

Exhibit D-1

Invalidity Chart for U.S. Patent Application Pub. No. 2012/0016932 (“de Castro 932”) over U.S. Patent No. 9,998,607

U.S. Patent Application Pub. No. 2012/0016932 (“de Castro 932”) was published January 19, 2012. DeCastro qualifies as prior art to U.S. Patent No. 9,998,607 (“the ‘607 Patent”) at least under AIA 35 U.S.C. § 102(a)(1). de Castro 932 discloses and/or renders obvious one or more of claims 1-3, and 5-6 of the ‘607 Patent alone and/or in combination with other references, as set forth in the chart below. To the extent de Castro 932 does not disclose one or more limitations of the claims, such limitations would have been inherent and/or it would have been obvious to combine the disclosures of de Castro 932 with the knowledge of one of ordinary skill in the art and/or with one or more other references, including as identified in Appendix D, to render the claims at-issue in the ‘607 Patent invalid.

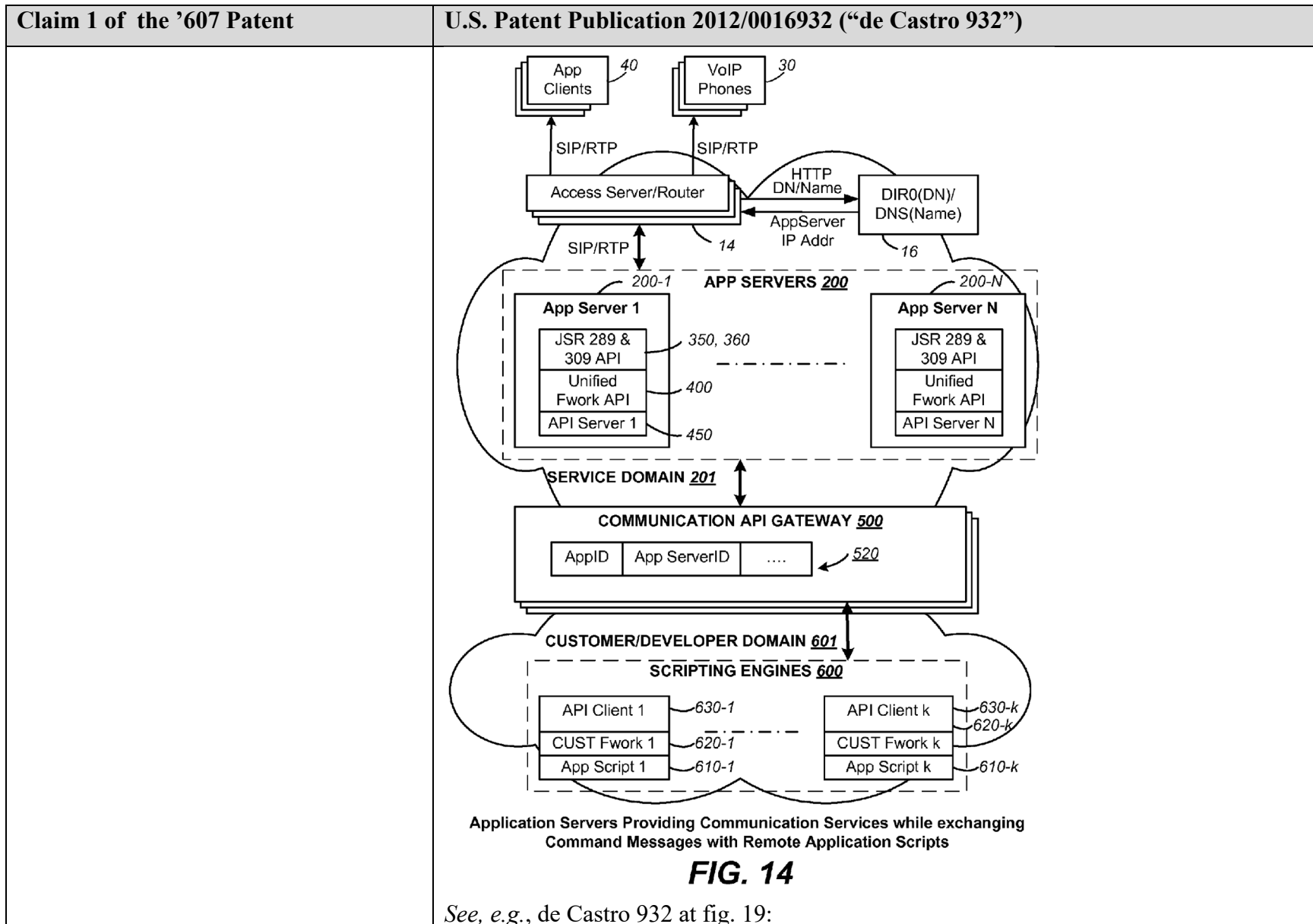
Defendant Cisco Systems Inc. (“Cisco”) incorporates in this chart all applicable qualifications, clarifications, and other statements made in Cisco’s Invalidity Contentions. This invalidity claim chart is based on Cisco’s present understanding of claims 1-3 and 5-6 and Plaintiff Zilkr Cloud Technologies, LLC’s (“Zilkr”) apparent construction of the claims, as set forth in Zilkr’s Infringement Contentions. These invalidity assertions are not an admission that Cisco concedes or acquiesces to any claim construction(s) implied or suggested by Zilkr the associated infringement claim charts. Nor is Cisco asserting any claim construction positions through these charts, including whether the preamble is a limitation. Cisco also does not concede or acquiesce that any asserted claim satisfies the requirements of 35 U.S.C. §§ 112 or 101, and submits these invalidity contentions only to the extent Zilkr’s assertions may be understood.

The excerpts cited herein are exemplary. For any claim limitation, Cisco may rely on excerpts cited for any other limitation and/or additional excerpts not set forth fully herein to the extent necessary to provide a more comprehensive explanation for a reference’s disclosure of a limitation. Where an excerpt refers to or discusses a figure or figure items, that figure and any additional descriptions of that figure should be understood to be incorporated by reference as if set forth fully herein. To the extent any cell lacks citations to the charted reference(s), this should not be taken as an admission that the reference(s) do(es) not disclose the corresponding limitation but rather indicates that Cisco does not presently have sufficient information regarding the charted reference(s) based on Cisco’s present understanding of the claim limitation.

These invalidity contentions are not an admission by Cisco that the accused products or components, including any current or past version of the products or components, are covered by, or infringe the asserted claims, particularly when these claims are properly construed and applied.

1. Claim 1.

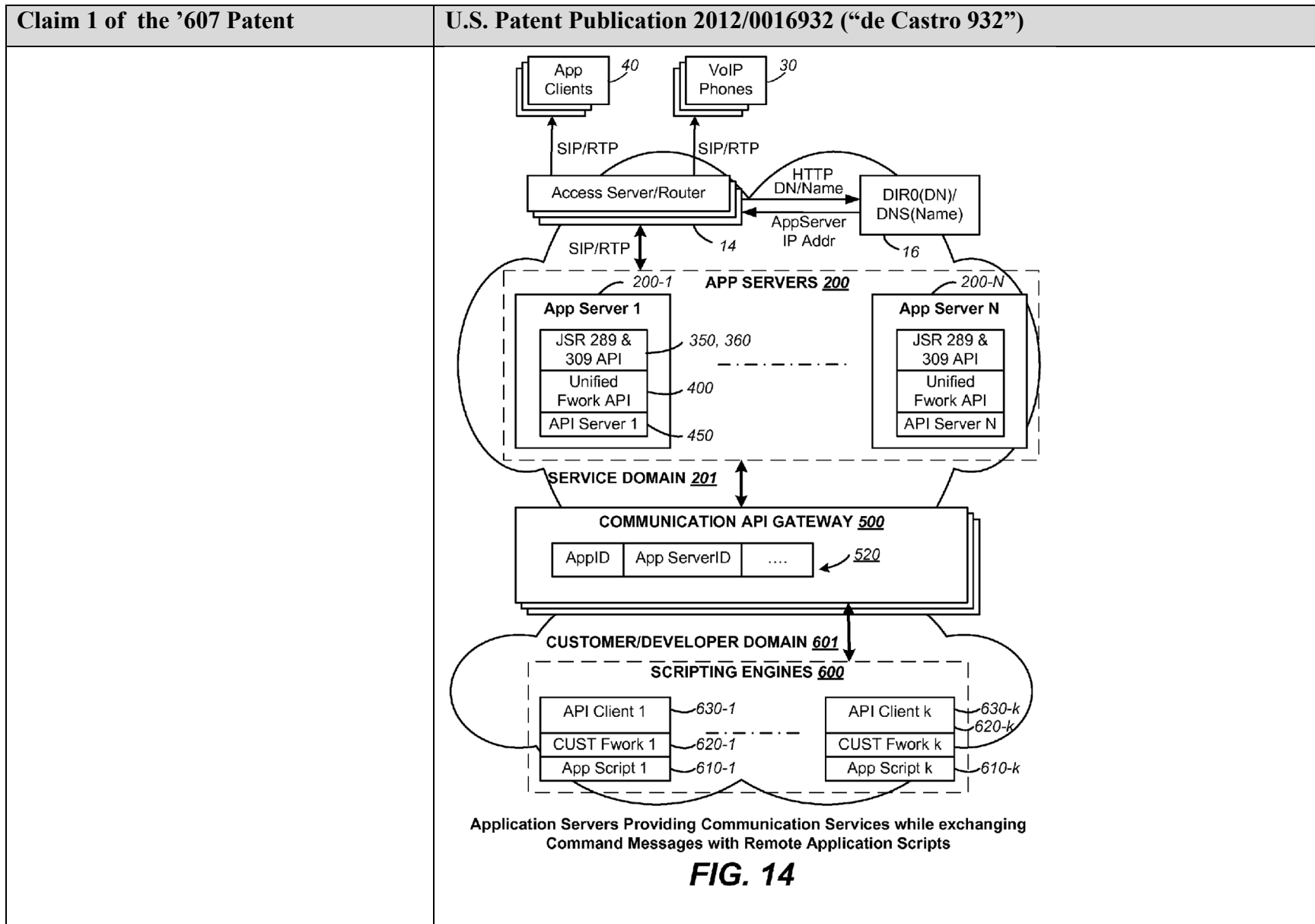
Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
1[p] A method of provisioning a plurality of applications from different application developer systems to a network provider system through a unified services platform, the method comprising:	de Castro 932 discloses a method of provisioning a plurality of applications from different application developer systems to a network provider system through a unified services platform, the method comprising. <i>See, e.g.,</i> de Castro 932 at fig. 14:



Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)												
	Registry 520 <table><tr><th colspan="3">SERVICE DOMAIN</th><th colspan="3">USER DOMAIN</th></tr><tr><td>App ServerID</td><td>App Server Properties/ Status</td><td>....</td><td>App ID</td><td>App Properties/ Status</td><td>....</td></tr></table> FIG. 19 See, e.g., de Castro 932 at [0115]: [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers. See, e.g., de Castro 932 at [0129]-[0130]: [0129] FIG. 18 illustrates in more detail the communication API gateway shown in FIG. 14. One or more communication API gateway 500 serves as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. The communication API gateway includes a registry 520, a resource load balancer 530, and a messaging server 540. The resource load balance 530 is responsible for locating a communication application server 200 best suited for executing the commands from a given API client 630. In the preferred embodiment, the messaging server 540 is an XMPP server. Messaging under the XMPP protocol is expedient in that the persistent connection is more conducive to the call environment. Also, it has well developed protocol and facilities for federation between different messaging domains.	SERVICE DOMAIN			USER DOMAIN			App ServerID	App Server Properties/ Status		App ID	App Properties/ Status	
SERVICE DOMAIN			USER DOMAIN										
App ServerID	App Server Properties/ Status		App ID	App Properties/ Status									

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. <i>See, e.g., de Castro 932 at [0102]:</i> [0102] FIG. 13 illustrates a typical implementation of deploying communication applications on an IP network such as the internet. A plurality of application platforms 100 is deployed on the network to provide communication services when hosting any number of communication applications defined by the application scripts 210. Clients such as VoIP phones 30, application clients 40 and other terminal equipment is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to an application platform, such as application platform 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of an available application platform with the necessary resources to execute the specific communication application. Once application platform 1 is selected, the application server in the platform will retrieve the application script 210 for the specific application associated with the dialed number. This is accomplished by looking up another directory service DIR1 18 for a pointer to the URL of the application script. For example, the dialed number may be associated with application script 1 and it is retrieved into the application platform 1. <i>See, e.g., de Castro 932 at [0039]:</i>

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0039] In a preferred embodiment, the communication application server is among a group of similar communication application servers on the network to provide telephony and communication services to a plurality of customers with application scripts hosted on scripting engines. One or more communication API gateway for the group of communication application servers is deployed on the network to serve as a messaging broker between the plurality of customers with application scripts and the group of communication application servers. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[a] establishing a plurality of service accounts for a corresponding plurality of application service provider systems, wherein the plurality of application service provider systems provide services for use in association with a telephone number of a subscriber of telecommunication services;	de Castro 932 discloses establishing a plurality of service accounts for a corresponding plurality of application service provider systems, wherein the plurality of application service provider systems provide services for use in association with a telephone number of a subscriber of telecommunication services. <i>See, e.g.,</i> de Castro 932 at fig. 14:

