

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US2007/071408

International filing date: 15 June 2007 (15.06.2007)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/814,150
Filing date: 16 June 2006 (16.06.2006)

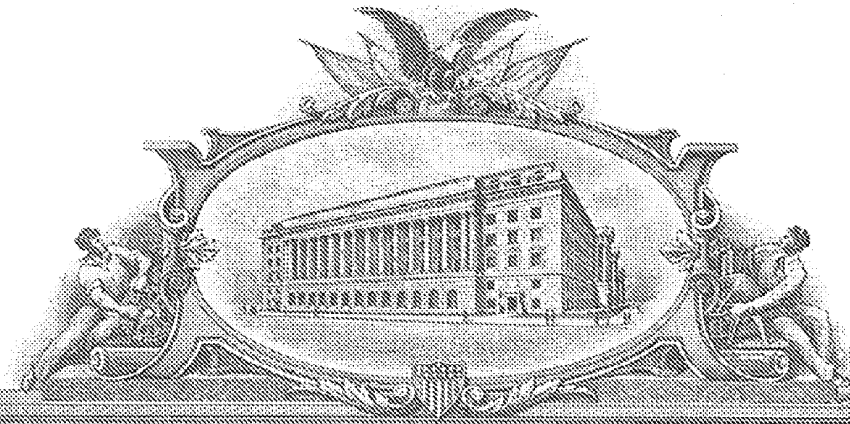
Date of receipt at the International Bureau: 29 August 2007 (29.08.2007)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

1653165



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

August 23, 2007

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: 60/814,150

FILING DATE: June 16, 2006

RELATED PCT APPLICATION NUMBER: PCT/US07/71408

THE COUNTRY CODE AND NUMBER OF YOUR PRIORITY APPLICATION, TO BE USED FOR FILING ABROAD UNDER THE PARIS CONVENTION, IS US60/814,150



Certified by

Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

061606



04788 U.S. PTO

Provisional Application COVER SHEET

Attorney Docket No.: NELTEC.003PR

First Named Inventor: Colin C. Quon

Title: ONLINE COMMUNITY AND IDENTITY EXTENSION TO
MOBILE DEVICES

Express Mail Label No.: EV 915 027 344 US

113374 U.S. PTO
60/814150

061606

Direct all correspondence to Customer No.: 20995

Date: June 16, 2006

Page 1 of 2

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

The following enclosures are transmitted herewith to be filed in the Provisional Patent Application of:

Inventors:

First Inventor : Colin C. Quon
Residence Address : 2031 E. 23rd Avenue, Vancouver, BC V5N 2T9

Second Inventor : Jeff A. LaPorte
Residence Address : #1611-3528 Vanness Avenue, Vancouver, BC V5R 6G4

APPLICATION ELEMENTS:

(X) Specification in 72 pages.

(X) Return prepaid postcard.

FILING FEES:

FEE CALCULATION				
FEE TYPE		LARGE FEE	CALCULATION	TOTAL
Basic Filing	37 CFR § 1.16(d)	1005 (\$200)		\$200
Application Size Fee	72 - 100 = 0	1681 (\$250) ⁺	0 x 250 =	\$0
			TOTAL FEE DUE	\$200

(X) A check in the amount of \$200 is enclosed to cover the filing fee.

This invention WAS NOT made by an agency of the United States Government or under a contract with an agency of the United States Government.

Provisional Application COVER SHEET	Attorney Docket No.: NELTEC.003PR
	First Named Inventor: Colin C. Quon
	Title: ONLINE COMMUNITY AND IDENTITY EXTENSION TO MOBILE DEVICES
	Express Mail Label No.: EV 915 027 344 US
Direct all correspondence to Customer No.: 20995	
Date: June 16, 2006	
Page 2 of 2	

Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Account No. 11-1410.



