the two users are still discussing where to go.

[0063] In other embodiments, the communication event from which intent is conveyed may be a historic event.

[0064] In certain embodiments, the user terminal of the user may also have the capability to access context data of the user and make this context data available to the bot. Such context data may be, by way of example, credit card details or location data.

[0065] In certain embodiments, the bot may first require permission to access the user's context data. For example, the bot may request a permission to access the user's context data. This permission request may be displayed at the user terminal whereby a user can engage with the request. Alternatively, permission may be granted via a setting available

at the user end which may grant permission to the bot to access all or certain groups of the user's context data. The setting could be maintained at the user terminal or on a server, e.g. as part of a user profile. In another embodiment, the permission may be granted implicitly by the user having invited the bot, as a contact, to be one of the participants in the conversation.

[0066] In certain embodiments a user may convey an authentication token to the bot upon signing on with the communication client. In other embodiments, the user may receive a request for authentication of the user by the ASA and supply this authentication token in response.

[0067] In the situation described above where the bot 308 accesses a communication event between two or more users, access to the users' context data may typically be granted in one of two different ways, as follows.

a) The bot 308 may have automatic permission to access the context data of all the user clients involved in a communication event by virtue of having access to the user client of a single user. In this scenario, the contact between the bot and the users may be direct or via the bot provisioning service, as described with reference to FIG 1.

b) The bot 308 may need to be granted permission by each of the other users of the communication event individually. In this scenario, at least the initial permission process is mediated by the bot provisioning service. Once the permission granting process has been completed, the bot 308 has access to the context data of all the users involved in the communication event.

Authentication

[0068] A first authentication process is shown in FIG 4a. FIG 4a shows an authentication flow when sign in to the communication service (e.g. VoIP and/or IM service) is possible.

[0069] First, the user signs in to the communication service by entering their authentication details.

[0070] Then the user receives from the communication service an authentication token - such as, by way of example, a Skype sign in. That is, the authentication token is provided to the client on a user terminal from the Skype backend when a user signs into Skype.

[0071] Once the authentication token has been received by the user, the user is able to send a chat message to the agent. This chat message is sent with the authentication token.

[0072] The message and authentication token are then forwarded onto an agent endpoint. Once this has been completed, the agent is enabled to forward the user's message to the agent back end. In the present embodiment a Representational State Transfer (REST)-like interface is used to pass tokens into a dynamic link library, but it will be understood that a bot doesn't necessarily have to be coded using windows technologies. A Hypertext Preprocessor (PHP) script, for example, could be used to handle this scenario and implement an agent, in which case the agent endpoint could be embodied in a different way.

[0073] In this embodiment, the authentication token can be used in the headers of messages in a communication event such as an IM session between the user terminal and the bot. Note that the authentication tokens can be first or third party tokens – i.e. the current user terminal associated with the bot, or the remote third party with which the first party is engaged in a conversation.

[0074] A second, alternative authentication process 500 is depicted in FIG 4.

[0075] FIG 4 illustrates an authentication technique when token exchange is not handled at the communication service sign in (e.g. where the sign in to the communication service is not possible) and the user signs in using an authentication service such as, by way of example, mail submission agent (MSA).

[0076] A key issue with the use of bots such as bot 108 and bot 308, described with reference to FIGs 1 and 3, is security. The systems 100 and 300 of FIGs 1 and 3 ensure, via the authentication methods described herein with reference to FIGs 5 and 6 that bots 108, 308 must first be authenticated by the user before it acquires access to the user's data and personal information as well as to the back end services 120, 320.

[0077] The back end service (or collection of services) require user authentication and authorization prior to access for both first and third parties.

[0078] According to this authentication process 500, the user first signs in with the communication systems 100, 300, in step 502, using a mail submission agent (MSA).

[0079] Upon signing in with the communication system, step 502, the user also requests and receives, in step 504, an authentication token suitable for agent back end authentication. This authentication token can be used to allow back end access to a bot 108, 308 or agent. This token may, by way of example, be a Relying Party Suite, (RPS) authentication token.

[0080] The authentication token is then transmitted, in step 506, from the user to a secure end point accessible by the agent or bot. The authentication token is then stored, in

step 508, at the personal data platform 117, 317 of FIGs 1 and 3 which holds all personal data, and is accessible by the bot 108, 308.

[0081] These steps may be performed automatically upon a user authenticating, or signing in with a communication system 100, 300, thus automatically authenticating the agent or bot to access.

[0082] After the above-described procedure has been completed, when the user transmits a message to the agent in step 510, the agent or bot simply retrieves the user's authentication token from the personal data platform where it is *already* stored. The agent or bot is thus instantly authenticated and allowed access to the back end, and the user's message and authentication token may be forwarded to the agent backend. A sign in flow could be modified to include retrieving additional tokens or scopes.

[0083] A third authentication method is described with reference to FIG 5 which is an alternative to authentication method of FIG 4.

[0084] FIG 5 shows token exchange when sign in to the communication service (e.g. the VoIP and/or IM service) is not possible and the user signs in via a proxy, referred to herein as Trouter or TCP/IP Router. The name Trouter is used herein to denote a proxy that clients connect to on start-up. Once connected, Trouter maintains this socket to allow the communication service's backend to selectively push or receive signals to/from that client, effectively tunnelling through any firewalls or routers that a given client may be using. It can be used to notify a running communication client instance that there is an incoming call, but it can also function as a general purpose communications channel.

[0085] According to the authentication method described below, a bot 108, 308 or agent does not acquire authentication immediately upon the user's login, but only if the user "summons" the bot 108, 308 or agent.

[0086] The user first signs in with the communication system 100, 300 using a mail submission agent (MSA) via the Trouter proxy through any firewalls or routers the client may be using. The Trouter proxy then responds by transmitting to the user a Trouter URL.

[0087] The user then transmits, in step 504, the Trouter URL to the agent or bot. The agent or bot stores, in step 506, the user's Trouter URL at the personal data platform 117, 317 of FIGs 1 and 3 which may further comprise a Trouter URL store, and is accessible by the bot 108, 308.

[0088] The above-described sequence of events takes place immediately upon a user signing in with the communication network. After this sequence of events has been

completed, when the user wishes to communicate with the agent or bot, the following sequence takes place.

[0089] The user transmits, in step 508, a message directed to the agent.

[0090] This message may be forwarded via instant messaging service to the agent or bot. Once the message is received at the agent or bot, it is examined, in step 510, whether the agent already has an authentication token for the user. This may be the case if, for example, the user has already used the agent or bot earlier in the communication session.

[0091] If the agent already has an authentication token for the user, then the agent or bot may connect to the back end without any further authentication steps (step 512). The user's message is forwarded to the backend and the user request is actioned accordingly.

[0092] If the agent does not already have the user's authentication token, then the agent retrieves, in step 514, the trout URL associated with the user that has been stored at an agent database comprised in the personal data platform 117, 317.

[0093] Once the user's Trouter URL has been retrieved from the agent database, the agent transmits, in step 516, an authentication token request to the user. The authentication token request informs the user end that an authentication token must be transmitted to the agent.

[0094] It is then examined, at the user end, in step 518, this is a check to determine whether the agent is actually allowed to possess ticketing information.

[0095] The purpose of this is two-fold.

[0096] First, it provides an (optional) opportunity for the communication client to present a consent dialog to the user on his display ("WidgetBot is requesting to perform an operation on your behalf, allow? Y/N"). No consent = no ticket = no backend access. The second is a security measure to guard against malicious agents. In order to be addressable inside the communication service, a bot is provisioned in a portal. At this point, a bot's capabilities are registered (which will include whether they are allowed 1st or 3rd party MSA authentication tokens). Armed with this information, a client can determine if a given agent should even be making this kind of request and whether to expose sensitive ticketing data to it or not.

[0097] If the agent 108, 308 is not already authenticated with the user, then nothing happens.

[0098] If the agent is already authenticated with the user, then the user retrieves its authentication token from its authentication service and transmits its authentication token to the agent. Trouter is one mechanism by which an agent could request the ticket from a

communication client at run-time. However, it does not have to go via the Trouter proxy. The resulting ticket response could go through Trouter, or CHAT service or any intermediary service that then forward the appropriate data to the agent.

[0099] The agent then stores the user's authentication token in an agent database that comprises an authentication token store.

[0100] The agent then acquires authentication and the user's message can then be transmitted to the agent back end and be actioned.

[0101] If secondary bots 121-124 require access to the back end, as was described with reference to FIG 1, then it is the first bot which manages the authentication of these secondary bots 121-124 using an authentication method analogous to the method described above.

[0102] The agent first accesses the personal data platform 117, 317, to examine whether the secondary bots 121-124 already have an authentication token associated with the specific user. If the secondary bots 121-124 do not already have an authentication token associated with the user, the bot 108, 308, retrieves the trout URL associated with the user that has been stored at the agent database comprised in the personal data platform 117, 317.

[0103] Once the user's Trouter URL has been retrieved from the agent database, the agent transmits an authentication token request to the user requesting an authentication token for the additional bots 121-124. The authentication token request informs the user end that an authentication token for the secondary bots 121-124 must be transmitted to the agent or bot 108 or 308.

[0104] The user retrieves the authentication token for the secondary bots 121-124 from its authentication service and transmits the authentication token to the agent or bot 108, 308. Trouter is one mechanism by which an agent could request the ticket from a communication client at run-time. However, it does not have to go via the Trouter proxy. The resulting ticket response could go through Trouter, or CHAT service or any intermediary service that then forward the appropriate data to the agent. The agent or bot 108 then stores the user's authentication token associated with the secondary bots 121-124 in the agent database agent database comprised in the personal data platform 117, 317.

[0105] The secondary bots 121-124 then acquire access authentication for the back end.

Selecting bots

[0106] Turning back to the system of FIG 1, below is described a method with which the primary bot 108 may enable other, secondary bots 121-124 to establish a connection

with the user 104. The different bots 108 and 121-124, may be distinct from one another in that they are provided by different parties (e.g. different companies); and/or they may be distinct from one another in that they appear as different contacts in the user's contact list; and/or as different participants in a given communication event (given session, e.g. given IM chat conversation or VoIP call); and/or the different bots may be distinct from one another in that they require separate permission and/or separate authentication to be involved in a given communication event (a given session such as a given IM chat conversation or VoIP call), e.g. to be a participant or to listen in on it, and/or to share context data (see later).

[0107] Initially the bot receives, in step 600, an intent from the user 104. As mentioned, the intent could be conveyed directly to the bot in a chat message, or derived from a conversation with a third party.

[0108] Subsequently the bot transmits, in step 602, the received intent to the bot provisioning service 110.

[0109] The bot provisioning service 110 is configured to match the received intent to a category. The bot provisioning service 110 queries the account database 115 in order to establish whether any of the secondary bots 121-124 match the intent content. If the query establishes that one or more secondary bots match the intent, the bot provisioning service 110 retrieves the contact data of the matching secondary bots 121-124 and forwards them to the bot 108 (step 604).

[0110] Once the relevant contact data has been received by the bot 108, it is transmitted, in step 606, to the user terminal 106 for presentation on the user display.

[0111] By way of example, if the content of the intent relates to booking a hotel, the bot provisioning service may retrieve bots associated with booking services or bots associated with a specific hotel. The bot 108 may then forward the contact data of these matching bots, such as by means of a swiftcard.

[0112] In some embodiments, the bot 108 may further be configured to extract context data from the user terminal and supply the context data to the selected bot to implement the action. In the context of the hotel booking example this may involve, once the user has selected the bot of preference, retrieving the user's credit card details in order for the booking by the secondary bot to take place without the need for the user to enter any additional information.

[0113] In some embodiments, the bot 108 may further be configured to obtain information from the secondary bot and provide said information to the user terminal. In the

context of the hotel booking example this may involve obtaining price quotes from the selected bots and presenting the quotes directly to the user.

[0114] In some embodiments the bot provisioning service may be located at the first network node. Alternatively, the bot provisioning service may be located at another network node. In certain embodiments, the bot may be configured to send an intent to the bot provisioning service.

[0115] In some embodiments, the bot 108 may be configured to capture the intent conveyed by a conversation in which the user is a participant.

[0116] In certain embodiments, the bot 108 may be authorized to conduct the selection of the secondary bot without user input.

[0117] In certain embodiments, to implement an action corresponding to the intent conveyed, the first network node accesses the bot provisioning service which enables access to a plurality of servicing autonomous software agents, each capable of implementing an action, and the first network node responds to the received intent by selecting one of the servicing autonomous agents to implement an action corresponding to the intent. In certain embodiments the action may be implemented by transmitting a message to the user terminal in a communication event established between the user terminal and the first network node by the communication client based on selection of the contact identifier.