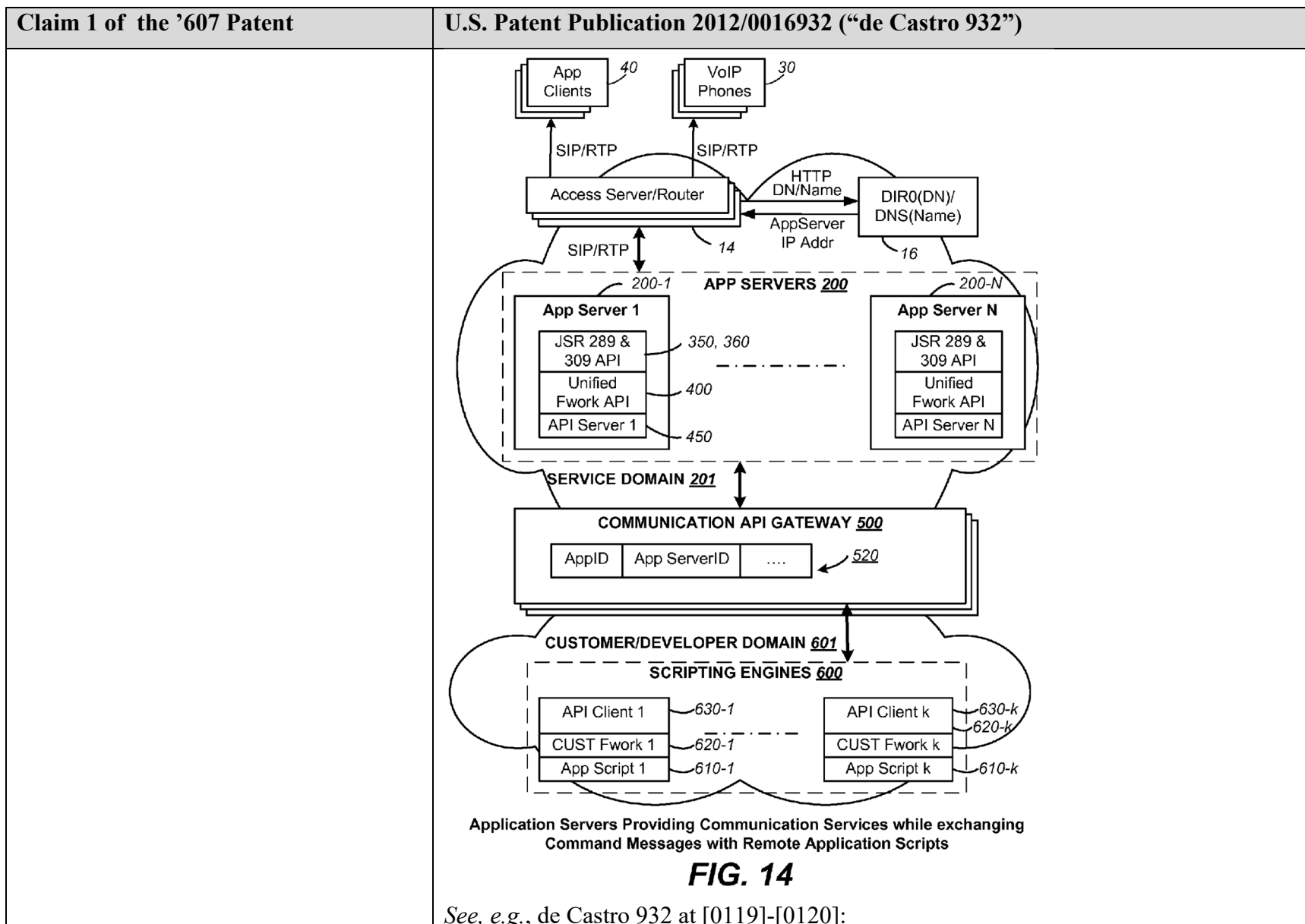


Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)												
	See, e.g., de Castro 932 at fig. 19: Registry 520 <table><tr><th colspan="3">SERVICE DOMAIN</th><th colspan="3">USER DOMAIN</th></tr><tr><td>App ServerID</td><td>App Server Properties/ Status</td><td>....</td><td>App ID</td><td>App Properties/ Status</td><td>....</td></tr></table> FIG. 19 See, e.g., de Castro 932 at [0129]-[0130]: [0129] FIG. 18 illustrates in more detail the communication API gateway shown in FIG. 14. One or more communication API gateway 500 serves as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. The communication API gateway includes a registry 520, a resource load balancer 530, and a messaging server 540. The resource load balance 530 is responsible for locating a communication application server 200 best suited for executing the commands from a given API client 630. In the preferred embodiment, the messaging server 540 is an XMPP server. Messaging under the XMPP protocol is expedient in that the persistent connection is more conducive to the call environment. Also, it has well developed protocol and facilities for federation between different messaging domains. [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having	SERVICE DOMAIN			USER DOMAIN			App ServerID	App Server Properties/ Status		App ID	App Properties/ Status	
SERVICE DOMAIN			USER DOMAIN										
App ServerID	App Server Properties/ Status		App ID	App Properties/ Status									

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. <i>See, e.g., de Castro 932 at [0039]:</i> [0039] In a preferred embodiment, the communication application server is among a group of similar communication application servers on the network to provide telephony and communication services to a plurality of customers with application scripts hosted on scripting engines. One or more communication API gateway for the group of communication application servers is deployed on the network to serve as a messaging broker between the plurality of customers with application scripts and the group of communication application servers. <i>See, e.g., de Castro 932 at [0102]-[0104]:</i> [0102] FIG. 13 illustrates a typical implementation of deploying communication applications on an IP network such as the internet. A plurality of application platforms 100 is deployed on the network to provide communication services when hosting any number of communication applications defined by the application scripts 210. Clients such as VoIP phones 30, application clients 40 and other terminal equipment is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to an application platform, such as application platform 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of an available application platform with the necessary resources to execute the specific communication application. Once application platform 1 is selected, the application server in the platform will retrieve the application script 210 for the specific application associated with the dialed number. This is accomplished by looking up another directory

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	service DIR1 18 for a pointer to the URL of the application script. For example, the dialed number may be associated with application script 1 and it is retrieved into the application platform 1. [0103] Once the “dialed” application script 1 is retrieved into the application platform 1, the application server 200 (see also FIG. 1 and FIG. 4) will process the script. First a browser, compatible with the language the script, will interpret or compile the script into execution codes which will invoke the class objects and make function calls to the unified API. The execution is performed in a Java virtual machine supported by the application server 200. [0104] In this manner, communication applications such as voice-centers, IVR, voice-enabled self-help applications web bots can be deployed by such an application platform on the internet by leveraging web-like technologies and practices and employing standards-based protocol and languages. Variations of such communication platforms and network architecture of communication resources have been disclosed in U.S. Pat. No. 6,922,411, United States Patent Application Publication No. US 2011/0046960, and U.S. patent application Ser. No. 13/088,396 filed on Apr. 17, 2011, the entire disclosure of said publications and Applications are incorporated herein by reference. <i>See, e.g., de Castro 932 at [0139]-[0141]:</i> [0139] The Rayo protocol primarily deals with calls. Inbound calls originate from the PSTN or via SIP and are offered to Rayo clients via XMPP using a Jabber Identifier (JID). Each call is in turn represented by its own unique JID allowing a two way conversation between the Rayo client and the server that's handling the call signaling and media. JID Format [0140] The JID follows a specific format. In XMPP the JID is constructed as <node>@<domain>/<resource> For Rayo, the <node> portion of the JID always represents the call ID. The <resource>, when present, represents the affected command ID.

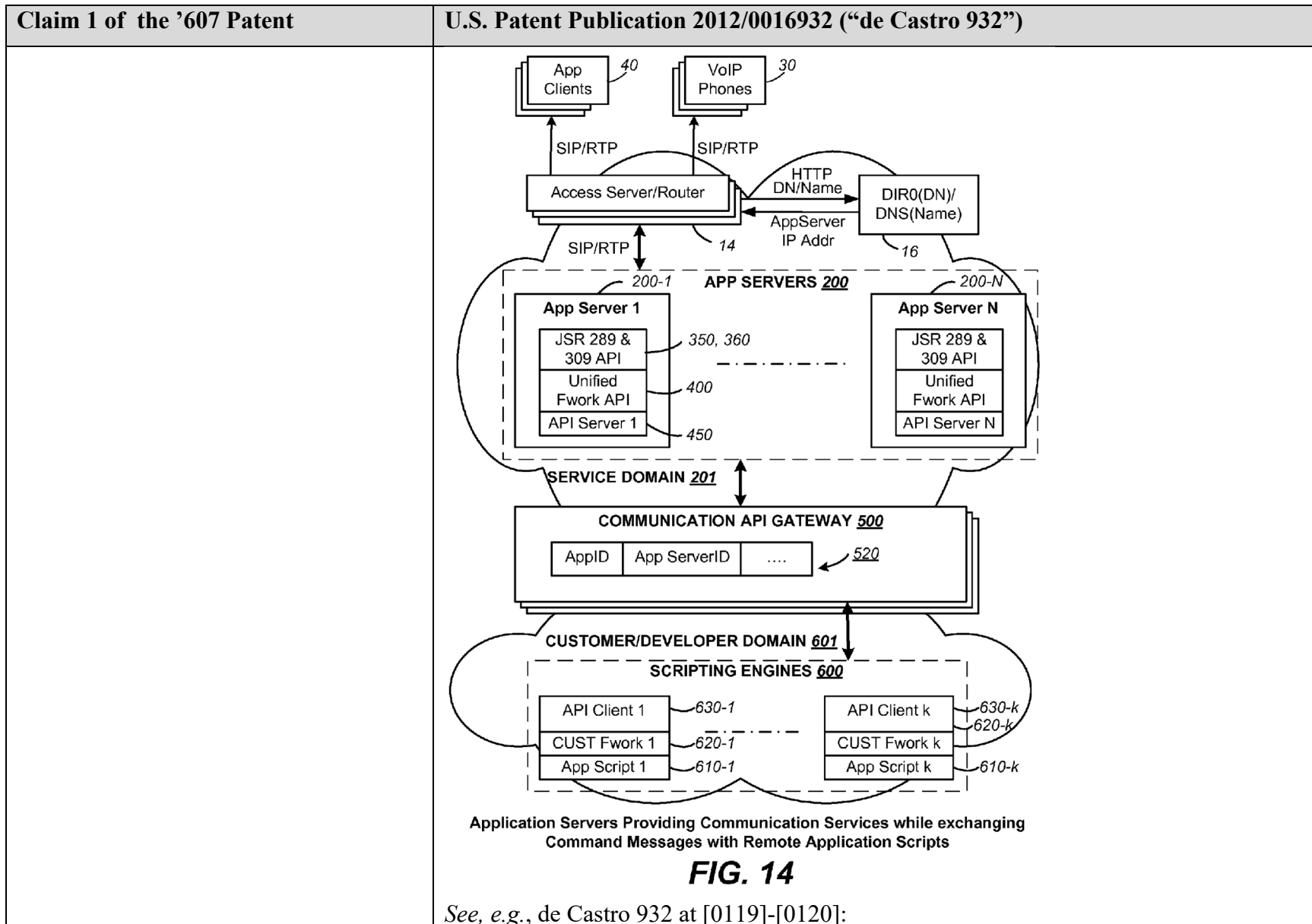
Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	Incoming Calls [0141] <!-- Message comes from the Call's JID --> <presence to='16577@app.Rayo.net/1' from='9f00061@call.Rayo.net/1'> <offer xmlns='urn:xmpp:Rayo:1' to='tel:+18003211212' from='tel:+13058881212'> <!-- Signaling (e.g. SIP) Headers --> <header name=' Via' value='192.168.0.1' /> <header name='Contact' value='192.168.0.1' /> </offer> </presence> To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[b] receiving, by a processor system, a plurality of API interfaces from the plurality of application service provider systems, wherein at least two of the API interfaces are different,	de Castro 932 discloses receiving, by a processor system, a plurality of API interfaces from the plurality of application service provider systems, wherein at least two of the API interfaces are different. <i>See, e.g., de Castro 932 at fig. 14:</i>



Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0119] On the other hand, in the customer/developer domain 601 is a plurality of scripting engines 600 for the customers to host and process the customers' application scripts 610. For example, a customer 1 has the scripting engines 600 host and process an application script such as App Script 1 610-1. The processing of App Script 1 is facilitated by a customer-specific framework 620, such as CUST Fwork 1 620-1 which supplies a set of customer-specific libraries. The processing is performed by an API client 630 such as API client 1 630-1 which interprets App Script 1 and renders it in terms of the commands of the finite set defined at the API server 450 of the communication application server 200. The API client 630 then packages the commands as messages in accordance with the messaging API protocol to be sent to a connected communication application server 200 such as App server 1 200-1. [0120] In a preferred embodiment, one or more communication API gateway 500 is deployed on the network to serve as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. <i>See, e.g., de Castro 932 at [0067]:</i> [0067] A software framework, in computer programming, is an abstraction in which common code providing generic functionality can be selectively specialized by user code providing specific functionality. Frameworks are a special case of software libraries in that they are reusable abstractions of code wrapped in a well-defined API, yet they contain some key distinguishing features that separate them from normal libraries. In this case, the unified communication API represents a further abstraction from the primitive call control and media control APIs that more closely models the real situation being addressed by the application. <i>See, e.g., de Castro 932 at [0038]:</i> [0038] According to a general aspect of the invention, a communication application server is

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	provided with a unified framework for call control and media control. The framework supports a unified API having class objects and functions conforming to a telephony object model. The class objects are invoked and manipulated by a finite set of commands and an application program essentially issues a series of such commands to operate the communication application server. More particularly, an API server on the communication application server defining a messaging API protocol enables an application script to pass commands remotely to the communication application server to operate it. This allows application scripts to be processed remotely by appropriate scripting engines. The resulting execution codes from the scripting are expressed in terms of the commands of the finite set, which are then sent to the communication application server. The API server at the communication application server parses out the commands from the messages to have the communication application server executes the commands as they are available for execution. <i>See, e.g., de Castro 932 at [0129]-[0130]:</i> [0129] FIG. 18 illustrates in more detail the communication API gateway shown in FIG. 14. One or more communication API gateway 500 serves as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. The communication API gateway includes a registry 520, a resource load balancer 530, and a messaging server 540. The resource load balance 530 is responsible for locating a communication application server 200 best suited for executing the commands from a given API client 630. In the preferred embodiment, the messaging server 540 is an XMPP server. Messaging under the XMPP protocol is expedient in that the persistent connection is more conducive to the call environment. Also, it has well developed protocol and facilities for federation between different messaging domains. [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having