Karen J. Lenker
Registration No. 54,618
Agent of Record
Customer No. 20,995
(949) 760-0404

2690269
061606

Knobbe Martens Olson & Bear LLP

Intellectual Property Law

2040 Main Street
Fourteenth Floor
Irvine, CA 92614
Tel 949-760-0404
Fax 949-760-9502
www.kmob.com

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

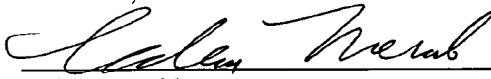
CERTIFICATE OF MAILING BY "EXPRESS MAIL"

Attorney Docket No. : NELTEC.003PR
Applicant(s) : Quon, et al.
For : ONLINE COMMUNITY AND IDENTITY
EXTENSION TO MOBILE DEVICES
Agent : Karen J. Lenker
"Express Mail" Label No. : EV 915 027 344 US
Date of Deposit : June 16, 2006

I hereby certify that the accompanying

Transmittal letter; specification in 72 pages; Check(s) for Filing Fee(s); Return
Prepaid Postcard

are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and are addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.


Nelson Merida

2690328
061606

San Diego
619-235-8550

San Francisco
415-954-4114

Los Angeles
310-551-3450

Riverside
951-781-9231

San Luis Obispo
805-547-5580

ONLINE COMMUNITY AND IDENTITY EXTENSION TO MOBILE DEVICESField of the Invention

This invention relates generally to the field of extending communications services between the user's mobile device and one or more of the user's online communities using the user's online names and identities, and more particularly to extending communications services between the user's mobile device and online dating and matchmaking.

Introduction

The disclosure provided in the following pages describe examples of some embodiments of the invention. The designs, figures, and description are non-limiting examples of some embodiments of the invention. In an embodiment, systems and methods relate to extending online identities and communities across disassociated networks and through various communications backplanes and allowing the services to be consumed on a mobile device. For example, the service can be extended to other end user devices and services such as PCs, TVs set-top boxes or gaming consoles. Moreover, disclosed advantages and benefits can apply to only some embodiments of the invention, and should not used to limit the invention.

Further descriptions of several embodiments of the invention are available in previously filed patent applications, including U.S. Patent Application No. 60/695,311, titled DYNAMIC AND CONTEXT DRIVEN CALL CONTROL AND SERVICE BRIDGING, filed on June 30, 2005, the entirety of which is hereby incorporated herein by reference and which is attached hereto as an appendix. The references cited herein are also incorporated by reference.

Background

Typically, person-to-person interaction, such as dating, starts with an intent and some context and then progresses through a number of stages towards live communications, and then, face-to-face meetings. Presently, a number of communications tools can be used for a caller to signal to a callee the communications intent. For example, communications tools that support disposable identities such as instant messaging (synchronous requiring both parties to be on a buddy list) and email (asynchronous and deliverable to any valid email address) can be used to signal caller intent such as "can we talk" or "I want to know more about the project status". Similarly, other communications tools with less disposable identities such as SMS (asynchronous where the callee does not have to answer), voice mail (asynchronous where the caller did not answer), and voice phone call (synchronous where the callee has to answer) can also be used by the caller to communicate intent or expression of interest in communicating with the callee. However, none of these tools support transient short-

lived communications sessions that allow users to easily exchange interaction context and do so without revealing significant personal information such as a personal mobile phone number.

Another problem is that these tools tend to tie the identity to the communications service. Suppose Joe on MySpace (or eBay or whatever community) has a service identity Jack on Skype (or Google or AOL or another communications services backplane) and that Jane on MySpace similarly has an identity Janice on Skype. If Joe clicks on Jane in MySpace, they can communicate through the backplane via their Skype identities as Jack to Janice, or they would have to create a new alias in Skype as Joe' and Jane'. In either case, it would be Jack or Joe' that is communicating with Janice or Jane' and not Joe communicating with Jane.

The problem solved by an embodiment of this invention is a method to allow two users to initiate communications with each other through original caller and callee identities through multi-modal communications services and alternative communications backplanes. It also provides a capability for a user to publish or send a session invitation handle called "EQO Me" which allows users to interact with rich context using online identities and through multi-modal communications services, such as messaging, IM chat, and voice.

