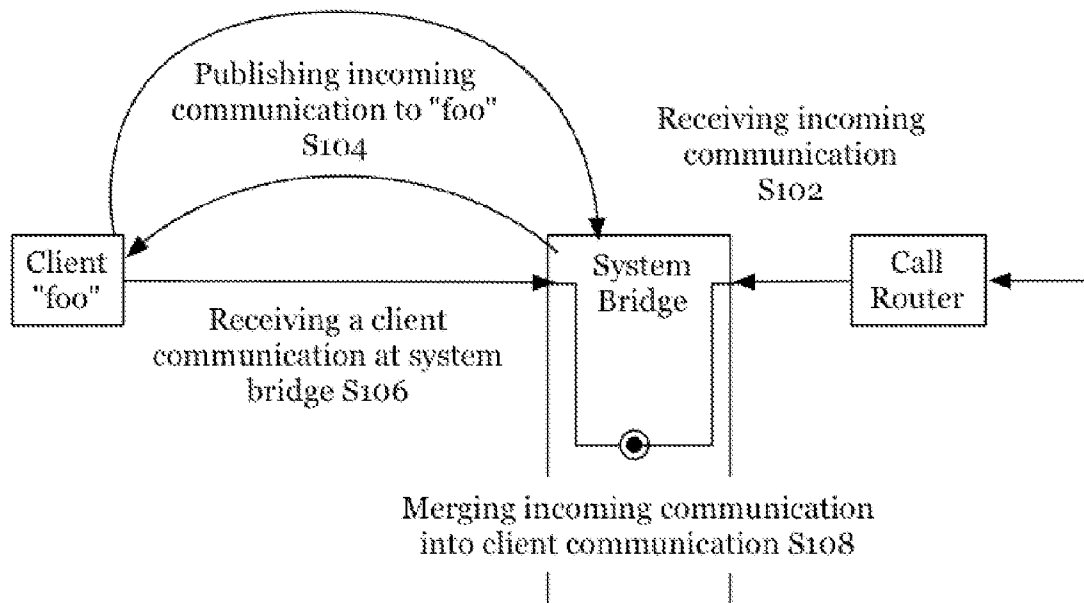


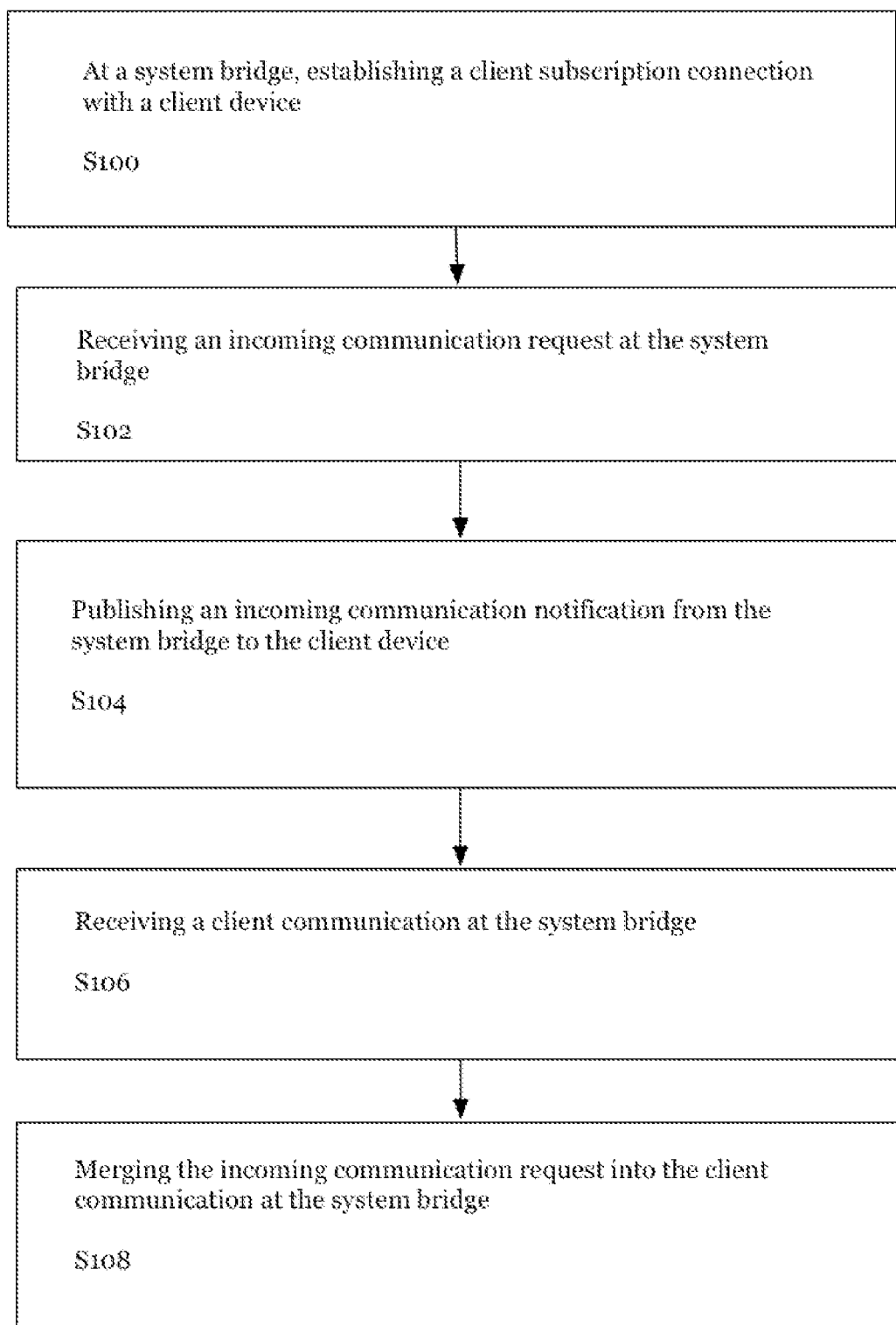


US 20130019297A1

(19) **United States**(12) **Patent Application Publication**
Lawson et al.(10) **Pub. No.: US 2013/0019297 A1**(43) **Pub. Date: Jan. 17, 2013**(54) **SYSTEM AND METHOD FOR
COMMUNICATING WITH A CLIENT
APPLICATION**(71) Applicant: **Twilio, Inc.**, San Francisco, CA (US)(72) Inventors: **Jeffrey Lawson**, San Francisco, CA
(US); **John Wolthuis**, San Francisco, CA
(US); **Evan Cooke**, San Francisco, CA
(US); **Jonas Borjesson**, San Francisco,
CA (US); **Rob Simutis**, San Francisco,
CA (US)(73) Assignee: **TWILIO, INC.**, San Francisco, CA
(US)(21) Appl. No.: **13/624,739**(22) Filed: **Sep. 21, 2012****Related U.S. Application Data**(63) Continuation-in-part of application No. 13/478,495,
filed on May 23, 2012.(60) Provisional application No. 61/489,189, filed on May
23, 2011, provisional application No. 61/500,549,
filed on Jun. 23, 2011, provisional application No.
61/537,217, filed on Sep. 21, 2011.**Publication Classification**(51) **Int. Cl.**
H04L 9/32 (2006.01)(52) **U.S. Cl.** **726/7**(57) **ABSTRACT**

A system and method for communicating with a client application that can include at a communication platform, receiving an authorization token of a first client application; verifying at least one permission associated with the authorization token; at a first server of the communication platform, accepting an incoming communication request; retrieving communication instructions from a server according to the incoming communication request; identifying an instruction to communicate with a communication destination of the first client application; and establishing communication with the first client application.

Establishing client subscription S100

**FIGURE 1**

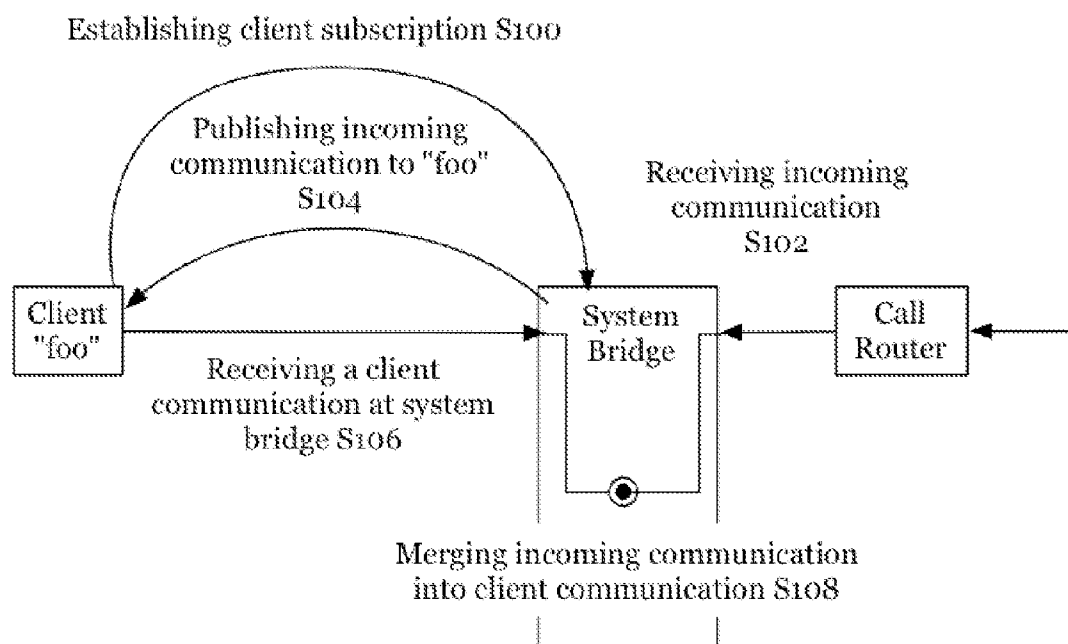
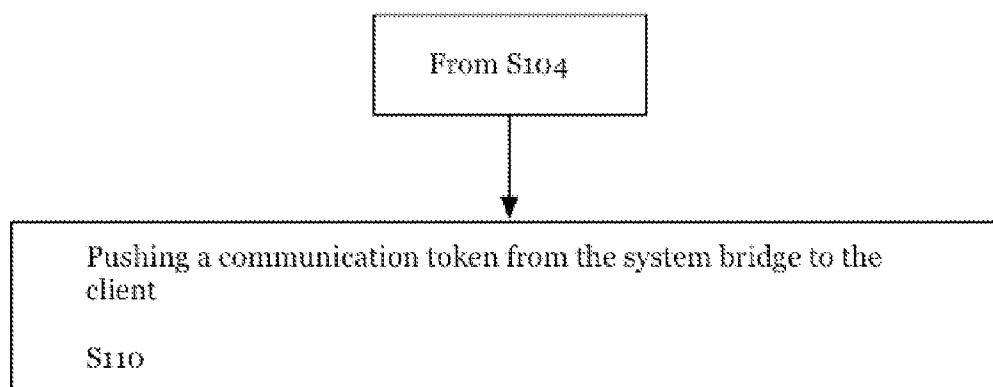
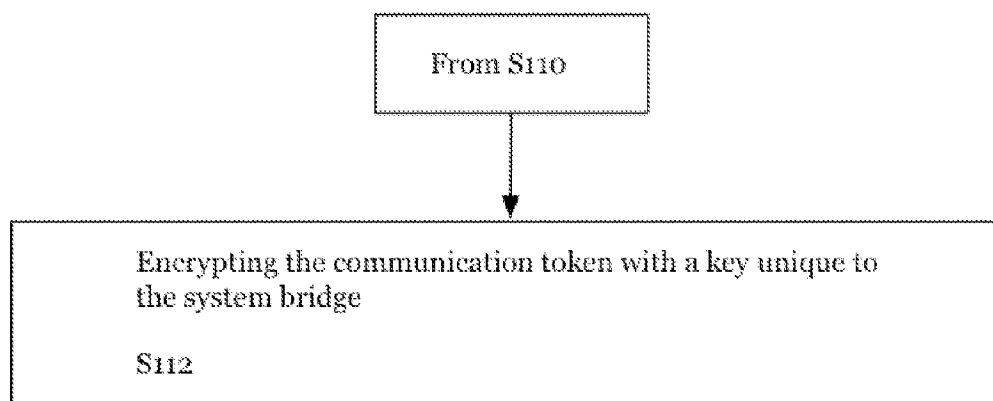
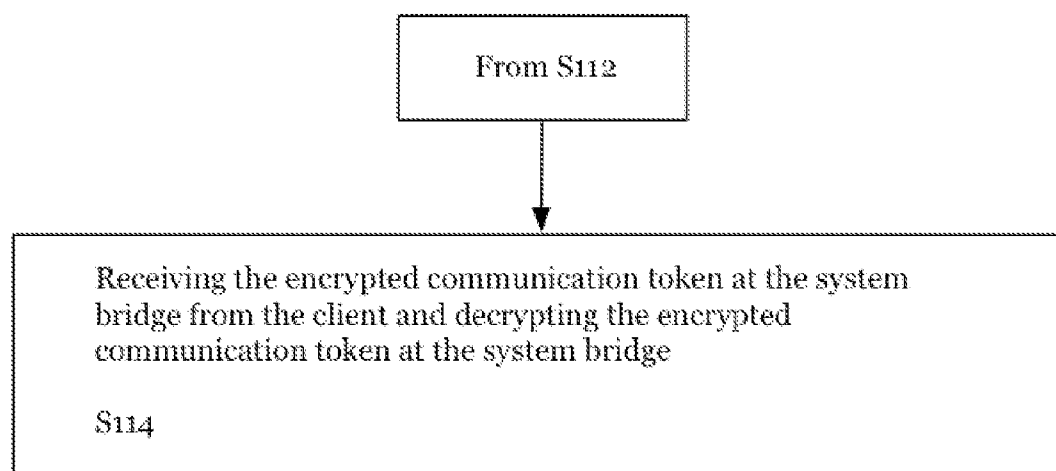