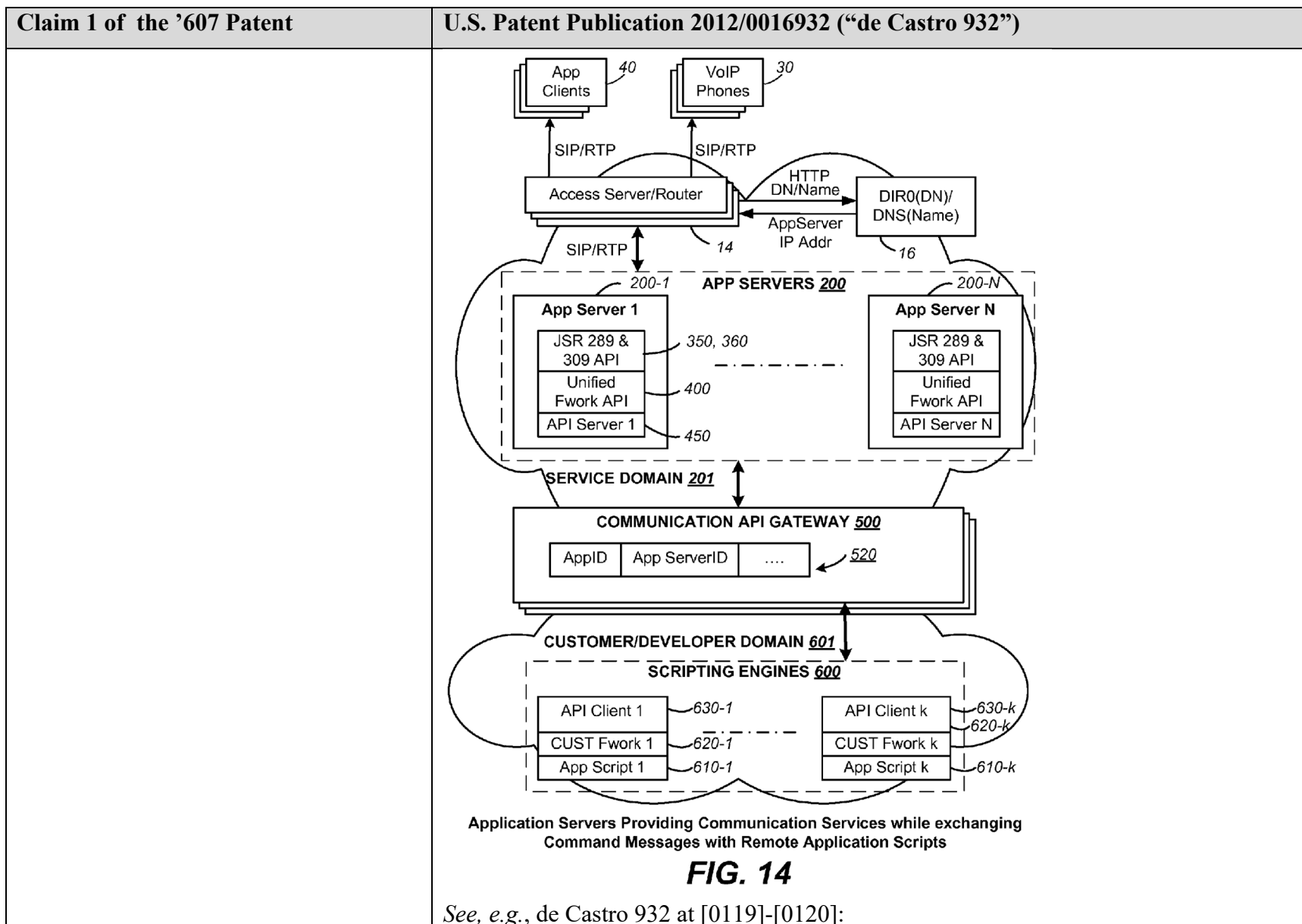
Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[c] integrating, by the processor system, each of the plurality of API interfaces with a unified services platform API interface;	de Castro 932 discloses integrating, by the processor system, each of the plurality of API interfaces with a unified services platform API interface. <i>See, e.g.,</i> de Castro 932 at fig. 14:



Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0119] On the other hand, in the customer/developer domain 601 is a plurality of scripting engines 600 for the customers to host and process the customers' application scripts 610. For example, a customer 1 has the scripting engines 600 host and process an application script such as App Script 1 610-1. The processing of App Script 1 is facilitated by a customer-specific framework 620, such as CUST Fwork 1 620-1 which supplies a set of customer-specific libraries. The processing is performed by an API client 630 such as API client 1 630-1 which interprets App Script 1 and renders it in terms of the commands of the finite set defined at the API server 450 of the communication application server 200. The API client 630 then packages the commands as messages in accordance with the messaging API protocol to be sent to a connected communication application server 200 such as App server 1 200-1. [0120] In a preferred embodiment, one or more communication API gateway 500 is deployed on the network to serve as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. <i>See, e.g., de Castro 932 at [0067]:</i> [0067] A software framework, in computer programming, is an abstraction in which common code providing generic functionality can be selectively specialized by user code providing specific functionality. Frameworks are a special case of software libraries in that they are reusable abstractions of code wrapped in a well-defined API, yet they contain some key distinguishing features that separate them from normal libraries. In this case, the unified communication API represents a further abstraction from the primitive call control and media control APIs that more closely models the real situation being addressed by the application. <i>See, e.g., de Castro 932 at [0038]:</i> [0038] According to a general aspect of the invention, a communication application server is

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	provided with a unified framework for call control and media control. The framework supports a unified API having class objects and functions conforming to a telephony object model. The class objects are invoked and manipulated by a finite set of commands and an application program essentially issues a series of such commands to operate the communication application server. More particularly, an API server on the communication application server defining a messaging API protocol enables an application script to pass commands remotely to the communication application server to operate it. This allows application scripts to be processed remotely by appropriate scripting engines. The resulting execution codes from the scripting are expressed in terms of the commands of the finite set, which are then sent to the communication application server. The API server at the communication application server parses out the commands from the messages to have the communication application server executes the commands as they are available for execution. <i>See, e.g., de Castro 932 at [0129]-[0130]:</i> [0129] FIG. 18 illustrates in more detail the communication API gateway shown in FIG. 14. One or more communication API gateway 500 serves as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. The communication API gateway includes a registry 520, a resource load balancer 530, and a messaging server 540. The resource load balance 530 is responsible for locating a communication application server 200 best suited for executing the commands from a given API client 630. In the preferred embodiment, the messaging server 540 is an XMPP server. Messaging under the XMPP protocol is expedient in that the persistent connection is more conducive to the call environment. Also, it has well developed protocol and facilities for federation between different messaging domains. [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having

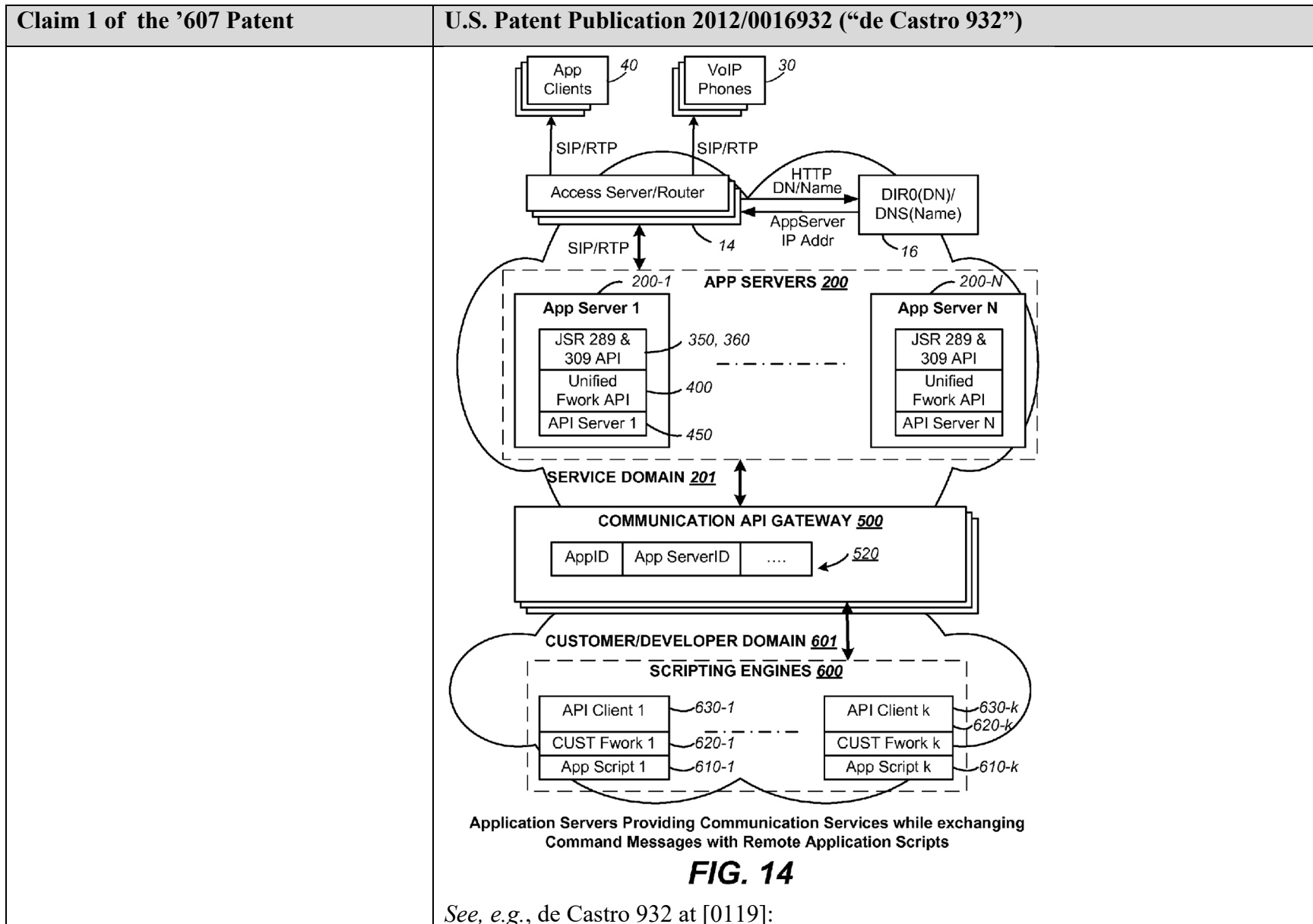
Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[d] receiving a provider API interface for a network provider system;	de Castro 932 discloses receiving a provider API interface for a network provider system. <i>See, e.g.,</i> de Castro 932 at fig. 14:



Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0119] On the other hand, in the customer/developer domain 601 is a plurality of scripting engines 600 for the customers to host and process the customers' application scripts 610. For example, a customer 1 has the scripting engines 600 host and process an application script such as App Script 1 610-1. The processing of App Script 1 is facilitated by a customer-specific framework 620, such as CUST Fwork 1 620-1 which supplies a set of customer-specific libraries. The processing is performed by an API client 630 such as API client 1 630-1 which interprets App Script 1 and renders it in terms of the commands of the finite set defined at the API server 450 of the communication application server 200. The API client 630 then packages the commands as messages in accordance with the messaging API protocol to be sent to a connected communication application server 200 such as App server 1 200-1. [0120] In a preferred embodiment, one or more communication API gateway 500 is deployed on the network to serve as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. <i>See, e.g., de Castro 932 at [0067]:</i> [0067] A software framework, in computer programming, is an abstraction in which common code providing generic functionality can be selectively specialized by user code providing specific functionality. Frameworks are a special case of software libraries in that they are reusable abstractions of code wrapped in a well-defined API, yet they contain some key distinguishing features that separate them from normal libraries. In this case, the unified communication API represents a further abstraction from the primitive call control and media control APIs that more closely models the real situation being addressed by the application. <i>See, e.g., de Castro 932 at [0038]:</i> [0038] According to a general aspect of the invention, a communication application server is

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	provided with a unified framework for call control and media control. The framework supports a unified API having class objects and functions conforming to a telephony object model. The class objects are invoked and manipulated by a finite set of commands and an application program essentially issues a series of such commands to operate the communication application server. More particularly, an API server on the communication application server defining a messaging API protocol enables an application script to pass commands remotely to the communication application server to operate it. This allows application scripts to be processed remotely by appropriate scripting engines. The resulting execution codes from the scripting are expressed in terms of the commands of the finite set, which are then sent to the communication application server. The API server at the communication application server parses out the commands from the messages to have the communication application server executes the commands as they are available for execution. <i>See, e.g., de Castro 932 at [0129]-[0130]:</i> [0129] FIG. 18 illustrates in more detail the communication API gateway shown in FIG. 14. One or more communication API gateway 500 serves as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. The communication API gateway includes a registry 520, a resource load balancer 530, and a messaging server 540. The resource load balance 530 is responsible for locating a communication application server 200 best suited for executing the commands from a given API client 630. In the preferred embodiment, the messaging server 540 is an XMPP server. Messaging under the XMPP protocol is expedient in that the persistent connection is more conducive to the call environment. Also, it has well developed protocol and facilities for federation between different messaging domains. [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[e] integrating the provider API interface with the unified services platform API interface, wherein at least one of the plurality of API interfaces from the plurality of application service provider systems is not compatible with the provider API interface;	de Castro 932 discloses integrating the provider API interface with the unified services platform API interface, wherein at least one of the plurality of API interfaces from the plurality of application service provider systems is not compatible with the provider API interface. <i>See, e.g., de Castro 932 at fig. 14:</i>

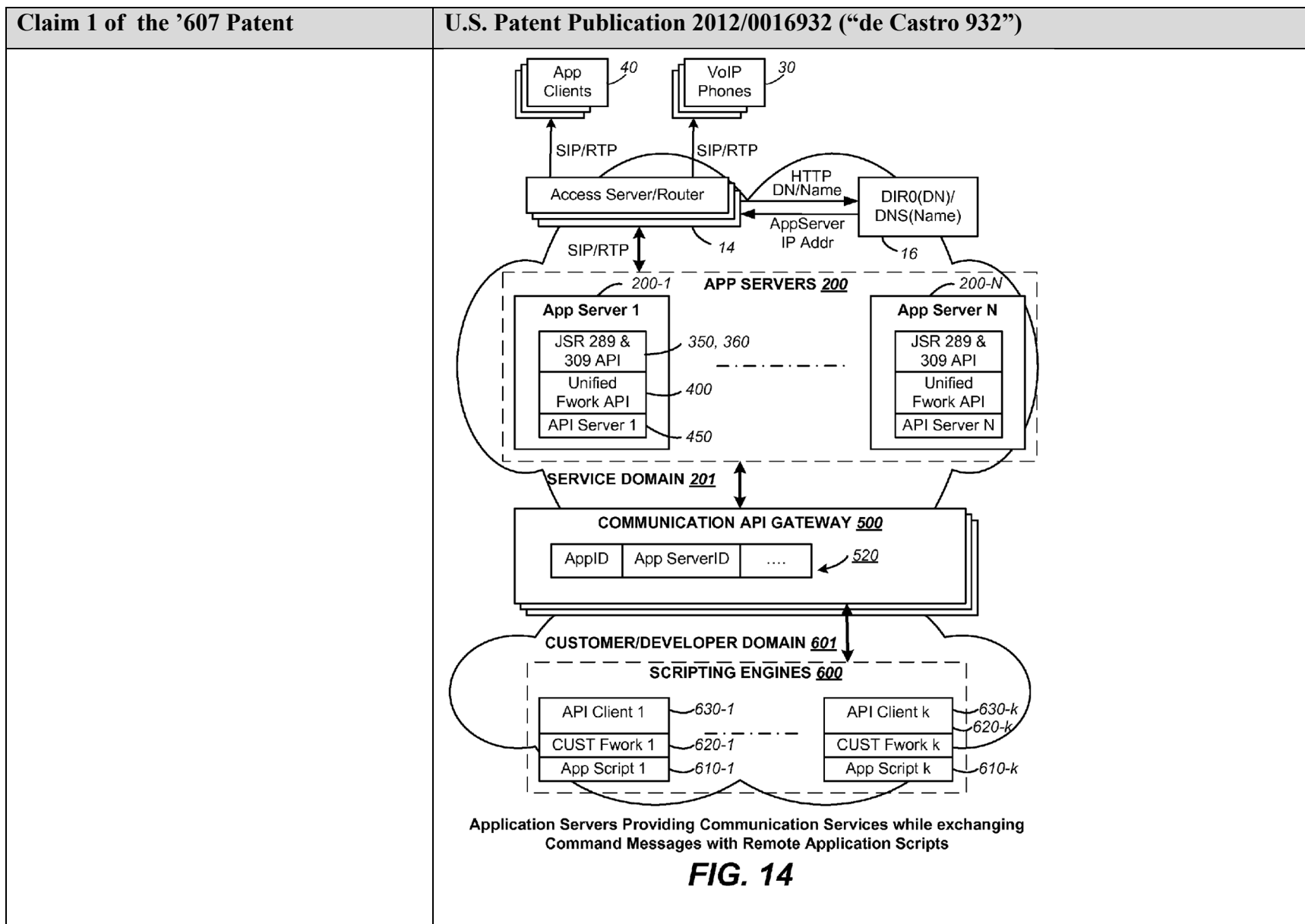


Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0119] On the other hand, in the customer/developer domain 601 is a plurality of scripting engines 600 for the customers to host and process the customers' application scripts 610. For example, a customer 1 has the scripting engines 600 host and process an application script such as App Script 1 610-1. The processing of App Script 1 is facilitated by a customer-specific framework 620, such as CUST Fwork 1 620-1 which supplies a set of customer-specific libraries. The processing is performed by an API client 630 such as API client 1 630-1 which interprets App Script 1 and renders it in terms of the commands of the finite set defined at the API server 450 of the communication application server 200. The API client 630 then packages the commands as messages in accordance with the messaging API protocol to be sent to a connected communication application server 200 such as App server 1 200-1. <i>See, e.g., de Castro 932 at [0067]:</i> [0067] A software framework, in computer programming, is an abstraction in which common code providing generic functionality can be selectively specialized by user code providing specific functionality. Frameworks are a special case of software libraries in that they are reusable abstractions of code wrapped in a well-defined API, yet they contain some key distinguishing features that separate them from normal libraries. In this case, the unified communication API represents a further abstraction from the primitive call control and media control APIs that more closely models the real situation being addressed by the application. <i>See, e.g., de Castro 932 at [0038]:</i> [0038] According to a general aspect of the invention, a communication application server is provided with a unified framework for call control and media control. The framework supports a unified API having class objects and functions conforming to a telephony object model. The class objects are invoked and manipulated by a finite set of commands and an application program essentially issues a series of such commands to operate the communication application server. More particularly, an API server on the