Context data supplied to a servicing entity

[0118] Turning to FIG 7, the method by which the primary bot 108 may receive and supply the aforementioned intent and context data will be described in more detail.

[0119] A communication event is initially established by a user terminal 106, 306 over the communications network 100, 300 (step 700). The communication event may, by way of example, be a communication session such as an IM conversation of a video or audio call between the user terminal and the secondary bot. The communication event may be a group communication event in which another user terminal is connected. By way of example, the communication event may be a chat between three human users.

[0120] The primary bot 108 or 308 then receives, in step 702, in a message conveyed in the communication event, a user intent and user context data. The primary bot 108 or 308 may be an AI software agent. In certain embodiments, the intent is derived by the bot from an exchange of messages in a group communication event between the user terminals. By way of example, the bot may use language recognition techniques, such as natural language processing, to process the messages or audio of the dialogue between the parties of the communication event and derive the intent. Similar techniques may also be used to derive

context data. In the case of context data, this might also be harvested from a user's profile information such as gender and status (e.g. connected, busy, offline, etc.). It should be noted that the primary bot 108 or 308 may not necessarily be a participant in the communication event as such- by way of example the primary bot may have access to the real time data or transcripts of a communication event or session that is established between two users.

[0121] The primary bot 108 or 308 then selects, in step 704, a secondary bot 121-124 to perform an action corresponding to the intent. The secondary bot 121-124 may be provided at a separate network node or at a common network node. In certain embodiments, the secondary bot 121-124, or other servicing entity, is selected based on matching a category of intent obtained from the user intent to a category of servicing entity.

[0122] The bot 108, 308 then generates, in step 706, a message containing the context data provided by the user terminal 106, 306.

[0123] Subsequently, the primary bot 108 or 308 transmits, in step 708, the message containing the context data to the secondary bot 121-124.

[0124] In certain embodiments, the secondary bot 121-124 may be an autonomous software agent selected by the computer program product to deliver the action autonomously to the user terminal via a separate communication event. The secondary bot 121-124 may in certain cases be replaced by another servicing entity such as a website.

[0125] In certain embodiments, a communication event may be established with the servicing entity at the node and the message containing the context data may be transmitted within the communication event. In certain embodiments, establishing a communication event with the servicing entity may involve receiving and responding to a request received from the user terminal.

[0126] In certain embodiments, the permissions associated with the user of the user terminal 106 or 306 and the secondary bot 121-124 may be determined.

[0127] In certain embodiments, users obtain permission to transmit context data by accessing the personal data platform 117, 317 of FIGs 1 and 3 which holds the permissions for users.

[0128] In certain embodiments, the secondary bot 121-124 may deliver the action to the user terminal by establishing a separate communication event between the secondary bot 121-124 and the user terminal 106 or 306.

[0129] In certain embodiments, the primary bot 108, 308 may autonomously connect with the secondary bot 121-124 and transmit the message containing the context data to the secondary bot 121-124.

[0130] In certain embodiments, upon establishing a communication event between the user terminal 106 or 306 and the secondary bot 121-124, the user terminal 106 or 306 may also receive a request for permission for the servicing entity to receive context data from the user.

[0131] It should be noted that different permissions may be associated with different types of context data and personal information, and different permissions may apply to different secondary bots and different users.

[0132] To illustrate what is meant by different permissions may apply to different types of context data, an example would be that a user may give permission to the bot 108 or 308 or secondary bot 121-124 to have access to his location information but may not to his credit card information. In a different example where different permissions apply to different bots, the primary bot 108 or 308 may have access to bot location and credit card information, whereas the secondary bot 121-124 may only have access to location information.

[0133] To illustrate the "different permissions for different users" concept, an example would be a conversation between users A and B in which the two users discuss booking a hotel and the primary bots 108 or 308 actions the intent by booking the hotel using user A's credit card details and returns a booking confirmation to the conversation between the two users, the booking confirmation containing the details of the credit card that was used. It may be the case that user A is happy to share the credit card details with user B. In this case, the credit card details will be redacted from the image the booking confirmation shown in the conversation.

[0134] In certain embodiments, the bot may determine that additional information is required to deliver the action of step 704. In this case, the bot may transmit a request to be surfaced at the user terminal to request the additional information.

[0135] In certain embodiments, when selecting a bot or other servicing entity for a specific intent, the primary bot 108 or 308 may do this using the user's current context, such as the user's current location, the user's past history and the user's preferences. By way of example, when the intent is to order a taxi and the user's location is San Francisco, the bot may take this location into account and e.g. consider that Uber is the common service for reserving a taxi in SF, that the user's past history indicates that the user has reserved a taxi using Uber before and also take into account the user's preferences, e.g. the user may already have the Uber app installed or may have indicated that he prefers using Uber.

[0136] When selecting a bot, the primary bot 108 or 308 may pass the current context to the secondary bot 121-124 so the user is not required to communicate it again. E.g. if the user has indicated to the primary bot 108 or 308 "I need a taxi to pier 39", the bot already knows this and it does not need to request information from the user regarding this matter again.

[0137] In certain embodiments, after the user completes a transaction with the secondary bot 121-124, information about the transaction may be passed back to the primary bot 108 or 308 for future reference and for affecting the user preferences.

[0138] FIGs 8A-8D illustrate an example of an interaction between users (i.e. humans) and bots which might take place e.g. over the course of a day. In this regard, FIGs 8A-8D illustrate successive instances of message exchange (i.e. FIG 8B takes place after FIG 8A, FIG 8C after 8B, and FIG 8D after 8C). In FIGs 8A-8D, User 802 has a personal assistant Bot 803. A further user, User 801 is shown who may or may not have a personal assistant bot of his own. A further bot, Bot 804 is also shown.

[0139] In the example interaction shown in FIG 8A, User 802 wishes to order pizza. To accomplish this, User 802 sends a message 805 to Bot 803 telling the bot to order pizza. Bot 803 can then request further details if required (such as what type of pizza, and where from). Bot 803 has access to historical data about User 802 which allows the bot to send message 807 asking if User 802 wants "the usual" (i.e. a specific pizza order the user has ordered at least once before). User 802 confirms with message 809. At this point, Bot 803 has been tasked with a pizza order and can contact another bot (Bot 804) as appropriate. For example, FIG 8A relates to a pizza order, so Bot 804 will be a pizza-related bot (e.g. a business-specific bot associated with the particular pizza shop to which the order is being sent, or a goods/services-specific bot such as a food ordering bot). In FIG 8A, the pizza order 811 is sent to Bot 804 and thus the order is placed.

[0140] Later, User 802 receives a message 813 from User 801, as shown in FIG 8B. This message might have some context data 814 as shown in FIG 8B where the message 813 be an invitation to an event and the context data 804 relates to the event (e.g. title, time, place etc.). Although not shown in FIG 8B, it is appreciated that Bot 803 can "listen in" to this message as received by User 802 and therefore be aware of the context data (e.g. store it locally for use as described later). Bot 803 may save the context data as particularly relevant if the user responds in the affirmative as shown by message 815 in FIG 8B.

[0141] In FIG 8C, the pizza ordered in FIG 8A is now ready to be delivered but the pizza shop does not have the current location of User 802. In this case, Bot 804

communicates with Bot 803 by sending a location request 817 to Bot 803. Bot 803 may then inform User 802 by sending a message 819 to the user indicating that the pizza bot, Bot 804 wants his location. User 802 then confirms with message 821 that Bot 803 has permission to share his location and thus Bot 803 can send the location in message 823 to Bot 804. The pizza delivery can then be completed as the pizza shop (via its bot) now has the location of User 802.

[0142] Later, as shown in FIG 8D, User 802 wishes to add User 801's event to his calendar. To do this, User 802 simply sends a message 825 to Bot 803 instructing it to add the event to his calendar. Bot 803 can recognise the event which the user is referencing due to "listening in" on previous conversations (i.e. the conversation shown in FIG 8B) and storing relevant data, e.g. temporarily caching context data 814. In the example of FIG 8D, Bot 803 has recognised the event by name ("Codess") and retrieved the context data 814 from memory. Bot 803 can thus check with User 802, using message 827, that it has the correct event before adding it to the calendar. Once the user confirms with message 829, the event is added to the calendar. The context data in this example includes information relating to the date/time of the event but it is appreciated that the context data might include any relevant context data (e.g. event location). If the required context data for a particular action is missing (e.g. if the user asked the bot to add the event to his calendar but the context data did not include date/time) then the bot can send additional messages (not shown in FIG 8D) in order to request this data from the user.

[0143] In the context of the embodiments in which no communication event is established between the user terminal 106 or 306 and the secondary bot 121-124, a permission request may be sent to the user terminal 106 or 306 for the user 104 or 304 to grant permission for the secondary bot 121-124 to receive context data.

[0144] FIGs 12A-I show a similar user-bot interaction session to that described in relation to FIGs 8A-D. Specifically, FIGs 12A-I illustrate the messages as appearing on the user interface display of the user terminal. In FIGs 12A-I, the user's personal assistant bot ("Cortana") relays information between the user and "Cups and Cakes bot" in order to facilitate the user ordering a cupcake. Cortana also relays information and permissions relating to delivery tracking. As shown in FIGs 12A-I, the user's interaction with the bot is substantially conversational in the sense that the user is able to message the bot in the same way, and through the same messaging client, as they would message a human user.

Bot Portal

[0145] It is appreciated from the above description that many bots may serve a variety of general and specific functions. The present invention further recognizes that users (developers) may be able to create their own bots. In this case, other users (including non-developers) may wish to share bots with each other. The present invention therefore provides a bot provisioning service or "bot portal" that allows a developer to either upload or identify the location of a bot. The portal also allows the user to specify the capabilities of the bot.

[0146] The portal provides a bot provisioning service in the form of a computer system (also called a "back end") comprising one or more computer devices, the computer system providing the provisioning service of autonomous software agents (bots), the computer device comprises a user interface generating component, a storage interface component, and an access component.

[0147] The user interface generating component provides the portal to a human user via a display, i.e. the user interface generating component is a controller operable to control a display in order to display the portal to a human user. That display may be the display on a user's personal device or may be local to the computer system back end itself. In either case, the display might be a display of a developer's device (a bot developer) via which the developer can access the portal in order to upload, access and/or edit his bots. The portal has entry fields for receiving agent data from the human user (e.g. developer)

[0148] The storage interface component accesses computer storage that stores autonomous software agents. That is, the bots themselves can be stored anywhere (not necessarily at the computer system itself, though that is possible). The bots may also be stored in a distributed fashion (see the discussion of calling/audio webhooks later). Hence, the storage interface component allows the computer system to access the bots from the computer storage.

[0149] The access component holds an association between the agent data and a network address of an agent., e.g. a list of bot names and the network addresses at which they are each stored. Note that each bot may have more than one network address defining a location of the computer storage in a computer network at which the agent is stored because, as mentioned above, the bots may be stored in a distributed manner. When an entity (such as a developer or other user) selects an agent based on the agent data, the access component enables automated access to the agent based on the network address.

[0150] The portal takes bot metadata (such as name, endpoint, etc.) and makes it available in a lookup that a user may optionally consume (via Skype) to add the bot as a contact into their address book. That is, the portal provides a registry of all the registered bots. Users can then access the portal and find bots to choose to add as contacts on their messaging client. Note that there may be cases in which a bot may automatically be added to a user's address book, such as:

[0151] Cortana bot (this was a business decision, not a technical one)

[0152] Concierge bot (new users only, this is a bot that provides helpful hints and tips)

[0153] Developers who own or are engineering a given bot.

[0154] Regarding the developers (above), a developer of a "WidgetBot", might automatically get WidgetBot propagated as a contact to his address book because he is a developer and has made his IM ID known to the portal. WidgetBot would not be automatically propagated to someone else's address book, but once the developer has published WidgetBot, other users would be able to add it as a contact using an IM client like any other bot.

[0155] FIG 9 illustrates an example of a portal screen which is displayed to a user 104 on the display 205 their respective user terminal 106, though it is not excluded that the user 104 may be able to access the portal via any other computing device.

[0156] Specifically, FIG 9 shows an example form by which a user may upload a bot to the portal. The form includes fields for a bot name 901, a description 902, a profile picture 903, an ID 904, and a section relating to the bot's capabilities 905. The capabilities section allows the user to specify the messaging capabilities of the bot, shown are a one-to-one messaging capability 907 (i.e. can the bot function in a one bot to one user messaging scenario), a group messaging capability 908 (i.e. can the bot function in a one bot to multiple user group messaging scenario), and an audio calling capability 910. The audio calling capability 910 is shown as a one-to-one capability, but it is not excluded than a group audio calling capability may be provided.