FIGURE 2

**FIGURE 3****FIGURE 4**

**FIGURE 5**

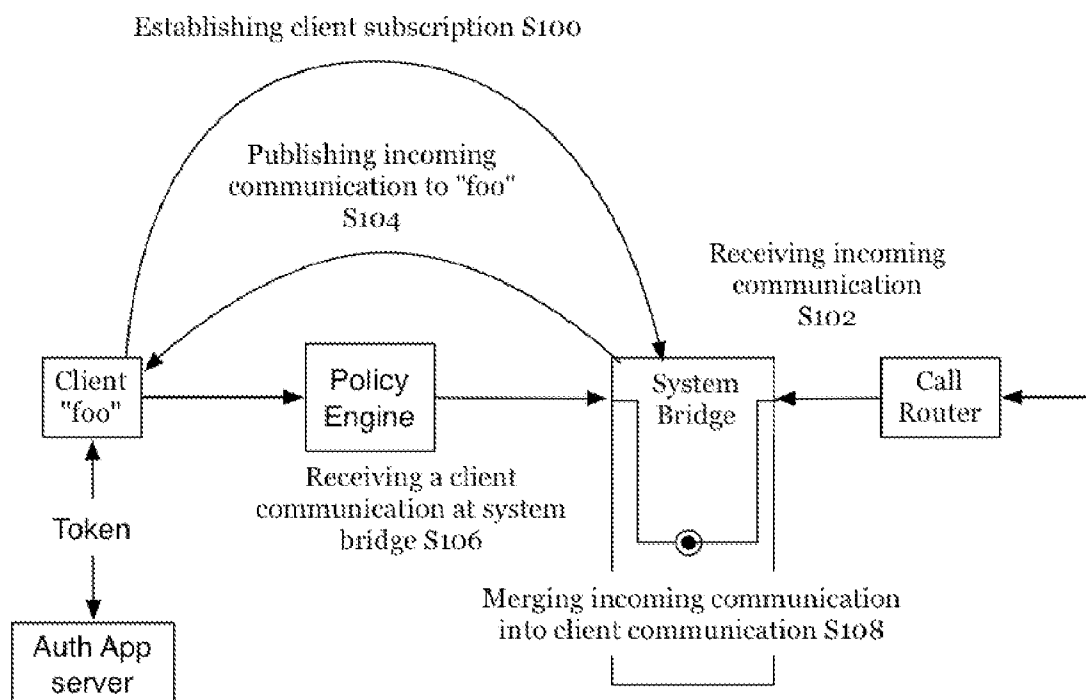
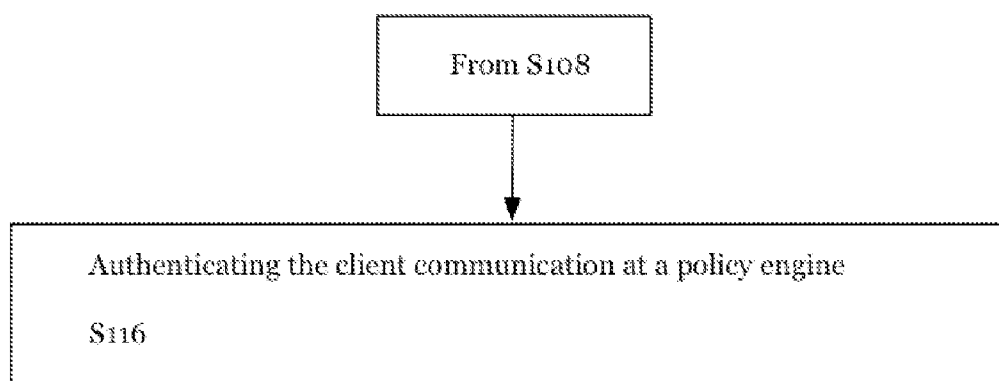
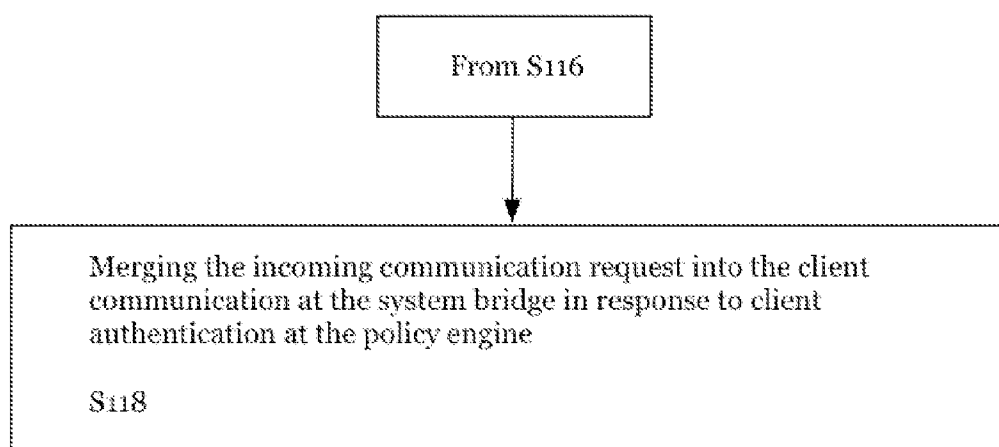
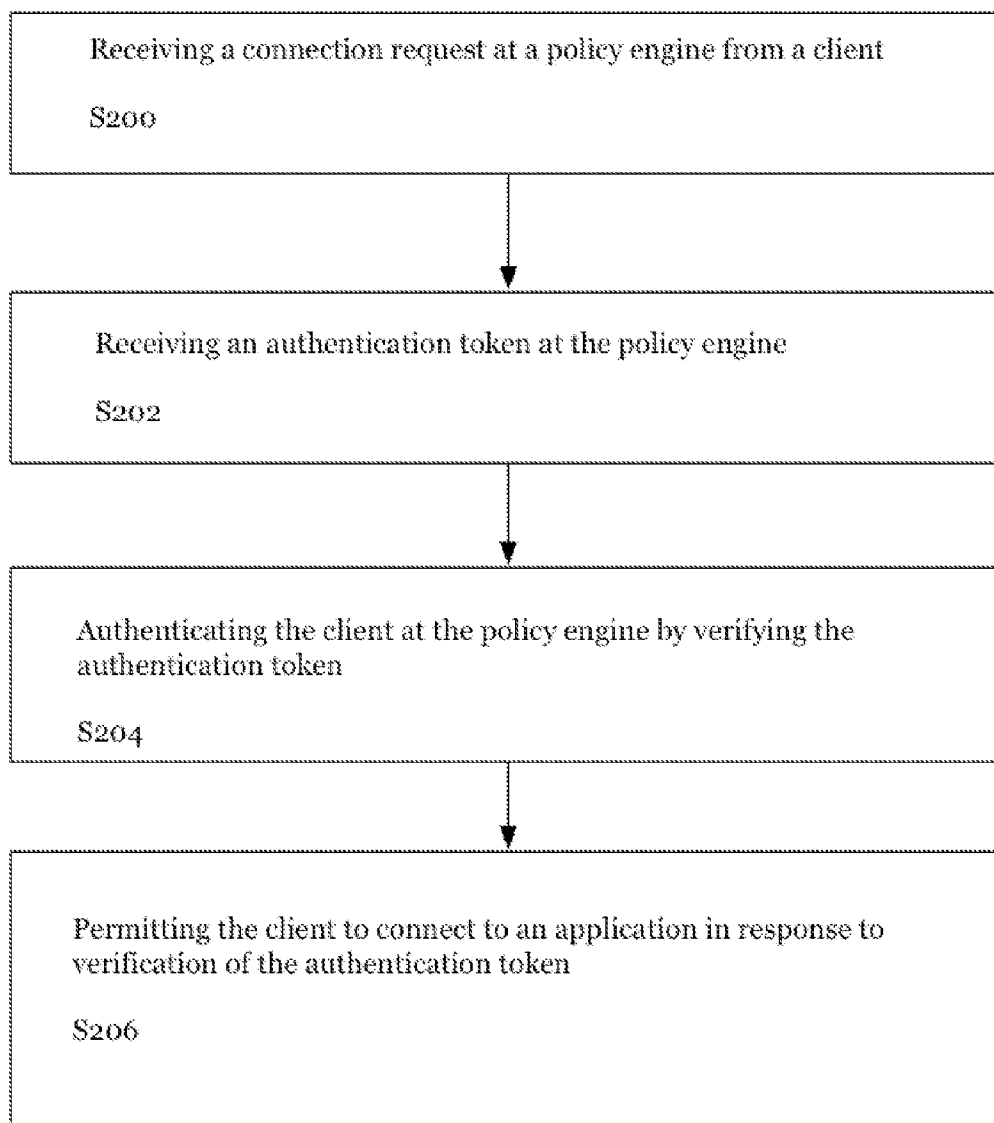