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	communication application server defining a messaging API protocol enables an application script to pass commands remotely to the communication application server to operate it. This allows application scripts to be processed remotely by appropriate scripting engines. The resulting execution codes from the scripting are expressed in terms of the commands of the finite set, which are then sent to the communication application server. The API server at the communication application server parses out the commands from the messages to have the communication application server executes the commands as they are available for execution. <i>See, e.g., de Castro 932 at [0130]:</i> [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would

Zilkr v Cisco Systems, Inc.
Eastern District of Texas No. 2:22-cv-00166-JRG-RSP

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	yield predictable results.
[f] receiving a definition of a service and a plan for each of the plurality of application service provider systems;	de Castro 932 discloses receiving a definition of a service and a plan for each of the plurality of application service provider systems. <i>See, e.g.,</i> de Castro 932 at fig. 14:



Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)												
	See, e.g., de Castro 932 at fig. 19: Registry <u>520</u> <table><tr><th colspan="3">SERVICE DOMAIN</th><th colspan="3">USER DOMAIN</th></tr><tr><td>App ServerID</td><td>App Server Properties/ Status</td><td>....</td><td>App ID</td><td>App Properties/ Status</td><td>....</td></tr></table> FIG. 19 See, e.g., de Castro 932 at [0129]-[0130]: [0129] FIG. 18 illustrates in more detail the communication API gateway shown in FIG. 14. One or more communication API gateway 500 serves as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. The communication API gateway includes a registry 520, a resource load balancer 530, and a messaging server 540. The resource load balance 530 is responsible for locating a communication application server 200 best suited for executing the commands from a given API client 630. In the preferred embodiment, the messaging server 540 is an XMPP server. Messaging under the XMPP protocol is expedient in that the persistent connection is more conducive to the call environment. Also, it has well developed protocol and facilities for federation between different messaging domains. [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by	SERVICE DOMAIN			USER DOMAIN			App ServerID	App Server Properties/ Status		App ID	App Properties/ Status	
SERVICE DOMAIN			USER DOMAIN										
App ServerID	App Server Properties/ Status		App ID	App Properties/ Status									

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. <i>See, e.g., de Castro 932 at [0006]- [0012]:</i> [0006] An example platform for easy deployment of telephony applications is described in U.S. Pat. No. 6,922,411, which entire disclosure is incorporated herein by reference. Essentially, a networked telephony system allows users to deploy on the Internet computer telephony applications associated with designated telephone numbers. The telephony application is easily created by a user in XML (Extended Markup Language) with predefined telephony XML tags (e.g. VoiceXML) and easily deployed on a website. The telephony XML tags include those for call control and media manipulation. A call to anyone of these designated telephone numbers may originate from anyone of the networked telephone system such as the PSTN (Public Switched Telephone System), a wireless network, or the Internet. The call is received by an application gateway center (AGC) installed on the Internet. Analogous to a web browser, the AGC provides facility for retrieving the associated XML application from its website and processing the call accordingly. [0007] This type of telephony platform allows very powerful yet simple telephony applications to be built and deployed on the Internet. The following are some examples of the telephony applications deployed on this platform. A “Follow me, find me” application sequentially calls a series of telephone numbers as specified by a user until one of the numbers answers and then connects the call. Otherwise, it does something else such as takes a message or sends e-mail or sends the call to a call center, etc. In another example, a Telephonic Polling application looks up from a database the telephone numbers of a population to be polled. It then calls the numbers in parallel, limited only by the maximum number of concurrent sessions supported, and plays a series of interactive voice prompts/messages in response to the called party's responses and

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	records the result in a database, etc. In another example, a Help Desk application plays a series of interactive voice prompts/messages in response to the called party's responses and possibly connects the call to a live agent as one option, etc. In yet another example, a Stock or Bank Transactions application plays a series of interactive voice prompts/messages in response to the called party's responses and conducts appropriate transactions with a backend database or web application, etc. [0008] The latter examples are generally referred to as self-help applications. In the voice context, a self-help application is referred to as IVR. IVR refers to Interactive Voice Response and is a technology that automates interaction with telephone callers. Enterprises are increasingly turning to IVR to reduce the cost of common sales, service, collections, inquiry and support calls to and from their company. [0009] IVR solutions enable users using voice as a medium or other form of inputs through a voice channel to retrieve information including bank balances, flight schedules, product details, order status, movie show times, and more from any telephone. Additionally, IVR solutions are increasingly used to place outbound calls to deliver or gather information for appointments, past due bills, and other time critical events and activities. [0010] FIG. 1 illustrates schematically a communication application environment. The communication application environment 10 includes one or more client interacting with a communication application server 200 in an application platform 100. The application platform 100 hosts an application specified by an application script 210 coded in object-oriented software. The communication application server 200 includes a browser 220 for interpreting and executing the application script 210. The execution of the application script invokes one or more server-side components 310 in the application server 200. Among the clients and the communication server, these components 310 provide services for call control, media control with one or more media server 230 and interactions with back-end systems 240 such as databases, and business logic and legacy systems such as CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning). One example of the platform is to host an IVR application which interacts with voice, text messaging and other clients in a multi-channel environment.

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0011] The communication application platform provides a third-party call control between any numbers of clients 20, 22, 30. The application script 210 defines the communication application 300 and directs how a call is to be handled. For example, when a user makes a call through a voice client such as a handset 20 or a VoIP phone 22 to the IVR, the voice application script 210 associated with the call number is retrieved. The browser 220 executes or renders the retrieved voice application script to allow the user to interact with the voice application 300. [0012] Communication of Multimedia information among endpoints and a third-party call controller generally require call control and media control. <i>See, e.g., de Castro 932 at [0075]:</i> [0075] The call control model of the unified communication framework is designed for calls and media that are controlled by a 3rd party server application, such as PBX, IVR, Conferencing, and Call Center applications. It assumes all the calls have at least their signals controlled by the communication application. In most cases, media should preferably be controlled by the communication application as well. <i>See, e.g., de Castro 932 at [0114]- [0116]:</i> [0114] FIG. 14 illustrates a network architecture suitable for practicing the present invention. Essentially, the network is considered as comprising a service domain 201 and a customer/developer domain 601. [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers. [0116]

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	Similar to that shown in FIG. 13, application clients such as VoIP phones 30, is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to one of the communication application servers 200, such as App server 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of the communication application server executing the specific communication application. <i>See, e.g., de Castro 932 at [0039]:</i> [0039] In a preferred embodiment, the communication application server is among a group of similar communication application servers on the network to provide telephony and communication services to a plurality of customers with application scripts hosted on scripting engines. One or more communication API gateway for the group of communication application servers is deployed on the network to serve as a messaging broker between the plurality of customers with application scripts and the group of communication application servers. <i>See, e.g., de Castro 932 at [0130]:</i> [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. To the extent it is found that de Castro 932 does not expressly disclose this limitation,

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[g] receiving one or more definitions of actions for the service, wherein the actions comprise responses to triggering events occurring with a telephone number; and	de Castro 932 discloses receiving one or more definitions of actions for the service, wherein the actions comprise responses to triggering events occurring with a telephone number. <i>See, e.g.,</i> de Castro 932 at [0122]: [0122] Thus, the communication application server 200 communicates with the scripting engines 600 using a messaging API protocol. When a command carried in a message is received at the communication application server 200, the API server 450 will parse out the command in accordance with the messaging API protocol and allow the command to be queued with the associated call (actor) to be executed. <i>See, e.g.,</i> de Castro 932 at [0141]: [0141] The Rayo client can now control the call by using one of the following commands. <!-- Accept (e.g. SIP 180/Ringing). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <accept xmlns='urn:xmpp:Rayo:1'> <!-- Sample Headers (optional) --> <header name="x-skill" value="agent" />

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	<pre> <header name="x-customer-id". value="8877" /> </accept> </iq> <!-- Answer (e.g. SIP 200/OK). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <answer xmlns='urn:xmpp:Rayo:1'> <!-- Sample Headers (optional) --> <header name="x-skill" value="agent" /> <header name="x-customer-id" value="8877" /> </answer> </iq> <!-- Redirect(e.g. SIP 302/Redirect). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <redirect to='tel:+14152226789' xmlns='urn:xmpp:Rayo:1'> <!-- Sample Headers(optional) --> <header name="x-skill" value="agent" /> <header name="x-customer-id" value="8877" /> </redirect> </iq> A call can also be rejected. Rejections can include an optional rejection reason. Rejection reasons are one of <busy/>, <decline/> or <error/>. If not specified, <decline/> is used as the default reason. <!-- Decline (.g. SIP 603/Decline). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <reject xmlns='urn:xmpp:Rayo:1'> </pre>

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	<pre> <decline /> <!-- Sample Headers (optional) --> <header name="x-reason-internal" value="bad-skill" /> </reject> </iq> <!-- Busy (.g. SIP 486/Busy). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <reject xmlns='urn:xmpp:Rayo:1'> <busy /> <!-- Sample Headers (optional) --> <header name="x-busy-detail" value="out of licenses" /> </reject> </iq> <!-- Error (.g. SIP 500/Internal Server Error). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <reject xmlns='urn:xmpp:Rayo:1'> <error /> <!-- Sample Headers (optional) --> <header name="x-error-detail" value="soem descriptive error message" /> </reject> </iq> </pre> <i>See, e.g., de Castro 932 at [0006]- [0012]:</i> [0006] An example platform for easy deployment of telephony applications is described in U.S. Pat. No. 6,922,411, which entire disclosure is incorporated herein by reference. Essentially, a networked telephony system allows users to deploy on the Internet

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	computer telephony applications associated with designated telephone numbers. The telephony application is easily created by a user in XML (Extended Markup Language) with predefined telephony XML tags (e.g. VoiceXML) and easily deployed on a website. The telephony XML tags include those for call control and media manipulation. A call to anyone of these designated telephone numbers may originate from anyone of the networked telephone system such as the PSTN (Public Switched Telephone System), a wireless network, or the Internet. The call is received by an application gateway center (AGC) installed on the Internet. Analogous to a web browser, the AGC provides facility for retrieving the associated XML application from its website and processing the call accordingly. [0007] This type of telephony platform allows very powerful yet simple telephony applications to be built and deployed on the Internet. The following are some examples of the telephony applications deployed on this platform. A “Follow me, find me” application sequentially calls a series of telephone numbers as specified by a user until one of the numbers answers and then connects the call. Otherwise, it does something else such as takes a message or sends e-mail or sends the call to a call center, etc. In another example, a Telephonic Polling application looks up from a database the telephone numbers of a population to be polled. It then calls the numbers in parallel, limited only by the maximum number of concurrent sessions supported, and plays a series of interactive voice prompts/messages in response to the called party's responses and records the result in a database, etc. In another example, a Help Desk application plays a series of interactive voice prompts/messages in response to the called party's responses and possibly connects the call to a live agent as one option, etc. In yet another example, a Stock or Bank Transactions application plays a series of interactive voice prompts/messages in response to the called party's responses and conducts appropriate transactions with a backend database or web application, etc. [0008] The latter examples are generally referred to as self-help applications. In the voice context, a self-help application is referred to as IVR. IVR refers to Interactive Voice Response and is a technology that automates interaction with telephone callers. Enterprises are increasingly turning to IVR to reduce the cost of common sales, service,