[0157] Fields for entering a messaging webhook 909 and a calling webhook 912 are also shown. The webhooks are the network addresses to which user messages and calls, respectively, will be routed by the IM client. In this sense, they are the equivalents of a normal (human) user's contact addresses. Separate calling and messaging webhooks may be used when the bot is stored in a distributed fashion. In which case audio calls may be routed differently than textual messages.

[0158] The information relating to the bot entered by the user may be considered metadata which may include both logical and functional data. Logical data is information relating to the bot category (e.g. hotel, food, flights, etc.) of what service the bot provides to the user. Functional data is data relating to the messaging client specifically (e.g. calling capabilities, network addresses, etc.). Other information may also be included such as bot name, a description, and/or a profile picture.

[0159] Once a user has entered the bot metadata (e.g. using a form such as the one shown in FIG 9), the bot may be published to the portal. This allows other users to search for and find the bot. An example of how a bot might appear in the portal is shown in FIG 10.

[0160] In FIG 10, the bot is shown as a list of bot metadata: name 1001, description 1002, status 1003, Bot ID 1004, add link 1005, capabilities 1006, application ID 1007, messaging webhook 1008, calling webhook 1009, and visibility 1010. Any or all of the metadata may be searchable by users in order to find a bot. Note however that the visibility 1010 may be turned off in order to hide a bot from search results.

[0161] The description 1002 should include details about logical capabilities of the bot, i.e. what the bot actually does, preferably in human-readable text. For example, specifying that the bot is a "hotel booking bot" or a "pizza ordering bot". The logical capabilities of the bot may be categorised to ease searching, e.g. "hotel" or "food". The logical capabilities may also be more specific such as specifying the particular company or brand which the bot represents.

[0162] The status 1003 allows users to see the status of the bot. For example, there may be a review period for new bots in which the admin of the bot provisioning service review bots before allowing them to be accessed from the portal. In this case a bot may be listed on the portal as "in review", "approved", "rejected", etc. (though preferably, rejected bots would be removed from the portal entirely). Alternatively, a bot may be (perhaps temporarily) "blocked" in which case a user may see that the bot exists on the portal but may not be able to add it as a contact.

[0163] The bot ID 1004 uniquely identifies the bot on the portal. The ID may be provided by the admin of the bot provisioning service.

[0164] The add link 1005 specifies a link which allows a user to add the bot as an IM contact, either directly (e.g. a hyperlink) or indirectly (e.g. by specifying the bot's address which the user needs to add as a contact, in the same way that the user may add a human contact).

[0165] The capabilities 1006 of the bot include at least what type of messaging the bot provides, e.g. text messaging, group chat, audio/video calls etc.

[0166] The application ID 1007 uniquely identifies the application.

[0167] Messaging webhook 1008 and calling webhook 1009 are the network addresses to which text and voice/video messages sent by the user will be routed, respectively.

[0168] FIG 11 shows an example of a portal displaying a list of (multiple) bots created by a user. That is, a user who has created multiple bots (i.e. a developer) and submitted them to the portal.

[0169] In this regard, the portal may present the user with a list of his bots and a summary of their status (shown as name 1101 and status 1102, though other metadata might also be included). The portal allows the user to edit their bots, view their status (e.g. in review), manage privacy settings, and delete their bots.

[0170] It is appreciated, as outlined above, that bots may require permission to access certain information relating to a user (e.g. credit card details). Hence, it is understood that the bot metadata stored on the portal may include information pertaining to said permission (e.g. whether the bot requires access to a user's credit card details), and therefore that the user must provide this permission in order for the bot to function. The permission metadata may also include hardware permissions such as whether the bot requires access to a webcam or speakers or a user's terminal. The permission metadata may also include other permission data such as legal waivers.

[0171] Generally, any of the functions described herein can be implemented using software, firmware, hardware (e.g., fixed logic circuitry), or a combination of these implementations. The terms "module," "functionality," "component", and "logic" as used herein generally represent software, firmware, hardware, or a combination thereof. In the case of a software implementation, the module, functionality, or logic represents program code that performs specified tasks when executed on a processor (e.g. CPU or CPUs). The program code can be stored in one or more computer readable memory devices. The features of the techniques described below are platform-independent, meaning that the techniques may be implemented on a variety of commercial computing platforms having a variety of processors.

[0172] For example, the user terminals may also include an entity (e.g. software) that causes hardware of the user terminals to perform operations, e.g., processors functional blocks, and so on. For example, the user terminals may include a computer-readable

medium that may be configured to maintain instructions that cause the user terminals, and more particularly the operating system and associated hardware of the user terminals to perform operations. Thus, the instructions function to configure the operating system and associated hardware to perform the operations and in this way result in transformation of the operating system and associated hardware to perform functions. The instructions may be provided by the computer-readable medium to the user terminals through a variety of different configurations.

[0173] One such configuration of a computer-readable medium is signal bearing medium and thus is configured to transmit the instructions (e.g. as a carrier wave) to the computing device, such as via a network. The computer-readable medium may also be configured as a computer-readable storage medium and thus is not a signal bearing medium. Examples of a computer-readable storage medium include a random-access memory (RAM), read-only memory (ROM), an optical disc, flash memory, hard disk memory, and other memory devices that may us magnetic, optical, and other techniques to store instructions and other data.

[0174] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

Claims

1. A communication system comprising:

a user terminal having a processor configured to execute a communication client installed at the terminal, and a display wherein the communication client is configured to cause contact identifiers to be displayed on the display, each contact identifier being selectable to initiate a communication event with a network node addressed by the contact identifier;

a first network node providing a first autonomous software agent (ASA) and addressable by a first contact identifier displayed on the user terminal, the autonomous software agent configured to receive an intent conveyed by a user at the user terminal;

an agent provisioning service component accessible by the first network node and enabling access to a plurality of servicing autonomous software agents (SASA), each capable of implementing an action,

wherein the first network node is configured to respond to the received intent and to select one of the servicing autonomous software agents to implement an action corresponding to the user intent.

2. A system according to Claim 1 which comprises a plurality of servicing autonomous software agents each at a respective network node.

3. A system according to Claim 1 which comprises a plurality of servicing autonomous agents at a common node.

4. A system according to Claim 1 wherein the agent provisioning service comprises a database storing details of the plurality of autonomous software agents.

5. A system according to Claim 1 wherein the Provisioning service component (PSC) holds category information of SASAs and wherein the first ASA is configured to match intent to a category.

6. A system according to Claim 5 wherein the category information includes the function of an ASA and/ or the location which the ASA is capable of servicing.

7. A system according to Claim 1 wherein the first ASA is configured to return contact data of a SASA for presentation on the display of the user terminal ; responsive to the user selecting a contact identifier of the contact data the user terminal sets up communication event with SASA addressed by contact data.

8. A system according to Claim 1 wherein the first ASA is configured to access the selected ASA and obtain information responsive to the intent to the user terminal.

9. A system according to Claim 1 wherein the first ASA is configured to extract context data from the user terminal and supply the context data to the selected SASA to implement the action.

10. A system according to Claim 1 wherein the provisioning service component is located at the first network node.

11. A system according to Claim 1 wherein the provisioning component is located at another network node in a communication network in which the network nodes are interconnected.

12. A system according to Claim 11 wherein the first ASA is configured to send the intent to PSC at the other network node

13. A system according to Claim 1 wherein the first ASA is configured to capture a user intent conveyed by a conversation in which the user is a participant.

14. A method of implementing an action corresponding to an intent conveyed by a user engaging with a user interface (UI) at a user terminal, wherein a communication client installed at the user terminal has detected that a contact identifier displayed on a display at the user terminal has been selected, the method comprising:

at an autonomous software agent addressable by the contact identifier and located at a first network node, receiving the user intent in a message;

the first network node accessing an agent provisioning service component which enables access to a plurality of servicing autonomous software agents, each capable of implementing an action;

wherein the first network node responds to the received intent by selecting one of the servicing autonomous agents to implement an action corresponding to the intent .

15. A computer program product comprising computer code stored on computer readable media and executable by one or more processors in a communication network to implement a method of implementing an action corresponding to an intent conveyed by a user engaging with a user interface (UI) at a user terminal, the method comprising

detecting by a computer code of communication client installed at the user terminal that a contact identifier displayed on a display at the user terminal has been selected;

transmitting the user intent in a message to an autonomous software agent addressable by the contact identifier and located at a first network node;

by computer code at the first network node accessing an agent provisioning service component which enables access to a plurality of servicing autonomous software agents, each capable of implementing an action ,

wherein the first network node responds to the received intent by selecting one of the servicing autonomous agents to implement an action corresponding to the intent .

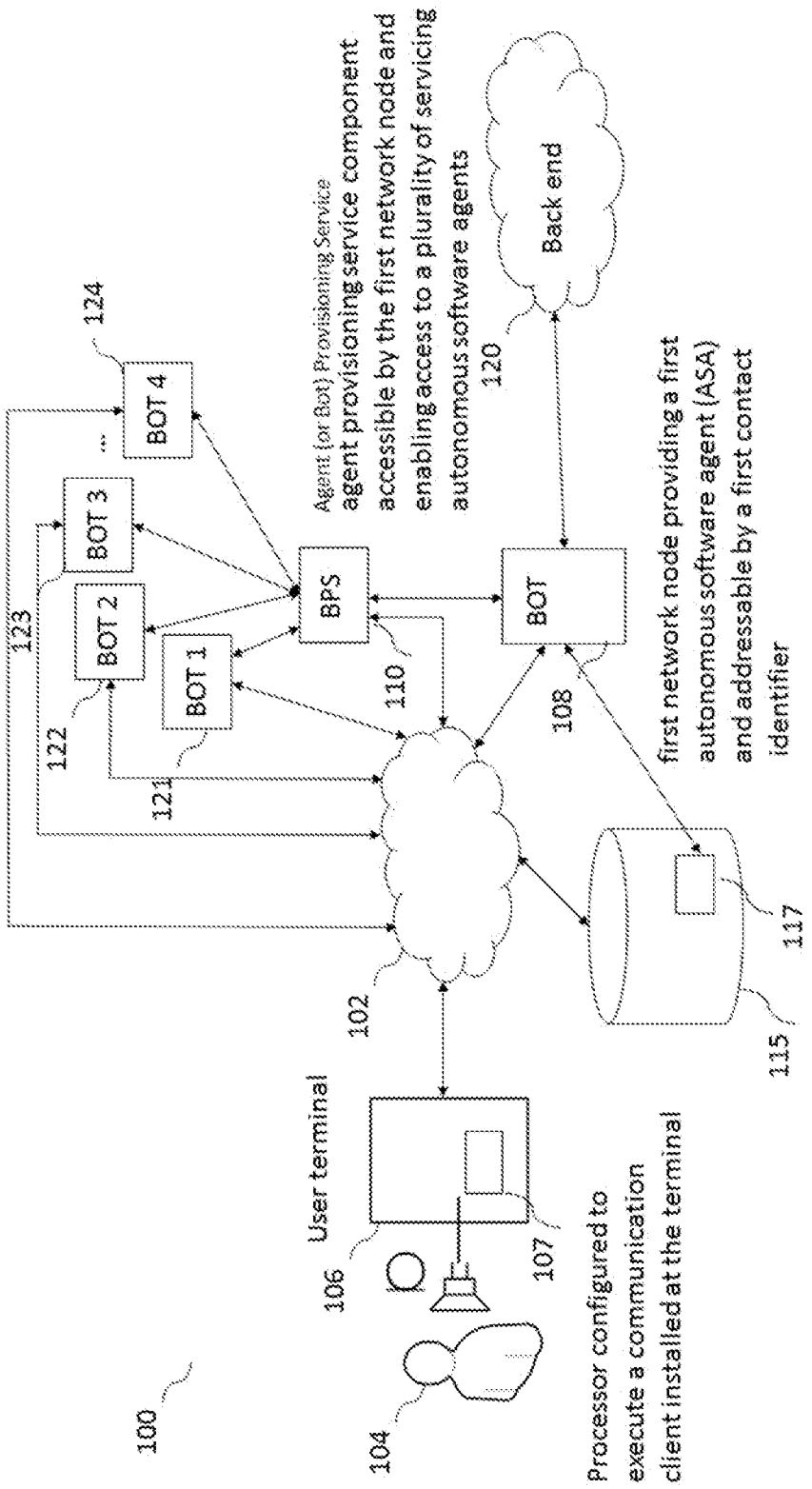
100
104
106
User terminal
107
Processor configured to execute a communication client installed at the terminal
102
115
117
first network node providing a first autonomous software agent (ASA) and addressable by a first contact identifier
108
BOT
110
BPS
Agent (or Bot) Provisioning Service
agent provisioning service component accessible by the first network node and enabling access to a plurality of servicing autonomous software agents
120
Back end
121
122
123
BOT 1
BOT 2
BOT 3
...
BOT 4
124