FIGURE 6

**FIGURE 7****FIGURE 8**

**FIGURE 9**

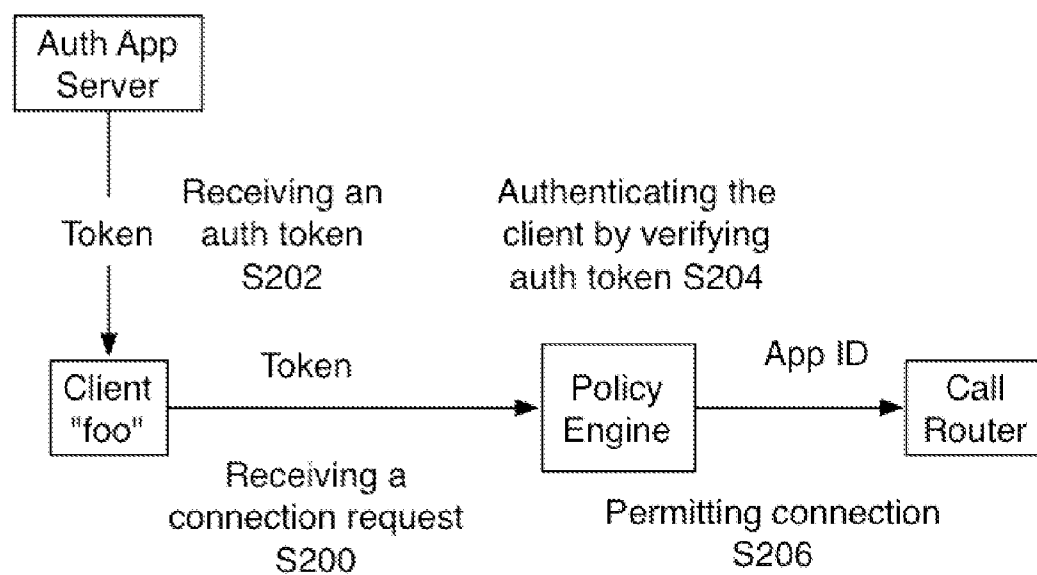
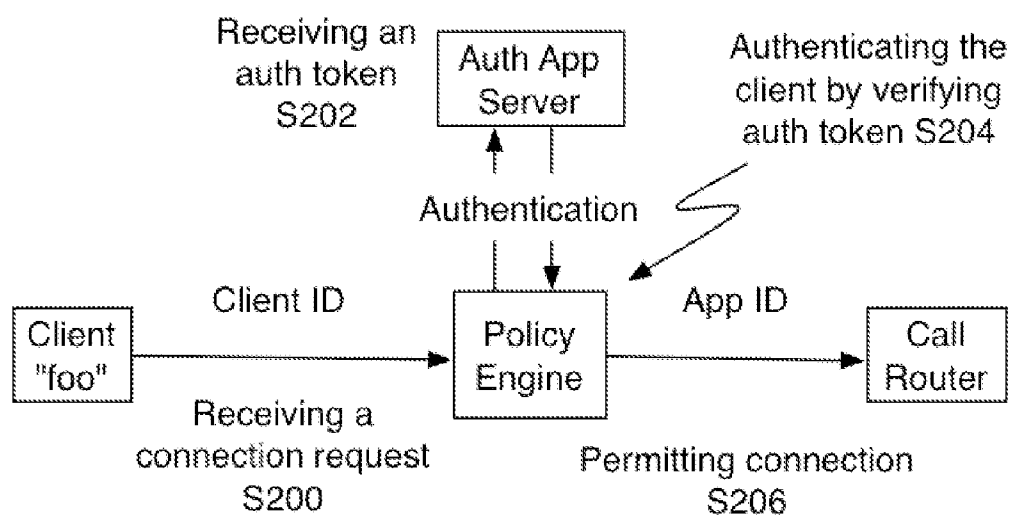
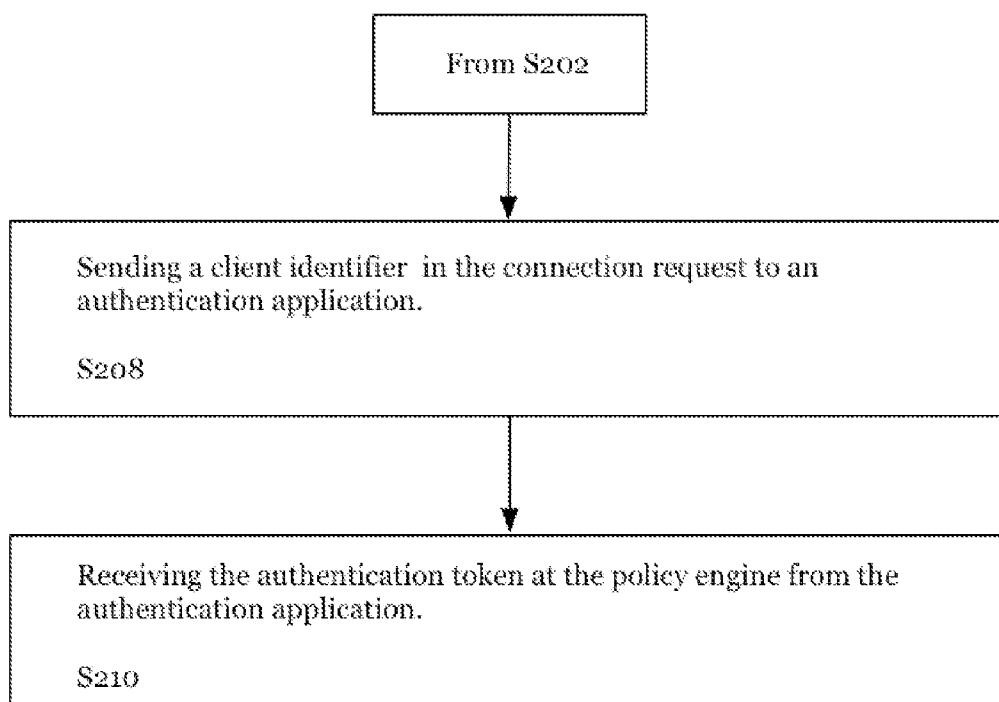
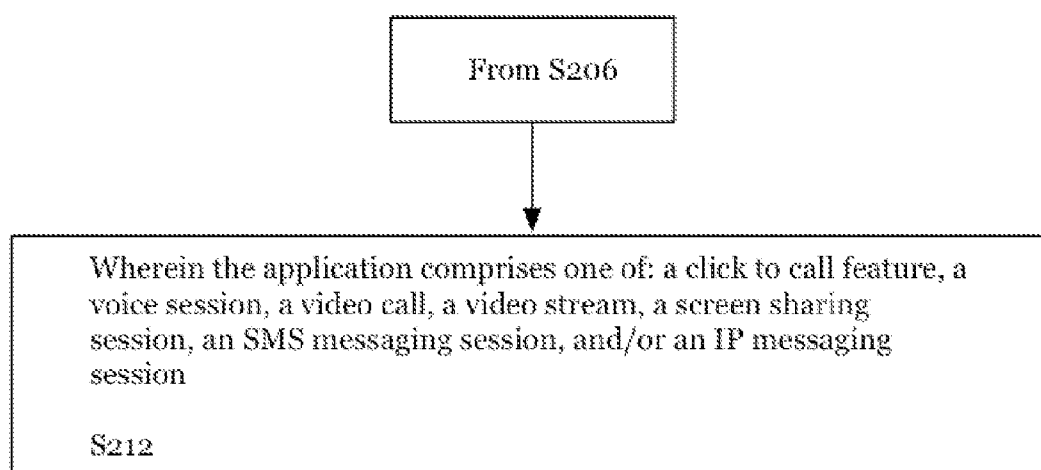


FIGURE 10

**FIGURE 11**

**FIGURE 12**

**FIGURE 13**

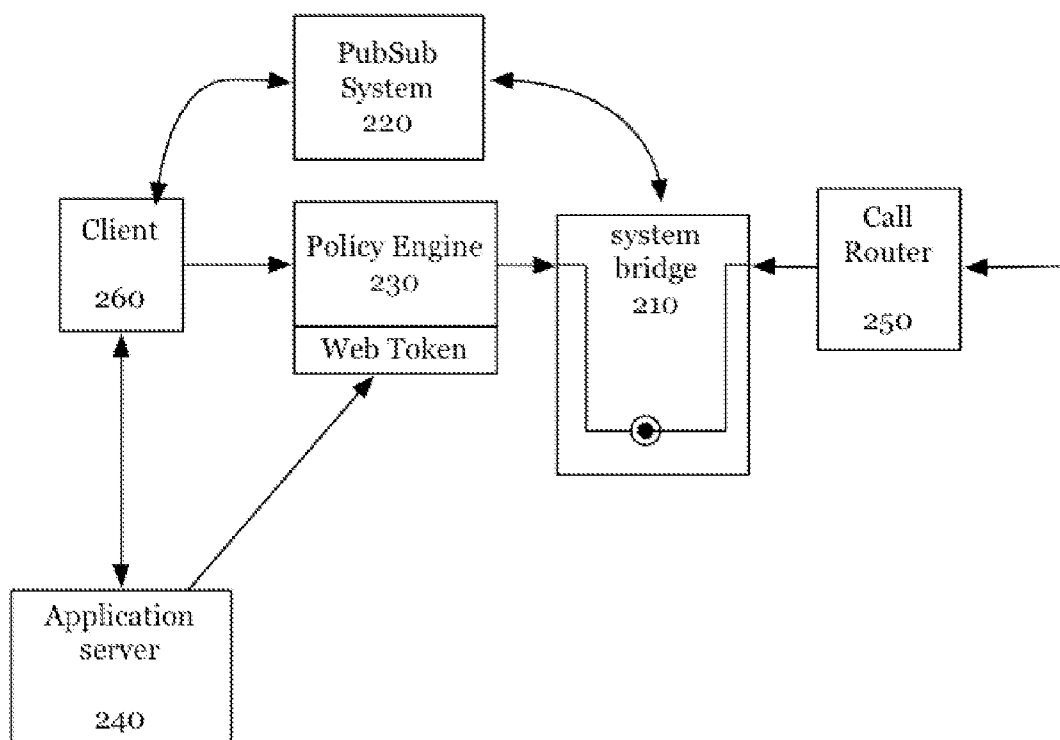


FIGURE 14

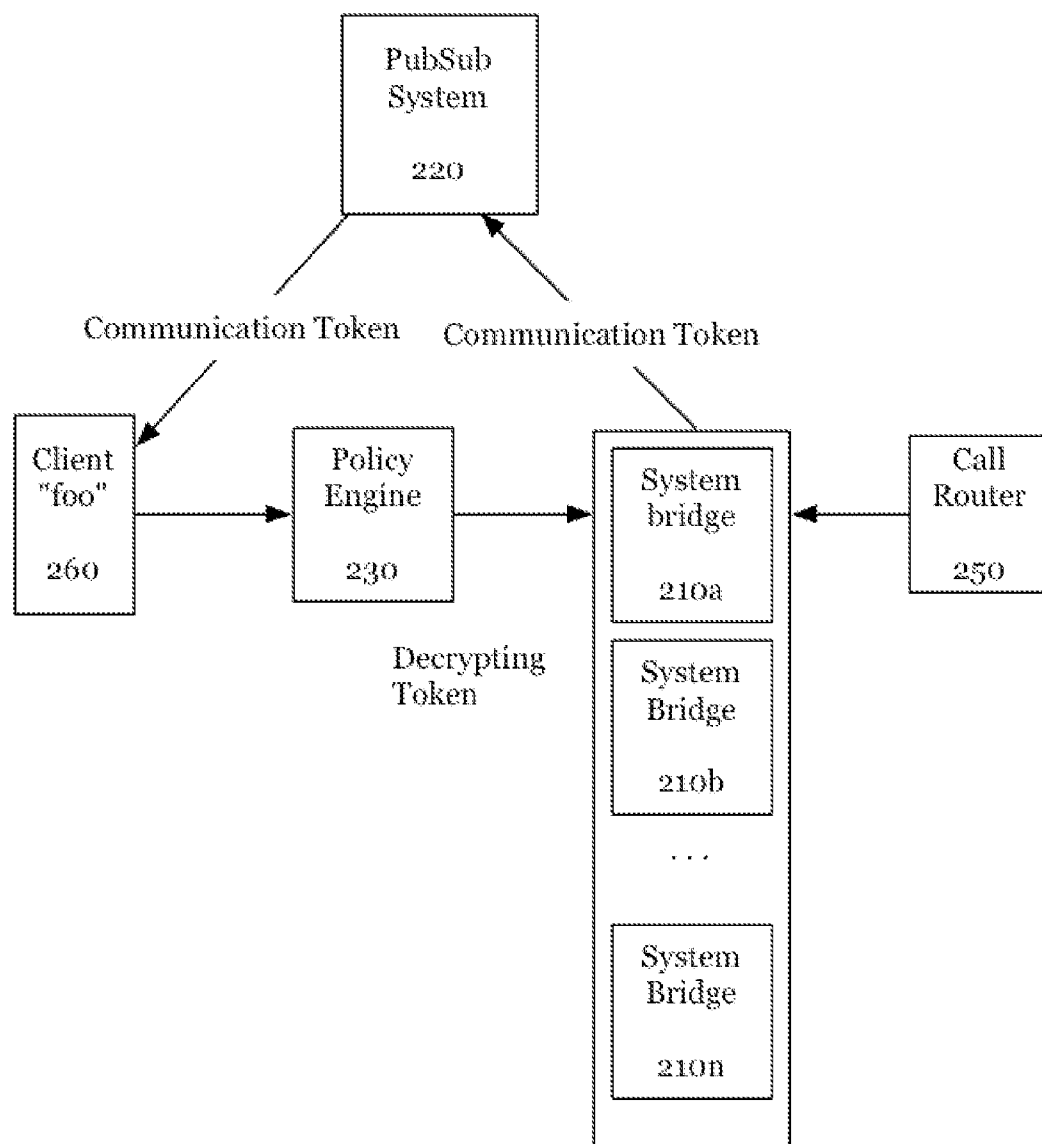
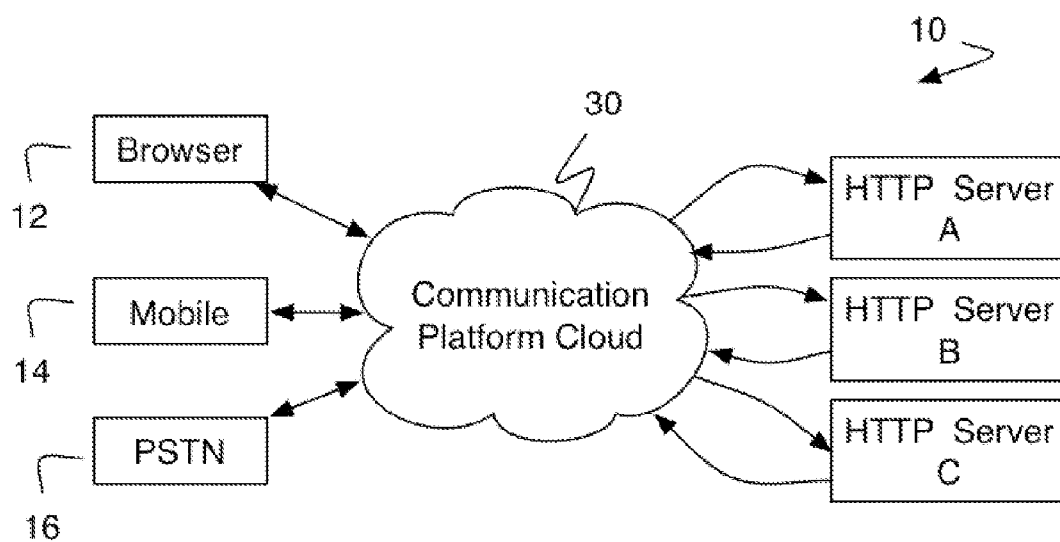