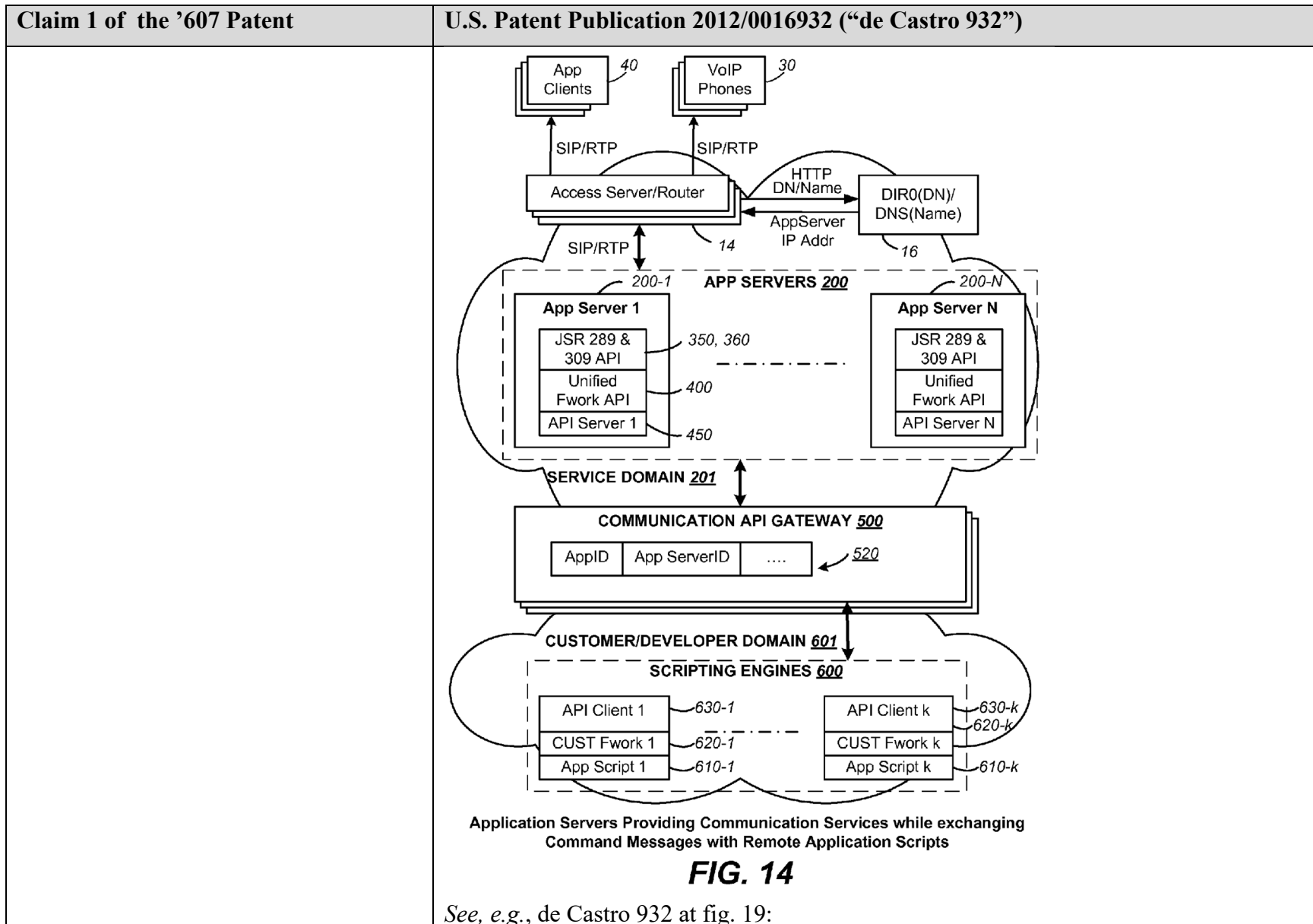
Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	collections, inquiry and support calls to and from their company. [0009] IVR solutions enable users using voice as a medium or other form of inputs through a voice channel to retrieve information including bank balances, flight schedules, product details, order status, movie show times, and more from any telephone. Additionally, IVR solutions are increasingly used to place outbound calls to deliver or gather information for appointments, past due bills, and other time critical events and activities. [0010] FIG. 1 illustrates schematically a communication application environment. The communication application environment 10 includes one or more client interacting with a communication application server 200 in an application platform 100. The application platform 100 hosts an application specified by an application script 210 coded in object-oriented software. The communication application server 200 includes a browser 220 for interpreting and executing the application script 210. The execution of the application script invokes one or more server-side components 310 in the application server 200. Among the clients and the communication server, these components 310 provide services for call control, media control with one or more media server 230 and interactions with back-end systems 240 such as databases, and business logic and legacy systems such as CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning). One example of the platform is to host an IVR application which interacts with voice, text messaging and other clients in a multi-channel environment. [0011] The communication application platform provides a third-party call control between any numbers of clients 20, 22, 30. The application script 210 defines the communication application 300 and directs how a call is to be handled. For example, when a user makes a call through a voice client such as a handset 20 or a VoIP phone 22 to the IVR, the voice application script 210 associated with the call number is retrieved. The browser 220 executes or renders the retrieved voice application script to allow the user to interact with the voice application 300. [0012] Communication of Multimedia information among endpoints and a third-party call controller generally require call control and media control.

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	<i>See, e.g.,</i> de Castro 932 at [0075]: [0075] The call control model of the unified communication framework is designed for calls and media that are controlled by a 3rd party server application, such as PBX, IVR, Conferencing, and Call Center applications. It assumes all the calls have at least their signals controlled by the communication application. In most cases, media should preferably be controlled by the communication application as well. <i>See, e.g.,</i> de Castro 932 at [0114]- [0116]: [0114] FIG. 14 illustrates a network architecture suitable for practicing the present invention. Essentially, the network is considered as comprising a service domain 201 and a customer/developer domain 601. [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers. [0116] Similar to that shown in FIG. 13, application clients such as VoIP phones 30, is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to one of the communication application servers 200, such as App server 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of the communication application server executing the specific communication application. <i>See, e.g.,</i> de Castro 932 at [0039]: [0039] In a preferred embodiment, the communication application server is among a group of

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	similar communication application servers on the network to provide telephony and communication services to a plurality of customers with application scripts hosted on scripting engines. One or more communication API gateway for the group of communication application servers is deployed on the network to serve as a messaging broker between the plurality of customers with application scripts and the group of communication application servers. <i>See, e.g.,</i> de Castro 932 at [0130]: [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[h] publishing, after integrating each of	de Castro 932 discloses publishing, after integrating each of the plurality of API

Zilkr v Cisco Systems, Inc.
Eastern District of Texas No. 2:22-cv-00166-JRG-RSP

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
the plurality of API interfaces with the unified services platform, the definition of the service, the definition of the plan, and the one or more definitions of actions for each application service provider in a provisioning catalog.	interfaces with the unified services platform, the definition of the service, the definition of the plan, and the one or more definitions of actions for each application service provider in a provisioning catalog. <i>See, e.g., de Castro 932 at fig. 14:</i>



Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)												
	Registry 520 <table><tr><th colspan="3">SERVICE DOMAIN</th><th colspan="3">USER DOMAIN</th></tr><tr><td>App ServerID</td><td>App Server Properties/ Status</td><td>....</td><td>App ID</td><td>App Properties/ Status</td><td>....</td></tr></table> FIG. 19 [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers. <i>See, e.g., de Castro 932 at [0129]-[0130]:</i> [0129] FIG. 18 illustrates in more detail the communication API gateway shown in FIG. 14. One or more communication API gateway 500 serves as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. The communication API gateway includes a registry 520, a resource load balancer 530, and a messaging server 540. The resource load balance 530 is responsible for locating a communication application server 200 best suited for executing the commands from a given API client 630. In the preferred embodiment, the messaging server 540 is an XMPP server. Messaging under the XMPP protocol is expedient in that the persistent connection is more conducive to the call environment. Also, it has well developed protocol and facilities for federation between different messaging domains.	SERVICE DOMAIN			USER DOMAIN			App ServerID	App Server Properties/ Status		App ID	App Properties/ Status	
SERVICE DOMAIN			USER DOMAIN										
App ServerID	App Server Properties/ Status		App ID	App Properties/ Status									

Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. [0102] FIG. 13 illustrates a typical implementation of deploying communication applications on an IP network such as the internet. A plurality of application platforms 100 is deployed on the network to provide communication services when hosting any number of communication applications defined by the application scripts 210. Clients such as VoIP phones 30, application clients 40 and other terminal equipment is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to an application platform, such as application platform 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of an available application platform with the necessary resources to execute the specific communication application. Once application platform 1 is selected, the application server in the platform will retrieve the application script 210 for the specific application associated with the dialed number. This is accomplished by looking up another directory service DIR1 18 for a pointer to the URL of the application script. For example, the dialed number may be associated with application script 1 and it is retrieved into the application platform 1. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not

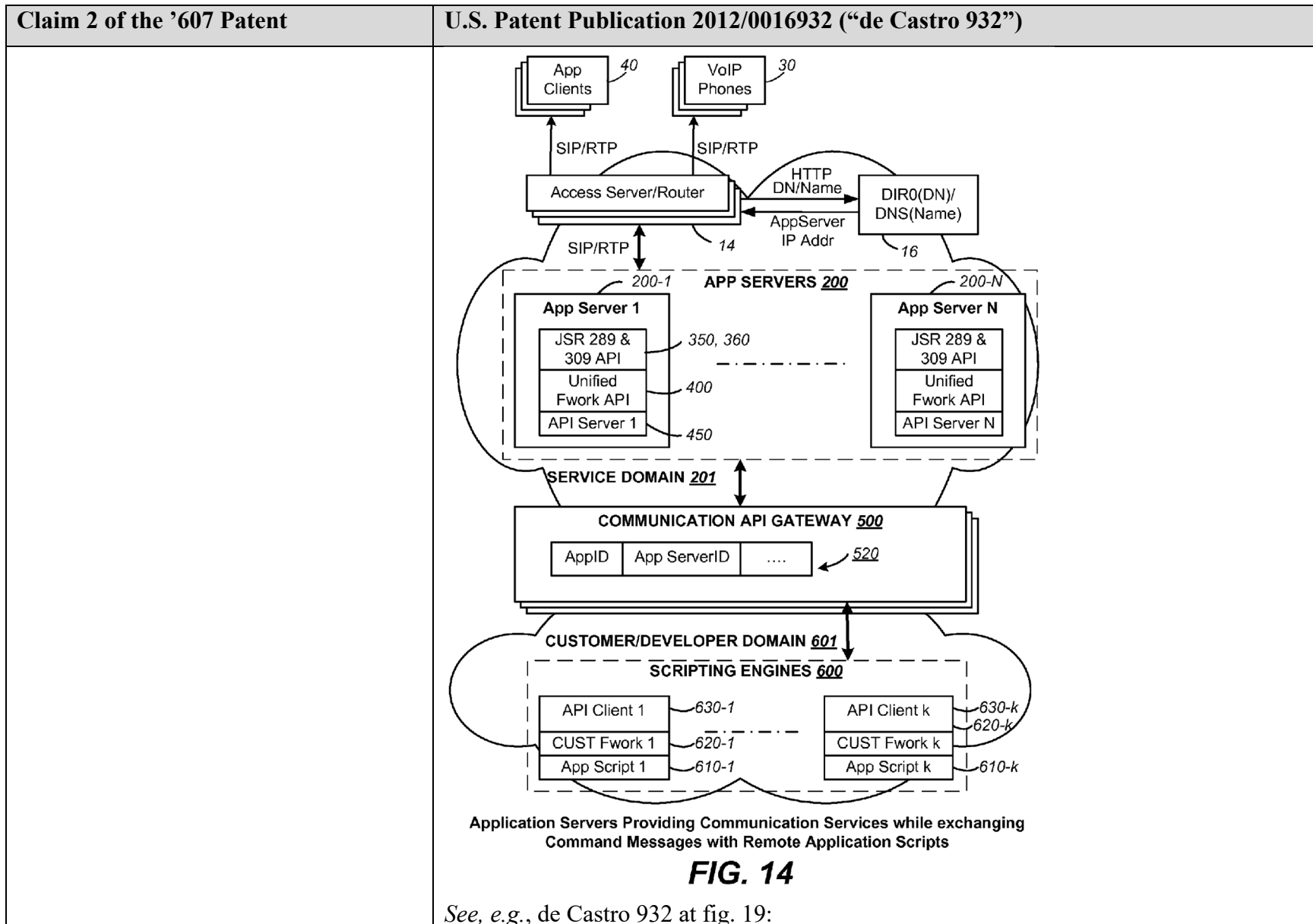
Claim 1 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.

2. Claim 2.

Claim 2 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
2. The method of claim 1, further comprising: receiving, from the network provider system, a selection of one or more services from the plurality of the application services provider systems; and	de Castro 932 discloses the method of claim 1, further comprising: receiving, from the network provider system, a selection of one or more services from the plurality of the application services provider systems. <i>See, e.g.,</i> de Castro 932 at [0102]-[0104]: [0102] FIG. 13 illustrates a typical implementation of deploying communication applications on an IP network such as the internet. A plurality of application platforms 100 is deployed on the network to provide communication services when hosting any number of communication applications defined by the application scripts 210. Clients such as VoIP phones 30, application clients 40 and other terminal equipment is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to an application platform, such as application platform 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of an available application platform with the necessary resources to execute the specific communication application. Once application platform 1 is selected, the application server in the platform will retrieve the application script 210 for the specific application

Claim 2 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	associated with the dialed number. This is accomplished by looking up another directory service DIR1 18 for a pointer to the URL of the application script. For example, the dialed number may be associated with application script 1 and it is retrieved into the application platform 1. [0103] Once the “dialed” application script 1 is retrieved into the application platform 1, the application server 200 (see also FIG. 1 and FIG. 4) will process the script. First a browser, compatible with the language the script, will interpret or compile the script into execution codes which will invoke the class objects and make function calls to the unified API. The execution is performed in a Java virtual machine supported by the application server 200. [0104] In this manner, communication applications such as voice-centers, IVR, voice-enabled self-help applications web bots can be deployed by such an application platform on the internet by leveraging web-like technologies and practices and employing standards-based protocol and languages. Variations of such communication platforms and network architecture of communication resources have been disclosed in U.S. Pat. No. 6,922,411, United States Patent Application Publication No. US 2011/0046960, and U.S. patent application Ser. No. 13/088,396 filed on Apr. 17, 2011, the entire disclosure of said publications and Applications are incorporated herein by reference. <i>See, e.g., de Castro 932 at [0114]-[0116]:</i> [0114] FIG. 14 illustrates a network architecture suitable for practicing the present invention. Essentially, the network is considered as comprising a service domain 201 and a customer/developer domain 601. [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers.