FIG. 1

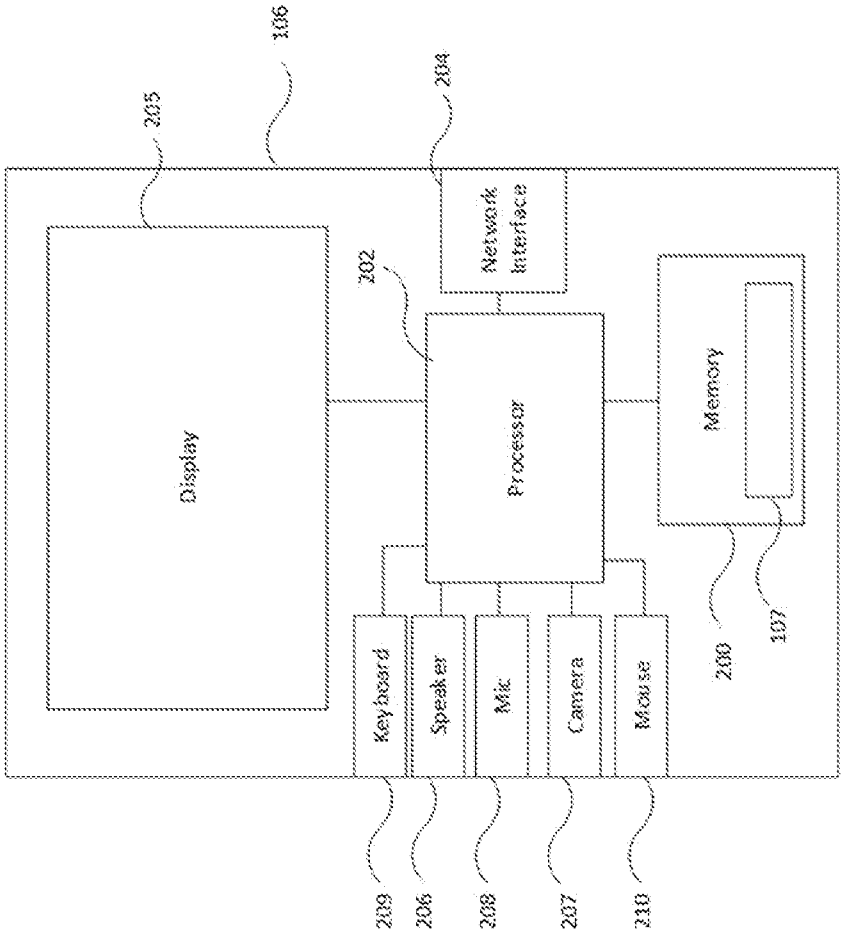
106
205
204
Network Interface
202
Display
Processor
Memory
Keyboard
Speaker
Mic
Camera
Mouse
208
107
209
206
208
207
210

FIG. 2

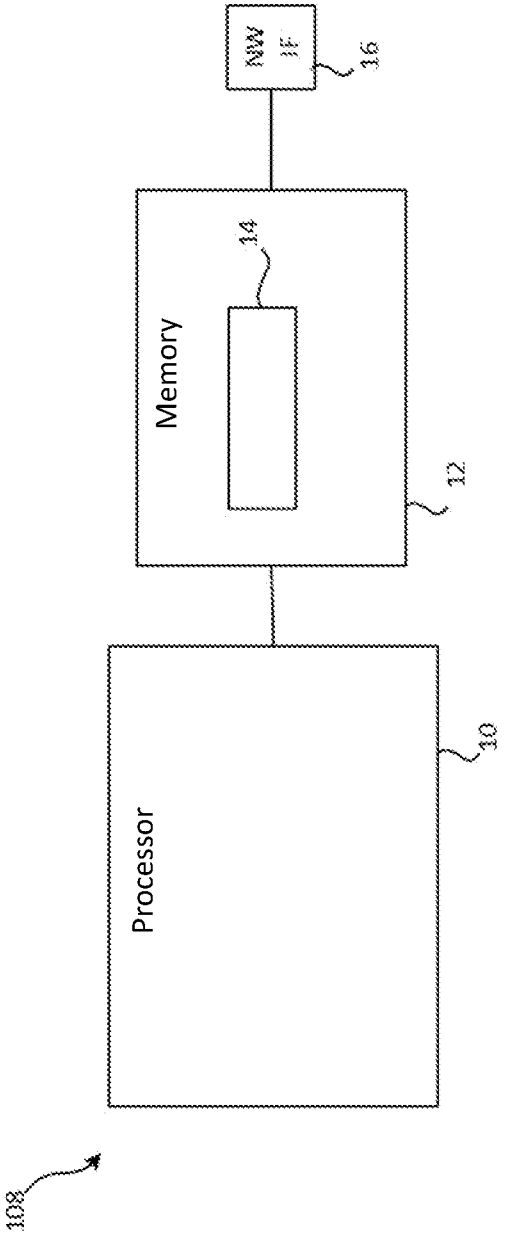

FIG. 2A

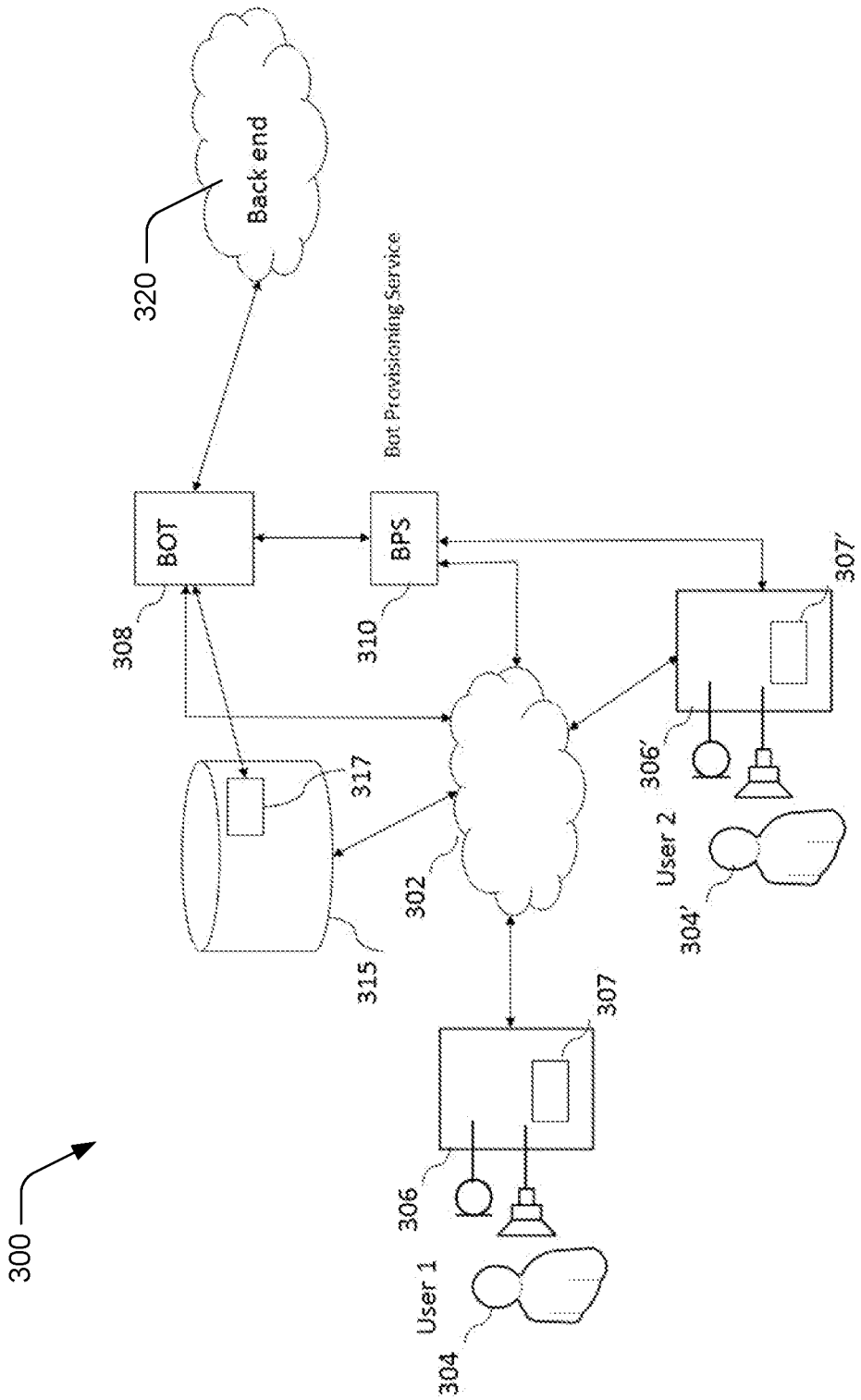
300
Back end
320
Bot Provisioning Service
BOT
BPS
308
310
317
315
302
306
307
User 1
304
306'
307'
User 2
304'

FIG. 3

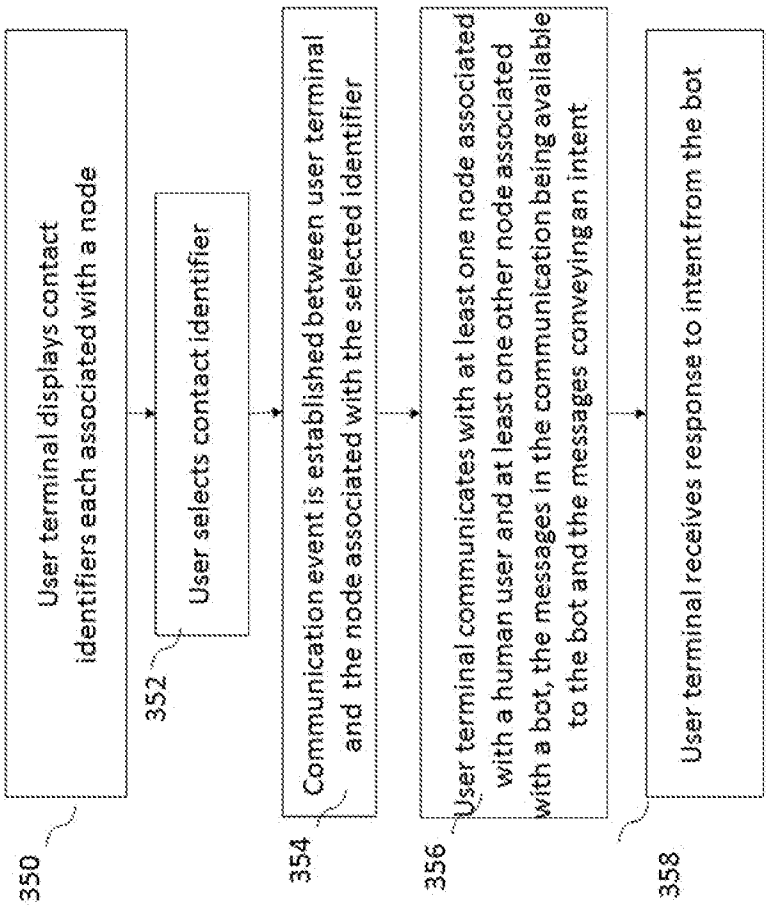
User terminal displays contact identifiers each associated with a node
User selects contact identifier
Communication event is established between user terminal and the node associated with the selected identifier
User terminal communicates with at least one node associated with a human user and at least one other node associated with a bot, the messages in the communication being available to the bot and the messages conveying an intent
User terminal receives response to intent from the bot
350
352
354
356
358