FIGURE 15

FIGURE 16

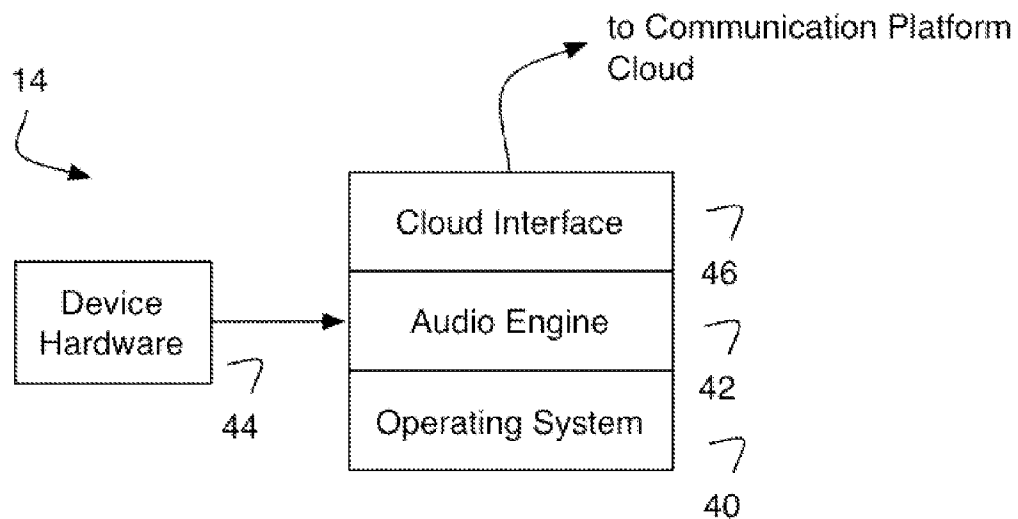


FIGURE 17

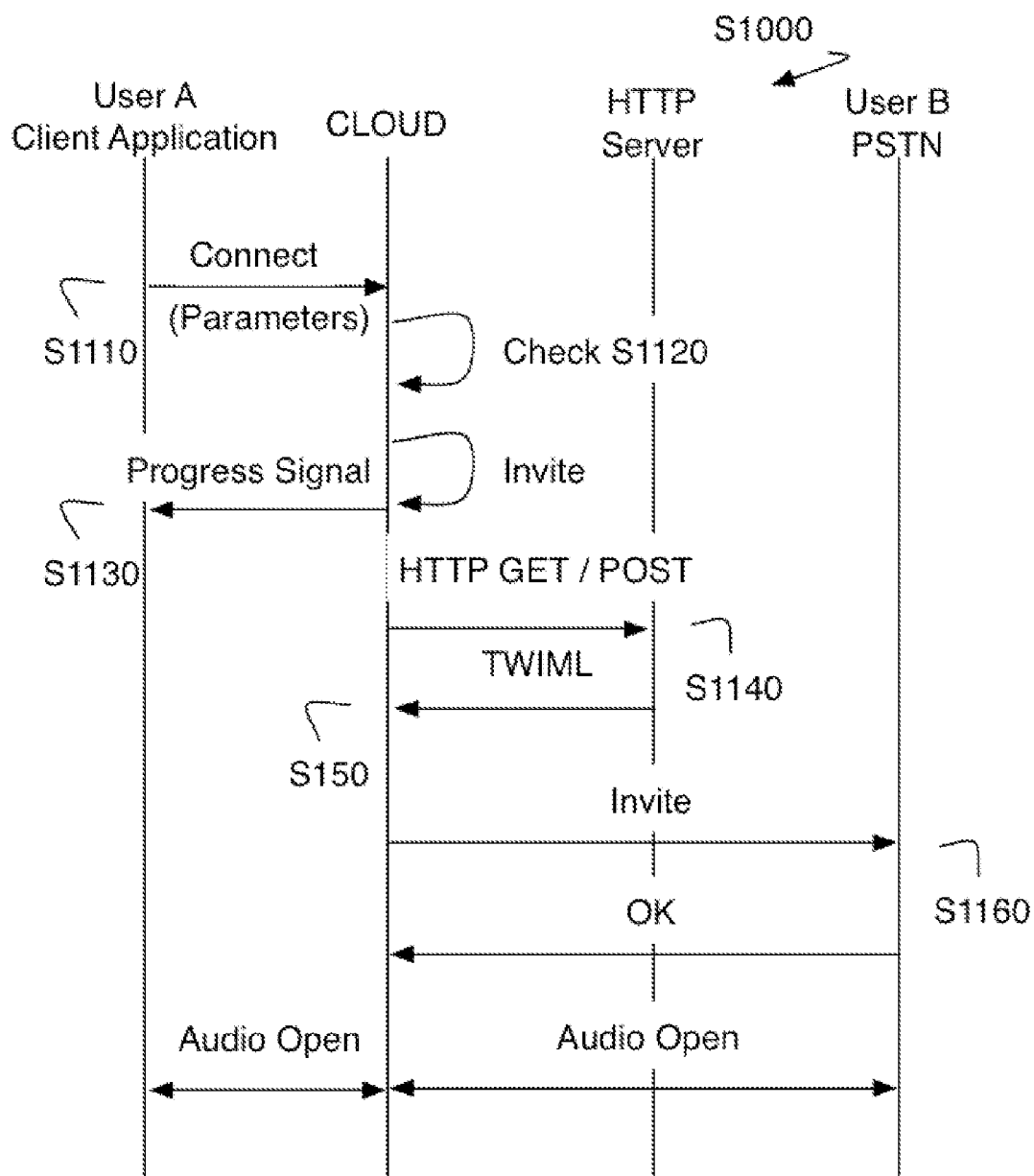


FIGURE 18

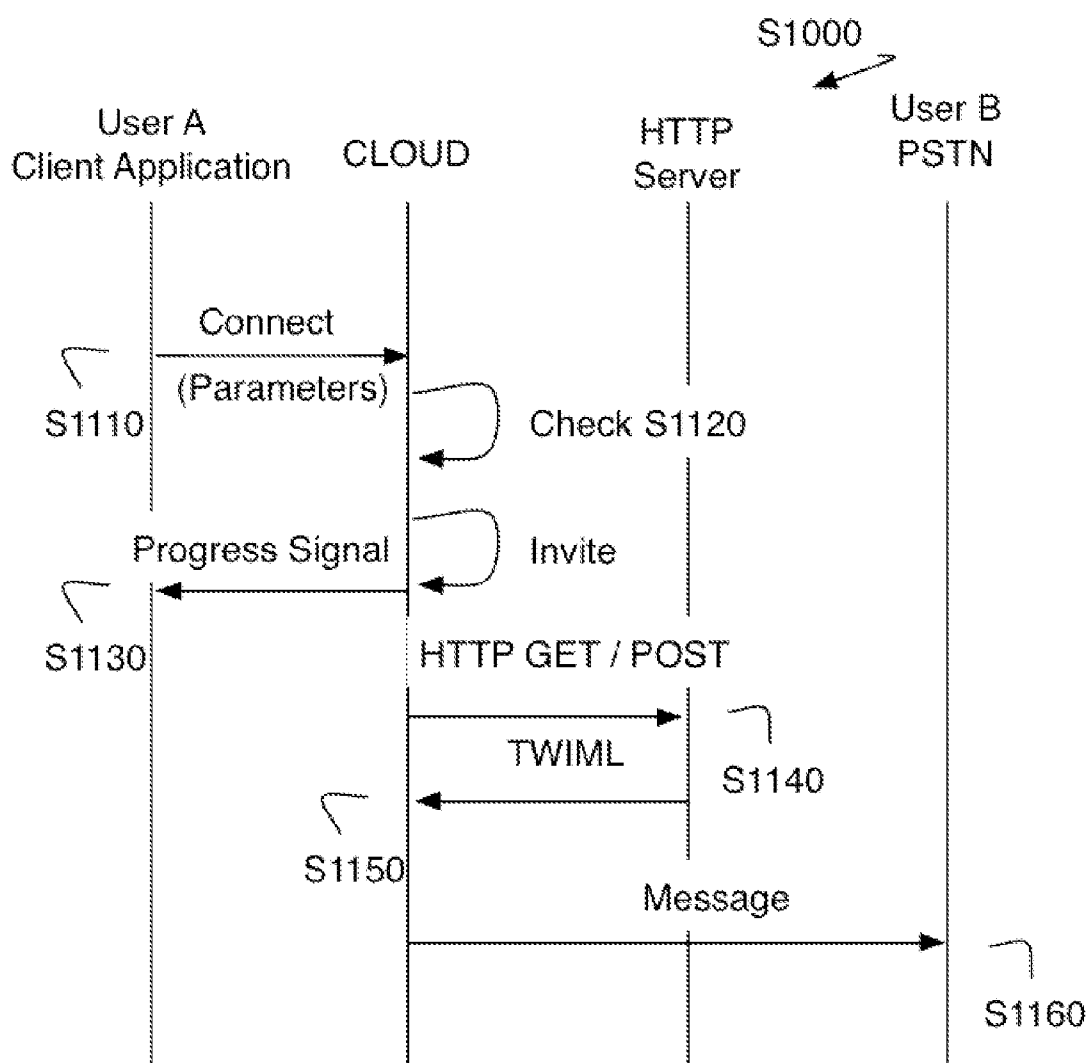


FIGURE 19

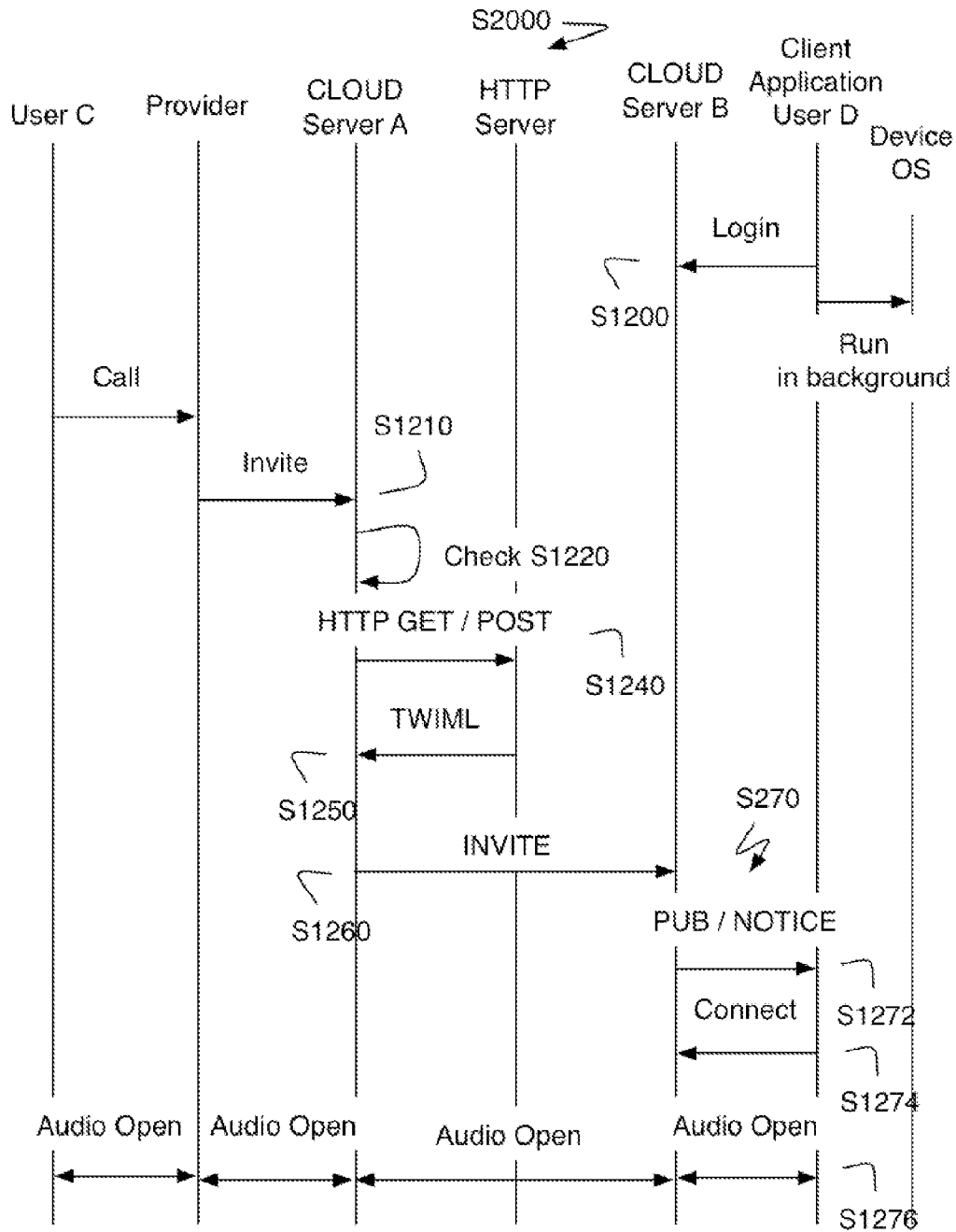


FIGURE 20

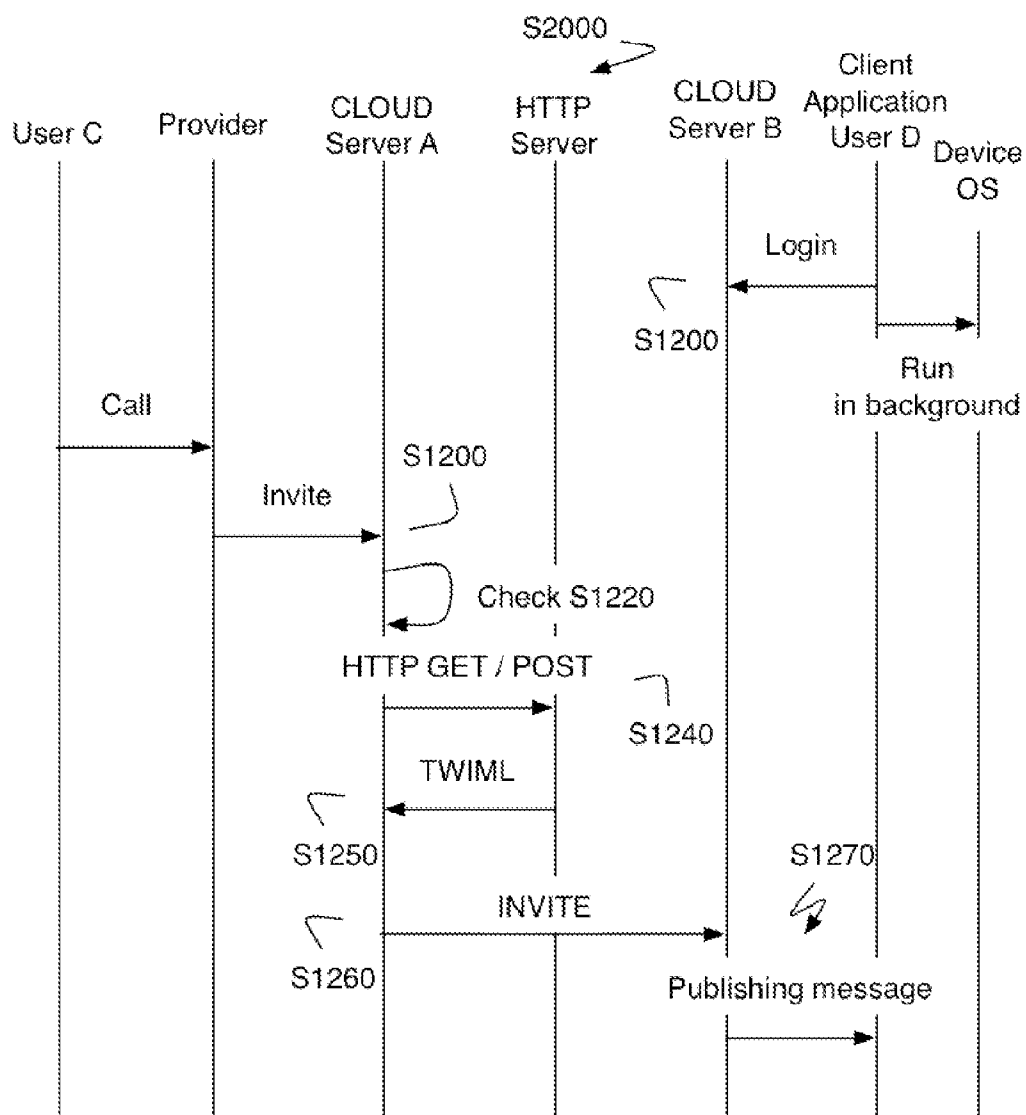


FIGURE 21

SYSTEM AND METHOD FOR COMMUNICATING WITH A CLIENT APPLICATION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation in part of prior application Ser. No. 13/478,495, filed 23 May 2012, entitled “SYSTEM AND METHOD FOR CONNECTING A COMMUNICATION TO A CLIENT” and which claims the benefit of U.S. Provisional Patent Application Ser. No. 61/489,189 entitled “System and Method for Connecting a Call to a Client” and filed on 23 May 2011; and U.S. Provisional Patent Application Ser. No. 61/500,549 entitled “System and Method for Connecting a Call to a Client” and filed on 23 Jun. 2011, which are all incorporated in their entirety by this reference.

[0002] This application claims the benefit of U.S. Provisional Application No. 61/537,217, entitled “SYSTEM AND METHOD FOR CONNECTING A CALL TO A CLIENT”, filed 21 Sep. 2011, which is incorporated in its entirety by this reference.

TECHNICAL FIELD

[0003] This invention relates generally to the telephony field, and more specifically to a new and useful system and method for communication with a client application in the telephony field.

BACKGROUND

[0004] In recent years, telephony applications and Voice over Internet Protocol (VoIP) have found applications in numerous settings. Such technology has enabled clients to establish communication to outside devices such as phones or applications. However, the nature of most network configurations requires usage of a traditional mobile telephony or landline telephony platform. This seriously limits the type of applications of internet-based telephony. Thus, there is a need in the telephony field to create a new and useful system and method for communication with a client application, such that the client can utilize an Internet browser and/or native application on a computer and/or mobile device for making and/or receiving phone communications. This invention provides such a new and useful system and method.

BRIEF DESCRIPTION OF THE FIGURES

[0005] FIG. 1 is a flowchart depicting a first method for connecting a communication to a client in accordance with a preferred embodiment;

[0006] FIG. 2 is schematic representations of preferred method for connecting a communication to a client in accordance with a preferred embodiment;

[0007] FIG. 3 is a flowchart depicting a variation of the first preferred method for connecting a communication to a client;

[0008] FIG. 4 is a flowchart depicting another variation of the first preferred method for connecting a communication to a client;

[0009] FIG. 5 is a flowchart depicting another variation of the first preferred method for connecting a communication to a client;

[0010] FIG. 6 is schematic representations of preferred methods for connecting a communication to a client in accordance with a preferred embodiment;

[0011] FIG. 7 is a flowchart depicting another variation of the first preferred method for connecting a communication to a client;

[0012] FIG. 8 is a flowchart depicting another variation of the first preferred method for connecting a communication to a client;

[0013] FIG. 9 is a flowchart depicting a second method for connecting a communication to a client in accordance with a preferred embodiment;

[0014] FIG. 10 is a schematic representation depicting a variation of the second preferred method for connecting a communication a communication to a client;

[0015] FIG. 11 is a schematic representation depicting another variation of the second preferred method for connecting a communication a communication to a client;

[0016] FIG. 12 is a flowchart depicting a variation of the second preferred method for connecting a communication to a client;

[0017] FIG. 13 is a flowchart depicting another variation of the second preferred method for connecting a communication to a client;

[0018] FIG. 14 is a schematic block diagram of a system for connecting a communication to a client in accordance with a preferred embodiment;

[0019] FIG. 15 is a schematic block diagram of another system for connecting a communication to a client in accordance with a preferred embodiment;

[0020] FIG. 16 is a schematic block diagram of a system for connecting a call in accordance with a preferred embodiment of the present invention;

[0021] FIG. 17 is a schematic block diagram of a mobile architecture operable in the system for connecting a call of the preferred embodiment of the present invention;

[0022] FIG. 18 is a schematic diagram of a method for making an outgoing media channel communication from a client application in accordance with a preferred embodiment of the present invention;

[0023] FIG. 19 is a schematic diagram of a method for sending an outgoing message from a client application in accordance with a preferred embodiment of the present invention;

[0024] FIG. 20 is a schematic diagram of a method for establishing a media channel communication with a client application in accordance with a preferred embodiment of the present invention; and

[0025] FIG. 21 is a schematic diagram of a method for sending a message communication to a client application in accordance with a preferred embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0026] The following description of the preferred embodiments of the invention is not intended to limit the invention to these preferred embodiments, but rather to enable any person skilled in the art to make and use this invention.

1. Method for Connecting a Client

[0027] As shown in FIGS. 1 and 2, a first preferred method for connecting a communication to a client of a preferred embodiment can include: at a system bridge, establishing a client subscription connection with a client device in block S100; receiving an incoming communication request at the

system bridge in block S102, publishing an incoming communication notification from the system bridge to the client device in block S104; receiving a client communication at the system bridge in block S106; and merging the incoming communication request into the client communication at the system bridge in block S108. The first method preferably functions to connect incoming communications to a client utilizing a subscription connection that prompts a client to call out. The first method preferably creates a substantially persistent channel of communication such that a client can actively establish a connection when an incoming message arrives, which in turn preferably enables web services, mobile devices, and platforms that want to implement features for incoming communications to circumvent incoming communication issues caused by firewalls and routers. In one alternative implementation of the first preferred method, client communication is preferably initiated by a client as opposed to a client directly responding to a received incoming call.