Claim 2 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0116] Similar to that shown in FIG. 13, application clients such as VoIP phones 30, is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to one of the communication application servers 200, such as App server 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of the communication application server executing the specific communication application. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[a] publishing the selection of the one or more services in a services catalog, wherein the services catalog is accessible by one or more subscribers of the network provider system.	de Castro 932 discloses publishing the selection of the one or more services in a services catalog, wherein the services catalog is accessible by one or more subscribers of the network provider system. <i>See, e.g.,</i> de Castro 932 at fig. 14:



Claim 2 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)												
	Registry 520 <table><tr><th colspan="3">SERVICE DOMAIN</th><th colspan="3">USER DOMAIN</th></tr><tr><td>App ServerID</td><td>App Server Properties/ Status</td><td>....</td><td>App ID</td><td>App Properties/ Status</td><td>....</td></tr></table> FIG. 19 See, e.g., de Castro 932 at [0115]: [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers. See, e.g., de Castro 932 at [0129]-[0130]: [0129] FIG. 18 illustrates in more detail the communication API gateway shown in FIG. 14. One or more communication API gateway 500 serves as a messaging broker between the plurality of customers with application scripts in the customer/developer domain 601 and the group of communication application servers 200 in the service domain 201. The communication API gateway includes a registry 520, a resource load balancer 530, and a messaging server 540. The resource load balance 530 is responsible for locating a communication application server 200 best suited for executing the commands from a given API client 630. In the preferred embodiment, the messaging server 540 is an XMPP server. Messaging under the XMPP protocol is expedient in that the persistent connection is more conducive to the call environment. Also, it has well developed protocol and facilities for federation between different messaging domains.	SERVICE DOMAIN			USER DOMAIN			App ServerID	App Server Properties/ Status		App ID	App Properties/ Status	
SERVICE DOMAIN			USER DOMAIN										
App ServerID	App Server Properties/ Status		App ID	App Properties/ Status									

Claim 2 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. <i>See, e.g., de Castro 932 at [0102]-[0104]:</i> [0102] FIG. 13 illustrates a typical implementation of deploying communication applications on an IP network such as the internet. A plurality of application platforms 100 is deployed on the network to provide communication services when hosting any number of communication applications defined by the application scripts 210. Clients such as VoIP phones 30, application clients 40 and other terminal equipment is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to an application platform, such as application platform 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of an available application platform with the necessary resources to execute the specific communication application. Once application platform 1 is selected, the application server in the platform will retrieve the application script 210 for the specific application associated with the dialed number. This is accomplished by looking up another directory service DIR1 18 for a pointer to the URL of the application script. For example, the dialed number may be associated with application script 1 and it is retrieved into the application platform 1. [0103] Once the “dialed” application script 1 is retrieved into the application platform 1, the

Claim 2 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	application server 200 (see also FIG. 1 and FIG. 4) will process the script. First a browser, compatible with the language the script, will interpret or compile the script into execution codes which will invoke the class objects and make function calls to the unified API. The execution is performed in a Java virtual machine supported by the application server 200. [0104] In this manner, communication applications such as voice-centers, IVR, voice-enabled self-help applications web bots can be deployed by such an application platform on the internet by leveraging web-like technologies and practices and employing standards-based protocol and languages. Variations of such communication platforms and network architecture of communication resources have been disclosed in U.S. Pat. No. 6,922,411, United States Patent Application Publication No. US 2011/0046960, and U.S. patent application Ser. No. 13/088,396 filed on Apr. 17, 2011, the entire disclosure of said publications and Applications are incorporated herein by reference. <i>See, e.g., de Castro 932 at [0114]-[0116]:</i> [0114] FIG. 14 illustrates a network architecture suitable for practicing the present invention. Essentially, the network is considered as comprising a service domain 201 and a customer/developer domain 601. [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers. [0116] Similar to that shown in FIG. 13, application clients such as VoIP phones 30, is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to one of the communication application servers 200, such as App server 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to

Claim 2 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	return the address of the communication application server executing the specific communication application. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.

3. Claim 3.

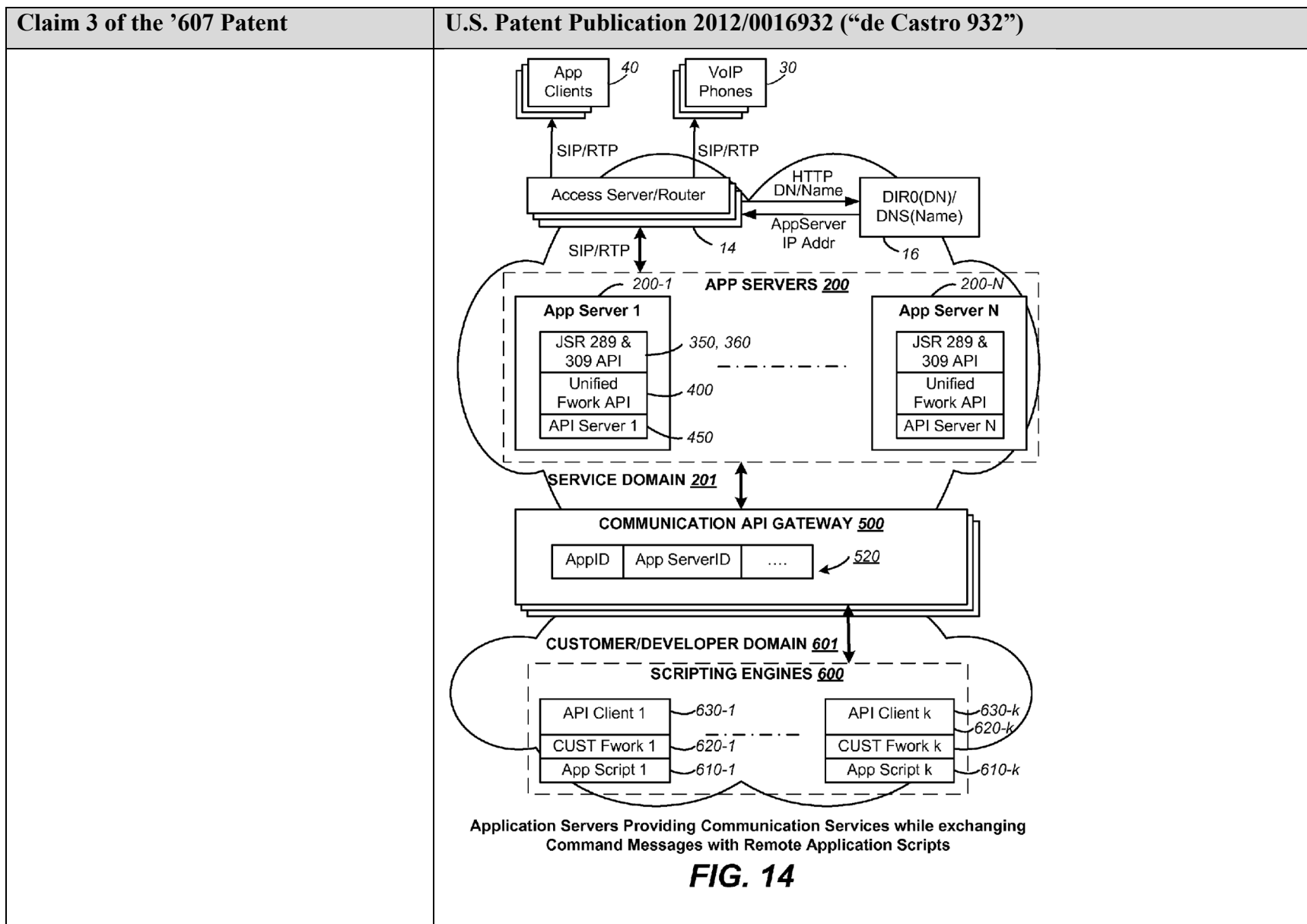
Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
3. The method of claim 1, further comprising: receiving a request for associating at least one service from the plurality of application service provider systems with a telephone number of a subscriber;	de Castro 932 discloses the method of claim 1, further comprising: receiving a request for associating at least one service from the plurality of application service provider systems with a telephone number of a subscriber. <i>See, e.g.,</i> de Castro 932 at figure 13:

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 ("de Castro 932")
	Application Platform having App Servers Providing Communication Services while Processing Application Scripts FIG. 13 <i>See, e.g., de Castro 932 at [0102]-[0104]:</i> [0102] FIG. 13 illustrates a typical implementation of deploying communication applications on an IP network such as the internet. A plurality of application platforms 100 is

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	deployed on the network to provide communication services when hosting any number of communication applications defined by the application scripts 210. Clients such as VoIP phones 30, application clients 40 and other terminal equipment is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to an application platform, such as application platform 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of an available application platform with the necessary resources to execute the specific communication application. Once application platform 1 is selected, the application server in the platform will retrieve the application script 210 for the specific application associated with the dialed number. This is accomplished by looking up another directory service DIR1 18 for a pointer to the URL of the application script. For example, the dialed number may be associated with application script 1 and it is retrieved into the application platform 1. [0103] Once the “dialed” application script 1 is retrieved into the application platform 1, the application server 200 (see also FIG. 1 and FIG. 4) will process the script. First a browser, compatible with the language the script, will interpret or compile the script into execution codes which will invoke the class objects and make function calls to the unified API. The execution is performed in a Java virtual machine supported by the application server 200. [0104] In this manner, communication applications such as voice-centers, IVR, voice-enabled self-help applications web bots can be deployed by such an application platform on the internet by leveraging web-like technologies and practices and employing standards-based protocol and languages. Variations of such communication platforms and network architecture of communication resources have been disclosed in U.S. Pat. No. 6,922,411, United States Patent Application Publication No. US 2011/0046960, and U.S. patent application Ser. No. 13/088,396 filed on Apr. 17, 2011, the entire disclosure of said publications and Applications are incorporated herein by reference. <i>See, e.g., de Castro 932 at [0116]:</i>

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0116] Similar to that shown in FIG. 13, application clients such as VoIP phones 30, is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to one of the communication application servers 200, such as App server 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of the communication application server executing the specific communication application. <i>See, e.g., de Castro 932 at [0124]-[0126]:</i> [0124] The call manager 260 manages the cooperation among the modules. When a call 1 is received into the communication application server 200, the call manager register the Call ID into a call ID registry 262 and start a call actor 1 270-1, and similarly for call 2, . . . , and call i. As command messages from the API clients 630 (see FIG. 14) are received by the application server 200, they are identified with a call actor and queued in an associated message queue 272, which are then applied to the call in order. As will be described in more detail later, the API server parses out the command in each message for execution in turn on the associate call actor. [0125] The admin module 280 includes sub modules such as a monitoring module 282, a statistics modules 284, a QoS (Quality of Service) modules 286 and Billing module 288. [0126] FIG. 16 illustrates in more detail the callID stored in the callID registry shown in FIG. 15. Essentially it includes a GUID (Globally Unique Identifier) and a server IP address. In this way, the call and the handling communication application server are readily identified. <i>See, e.g., de Castro 932 at [0139]-[0141]:</i> [0139] The Rayo protocol primarily deals with calls. Inbound calls originate from the PSTN or via SIP and are offered to Rayo clients via XMPP using a Jabber Identifier (JID). Each

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	call is in turn represented by its own unique JID allowing a two way conversation between the Rayo client and the server that's handling the call signaling and media. JID Format [0140] The JID follows a specific format. In XMPP the JID is constructed as <node>@<domain>/<resource> For Rayo, the <node> portion of the JID always represents the call ID. The <resource>, when present, represents the affected command ID. Incoming Calls [0141] <pre> <!-- Message comes from the Call's JID --> <presence to='16577@app.Rayo.net/1' from='9f00061@call.Rayo.net/1'> <offer xmlns='urn:xmpp:Rayo:1' to='tel:+18003211212' from='tel:+13058881212'> <!-- Signaling (e.g. SIP) Headers --> <header name=' Via' value='192.168.0.1' /> <header name='Contact' value='192.168.0.1' /> </offer> </presence> </pre> <i>See, e.g., de Castro 932 at fig. 14:</i>



Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	<i>See, e.g.,</i> de Castro 932 at [0006]- [0012]: [0006] An example platform for easy deployment of telephony applications is described in U.S. Pat. No. 6,922,411, which entire disclosure is incorporated herein by reference. Essentially, a networked telephony system allows users to deploy on the Internet computer telephony applications associated with designated telephone numbers. The telephony application is easily created by a user in XML (Extended Markup Language) with predefined telephony XML tags (e.g. VoiceXML) and easily deployed on a website. The telephony XML tags include those for call control and media manipulation. A call to anyone of these designated telephone numbers may originate from anyone of the networked telephone system such as the PSTN (Public Switched Telephone System), a wireless network, or the Internet. The call is received by an application gateway center (AGC) installed on the Internet. Analogous to a web browser, the AGC provides facility for retrieving the associated XML application from its website and processing the call accordingly. [0007] This type of telephony platform allows very powerful yet simple telephony applications to be built and deployed on the Internet. The following are some examples of the telephony applications deployed on this platform. A “Follow me, find me” application sequentially calls a series of telephone numbers as specified by a user until one of the numbers answers and then connects the call. Otherwise, it does something else such as takes a message or sends e-mail or sends the call to a call center, etc. In another example, a Telephonic Polling application looks up from a database the telephone numbers of a population to be polled. It then calls the numbers in parallel, limited only by the maximum number of concurrent sessions supported, and plays a series of interactive voice prompts/messages in response to the called party's responses and records the result in a database, etc. In another example, a Help Desk application plays a series of interactive voice prompts/messages in response to the called party's responses and possibly connects the call to a live agent as one option, etc. In yet another example, a Stock or Bank Transactions application plays a series of interactive voice prompts/messages in response to the called party's responses and conducts appropriate

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	transactions with a backend database or web application, etc. [0008] The latter examples are generally referred to as self-help applications. In the voice context, a self-help application is referred to as IVR. IVR refers to Interactive Voice Response and is a technology that automates interaction with telephone callers. Enterprises are increasingly turning to IVR to reduce the cost of common sales, service, collections, inquiry and support calls to and from their company. [0009] IVR solutions enable users using voice as a medium or other form of inputs through a voice channel to retrieve information including bank balances, flight schedules, product details, order status, movie show times, and more from any telephone. Additionally, IVR solutions are increasingly used to place outbound calls to deliver or gather information for appointments, past due bills, and other time critical events and activities. [0010] FIG. 1 illustrates schematically a communication application environment. The communication application environment 10 includes one or more client interacting with a communication application server 200 in an application platform 100. The application platform 100 hosts an application specified by an application script 210 coded in object-oriented software. The communication application server 200 includes a browser 220 for interpreting and executing the application script 210. The execution of the application script invokes one or more server-side components 310 in the application server 200. Among the clients and the communication server, these components 310 provide services for call control, media control with one or more media server 230 and interactions with back-end systems 240 such as databases, and business logic and legacy systems such as CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning). One example of the platform is to host an IVR application which interacts with voice, text messaging and other clients in a multi-channel environment. [0011] The communication application platform provides a third-party call control between any numbers of clients 20, 22, 30. The application script 210 defines the communication application 300 and directs how a call is to be handled. For example, when a user makes a call through a voice client such as a handset 20 or a VoIP phone 22 to the IVR, the

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	voice application script 210 associated with the call number is retrieved. The browser 220 executes or renders the retrieved voice application script to allow the user to interact with the voice application 300. [0012] Communication of Multimedia information among endpoints and a third-party call controller generally require call control and media control. <i>See, e.g., de Castro 932 at [0075]:</i> [0075] The call control model of the unified communication framework is designed for calls and media that are controlled by a 3rd party server application, such as PBX, IVR, Conferencing, and Call Center applications. It assumes all the calls have at least their signals controlled by the communication application. In most cases, media should preferably be controlled by the communication application as well. <i>See, e.g., de Castro 932 at [0114]- [0115]:</i> [0114] FIG. 14 illustrates a network architecture suitable for practicing the present invention. Essentially, the network is considered as comprising a service domain 201 and a customer/developer domain 601. [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers. <i>See, e.g., de Castro 932 at [0130]:</i> [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[a] associating the telephone number with the at least one service; and	de Castro 932 discloses associating the telephone number with the at least one service. <i>See, e.g.,</i> de Castro 932 at figure 13:

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 ("de Castro 932")
	Application Platform having App Servers Providing Communication Services while Processing Application Scripts FIG. 13 <i>See, e.g., de Castro 932 at [0102]-[0104]:</i> [0102] FIG. 13 illustrates a typical implementation of deploying communication applications on an IP network such as the internet. A plurality of application platforms 100 is