FIG. 3A

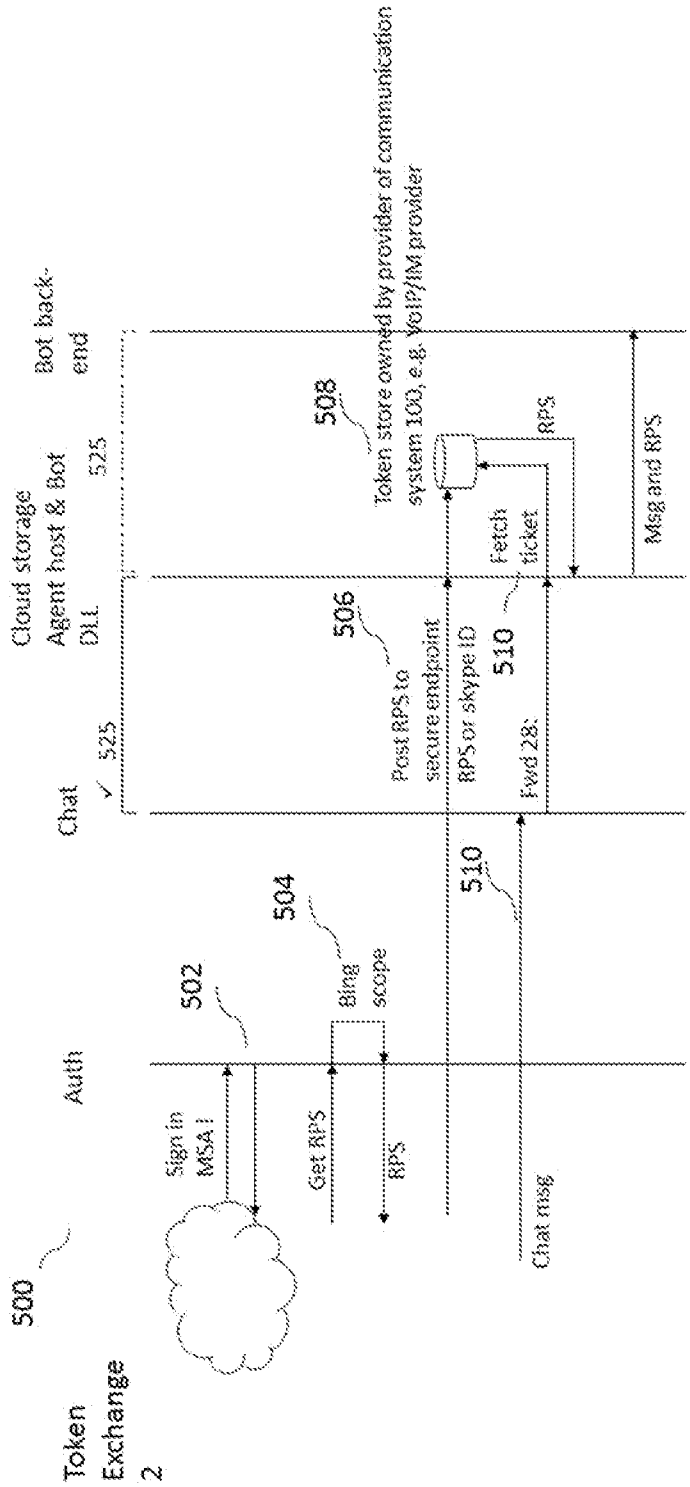
500
Token Exchange 2
Auth
Chat
Cloud storage
Agent host & Bot
DLL
Bot back-end
525
525
502
504
506
508
510
510
Sign in MSA I
Get RPS
RPS
Bing scope
Post RPS to secure endpoint
RPS or skype ID
Fwd 28:
Chat msg
Token store owned by provider of communication system 100, e.g. VoIP/IM provider
Fetch ticket
RPS
Msg and RPS

FIG. 4

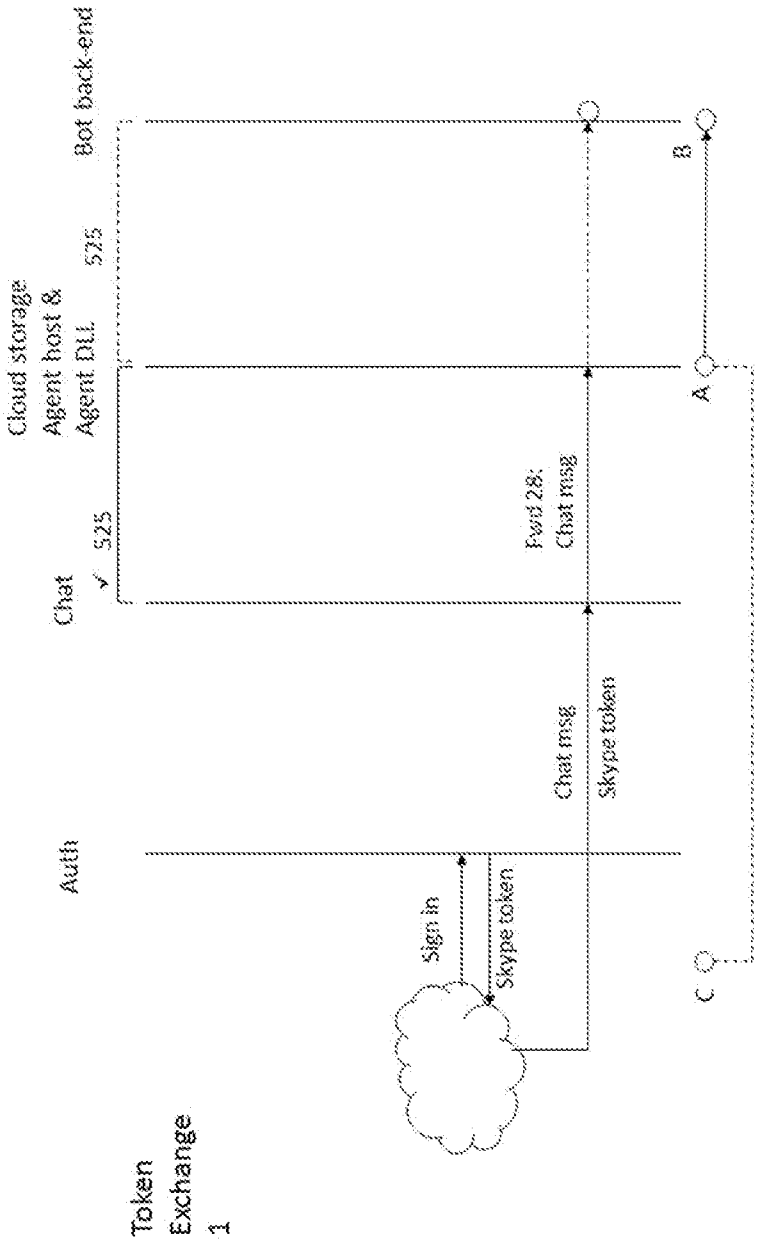
Token Exchange 1
Auth
Chat
525
Cloud storage
Agent host &
Agent DLL
525
Bot back-end
Sign in
Skype token
Chat msg
Skype token
Fwd 28:
Chat msg
A
B
C

FIG. 4A

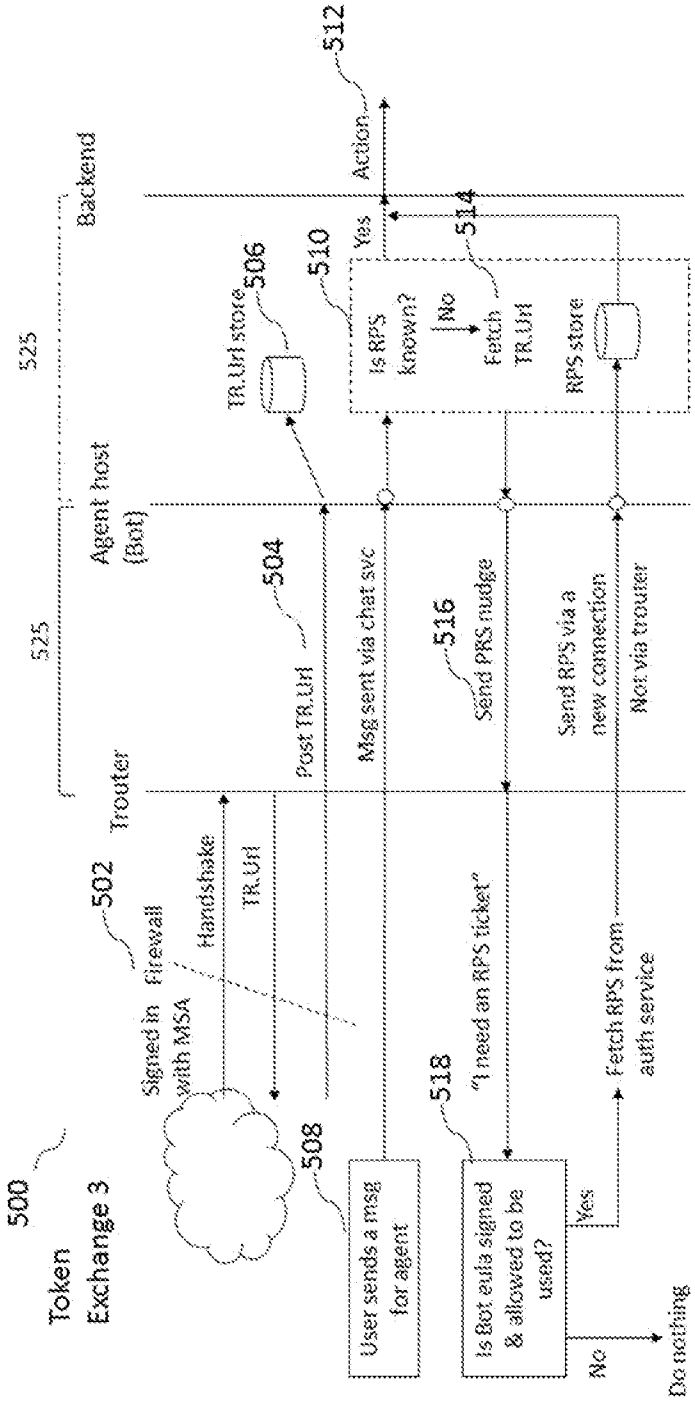
500
Token Exchange 3
502
Firewall
Handshake
Signed in with MSA
TR.Url
Trouter
525
Agent host (Bot)
525
Backend
504
Post TR.Url
Msg sent via chat svc
516
Send PRS nudge
Send RPS via a new connection
Not via trouter
TR.Url store
506
510
Is RPS known?
No
Fetch TR.Url
RPS store
Yes
514
512
Action
508
User sends a msg for agent
518
"I need an RPS ticket"
Is Bot eula signed & allowed to be used?
Yes
No
Do nothing
Fetch RPS from auth service

FIG. 5

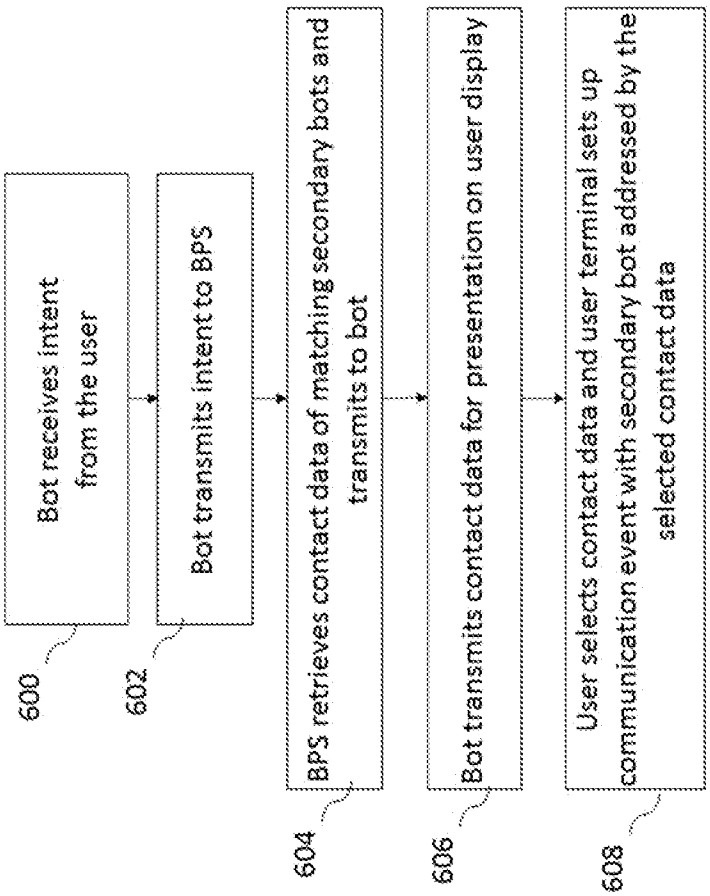
Bot receives intent from the user
Bot transmits intent to BPS
BPS retrieves contact data of matching secondary bots and transmits to bot
Bot transmits contact data for presentation on user display
User selects contact data and user terminal sets up communication event with secondary bot addressed by the selected contact data
600
602
604
606
608