[0028] Preferably, the first preferred method can employ authentication and/or authorization security measures that function to secure the communication channels. An authentication layer preferably prevents malicious parties from posing as a client and/or incoming call. In one embodiment, an application (web or native) may facilitate the use of a token to authenticate a client connecting to an incoming call. The first preferred method is preferably used within internet-telephony platform, but may alternatively be used for any suitable applications such as Internet messaging or real-time applications that may need to establish an incoming communication channel. The method can be configured and/or adapted to function for any suitable type of communication, including telephony-based voice calls, Internet based voice calls, video calls, video streams, video sessions, screen sharing, screen sharing streams, screen sharing sessions, SMS messaging, IP messaging, alternative messaging, or any suitable form of communication. The term call should be understood to include any suitable application, and any suitable form of incoming communication to a client may be received and merged with the client through this method, such as video, screen sharing, chat, or text messaging.

[0029] As shown in FIG. 1, the preferred method can include block S100, which recites at the system bridge, establishing a client subscription connection with a client device. Block S100 preferably functions to create a substantially persistent connection between the client and the system bridge. The client subscription connection is preferably a publication/subscription Internet communication channel that can be used to push incoming call notifications to the client. The subscription channel can include one or more websockets, an HTTP server push, an Adobe flash socket, ajax long polling, ajax multipart streaming, a forever iframe, jsonp polling, Comet, XMPP, BOSH, or any suitable communication technology to facilitate a client subscription. The subscription connection is preferably persistently maintained. The client preferably establishes a client subscription connection by initially registering a subscription channel and then subscribing to that channel. The channel is preferably subscribed to using a method substantially similar to method for connecting a client to an application described below where the use of a token is used. Alternatively, any suitable technique to subscribe may be used. Once registered, the system bridge will preferably publish a notification to that subscription. The subscription connection is preferably established between a client and a system bridge through a

pubsub system, but any suitable subscription connection may be used. A client identifier is preferably created which may be used for the subscription channel but additionally as a handle or mapping for addressing incoming calls. The client identifier is preferably specified by the client, but may alternatively be an automatically assigned client identifier. The client identifier preferably includes a client ID, which may be signed with a shared secret. The client identifier may include various scopes that may take form as subdirectories or hierarchical layers of contexts. For example, one name space of "foo" may have a subdirectory of "foo/mobile" and a second subdirectory of "foo/web". In this example, "foo/mobile" may be registered for a mobile device client and "foo/web" may be registered for a desktop browser session. Incoming calls can preferably address these devices individually by using "foo/mobile" or "foo/web". Additionally, an incoming call may be simultaneously directed at both clients by using "foo/*". Additional attributes can be assigned to the name-spaced endpoint or to different scopes of the name-spaced endpoint. Additionally the client identifiers may be used to broadcast to multiple clients. Clients can preferably subscribe to client identifiers. For example a plurality of clients may subscribe to "foo" and each receive a request sent to "foo".

[0030] Block S102, which recites receiving an incoming communication request at the system bridge, functions to accept a communication from an outside entity directed to a client. The incoming call may have any suitable source. A cloud-based communication router preferably initially receives/initiates the incoming communication. The cloud-based communication router is preferably a call router call router of a telephony platform such as one substantially similar to the one described in published U.S. Patent Application No. 2009/0252159, titled "SYSTEM AND METHOD FOR PROCESSING TELEPHONY SESSIONS", which is hereby incorporated in its entirety by this reference, but the cloud-based communication router may alternatively be any suitable communication router. Alternatively, the system bridge may be integrated into the cloud-based router or call router architecture or alternatively into any suitable communication framework. The incoming call preferably specifies an identifier, and more preferably, the incoming call specifies a name-spaced client identifier. The identifier preferably corresponds to a client or more preferably a subscription channel. In one variation, the client may vary depending on what user devices are active, and thus the identifier is preferably not specific to a particular client device (e.g., addressing to "foo/*"). The identifier is preferably unique to a user, account, session, or any suitable entity. Preferably using the identifier, a subscription is identified and an incoming communication notification is generated for publishing in block S104. While the client is notified and calls out, the system bridge preferably puts the incoming communication into a holding-state. The holding state is preferably a temporary state where the incoming communication is received by the system bridge but the client (e.g., the callee) has not initiated an outgoing communication to connect to the caller. When the system bridge is ready to merge the incoming communication to a client, the incoming communication is pulled from the holding-state.