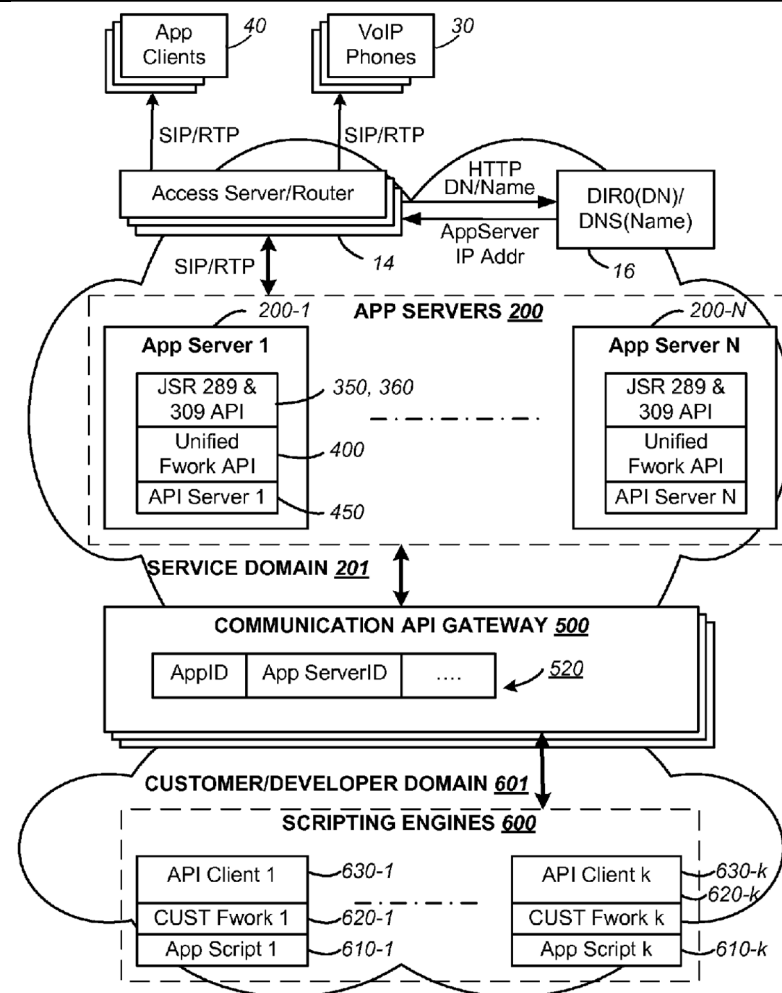
Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	deployed on the network to provide communication services when hosting any number of communication applications defined by the application scripts 210. Clients such as VoIP phones 30, application clients 40 and other terminal equipment is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to an application platform, such as application platform 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of an available application platform with the necessary resources to execute the specific communication application. Once application platform 1 is selected, the application server in the platform will retrieve the application script 210 for the specific application associated with the dialed number. This is accomplished by looking up another directory service DIR1 18 for a pointer to the URL of the application script. For example, the dialed number may be associated with application script 1 and it is retrieved into the application platform 1. [0103] Once the “dialed” application script 1 is retrieved into the application platform 1, the application server 200 (see also FIG. 1 and FIG. 4) will process the script. First a browser, compatible with the language the script, will interpret or compile the script into execution codes which will invoke the class objects and make function calls to the unified API. The execution is performed in a Java virtual machine supported by the application server 200. [0104] In this manner, communication applications such as voice-centers, IVR, voice-enabled self-help applications web bots can be deployed by such an application platform on the internet by leveraging web-like technologies and practices and employing standards-based protocol and languages. Variations of such communication platforms and network architecture of communication resources have been disclosed in U.S. Pat. No. 6,922,411, United States Patent Application Publication No. US 2011/0046960, and U.S. patent application Ser. No. 13/088,396 filed on Apr. 17, 2011, the entire disclosure of said publications and Applications are incorporated herein by reference. <i>See, e.g., de Castro 932 at [0116]:</i>

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0116] Similar to that shown in FIG. 13, application clients such as VoIP phones 30, is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to one of the communication application servers 200, such as App server 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of the communication application server executing the specific communication application. <i>See, e.g., de Castro 932 at [0124]-[0126]:</i> [0124] The call manager 260 manages the cooperation among the modules. When a call 1 is received into the communication application server 200, the call manager register the Call ID into a call ID registry 262 and start a call actor 1 270-1, and similarly for call 2, . . . , and call i. As command messages from the API clients 630 (see FIG. 14) are received by the application server 200, they are identified with a call actor and queued in an associated message queue 272, which are then applied to the call in order. As will be described in more detail later, the API server parses out the command in each message for execution in turn on the associate call actor. [0125] The admin module 280 includes sub modules such as a monitoring module 282, a statistics modules 284, a QoS (Quality of Service) modules 286 and Billing module 288. [0126] FIG. 16 illustrates in more detail the callID stored in the callID registry shown in FIG. 15. Essentially it includes a GUID (Globally Unique Identifier) and a server IP address. In this way, the call and the handling communication application server are readily identified. <i>See, e.g., de Castro 932 at [0139]-[0141]:</i> [0139] The Rayo protocol primarily deals with calls. Inbound calls originate from the PSTN or via SIP and are offered to Rayo clients via XMPP using a Jabber Identifier (JID). Each

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	call is in turn represented by its own unique JID allowing a two way conversation between the Rayo client and the server that's handling the call signaling and media. JID Format [0140] The JID follows a specific format. In XMPP the JID is constructed as <node>@<domain>/<resource> For Rayo, the <node> portion of the JID always represents the call ID. The <resource>, when present, represents the affected command ID. Incoming Calls [0141] <pre> <!-- Message comes from the Call's JID --> <presence to='16577@app.Rayo.net/1' from='9f00061@call.Rayo.net/1'> <offer xmlns='urn:xmpp:Rayo:1' to='tel:+18003211212' from='tel:+13058881212'> <!-- Signaling (e.g. SIP) Headers --> <header name=' Via' value='192.168.0.1' /> <header name='Contact' value='192.168.0.1' /> </offer> </presence> </pre> <i>See, e.g., de Castro 932 at fig. 14:</i>

Claim 3 of the '607 Patent

U.S. Patent Publication 2012/0016932 (“de Castro 932”)



Application Servers Providing Communication Services while exchanging
 Command Messages with Remote Application Scripts

FIG. 14

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	<i>See, e.g.,</i> de Castro 932 at [0006]- [0012]: [0006] An example platform for easy deployment of telephony applications is described in U.S. Pat. No. 6,922,411, which entire disclosure is incorporated herein by reference. Essentially, a networked telephony system allows users to deploy on the Internet computer telephony applications associated with designated telephone numbers. The telephony application is easily created by a user in XML (Extended Markup Language) with predefined telephony XML tags (e.g. VoiceXML) and easily deployed on a website. The telephony XML tags include those for call control and media manipulation. A call to anyone of these designated telephone numbers may originate from anyone of the networked telephone system such as the PSTN (Public Switched Telephone System), a wireless network, or the Internet. The call is received by an application gateway center (AGC) installed on the Internet. Analogous to a web browser, the AGC provides facility for retrieving the associated XML application from its website and processing the call accordingly. [0007] This type of telephony platform allows very powerful yet simple telephony applications to be built and deployed on the Internet. The following are some examples of the telephony applications deployed on this platform. A “Follow me, find me” application sequentially calls a series of telephone numbers as specified by a user until one of the numbers answers and then connects the call. Otherwise, it does something else such as takes a message or sends e-mail or sends the call to a call center, etc. In another example, a Telephonic Polling application looks up from a database the telephone numbers of a population to be polled. It then calls the numbers in parallel, limited only by the maximum number of concurrent sessions supported, and plays a series of interactive voice prompts/messages in response to the called party's responses and records the result in a database, etc. In another example, a Help Desk application plays a series of interactive voice prompts/messages in response to the called party's responses and possibly connects the call to a live agent as one option, etc. In yet another example, a Stock or Bank Transactions application plays a series of interactive voice prompts/messages in response to the called party's responses and conducts appropriate

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	transactions with a backend database or web application, etc. [0008] The latter examples are generally referred to as self-help applications. In the voice context, a self-help application is referred to as IVR. IVR refers to Interactive Voice Response and is a technology that automates interaction with telephone callers. Enterprises are increasingly turning to IVR to reduce the cost of common sales, service, collections, inquiry and support calls to and from their company. [0009] IVR solutions enable users using voice as a medium or other form of inputs through a voice channel to retrieve information including bank balances, flight schedules, product details, order status, movie show times, and more from any telephone. Additionally, IVR solutions are increasingly used to place outbound calls to deliver or gather information for appointments, past due bills, and other time critical events and activities. [0010] FIG. 1 illustrates schematically a communication application environment. The communication application environment 10 includes one or more client interacting with a communication application server 200 in an application platform 100. The application platform 100 hosts an application specified by an application script 210 coded in object-oriented software. The communication application server 200 includes a browser 220 for interpreting and executing the application script 210. The execution of the application script invokes one or more server-side components 310 in the application server 200. Among the clients and the communication server, these components 310 provide services for call control, media control with one or more media server 230 and interactions with back-end systems 240 such as databases, and business logic and legacy systems such as CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning). One example of the platform is to host an IVR application which interacts with voice, text messaging and other clients in a multi-channel environment. [0011] The communication application platform provides a third-party call control between any numbers of clients 20, 22, 30. The application script 210 defines the communication application 300 and directs how a call is to be handled. For example, when a user makes a call through a voice client such as a handset 20 or a VoIP phone 22 to the IVR, the

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	voice application script 210 associated with the call number is retrieved. The browser 220 executes or renders the retrieved voice application script to allow the user to interact with the voice application 300. [0012] Communication of Multimedia information among endpoints and a third-party call controller generally require call control and media control. <i>See, e.g., de Castro 932 at [0075]:</i> [0075] The call control model of the unified communication framework is designed for calls and media that are controlled by a 3rd party server application, such as PBX, IVR, Conferencing, and Call Center applications. It assumes all the calls have at least their signals controlled by the communication application. In most cases, media should preferably be controlled by the communication application as well. <i>See, e.g., de Castro 932 at [0114]- [0115]:</i> [0114] FIG. 14 illustrates a network architecture suitable for practicing the present invention. Essentially, the network is considered as comprising a service domain 201 and a customer/developer domain 601. [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers. <i>See, e.g., de Castro 932 at [0130]:</i> [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[b] invoking the at least one service in response to a communication event with the telephone number.	de Castro 932 discloses invoking the at least one service in response to a communication event with the telephone number. <i>See, e.g.,</i> de Castro 932 at [0102]-[0104]: [0102] FIG. 13 illustrates a typical implementation of deploying communication applications on an IP network such as the internet. A plurality of application platforms 100 is deployed on the network to provide communication services when hosting any number of communication applications defined by the application scripts 210. Clients such as VoIP phones 30, application clients 40 and other terminal equipment is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to an application platform, such as application platform 1. The routing is performed by looking up a directory

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	service DIR0 16 with the IP address/dialed number in order to return the address of an available application platform with the necessary resources to execute the specific communication application. Once application platform 1 is selected, the application server in the platform will retrieve the application script 210 for the specific application associated with the dialed number. This is accomplished by looking up another directory service DIR1 18 for a pointer to the URL of the application script. For example, the dialed number may be associated with application script 1 and it is retrieved into the application platform 1. [0103] Once the “dialed” application script 1 is retrieved into the application platform 1, the application server 200 (see also FIG. 1 and FIG. 4) will process the script. First a browser, compatible with the language the script, will interpret or compile the script into execution codes which will invoke the class objects and make function calls to the unified API. The execution is performed in a Java virtual machine supported by the application server 200. [0104] In this manner, communication applications such as voice-centers, IVR, voice-enabled self-help applications web bots can be deployed by such an application platform on the internet by leveraging web-like technologies and practices and employing standards-based protocol and languages. Variations of such communication platforms and network architecture of communication resources have been disclosed in U.S. Pat. No. 6,922,411, United States Patent Application Publication No. US 2011/0046960, and U.S. patent application Ser. No. 13/088,396 filed on Apr. 17, 2011, the entire disclosure of said publications and Applications are incorporated herein by reference. <i>See, e.g., de Castro 932 at [0006]- [0012]:</i> [0006] An example platform for easy deployment of telephony applications is described in U.S. Pat. No. 6,922,411, which entire disclosure is incorporated herein by reference. Essentially, a networked telephony system allows users to deploy on the Internet computer telephony applications associated with designated telephone numbers. The telephony application is easily created by a user in XML (Extended Markup Language)

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	with predefined telephony XML tags (e.g. VoiceXML) and easily deployed on a website. The telephony XML tags include those for call control and media manipulation. A call to anyone of these designated telephone numbers may originate from anyone of the networked telephone system such as the PSTN (Public Switched Telephone System), a wireless network, or the Internet. The call is received by an application gateway center (AGC) installed on the Internet. Analogous to a web browser, the AGC provides facility for retrieving the associated XML application from its website and processing the call accordingly. [0007] This type of telephony platform allows very powerful yet simple telephony applications to be built and deployed on the Internet. The following are some examples of the telephony applications deployed on this platform. A “Follow me, find me” application sequentially calls a series of telephone numbers as specified by a user until one of the numbers answers and then connects the call. Otherwise, it does something else such as takes a message or sends e-mail or sends the call to a call center, etc. In another example, a Telephonic Polling application looks up from a database the telephone numbers of a population to be polled. It then calls the numbers in parallel, limited only by the maximum number of concurrent sessions supported, and plays a series of interactive voice prompts/messages in response to the called party's responses and records the result in a database, etc. In another example, a Help Desk application plays a series of interactive voice prompts/messages in response to the called party's responses and possibly connects the call to a live agent as one option, etc. In yet another example, a Stock or Bank Transactions application plays a series of interactive voice prompts/messages in response to the called party's responses and conducts appropriate transactions with a backend database or web application, etc. [0008] The latter examples are generally referred to as self-help applications. In the voice context, a self-help application is referred to as IVR. IVR refers to Interactive Voice Response and is a technology that automates interaction with telephone callers. Enterprises are increasingly turning to IVR to reduce the cost of common sales, service, collections, inquiry and support calls to and from their company. [0009]

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	IVR solutions enable users using voice as a medium or other form of inputs through a voice channel to retrieve information including bank balances, flight schedules, product details, order status, movie show times, and more from any telephone. Additionally, IVR solutions are increasingly used to place outbound calls to deliver or gather information for appointments, past due bills, and other time critical events and activities. [0010] FIG. 1 illustrates schematically a communication application environment. The communication application environment 10 includes one or more client interacting with a communication application server 200 in an application platform 100. The application platform 100 hosts an application specified by an application script 210 coded in object-oriented software. The communication application server 200 includes a browser 220 for interpreting and executing the application script 210. The execution of the application script invokes one or more server-side components 310 in the application server 200. Among the clients and the communication server, these components 310 provide services for call control, media control with one or more media server 230 and interactions with back-end systems 240 such as databases, and business logic and legacy systems such as CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning). One example of the platform is to host an IVR application which interacts with voice, text messaging and other clients in a multi-channel environment. [0011] The communication application platform provides a third-party call control between any numbers of clients 20, 22, 30. The application script 210 defines the communication application 300 and directs how a call is to be handled. For example, when a user makes a call through a voice client such as a handset 20 or a VoIP phone 22 to the IVR, the voice application script 210 associated with the call number is retrieved. The browser 220 executes or renders the retrieved voice application script to allow the user to interact with the voice application 300. [0012] Communication of Multimedia information among endpoints and a third-party call controller generally require call control and media control. <i>See, e.g., de Castro 932 at [0075]:</i>

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0075] The call control model of the unified communication framework is designed for calls and media that are controlled by a 3rd party server application, such as PBX, IVR, Conferencing, and Call Center applications. It assumes all the calls have at least their signals controlled by the communication application. In most cases, media should preferably be controlled by the communication application as well. <i>See, e.g.,</i> de Castro 932 at [0114]- [0115]: [0114] FIG. 14 illustrates a network architecture suitable for practicing the present invention. Essentially, the network is considered as comprising a service domain 201 and a customer/developer domain 601. [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers. <i>See, e.g.,</i> de Castro 932 at [0130]: [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630.