FIG. 6

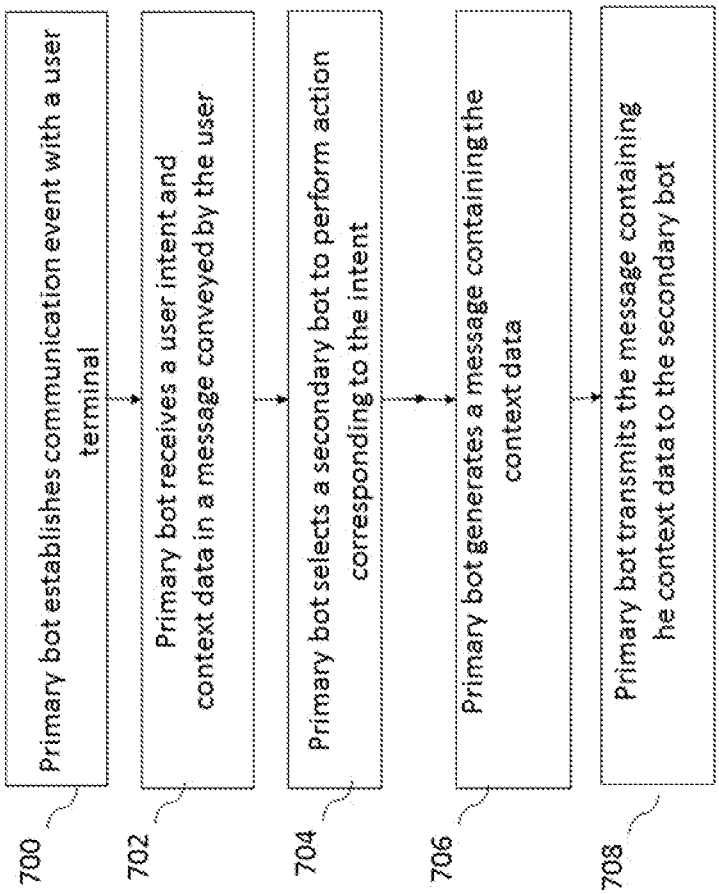
Primary bot establishes communication event with a user terminal
Primary bot receives a user intent and context data in a message conveyed by the user
Primary bot selects a secondary bot to perform action corresponding to the intent
Primary bot generates a message containing the context data
Primary bot transmits the message containing he context data to the secondary bot
700
702
704
706
708

FIG. 7

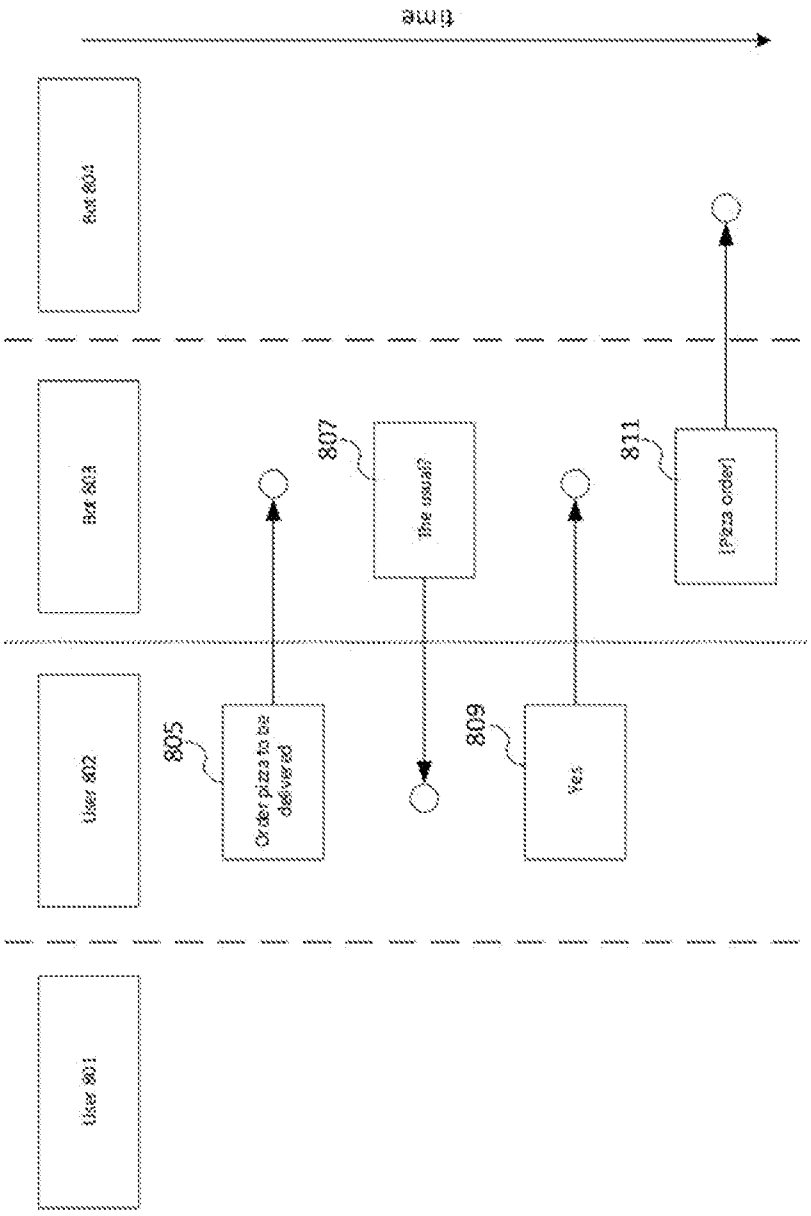
time
Bot 804
Bot 803
User 802
User 801
807
The usual?
811
Pizza order
805
Order pizza to be delivered
809
Yes

FIG. 8A

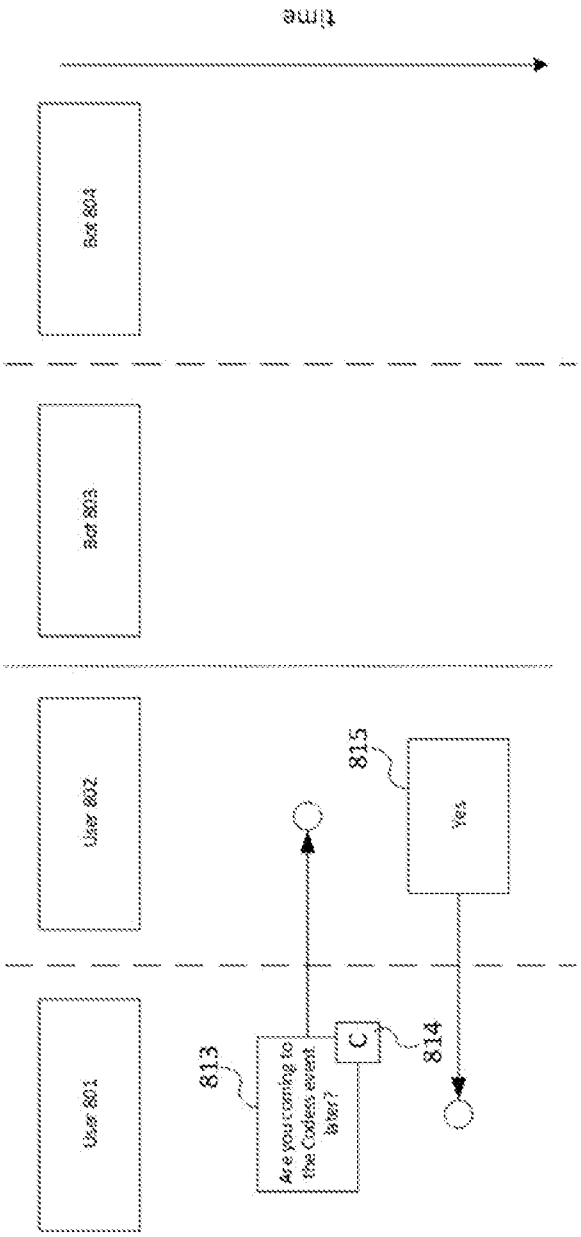
time
Bot 803
Bot 803
User 802
User 801
815
Yes
813
Are you coming to the Cadeau event later?
C
814

FIG. 8B

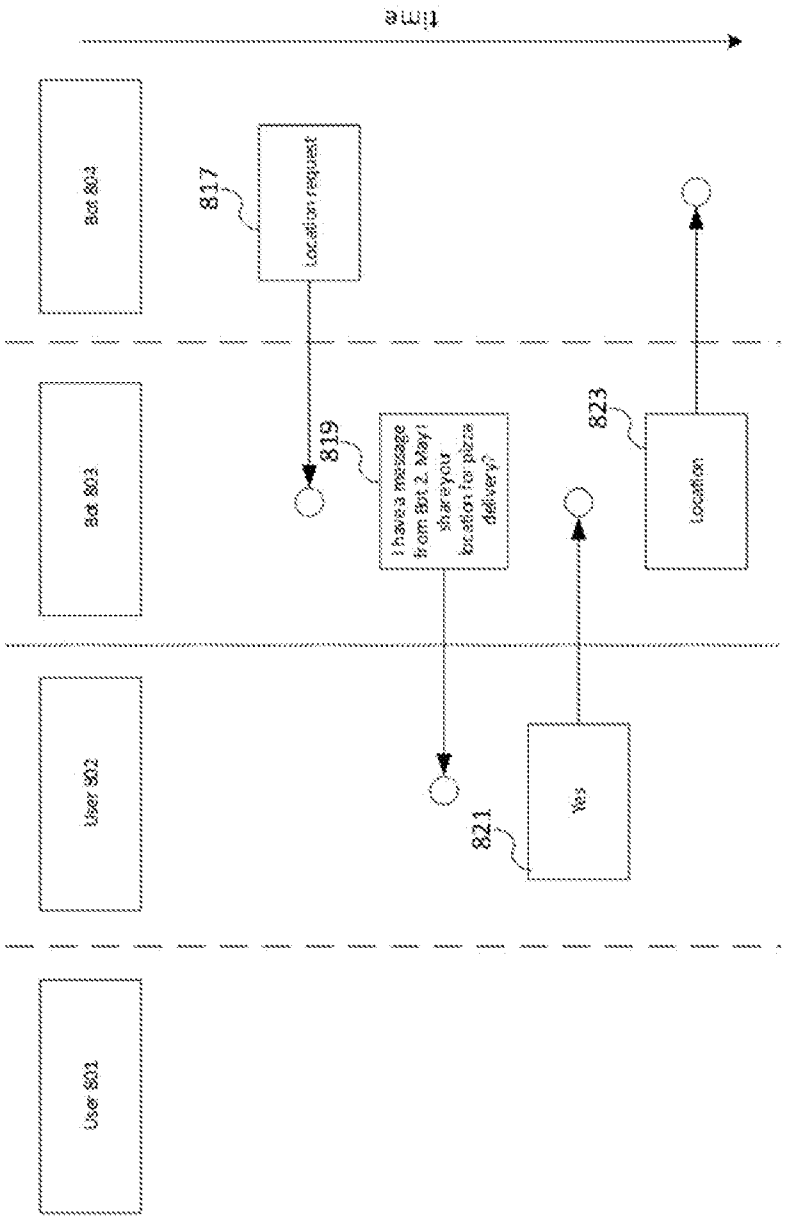
time
Bot 803
Bot 803
User 802
User 802
817
Location request
819
I have a message from Bot 2. May I share you location for pizza delivery?
823
Location
Yes
821