[0031] Block S104, which recites publishing an incoming communication notification from the system bridge to the client device, functions to push a message to the client through the subscription channel. The client subscription channel preferably facilitates simple notification because the connection is substantially persistent and the outside entity is

not required to independently establish the connection. There may additionally be a plurality of clients subscribed to the subscription channel, and the incoming communication notification may be published a plurality of clients. The incoming communication notification is preferably sufficient to initiate a client communication sent to the system bridge by the client. Alternatively, the communication notification may include additional parameters such as keys to authenticate the message is from the system bridge, call URI's to direct a call out, caller metadata, and/or any suitable parameter. The system bridge may additionally pass opaque data (from the perspective of the client) to the client. This data is preferably passed along when the client connects back in. In one variation, block S104 may include pushing a communication token from the system bridge to the client S110 as shown in FIG. 3. The passed communication token may be used to later identify which incoming communication to connect with the client communication. The communication token preferably includes a communication ID and a host (identifier for the instance) that allows the incoming communication to be uniquely identified across multiple bridge nodes. Block S110 preferably includes encrypting the communication token with a key unique to the system bridge in block S112 and as shown in FIG. 4. The variation preferably includes receiving the encrypted communication token at the system bridge from the client and decrypting the encrypted communication token at the system bridge in Block S114 as shown in FIG. 5. The passing of a communication token functions to remove the need for an external state, unnecessary because the state is preferably contained in the system bridge token. As described below, the communication token is preferably passed back to the system bridge from the client, and the communication token is then decrypted with the bridge system secret and the call (or connection) bridged as shown in FIG. 15. When there is a plurality of system bridges used in a scale infrastructure, the token preferably indicates which system bridge to pass the client communication message. If the subscription channel is encrypted and authorized, then the token may additionally function to provide security for incoming connections.

[0032] Block S106, which recites receiving a client communication at the system bridge, functions to have the client call out to the system bridge to be merged with the incoming communication. The client, upon receiving the communication notification from block S104, preferably initiates forming connections to the system bridge. The client may be any suitable device with a network connection to the system bridge. The client device may be running a native application or alternatively a web application. The call out message is preferably communicated through HTTP or HTTPS but any suitable transport layer may alternatively be used. Any additional parameters from the client are preferably included in a message to the system bridge as appropriate, such as an application identifier or application data. The application identifier is preferably a name-spaced endpoint. A name spaced endpoint is preferably a context that embodies various aspects for the endpoint (e.g., a client). Name-spaced endpoints preferably include a plurality of parameters including associated phone numbers, application URI's, status callback URI's (e.g., backup URI's if communication with a client application fails), or any suitable parameter. A name-spaced endpoint may be globally or locally unique for a given user. For example, a name-spaced endpoint may be unique for all communication occurring over a platform or protocol. The name-spaced endpoint may alternatively, be unique for an

instance of an application. Allowed users, pricing models, account-application settings, and/or any suitable aspects can be stored as part of the name-spaced endpoint. For example, only particular users may be allowed to call in or use a particular scope of a name-spaced endpoint while a second scope may be publicly accessible.

[0033] Additionally or alternatively, security measures are taken to authenticate the message from the client to the system bridge as shown in FIG. 6. Block S108 preferably includes authenticating the client communication at a policy engine S116 as shown in FIG. 7. The policy engine preferably authenticates a client communication from the client prior to merging the incoming communication. The policy engine preferably authenticated the client by using a token associated with the client and analyzing a signed client communication from the client. If the client communication satisfies the authentication, a preferred embodiment preferably includes merging the incoming communication request into the client communication at the system bridge in response to client authentication at the policy engine S118 as shown in FIG. 8. If the client communication fails to satisfy the authentication, an error may be logged and any suitable error response may be taken. Preferably, the token is sent from the client. In one variation, the client has an embedded token from when an application was instantiated. In another variation, a communication can be made to an authentication application to retrieve a token. Alternatively, a client identifier is sent to a policy engine that has an authentication application authenticate the credentials of the client identifier. Similarly a SIP backend authentication may alternatively be used. The authentication may occur for registration and/or outbound communications. The token preferably includes an application ID and any additional parameters. The contents of a token are preferably signed with a secret key. A policy engine preferably completes authentication of the token before allowing a connection to the system bridge. A policy engine preferably mediates the call out requests and authenticates any tokens, and may additionally verify permissions associated with application identifiers. Any suitable alternative forms of authentication may alternatively or additionally be integrated into the method.

[0034] Block S108, which recites merging the incoming communication request into the client communication at the system bridge, functions to connect the incoming communication request to the client. Once the client sends a client communication to the system bridge, the system bridge preferably identifies the corresponding incoming communication and establishes the communication channel between the client and the outside entity of the incoming communication. The incoming communication is preferably moved out of the holding-state and added to an active channel of communication with the intended client. Upon establishing the connection, the client and the outside entity can preferably continue to use the established connection for communication. For example, a VoIP call made to the system bridge may be connected to a client device, and then a VoIP session can preferably continue to use the channel established by the preferred method. The connection may be any suitable form of connection including a number of channels that may include audio, video, messaging, screen sharing, or any suitable channel for communication.

[0035] As shown in FIG. 9, a second preferred method for connecting a client to an application of a preferred embodiment includes receiving a connection request at a policy

engine from a client, the connection request including an authentication token retrieved by the client in block S200; receiving an authentication token at the policy engine S202, authenticating the client at the policy engine by verifying the authentication token in block S204; and permitting the client to connect to an application in response to verification of the authentication token in block S206. The method functions to enable a possibly untrusted client to securely access application services and resources. A client preferably describes a web browser, application session, or any suitable outside entity. The method is preferably implemented for use with a telephony platform but may alternatively be used for any suitable communication platform. The method may be used to implement browser based telephony application such as click to call features, voice sessions, video calls, video streams, screen sharing, sending of SMS messages, alternative messaging, or any suitable application. In an exemplary application, the client is enabled to initiate and interact with a telephony application by obtaining a token to authenticate access to the application. Additionally, the method of connecting a client to an application may be applied in cooperation with the method above for a client receiving incoming communications.

[0036] Block S200, which recites receiving a connection request at a policy engine from a client, functions to receive communication from a client wishing to use an application. As described below, an authentication token is preferably directly or indirectly communicated to the policy engine. The policy engine then can preferably allow or deny access to an application by a client based on the verification of the authentication token. The connection request may be accompanied by the token as shown in FIG. 10 or the connection request may come prior to obtaining a token as shown in FIG. 11.

[0037] Block S202, which recites receiving an authentication token at the policy engine, functions to obtain an authentication token on behalf of a client. The authentication token is preferably a data package that includes application ID and/or additionally parameters. The authentication token is preferably signed. The authentication token is more preferably signed with a secret key shared by the policy engine and an accessing entity (e.g., a web application for use of the telephony application). The application ID and/or the authentication token may be sent to client, which may then use the authentication token to connect to a web application. In another variation, the client may provide an identifier that enables a policy engine to validate with an authentication application. In the first variation, as shown in FIG. 10, a web application preferably sends an authentication token to the client. The client then communicates the authentication token directly when sending a connection request. The authentication token may be embedded in the webpage or application when instantiated. Alternatively, a client may dynamically request the authentication token such as by using AJAX upon a user-triggered event. In a second variation, as shown in FIG. 11, the client may send an identifier when sending a connection request for indirectly obtaining a token for a client. In this variation, block S202 preferably includes sending a client identifier contained in the connection request from the client to an authentication application S208 and receiving the authentication token at the policy engine from the authentication application S210 as shown in FIG. 12. A policy engine preferably connects with an authentication application. The authentication application can preferably use the identifier to authenticate or deny access by the client. The authentication

application then sends a response to the policy engine preferably includes an authentication token. The authentication token received at the policy engine from the authentication application preferably enables the client to be indirectly authorized to form a connection. The authentication application is preferably a server run by the web application entity, but may alternatively be a third party authentication application. The identifier may include an account name, code, or any suitable parameters that the authentication application requires to complete authentication.

[0038] Block S204, which recites authenticating the client at the policy engine by verifying the authentication token, functions to determine if a client should be allowed or denied access to an application. In the first variation where a token is received from the client, a shared secret between the application and the policy engine may be used to authenticate the token. In the second variation, the authentication application may send the authentication token, which may be authorized in a similar manner, or the authentication application may communicate to the policy engine if the client is allowed or denied.

[0039] Block S206, which includes permitting the client to connect to an application in response to verification of the authentication token, functions to allow the client to connect to the application or to deny the client access to the application. The connection request from the client is preferably forwarded on to the application if the verification of the authentication token allows access. If the connection request is denied, a communication may be sent back to the client or any suitable response may be made. The connection in one application is preferably establishing a voice session, video session, click to call feature, starting an outbound call, a video stream, a screen sharing session, SMS/MMS messaging, IP messaging session, and/or any suitable communication application as in block S212 shown in FIG. 13. In one exemplary application, a call router of a telephony platform preferably facilitates execution of the application.

2. System for Connecting a Client

[0040] As shown in FIG. 14, a system for connecting a call to a client of a preferred embodiment preferably includes a system bridge 210, a pub/sub system 220, and optionally a policy engine 230. The system preferably functions to facilitate connecting an outside entity to a client 260. The system preferably implements the above method. The system bridge 210 preferably includes an incoming channel to receive incoming calls or video such as from a PSTN device, another client device, or any suitable source. In one embodiment, the system bridge 210 includes a communication link to a call router 250 of a telephony platform. In another embodiment, the system bridge 210 is integrated into a call router 250. In another implementation, a SIP endpoint is used in place of a call router. As shown in FIG. 15, the system bridge 210 may be a cloud environment or system bridge cluster composed of a plurality of system bridges (210a, 210b, 210n). The system bridge 210 preferably additionally includes internet channels to be accessed by a client 260. The system bridge 210 preferably connects to the pub/sub system 220, and the pub/sub system 220 preferably maintains persistent connections to clients. The pub/sub system 220 may use any suitable technology such as websockets, HTTP server push, adobe flash sockets, AJAX long polling, AJAX multipart streaming, forever iframes, jsonp polling, Comet, XMPP, BOSH, or any suitable communication technology to facilitate subscription

and publication channels. Any suitable system may be used in place of the pub/sub system such as a queuing system. The client **260** preferably includes native or web application code that cooperates with the system to establish a subscription through the pub/sub system **220** to the system bridge **210** and to send a client communication that is received by the system bridge **210**. The client preferably uses HTTP or HTTPS or any suitable communication protocol. Additionally, a policy engine **230** may be an intermediary system for the communication channel between clients and the system bridge **210**. The policy engine **230** preferably authenticates signed messages using web tokens, but may alternatively be configured for any suitable form of authentication. An authentication application server **240** preferably facilitates the distribution and/or processing of authentication tokens.

[0041] The system preferably implements the above methods in a computer-readable medium storing computer-readable instructions. The instructions are preferably executed by computer-executable components preferably integrated with the system bridge **210**, the pub/sub system **220**, and the optional policy engine **230**. The computer-readable medium may be stored on any suitable computer readable media such as RAMs, ROMs, flash memory, EEPROMs, optical devices (CD or DVD), hard drives, floppy drives, or any suitable device. The computer-executable component is preferably a processor but the instructions may alternatively or additionally be executed by any suitable dedicated hardware device.

3. System for Communicating with a Client Application

[0042] As shown in FIG. 16, a system **10** for communicating with a client application can include one or more hardware interfaces, including for example a browser **12**, a mobile device **14**, or a public switched telephone network (PSTN) line **16**. In the system **10** of the preferred embodiment, each of the hardware interfaces **12**, **14**, **16** can be connected and/or interconnected with one another through a communication platform cloud-based service **30** such as that available from the assignee of the present application. Preferably, the cloud service **30** is configured according to an Application Programming Interface (API). As used herein, the term API should be understood to mean any combination of software, firmware, and/or hardware that allows two or more software applications (i.e., machine-readable instructions) to communicate with one another. An example API can be configured as a telephony platform such as one substantially similar to the one described in published U.S. Patent Application No. 2009/0252159, titled "SYSTEM AND METHOD FOR PROCESSING TELEPHONY SESSIONS", assigned to the assignee of the present application, and hereby incorporated in its entirety by this reference.

[0043] As shown in FIG. 16, the cloud system **30** functions to communicate with one or more client HTTP servers **18**, **20**, **22**. The cloud system **30** and HTTP server **18**, **20**, **22** communications are preferably RESTful in nature in both directions. RESTful is understood in this document to describe a Representational State Transfer architecture as is known in the art. The RESTful HTTP requests are preferably stateless, thus each message communicated from the call router to the application server preferably contains all necessary information for operation of the application server and response generation of the application server. Hardware communications elements such as routers and servers preferably do not need to remember or store previous communications to be aware of the state. Documents, media, and application state are preferably viewed as addressable resources, combined with data

provide to the resource via request parameter, such as HTTP GET or HTTP POST parameters, or request body contents. Such request data can include an updated representation of the call resource, or other call state data generated as a result of call router operation, such as digits pressed on the keypad or audio recordings generated.