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	<i>See, e.g.,</i> de Castro 932 at [0139]-[0141]: [0139] The Rayo protocol primarily deals with calls. Inbound calls originate from the PSTN or via SIP and are offered to Rayo clients via XMPP using a Jabber Identifier (JID). Each call is in turn represented by its own unique JID allowing a two way conversation between the Rayo client and the server that's handling the call signaling and media. JID Format [0140] The JID follows a specific format. In XMPP the JID is constructed as <node>@<domain>/<resource> For Rayo, the <node> portion of the JID always represents the call ID. The <resource>, when present, represents the affected command ID. Incoming Calls [0141] <pre> <!-- Message comes from the Call's JID --> <presence to='16577@app.Rayo.net/1' from='9f00061@call.Rayo.net/1'> <offer xmlns='urn:xmpp:Rayo:1' to='tel:+18003211212' from='tel:+13058881212'> <!-- Signaling (e.g. SIP) Headers --> <header name=' Via' value='192.168.0.1' /> <header name='Contact' value='192.168.0.1' /> </offer> </presence> </pre> To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would

Claim 3 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.

4. Claim 5.

Claim 5 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
5. The method of claim 3, wherein invoking the at least one service in response to the communication event with the telephone number comprises: invoking the at least one service associated with the telephone number in response to a first triggering event, wherein the first service is provided by a first service provided by a first application service provider system of the plurality of application service provider systems.	de Castro 932 discloses the method of claim 3, wherein invoking the at least one service in response to the communication event with the telephone number comprises: invoking the at least one service associated with the telephone number in response to a first triggering event, wherein the first service is provided by a first service provided by a first application service provider system of the plurality of application service provider systems. <i>See, e.g., de Castro 932 at [0102]-[0104]:</i> [0102] FIG. 13 illustrates a typical implementation of deploying communication applications on an IP network such as the internet. A plurality of application platforms 100 is deployed on the network to provide communication services when hosting any number of communication applications defined by the application scripts 210. Clients such as VoIP phones 30, application clients 40 and other terminal equipment is able to access a specific communication application by calling a dialed number or an URL. The call is routed by an access server/router 14 into the IP network 10 to an application platform, such as application platform 1. The routing is performed by looking up a directory service DIR0 16 with the IP address/dialed number in order to return the address of an available application platform with the necessary resources to execute the specific communication application. Once application platform 1 is selected, the application server in the platform will retrieve the application script 210 for the specific application associated with the dialed number. This is accomplished by looking up another directory

Claim 5 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	service DIR1 18 for a pointer to the URL of the application script. For example, the dialed number may be associated with application script 1 and it is retrieved into the application platform 1. [0103] Once the “dialed” application script 1 is retrieved into the application platform 1, the application server 200 (see also FIG. 1 and FIG. 4) will process the script. First a browser, compatible with the language the script, will interpret or compile the script into execution codes which will invoke the class objects and make function calls to the unified API. The execution is performed in a Java virtual machine supported by the application server 200. [0104] In this manner, communication applications such as voice-centers, IVR, voice-enabled self-help applications web bots can be deployed by such an application platform on the internet by leveraging web-like technologies and practices and employing standards-based protocol and languages. Variations of such communication platforms and network architecture of communication resources have been disclosed in U.S. Pat. No. 6,922,411, United States Patent Application Publication No. US 2011/0046960, and U.S. patent application Ser. No. 13/088,396 filed on Apr. 17, 2011, the entire disclosure of said publications and Applications are incorporated herein by reference. <i>See, e.g., de Castro 932 at [0006]- [0012]:</i> [0006] An example platform for easy deployment of telephony applications is described in U.S. Pat. No. 6,922,411, which entire disclosure is incorporated herein by reference. Essentially, a networked telephony system allows users to deploy on the Internet computer telephony applications associated with designated telephone numbers. The telephony application is easily created by a user in XML (Extended Markup Language) with predefined telephony XML tags (e.g. VoiceXML) and easily deployed on a website. The telephony XML tags include those for call control and media manipulation. A call to anyone of these designated telephone numbers may originate from anyone of the networked telephone system such as the PSTN (Public Switched Telephone System), a wireless network, or the Internet. The call is received by an

Claim 5 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	application gateway center (AGC) installed on the Internet. Analogous to a web browser, the AGC provides facility for retrieving the associated XML application from its website and processing the call accordingly. [0007] This type of telephony platform allows very powerful yet simple telephony applications to be built and deployed on the Internet. The following are some examples of the telephony applications deployed on this platform. A “Follow me, find me” application sequentially calls a series of telephone numbers as specified by a user until one of the numbers answers and then connects the call. Otherwise, it does something else such as takes a message or sends e-mail or sends the call to a call center, etc. In another example, a Telephonic Polling application looks up from a database the telephone numbers of a population to be polled. It then calls the numbers in parallel, limited only by the maximum number of concurrent sessions supported, and plays a series of interactive voice prompts/messages in response to the called party's responses and records the result in a database, etc. In another example, a Help Desk application plays a series of interactive voice prompts/messages in response to the called party's responses and possibly connects the call to a live agent as one option, etc. In yet another example, a Stock or Bank Transactions application plays a series of interactive voice prompts/messages in response to the called party's responses and conducts appropriate transactions with a backend database or web application, etc. [0008] The latter examples are generally referred to as self-help applications. In the voice context, a self-help application is referred to as IVR. IVR refers to Interactive Voice Response and is a technology that automates interaction with telephone callers. Enterprises are increasingly turning to IVR to reduce the cost of common sales, service, collections, inquiry and support calls to and from their company. [0009] IVR solutions enable users using voice as a medium or other form of inputs through a voice channel to retrieve information including bank balances, flight schedules, product details, order status, movie show times, and more from any telephone. Additionally, IVR solutions are increasingly used to place outbound calls to deliver or gather information for appointments, past due bills, and other time critical events and activities.

Claim 5 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	[0010] FIG. 1 illustrates schematically a communication application environment. The communication application environment 10 includes one or more client interacting with a communication application server 200 in an application platform 100. The application platform 100 hosts an application specified by an application script 210 coded in object-oriented software. The communication application server 200 includes a browser 220 for interpreting and executing the application script 210. The execution of the application script invokes one or more server-side components 310 in the application server 200. Among the clients and the communication server, these components 310 provide services for call control, media control with one or more media server 230 and interactions with back-end systems 240 such as databases, and business logic and legacy systems such as CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning). One example of the platform is to host an IVR application which interacts with voice, text messaging and other clients in a multi-channel environment. [0011] The communication application platform provides a third-party call control between any numbers of clients 20, 22, 30. The application script 210 defines the communication application 300 and directs how a call is to be handled. For example, when a user makes a call through a voice client such as a handset 20 or a VoIP phone 22 to the IVR, the voice application script 210 associated with the call number is retrieved. The browser 220 executes or renders the retrieved voice application script to allow the user to interact with the voice application 300. [0012] Communication of Multimedia information among endpoints and a third-party call controller generally require call control and media control. <i>See, e.g., de Castro 932 at [0075]:</i> [0075] The call control model of the unified communication framework is designed for calls and media that are controlled by a 3rd party server application, such as PBX, IVR, Conferencing, and Call Center applications. It assumes all the calls have at least their signals controlled by the communication application. In most cases, media should

Claim 5 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	preferably be controlled by the communication application as well. <i>See, e.g.,</i> de Castro 932 at [0114]- [0115]: [0114] FIG. 14 illustrates a network architecture suitable for practicing the present invention. Essentially, the network is considered as comprising a service domain 201 and a customer/developer domain 601. [0115] In the service domain 201 is a plurality of communication application servers 200, such as App Server 1 to App Server N. As in FIG. 13, the communication application servers 200 provide telephony and communication services to applications and allows application clients 40 such as, for example, VoIP phones to access the applications hosted in the communication application servers. <i>See, e.g.,</i> de Castro 932 at [0130]: [0130] FIG. 19 illustrates the main information maintained in the registry of the communication API gateway. The XMPP server 540 in the communication API gateway 500 mediates between the plurality of communication application servers 200 (each having an XMPP server 458) in the service domain 201 and the plurality of API clients 630 (each having an XMPP client not shown explicitly). The XMPP server 540 conducts federation by referencing the information in the registry 520, which registers the communication application servers 200 in the service domain and the customers' applications in the customer/developer domain 601. Also logged are the properties and status associated with each communication application server 200 and the properties and status associated with each API clients 630. <i>See, e.g.,</i> de Castro 932 at [0122]: [0122] Thus, the communication application server 200 communicates with the scripting engines 600 using a messaging API protocol. When a command carried in a message is received at the communication application server 200, the API server 450 will parse out

Claim 5 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	the command in accordance with the messaging API protocol and allow the command to be queued with the associated call (actor) to be executed. <i>See, e.g., de Castro 932 at [0141]:</i> [0141] The Rayo client can now control the call by using one of the following commands. <!-- Accept (e.g. SIP 180/Ringing). Only applies to incoming calls. --> <pre> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <accept xmlns='urn:xmpp:Rayo:1'> <!-- Sample Headers (optional) --> <header name="x-skill" value="agent" /> <header name="x-customer-id" value="8877" /> </accept> </iq> <!-- Answer (e.g. SIP 200/OK). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <answer xmlns='urn:xmpp:Rayo:1'> <!-- Sample Headers (optional) --> <header name="x-skill" value="agent" /> <header name="x-customer-id" value="8877" /> </answer> </iq> <!-- Redirect(e.g. SIP 302/Redirect). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <redirect to='tel:+14152226789' xmlns='urn:xmpp:Rayo:1'> <!-- Sample Headers(optional) --> </pre>

Claim 5 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	<pre> <header name="x-skill" value="agent" /> <header name="x-customer-id" value="8877" /> </redirect> </iq> A call can also be rejected. Rejections can include an optional rejection reason. Rejection reasons are one of <busy/>, <decline/> or <error/>. If not specified, <decline/> is used as the default reason. <!-- Decline (.g. SIP 603/Decline). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <reject xmlns='urn:xmpp:Rayo:1'> <decline /> <!-- Sample Headers (optional) --> <header name="x-reason-internal" value="bad-skill" /> </reject> </iq> <!-- Busy (.g. SIP 486/Busy). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <reject xmlns='urn:xmpp:Rayo:1'> <busy /> <!-- Sample Headers (optional) --> <header name="x-busy-detail" value="out of licenses" /> </reject> </iq> <!-- Error (.g. SIP 500/Internal Server Error). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> </pre>

Claim 5 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	<pre> <reject xmlns='urn:xmpp:Rayo:1'> <error /> <!-- Sample Headers (optional) --> <header name="x-error-detail" value="soem descriptive error message" /> </reject> </iq> </pre> To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.

5. Claim 6.

Claim 6 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
6. The method of claim 5, wherein the triggering event comprises a representational state transfer (REST) upon sending a message, a REST upon receiving a message, a REST before making a call, a REST after making a call, a REST before receiving a call, a REST after receiving a call, a Session	de Castro 932 discloses the method of claim 5, wherein the triggering event comprises a representational state transfer (REST) upon sending a message, a REST upon receiving a message, a REST before making a call, a REST after making a call, a REST before receiving a call, a REST after receiving a call, a Session Initiation Protocols (SIP) on receiving a call, a SIP on making a call, or any combination thereof. <i>See, e.g., de Castro 932 at [0139]:</i>

Claim 6 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
Initiation Protocols (SIP) on receiving a call, a SIP on making a call, or any combination thereof.	[0139] The Rayo protocol primarily deals with calls. Inbound calls originate from the PSTN or via SIP and are offered to Rayo clients via XMPP using a Jabber Identifier (JID). Each call is in turn represented by its own unique JID allowing a two way conversation between the Rayo client and the server that's handling the call signaling and media. [0141] The Rayo client can now control the call by using one of the following commands. <!-- Accept (e.g. SIP 180/Ringing). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <accept xmlns='urn:xmpp:Rayo:1'> <!-- Sample Headers (optional) --> <header name="x-skill" value="agent" /> <header name="x-customer-id" value="8877" /> </accept> </iq> <!-- Answer (e.g. SIP 200/OK). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <answer xmlns='urn:xmpp:Rayo:1'> <!-- Sample Headers (optional) --> <header name="x-skill" value="agent" /> <header name="x-customer-id" value="8877" /> </answer> </iq> <!-- Redirect(e.g. SIP 302/Redirect). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'>

Claim 6 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	<pre> <redirect to='tel:+14152226789' xmlns='urn:xmpp:Rayo:1'> <!-- Sample Headers(optional) --> <header name="x-skill" value="agent" /> <header name="x-customer-id" value="8877" /> </redirect> </iq> A call can also be rejected. Rejections can include an optional rejection reason. Rejection reasons are one of <busy/>, <decline/> or <error/>. If not specified, <decline/> is used as the default reason. <!-- Decline (.g. SIP 603/Decline). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <reject xmlns='urn:xmpp:Rayo:1'> <decline /> <!-- Sample Headers (optional) --> <header name="x-reason-internal" value="bad-skill" /> </reject> </iq> <!-- Busy (.g. SIP 486/Busy). Only applies to incoming calls. --> <iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <reject xmlns='urn:xmpp:Rayo:1'> <busy /> <!-- Sample Headers (optional) --> <header name="x-busy-detail" value="out of licenses" /> </reject> </iq> <!-- Error (.g. SIP 500/Internal Server Error). Only applies to incoming calls. --> </pre>

Claim 6 of the '607 Patent	U.S. Patent Publication 2012/0016932 (“de Castro 932”)
	<pre><iq type='set' to='9f00061@call.Rayo.net/1' from='16577@app.Rayo.net/1'> <reject xmlns='urn:xmpp:Rayo:1'> <error /> <!-- Sample Headers (optional) --> <header name="x-error-detail" value="soem descriptive error message" /> </reject> </iq></pre> To the extent it is found that de Castro 932 does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that de Castro 932 does not anticipate this claim, de Castro 932 renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine de Castro 932 with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.