FIG. 8C

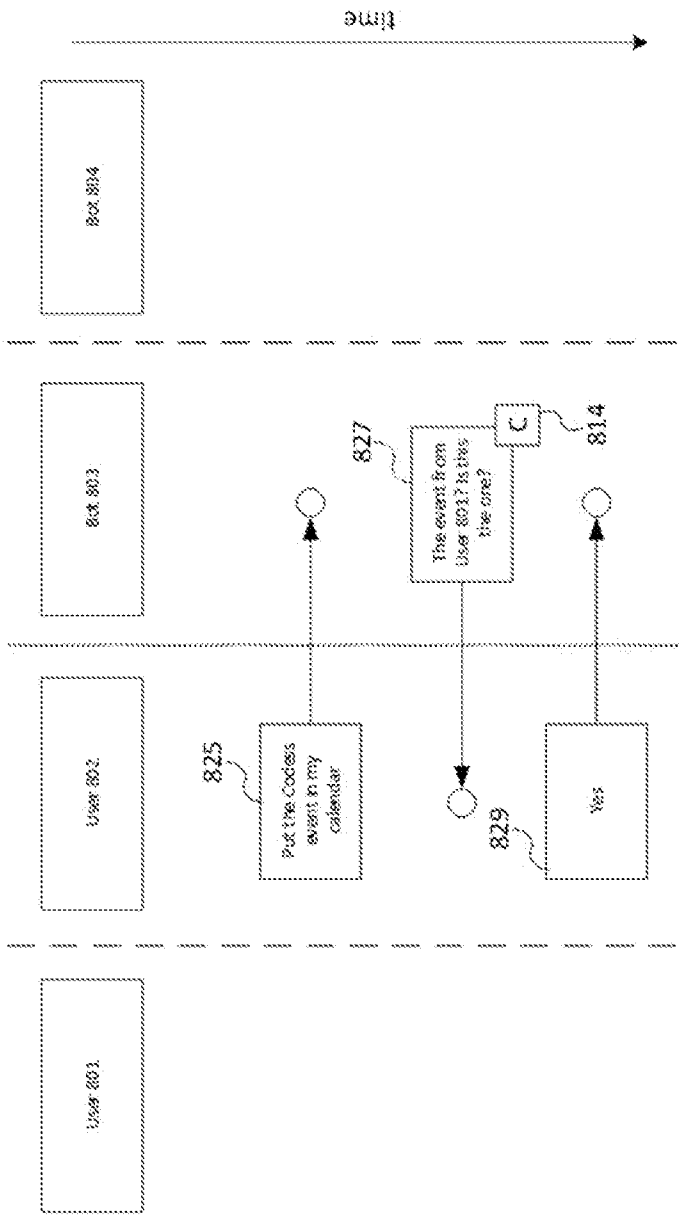
time
Bot 804
Bot 803
User 802
User 801
827
The event from User 803 is the one?
C
814
825
Put the Codes event in my calendar
829
Yes

FIG. 8D

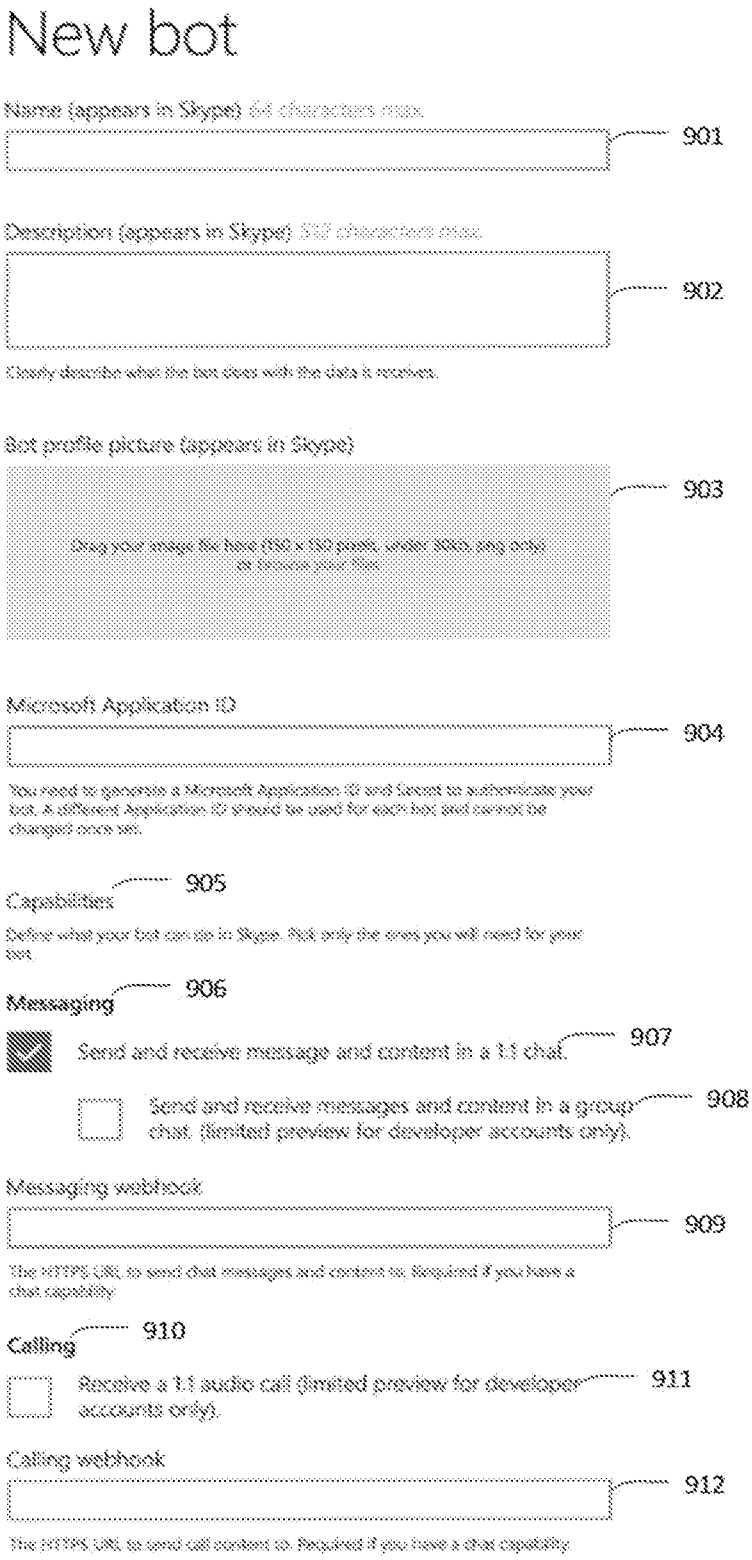
New bot
Name (appears in Skype) 64 characters max.
901
Description (appears in Skype) 512 characters max.
902
Clearly describe what the bot does with the data it receives.
Bot profile picture (appears in Skype)
903
Drag your image file here (192 x 192 pixels, under 30kb, png only)
or browse your files
Microsoft Application ID
904
You need to generate a Microsoft Application ID and Secret to authenticate your bot. A different Application ID should be used for each bot and cannot be changed once set.
Capabilities
905
Define what your bot can do in Skype. Not only the ones you will need for your bot.
Messaging
906
Send and receive message and content in a 1:1 chat.
907
Send and receive messages and content in a group chat. (limited preview for developer accounts only).
908
Messaging webhook
909
The HTTPS URL to send chat messages and content to. Required if you have a chat capability.
Calling
910
Receive a 1:1 audio call (limited preview for developer accounts only).
911
Calling webhook
912
The HTTPS URL to send call content to. Required if you have a chat capability.

FIG. 9

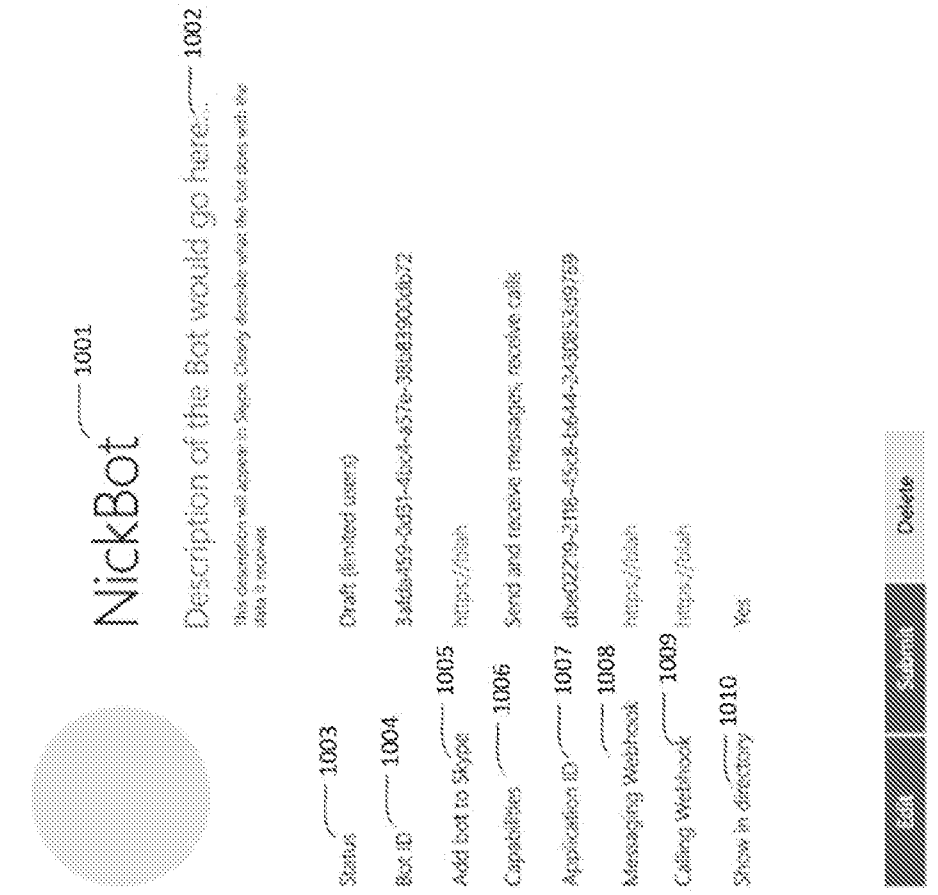
1001
NickBot
Description of the Bot would go here...
1002
1003
Status
Draft (limited users)
1004
Bot ID
1005
Add bot to Skype
https://skin
1006
Capabilities
Send and receive messages, receive calls
1007
Application ID
1008
Messaging Webhook
https://skin
1009
Calling Webhook
https://skin
1010
Show in directory
Yes
Edit
Delete

FIG. 10

My Bots
1101
1102
Name
Status
Actions
Draft pending
In review
Approved (published)
Blocked
Edit
Approve
Hide from Directory
Delete
Publish

FIG. 11

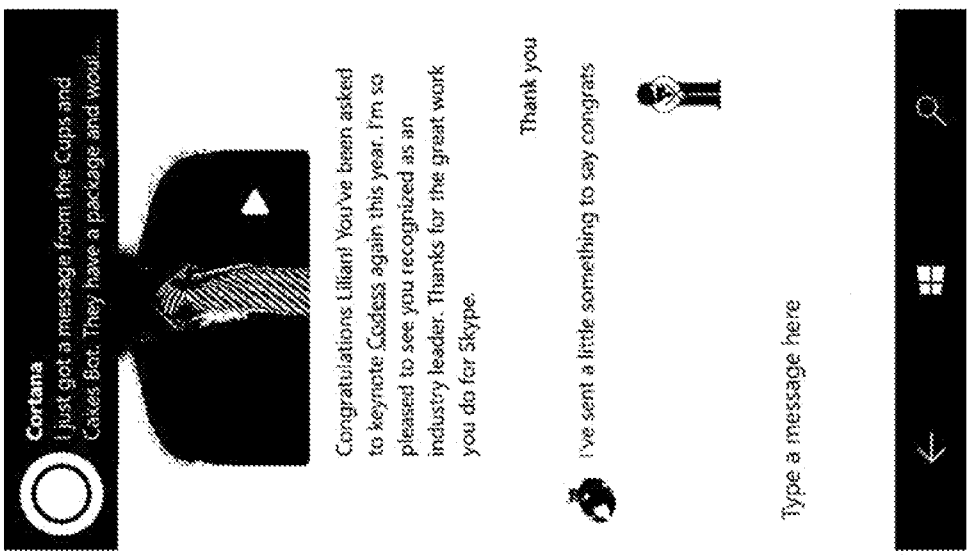
Cortana
I just got a message from the Cups and Cakes Bot. They have a package and woul...
Congratulations Lilian! You've been asked to keynote Codess again this year. I'm so pleased to see you recognized as an industry leader. Thanks for the great work you do for Skype.
Thank you
I've sent a little something to say congrats
Type a message here

FIG. 12A

11:49
Cortana
Joule Restaurant
Cortana, remind me to pick up my dry cleaning on the way home.
I'll take care of that for you!
Today
I just got a message from the Cups and Cakes Bot – they would like to know your location for delivery. May I share it?
Moscone Center
747 Howard St
San Francisco, CA 94103
Type a message here