[0044] State information included with each request can include a unique call identifier, call status data such as whether the call is in-progress or completed, the caller ID of the caller, the phone number called, geographic data about the callers, and/or any suitable data. Alternatively, a varying level of a RESTful communication (statelessness) can be used, such as by using cookies, session tracking, or any suitable devices to simulate a normal website visitor model. Preferably, data sent with each request can fully enable the application server to determine the next state of the call to execute. RESTfulness preferably does not preclude using an external data source, such as a database, to lookup additional data to log call meta data, or determine application logic.

[0045] The hardware interfaces **12**, **14**, **16** function to interface with the human element of the communication through voice, video, short message service (SMS), multimedia messaging service (MMS), proprietary or alternative IP messaging protocols, or other communication means and/or protocols. As an example, the browser **12** can include any web-enabled browser that is capable of receiving and/or transmitting voice signals such as through real time messaging protocol (RTMP) or other suitable TCP-type communication protocols. Alternatively, the browser **12** can include any plug-in enabled browser that is configured for session initiated protocol (SIP) communications that can be transmitted through TCP, UDP and/or other suitable protocols. Similarly, the mobile device **14** can include a SIP module or other suitable communication protocol. Preferably, each of the browser **12** and the mobile device **14** can be connected and/or connectable to the cloud service **30** through any suitable type of network, including but not limited to, wide area networks, local area networks, mobile telephony networks, or any suitable combination thereof. As an example, many current mobile devices **14**, such as tablet computers, are connectable to the Internet through both WiFi and mobile telephony (GSM and CDMA) networks.

[0046] The mobile device **14** of the system **10** of the preferred embodiment can include a series of modules relating to its communication function. As shown in FIG. 17, the mobile device **14** of the preferred embodiment can include an operating system module **40**, an audio engine **42**, and a cloud interface module **46**. The operating system module **40** of the mobile device **14** of the preferred embodiment functions to control the general functionality of the mobile device **14**, including but not limited to its communication capabilities. The audio engine **42** of the mobile device **14** of the preferred embodiment functions to control a majority of voice communication functions, including interfacing with any device hardware **44** such as speakers and microphones. In particular, the audio engine **42** can function to initiate and control communication sessions including voice communication sessions using Internet protocol (rather than a typical telephony communication protocol typically used in a mobile device).

[0047] The mobile device **14** of the preferred embodiment can further include a cloud interface module **46** that functions to cooperate with the audio engine **42** to communicate with the cloud service **30** and communicate with other hardware interfaces **12**, **14**, **16**. The cloud interface module **46** can be

configured within a browser function of the mobile device **14**, or alternatively as a native application or set of machine executable instructions residing permanently or quasi-permanently in a memory module (not shown) of the mobile device **14**. Preferably, the cloud interface module **46** can be designed according to the RESTful principles described above.

[0048] In one variation of the mobile device **14** of the preferred embodiment, the cloud interface module **46** can further include one or more security and/or resource management tools described in further detail below. Examples of security and/or resource management tools can include an authorization token, which can be defined as an electronic listing of possible privileges for its bearer. A developer or other suitable account holder preferably obtains an account token from the provider of the cloud service **30** and then signs the account token to generate an authorization token. The authorization token can be used to authorize actions when included in communication requests, client registration, client presence registration, and/or any suitable use. The cloud service or any suitable service can preferably verify the authorization token by validating the signature of the authorization token against the account key. Thus enabling possibly untrusted client applications to prove that they are authorized to perform particular actions. Example privileges conferred by a token can include, but are not limited to, a user's abilities to make and/or receive calls, an identification of the application and/or user, and one or more operational parameters that can further define a scope of functions and/or processes performable by the cloud interface module **46**. In one exemplary variation, an authorization token permits a communication request to send one or more communications. In another exemplary variation, an authorization token can permit a communication request to establish a communication session that supports communications selected from audio, video, chat, and/or other suitable communication mediums. In another variation an authorization token can permit presence information to be provided to other devices or to be obtained from other devices or a contact list. In one alternative embodiment, the token can be transmittable in its entirety in the clear. In another alternative embodiment, some or all aspects of the token are encrypted during transmission and are partially or entirely unknown to the user of the mobile device **14**. The cloud service **30** may be substantially agnostic relative to the contents of the token insofar as the contents of the token can be directed entirely to the HTTP server or to a authorization server indicating if the authorization token is allowed. Alternatively, the cloud service **30** can function in part as a gatekeeper and perform one or more security and/or authenticity checks prior to, during, and/or following a communication session.

[0049] The system preferably implements the below methods in a computer-readable medium storing computer-readable instructions. The instructions are preferably executed by computer-executable components preferably integrated with the mobile devices **12**, **14**, **16**, the cloud system **30**, and the HTTP servers **18**, **20**, **22**. The computer-readable medium can be stored on any suitable computer readable media such as RAMs, ROMs, flash memory, EEPROMs, optical devices (CD or DVD), hard drives, floppy drives, or any suitable device. The computer-executable component is preferably a processor but the instructions can alternatively or additionally be executed by any suitable dedicated hardware device.

4. Method for Connecting a Call to a Client

[0050] As shown in FIGS. **18** and **19**, a method **S1000** of a first preferred embodiment for making outgoing communications from a client application can include receiving a connection request from a first client application **S1110**; verifying at least one parameter of the communication request **S1120**, transmitting a progress signal to the first client application **S1130**, retrieving communication instructions from a server according to the incoming communication request **S1140**, identifying an instruction to make a call to a communication destination **S1150**, and establishing communication with the communication destination **S1160**. The method of the first preferred embodiment enables a client application to establish communication. The medium of communication is preferably a telephony based communication but may alternatively be any suitable communication medium such as video session, screen sharing, text messaging, and/or multimedia messaging. The method preferably establishes a media channel (i.e., communication session) such as an open voice channel between the first client application and at least a second endpoint. The method may alternatively perform one or more communication requests such as transmitting a message from the first client application to at least a second endpoint. As shown in FIG. **18**, the method may be used for connecting a USER A that is using a client application to a USER B that is using a PSTN device. As shown in FIG. **19**, the method may alternatively, be used for communicating a media message such as an SMS or MMS message from a USER A on a client application to a USER B using a PSTN device. The method as described herein, USER A can include a mobile device **14** of the type described above. USER B may be a PSTN connected device or a mobile device **14** of the type described above.

[0051] Step **S1110**, which includes receiving a connection request from a first client application, functions to connect a USER A of client application to the CLOUD. The CLOUD is preferably a communication platform and can include one or more functions and/or hardware systems described above with reference to cloud service **30**. Preferably, the CLOUD can include one or more separate servers and/or services configured for trafficking and connecting communications between the users in accordance with the principles of the API described above. The connection Step **S1110** functions to initiate communications over a predetermined protocol, such as SIP, RTMP and the like, between a mobile device and the CLOUD. The predetermined protocol is preferably an internet protocol utilizing an internet protocol such as TCP, UDP, and/or HTTP, but any suitable protocol may be used.

[0052] In one variation of the method of the preferred embodiment, the connecting step Sino can further include transmitting one or more permissions, parameters or tokens with the connection attempt. Preferably, the parameters can include one or more of a cloud system token (or security credential token) that can contain authentication credentials of USER A, a bridge token (which can have a null value in outgoing calls such as that shown in FIG. **18**), and/or one or more user-defined parameters. The authentication token is preferably associated with security credentials that can authenticate and/or authorize a communication request to perform particular actions. The authorization token can preferably be used to verify if a bearer can perform a particular action and in some variations which operations to limit/permit. A communication request can be permitted to execute one or more communications. Any suitable restriction on the

communications may additionally be associated with the authorization token such as number of communications, timing of communications, communication endpoints, and communication mediums. As described above, the authorization token is preferably created for a client application by a developer signing against their account key or some other suitable identifier. At the communication platform or at a service communicatively coupled to the communication platform, the authorization token can be verified against an account key provided to a developer or provider of a system using the client application. As an example, the user defined parameters can include additional constraints or conditions on the use of the system, such as for example particular users to whom calls can be placed, particular abilities of the token holder, identifiers of any applications or functions in use, or identifiers of any required recipients of communications, e.g., customer service or sales representatives. Preferably, any or all of the permissions can be consolidated into a single entity, such as the cloud system token; and any or all of the permissions can be sealed and kept invisible from USER A and partially or entirely transparent to any intermediary servers useable in the communication process.

[0053] Step S1120, which includes verifying at least one parameter of the communication request, functions to check if the connection request should be completed. Preferably, the connection request is sent with a security credential token for a user of the first client application. The security credential token is preferably checked at the CLOUD to ensure the client application is allowed to make a connection request on behalf of the user. Permissions can be stored in association with an authorization token, and the authorization token preferably received in Sino is used to retrieve the associated permissions. The permissions are then used to determine if the communication request is allowed. Additionally or alternatively, the user defined parameters may be checked to ensure they satisfy any requirements or meet expectations. Additionally, step S1120 can preferably function to check other diagnostic functions of the communication system and identify selected ports and/or communication channels for the impending communications. For example, the CLOUD may check to ensure that the connection request is not being made when the CLOUD is an overloaded state, and that the connection request satisfies an overload policy.

[0054] Step S1130, which includes transmitting a progress signal to the first client application, functions to transmit a ringtone or confirmation message to USER A. The progress signal may take different forms depending on the communication medium. For voice or video calls, the progress signal is preferably a ringtone played for the user of the client application. In an asynchronous messaging application, the progress signal may indicate the delivery status of the message (e.g., sending, delivered, read). In some variations, the progress signal is a confirmation transmission to confirm the connection for the client application. The confirmation transmission is preferably transparent to a user and is only an acknowledgment by the CLOUD for the client application. The progress signal is preferably transmitted in response to an invite transmission within one or more of the CLOUD service. Preferably, the invite can include an application to which connection is sought, the account identification of the recipient, as well as the user defined parameters that were previously received in the CLOUD. Additionally, the invite can include a caller-identification or user identification of USER A. The invite is preferably a SIP invite but may be any

suitable invite. In response to a smooth traffic flow of the invite within the one or more CLOUD servers, the CLOUD can preferably transmit the progress signal back to the mobile device via the SIP connection.

[0055] Step S1140, which includes retrieving communication instructions from a server according to the incoming communication request, functions to fetch instruction markup determining communication logic. Step S1140 preferably includes transmitting an HTTP request (such as a GET or POST) to the HTTP Server associated with the user account of the client application. Preferably the user account has a URL configured as a resource location of the communication instructions. Step S1140 can preferably be performed simultaneously or substantially simultaneously with one or more variations of step S1130, described above. In another variation of the method of the preferred embodiment, the HTTP request can include any or all of the permissions described above. Preferably, the HTTP request includes at least the user-defined parameters, which can be transmitted and received in an encrypted format. Alternatively, step S1140 can include verifying one or more of the user defined parameters at or by the HTTP server.

[0056] Step S1150, which includes identifying an instruction to make a call to a communication destination, functions to process the communication instructions and determine an outgoing communication destination. The communication instructions retrieved in Step S1140 are preferably transmitted as a markup language format message to the CLOUD from the HTTP Server. In one variation of the method of the preferred embodiment, the markup language format message can include one of an HTML message, an XHTML message, an SGML message, or any other suitable markup language or variation or version thereof. In another variation of the method of the preferred embodiment, the markup language format message can include a TWiML message for use with a cloud system of the assignee of the present application. The communication instructions may include any suitable logic. In making an outgoing call, the communication instructions preferably include an instruction to dial or communicate with a communication destination.

[0057] Step S1160, which includes establishing communication with the communication destination, functions to complete communication with the desired endpoint. Establishing communication can establish communication from the source of the communication request (e.g., USER A) and at least one party/endpoint. A plurality of destination endpoints/parties may alternatively be included in the established communication. In a first variation, the communication is preferably an open media channel (e.g., voice/audio, video, screen sharing or any suitable communication session) between the client application and a device of the communication destination. This variation preferably includes inviting the communication destination; and upon receiving acknowledgment of invite acceptance, opening a media channel between the communication destination and the first client application. The invite is preferably a SIP invite but may alternatively be any suitable invite to a media channel. This may be used to establish communication with a PSTN device and/or a second client application as described more below. In one example, a USER A is attempting to contact a PSTN USER B. Accordingly, the communication instructions can indicate that the recipient is a PSTN user, and therefore the CLOUD will preferably create an appropriate SIP invite and transmit it to an appropriate provider. As used herein, the PSTN network

and/or service provider can include both mobile and landline-based telephony networks. A response is preferably received from the provider at the CLOUD service, wherein the response can include an appropriate HTTP status code such as 200 OK. Following confirmation and cross-confirmation between one or more components of the CLOUD service and the service provider, the method can open a media channel between USER A and USER B. As shown in FIG. 18, the media channel bypasses and/or is transparent to the HTTP server, but can be transmitted through one or more components of the CLOUD service. It should be understood that the media channel can also include other types of media, including at least video, screen sharing, SMS, MMS, or any other suitable communication media.

[0058] In a second variation of Step S1160, establishing a communication with the communication destination can include performing one or more specific communication requests. Performing the communication request preferably includes sending an asynchronous message to the communication destination as shown in FIG. 19. An asynchronous message can be sent using protocols including SMS, MMS, IP-based messaging protocols, proprietary messaging protocols, and/or any suitable messaging protocol.

[0059] In another variation of the method of the preferred embodiment, USER A can be communicating on a mobile device having a native application, software program or set of machine executable instructions that functions to perform one or more of the steps outlined above. In one example configuration, the connect, check and invite steps S1110, S1120, S1130 or significant portions thereof can all be performed by such a native application configured with an operating system of the mobile device. As an example, certain invite and authentication procedures, including for example the cloud system token and user defined parameters associated with the application native to the user's mobile device. In such an example configuration, the mobile device can communicate directly with the HTTP server via one or more requests and/or messages of the type described above.

[0060] As shown in FIGS. 20 and 21, a method S2000 of a second preferred embodiment for receiving communication at a client application may include at a first server of a communication platform, receiving an executed login of a user of a first client application S1200, at a second server of the communication platform, accepting an incoming communication request S1210, retrieving communication instructions from a server according to the incoming communication request S1240, identifying an instruction to make a call to a communication destination of the user of the first client application S1250, forwarding the communication request to the first server of the communication platform S1260, and at the first server of the communication platform, establishing communication with the first client application S1270. The method functions to receive calls at a client application through the systems described above. The method is preferably employed to enable software applications or machine executable instructions running natively on a mobile device to interface with a telephony or communication platform. The method may be used such that a PSTN device or any suitable communication capable device may contact the client application. The method may be used for voice, video, SMS, MMS, IP based messaging, proprietary messaging, and/or any suitable communication medium. The method may be used in combination with the method above for making outgoing communications from a client application such that a

first client application may contact a client application. The pairs of steps S1110 and S1210, S1120 and S1220, S1140 and S1240, and S1150 and S1250 may in some variations be substantially similar to their corresponding step.

[0061] Step S1200, which includes at a first server of a communication platform, receiving an executed login of a user of a first client application, functions to establish a connection of a user to a communication platform/CLOUD system. The user preferably uses a mobile device having a native application (with a client application) of the type described above, but any suitable device with a client application may be used. Preferably, when USER D logs into his account on CLOUD B, a URL subscription <userid_sid>/USERD is established in CLOUD B such that USER D will receive any messages published to that URL. Example messages can include voice, video, SMS, MMS, email or any other suitable type of electronic communication audible or readable on a mobile device.

[0062] Step S1210, which includes at a second server of the communication platform, accepting an incoming communication request, functions to initiate a call from a caller (e.g., USER C) to the user of the first client application (e.g., USER D). As shown in FIG. 5, the call initiation can be processed at a service provider, such as for example a PSTN service provider for a mobile telephony or landline telephony network. The service provider can function to receive a telephonic communication (voice, SMS, MMS) and convert that incoming communication into an SIP request transmittable to CLOUD A. Alternatively, the caller may be initiated by device using a second client application, and the outgoing call may be established in manner substantially similar to that of the method for making outgoing communications from a client application. In step S1220 of the method of the preferred embodiment, CLOUD A will perform a check of the received SIP invite, which can include checking a recipient account, a voice URL for the account, an SMS or MMS URL for the account and/or any other account-specific or user specific permissions or preferences associated with the account of USER D. In the variation where a call is initiated from a second client application, the method S2000 may include transmitting a progress signal to the second client application, which is substantially similar to Step S1130.

[0063] Steps S1240 and S1250, which include retrieving communication instructions from a server according to the incoming communication request and identifying an instruction to make a call to a communication destination of the user of the first client application, function to process the application logic of a communication platform application. Step S1240 of the method of the preferred embodiment preferably includes transmitting an HTTP request from CLOUD A to the users HTTP Server in response to a successful check of the recipient's credentials. As an example, once CLOUD A identifies a voice URL associated with USER D's account, CLOUD A can transmit an HTTP GET/POST request to USER D's HTTP Server. Step S1250 of the method of the preferred embodiment preferably processes the communication instructions and determine an outgoing communication destination corresponding to the user of the first client application. A markup language format message is preferably received at CLOUD A from the HTTP Server. In one variation of the method of the preferred embodiment, the markup language format message can include one of an HTML message, an XHTML message, an SGML message, or any other suitable markup language or variation or version thereof. In

another variation of the method of the preferred embodiment, the markup language format message can include a TWiML message for use with a cloud system of the assignee of the present application.

[0064] Step S1260, which includes forwarding the communication request to the first server of the communication platform functions to communicate the incoming communication request to a portion of the CLOUD that manages the logged in user. For media channels communication requests such as voice or video calls, an invite request is transmitted from CLOUD A to CLOUD B in response to the receipt of the markup language format message from USER D's HTTP Server. Preferably, the invite request contains at least an account identification for USER D and a client name (USER D) associated with the account during the login step S1200. The invite request is preferably a SIP invite as is known in the art. In the case of an asynchronous messaging, the text, and/or media message may be contained in the communication message sent to the server of CLOUD B as shown in FIG. 20. The first server of the communication platform may additionally or alternatively include or function as a subscription manager of the communication platform. The communication request may subsequently be forwarded to an appropriate server to manage the subscription or login session of the user of the first client application.

[0065] Step S1270, which includes at the first server of the communication platform, establishing communication with the first client application; functions to complete the communication request initiated in Step S1200. Step S1270 preferably completes the communication request between the entity of the incoming communication request (e.g., USER C) and the user of the first client application (e.g., USER D). Establishing communication with the first client application preferably includes publishing a notification to the first client application S1272. In one variation of the method of the preferred embodiment, CLOUD B can publish an invite message to USER D's URL while simultaneously causing the caller, USER C, to hear a ringtone. In Step S1272 CLOUD B can receive the publication of the invite message and broadcast it to all registered listeners, i.e., those users associated with the account established in step S1200. Establishing communication with the first client application may include varying steps depending on if the method is employed for establishing a media channel between two parties or for sending a message from one entity to a user of the first client application.

[0066] Establishing communication for a media channel communication request can additionally include receiving a connection acceptance by the first client application S1274, and opening a media channel with the first client application upon connection S1276. If USER D agrees to accept the incoming call, then he will transmit a connection acceptance to CLOUD B as shown in step S1274. The acceptance can include the transmission of one or more permissions with the connection acceptance. Preferably, the permissions can include one or more of a cloud system token containing the security credentials of USER D (which can have a null value since the call is being received in FIG. 20), a bridge token (which might have been copied from the invitation from CLOUD B), and one or more user defined parameters. In another variation of the method of the preferred embodiment, CLOUD B step S1274 can also include checking the one or more permissions at CLOUD B, which can include for example checking if there are a predetermined number of

parameters (even though one or more can be null). Alternatively, step S1274 can further include unpacking the bridge token to identify an IP address associated with one or more CLOUD A or CLOUD B servers to ensure that communications are properly routed and merged through the appropriate portion of CLOUD A or CLOUD B, i.e., a bridge portion of one of the respective CLOUD services. In another variation of the method or the preferred embodiment, step S1274 can include bridging the call between USER C and USER D in response to a bridge token associated with the connection request from USER D.

[0067] In another variation of the method of the preferred embodiment, step S1274 can recite responding from the provider to the CLOUD service, wherein the response can include an appropriate HTTP status code such as 200 OK. Following confirmation and cross-confirmation between one or more components of the CLOUD service and the service provider, step S1216 of the method of the preferred embodiment recites opening an audio channel between USER C and USER D. As shown in FIG. 20, the audio channel bypasses and/or is transparent to the HTTP server, but can be transmitted through one or more components of the CLOUD A and CLOUD B components.

[0068] In another variation of the method of the preferred embodiment, USER D can be communicating on a mobile device having a native application, software program or set of machine executable instructions that functions to perform one or more of the steps outlined above. In one example configuration, the login, pub/notice and connect steps S1200, S1272, S1274 or significant portions thereof can all be performed by such a native application configured with an operating system of the mobile device. As an example, certain invite and authentication procedures, including for example the cloud system token and user defined parameters associated with the application native to the user's mobile device. In such an example configuration, the mobile device can communicate directly with the HTTP server via one or more requests and/or messages of the type described above.

[0069] As shown in FIGS. 20 and 21, the cloud services can be broken in to two or more segments, including for example CLOUD A and CLOUD B. Those of skill the art will appreciate that these designations are for illustrative and descriptive ease only, and that CLOUD A and CLOUD B can include a single computer/server having multiple functions or multiple discrete computers/servers having unique functions, depending upon the particular configuration sought by the user. The method of the preferred embodiment can be configured partially or entirely in a computer-readable medium storing computer-readable instructions. The instructions are preferably executed by computer-executable components preferably integrated with a mobile devices, one or more portions of the cloud system and the HTTP servers. The computer-readable medium can be stored on any suitable computer readable media such as RAMs, ROMs, flash memory, EEPROMs, optical devices (CD or DVD), hard drives, floppy drives, or any suitable device. The computer-executable component is preferably a processor but the instructions can alternatively or additionally be executed by any suitable dedicated hardware device.

[0070] As a person skilled in the art will recognize from the previous detailed description and from the figures and claims, modifications and changes can be made to the preferred embodiments of the invention without departing from the scope of this invention defined in the following claims.

We claim:

1. A method for making outgoing communications comprising:

receiving a connection request from a first client application, wherein the connection request contains at least an authorization token;
 verifying at least the authorization token;
 retrieving communication instructions from a server according to the incoming communication request;
 identifying an instruction to communicate with a communication destination; and
 establishing communication with the communication destination.

2. The method of claim 1, wherein verifying the authorization token includes permitting the connection request if the authorization token is successfully verified and denying the connection request if the authorization token is unsuccessfully verified.

3. The method of claim 2, further comprising providing an account key; and wherein verifying the authorization token includes validating a signature of the authorization token against the account key.

4. The method of claim 2, wherein establishing a communication includes inviting the communication destination; and upon receiving acknowledgement of invite acceptance, opening a communication channel between the communication destination and the first client application.

5. The method of claim 4, wherein the opened communication channel is a voice channel.

6. The method of claim 4, wherein the opened communication channel is a video channel.

7. The method of claim 4, further comprising transmitting a progress signal to the first client application while establishing communication with the communication destination.

8. The method of claim 2, wherein establishing a communication includes sending an asynchronous media message.

9. The method of claim 8, wherein the asynchronous media message is a text message.

10. The method of claim 8, wherein the asynchronous media message is a multimedia message.

11. The method of claim 2, further comprising at the first client application on a mobile device, passing the authorization key with a communication request that is received by a communication platform.

12. The method of claim 2, wherein receiving a connection request from a first client application and verifying at least the authorization token are performed at a distributed computing communication platform.

13. The method of claim 2, further comprising at a server of a communication platform receiving a second authorization token of a second client application; verifying the second authorization token; and wherein the communication destination is associated with the second client application and communication is established with the second client application.

14. The method of claim 13, wherein establishing communication includes publishing an invite request to the second client application from the server of the communication plat-

form; and upon receiving acknowledgement of invite acceptance by the second client application, opening a media channel between the second client application and the first client application.

15. The method of claim 13, wherein the invite message includes a media message; and establishing communication further includes publishing the media message to the second client application.

16. The method of claim 2, wherein identifying an instruction to make a communication to a communication destination includes identifying an instruction to make a communication to a plurality of communication destinations; and establishing communication includes establishing communication to the plurality of communication destinations.

17. The method of claim 2, wherein retrieving communication instructions from a server is retrieved making an HTTP request.

18. A method for receiving communications at a client application comprising:

at a communication platform, receiving an authorization token of a first client application;
 verifying at least one permission associated with the authorization token;

at a first server of the communication platform, accepting an incoming communication request;

retrieving communication instructions from a server according to the incoming communication request;

identifying an instruction to communicate with a communication destination of the first client application; and
 establishing communication with the first client application.

19. The method of claim 18, further comprising establishing a subscription for the first client application upon verifying the at least one permission associated with the authorization token of the first client application.

20. The method of claim 19, wherein establishing a communication with the first client application includes publishing a notification to the first client application.

21. The method of claim 20, wherein the published notification includes a media message to deliver to the first client application.

22. The method of claim 20, wherein the published notification is an invite request; and upon receiving acknowledgement of invite acceptance by the first client application, opening a media channel with the first client application.

23. The method of claim 22, wherein the opened media channel is a voice channel.

24. The method of claim 22, wherein the opened media channel is a video channel.

25. The method of claim 22, further comprising forwarding the communication request of the first server to a communication bridge; wherein the published notification includes a bridge token; and the received acknowledgement of invite acceptance also includes the bridge token; and further comprising bridging the media channel in the communication bridge according to the bridge token.

* * * * *