FIG. 12B

11:49
Cortana
Book Restaurant
Cortana, remind me to pick up my dry cleaning on the way home.
I'll take care of that for you!
Today
I just got a message from the Cups and Cakes bot -- they would like to know your location for delivery. May I share it?
Moscone Center
747 Howard St
San Francisco, CA 94103
Yes
Type a message here

FIG. 12C

11:49
Cortana
Cortana, remind me to pick up my dry cleaning on the way home.
I'll take care of that for you!
Today
I just got a message from the Cups and Cakes Bot – they would like to know your location for delivery. May I share it?
Moscone Center
747 Howard St
San Francisco, CA 94103
Yes
Would you like to track delivery?
Type a message here

FIG. 12D

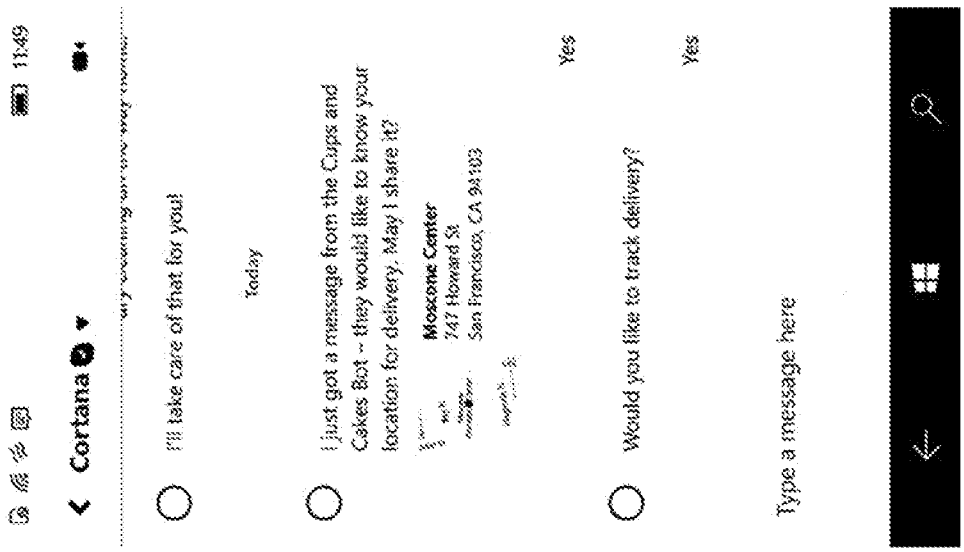
11:49
Cortana
I'll take care of that for you!
Today
I just got a message from the Cups and Cakes Bot - they would like to know your location for delivery. May I share it?
Moscone Center
747 Howard St
San Francisco, CA 94103
Yes
Would you like to track delivery?
Yes
Type a message here

FIG. 12E

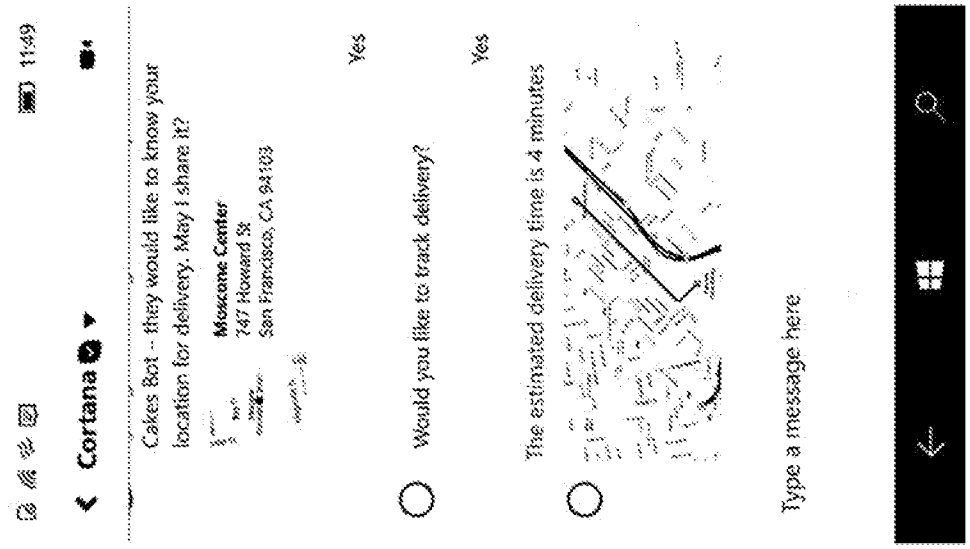
11:49
Cortana
Cakes Bot - they would like to know your location for delivery. May I share it?
Moscone Center
747 Howard St
San Francisco, CA 94103
Yes
Would you like to track delivery?
Yes
The estimated delivery time is 4 minutes
Type a message here

FIG. 12F

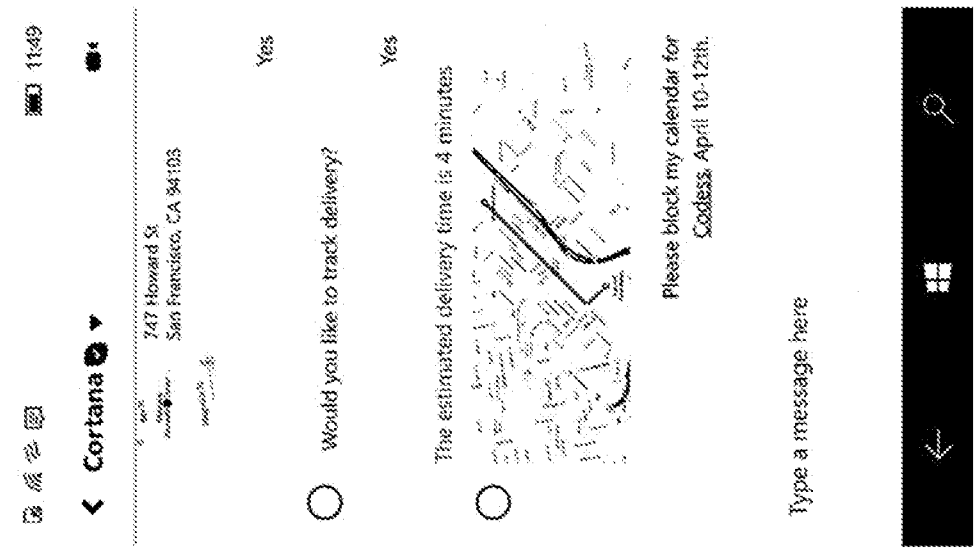
11:49
Cortana
747 Howard St
San Francisco, CA 94103
Yes
Would you like to track delivery?
Yes
The estimated delivery time is 4 minutes
Please block my calendar for
Codess April 10-12th.
Type a message here

FIG. 12G

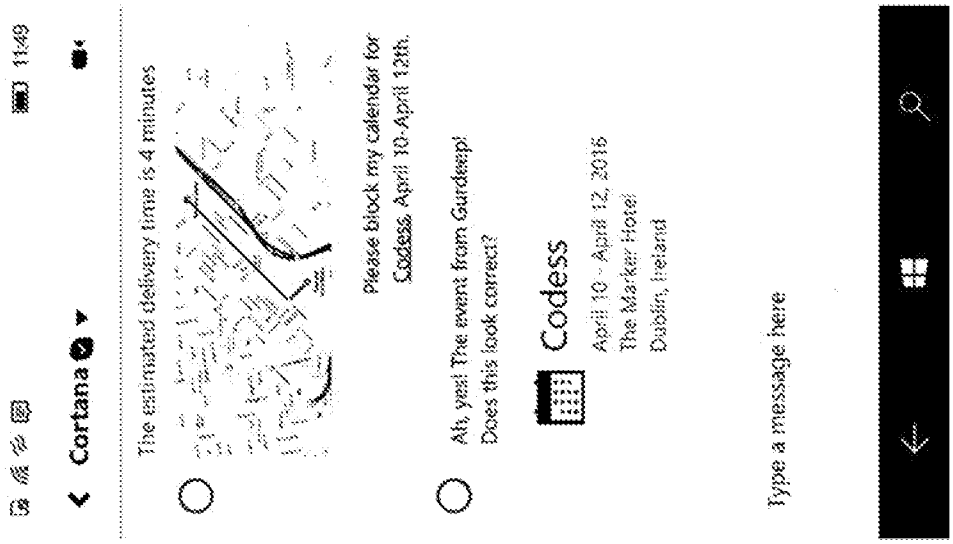
11:49
Cortana
The estimated delivery time is 4 minutes
Please block my calendar for Codess April 10-April 12th.
Ah, yes! The event from Gurdeep! Does this look correct?
Codess
April 10 - April 12, 2016
The Marker Hotel
Dublin, Ireland
Type a message here

FIG. 12H

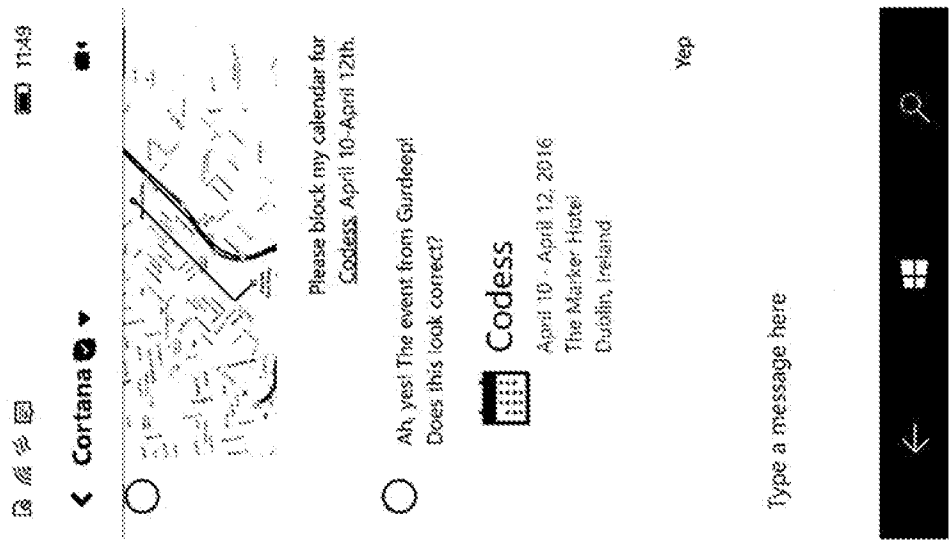
11:49
Cortana
Please block my calendar for Codess, April 10-April 12th.
Ah, yes! The event from Gurdeep! Does this look correct?
Codess
April 10 - April 12, 2016
The Marker Hotel
Dublin, Ireland
Yep
Type a message here

FIG. 12I

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/024381

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/46 G06F9/50
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 263 358 B1 (BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY) 17 July 2001 (2001-07-17) abstract column 1, lines 8-33 column 2, lines 22-28, 60-65 column 3, lines 26-58 column 5, lines 5-20 column 6, line 29 - column 7, line 4 column 7, lines 46-48 column 9, lines 13-29 -----	1-15
A	EP 0 964 334 A2 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 15 December 1999 (1999-12-15) the whole document -----	1-15

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 June 2017	11/07/2017

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Tomàs Blanch, F

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/024381

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 8 886 701 B1 (HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.) 11 November 2014 (2014-11-11) the whole document	1-15
A	US 2007/011485 A1 (CASSATT CORPORATION) 11 January 2007 (2007-01-11) the whole document	1-15
A	US 7 440 992 B1 (AGENT SOFTWARE CORPORATION) 21 October 2008 (2008-10-21) the whole document	1-15

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2017/024381

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6263358 B1	17-07-2001	AU 735024 B2	28-06-2001
		CA 2297935 A1	04-02-1999
		DE 69808633 D1	14-11-2002
		DE 69808633 T2	12-06-2003
		EP 0996887 A1	03-05-2000
		US 6263358 B1	17-07-2001
		WO 9905594 A1	04-02-1999
EP 0964334 A2	15-12-1999	CN 1237734 A	08-12-1999
		EP 0964334 A2	15-12-1999
		JP 3629548 B2	16-03-2005
		JP 2000029709 A	28-01-2000
		KR 20000005680 A	25-01-2000
		TW 412707 B	21-11-2000
		US 6549932 B1	15-04-2003
US 8886701 B1	11-11-2014	US 8886701 B1	11-11-2014
		US 2015067022 A1	05-03-2015
US 2007011485 A1	11-01-2007	US 2007011485 A1	11-01-2007
		US 2012192152 A1	26-07-2012
US 7440992 B1	21-10-2008	US 7440992 B1	21-10-2008
		US 2009157796 A1	18-06-2009