An embodiment of this invention is related to the dynamic and context-driven call control technology (called a nudge) that decouples the call request or "analogous to a tap on the shoulder" from the actual call and call path such that the original calling identities are used for call initiation, call acceptance, and call routing. A number of methods are available to alias the caller identities or retain the original caller identities, and then statically or dynamically route the actual call and multimedia session between the participating calling parties.

An embodiment of the invention illustrates the use of the nudge technology to not only enable dynamic call disposition but also allowing the initiating call nudge to be initiated with caller context such as, who the caller is, why the caller is calling, the importance of the call, the context of the call, and other related information, such as the call referring party; and also enables such context and intent to be published or pushed as an "EQO Me" message.

EQO Communications, an operating entity of Neltura Technology launched a service that extends online identities and communications to mobile and across disassociated networks in February 2006. The initial launch application links Skype identities and services onto J2ME equipped mobile devices on mobile carrier networks. The solution retains the original caller Skype identity where Skype is federated with the PSTN via SkypeOut but it does not specifically separate the calling identity such as eBay or MySpace caller identity from the Skype communications identity due to underlying limitations of the Skype network.

A number of companies including Tello are pushing IM and communications federation technology that allows users on one domain such as MSN to communicate with user on another domain such as Yahoo. Services such as GoogleTalk provides open communications infrastructure such as Jabber / XMPP for presence and contact list interfacing and libjingle for call signaling and control. Such technology allows web2.0 type solution enabling applications to leverage GoogleTalk as a communications backplane. However, such technology does not allow users on different communities to communicate via the original caller identities or express caller intent and calling context that is available via the call nudge technology.

Both Skype and Google have deployed Click-to-call functionality wherein a hyperlink on a web page can be clicked to initiate a call to a particular ID. In these applications, the called user's identity on the communications network is exposed to the caller, and the calling user's id is exposed to the called user.

Computer-Telephony Integration (CTI) technology has been deployed which can initiate calls from a computer application, typically from a contact management application such as Microsoft Outlook. In these applications, there is no bridging of services and the concept of initiating a communications session via indication on one service, which being provided the communications service via another service using a different identity which remains protected is missing altogether.

A number of click-to-call services exist today, such as Jajah and SkypeMe. For Jajah, it is envisioned that a user can click on a phone number which is represented as a callable URL, and a call is then made between the callable URL and the user's phone. For SkypeMe, a user with the Skype PC client can click on another user's SkypeMe button, and a call is then made between the caller and the callee on the Skype network. However, these services have a number of limitations. In the case of Jajah, the callee number is typically exposed, and there is no way to convey caller intention or the ability to use an online calling identity. For SkypeMe, both parties have to be Skype users, and the callee whom exposes the SkypeMe button has to enable any Skype users to call them, even those not on his or her Skype buddy list. Further, there is no call request context and generally is used where the calls are free end-to-end.

Original Called Number

On traditional PSTN networks such as ISDN PRI signaling, the original called number is presented even though alternate calling trunks might be used to carry the call. This solution is limited to PSTN services based on E.164 type calling numbers.

ENUM

NeuStar is one of the reference implementations of ENUM and a leading provider of clearinghouse services for communications. It enables the routing of telephone calls in North America, manages online digital identity efforts and number translation services, and operates the authoritative database for North American telephone numbers. However, NeuStar is a service that allows service providers to lookup and resolve user identities and numbers so that calls can be routed. This is distinct from a way to enable calls to be made via original calling identities.

Enhanced Original Calling Number Services

TalkPlus has a service that allows a user calling from one number (such as the user's mobile phone) through a set of service gateways that alter the number so that the call appears to come from another number. For instance, if Joe works for Goldman Sachs and wants to call his client from Joe's mobile phone, the service allows the call to arrive at the client's communication device and service with the original calling number and identity as "Goldman Sachs" instead of Joe's mobile number.

Session Border Gateways

A number of border session gateways such as AcmePacket are available that allow users in one domain, such as a carrier, to talk to users on another domain, such as a second carrier. However, session border gateway controllers are mostly limited to linking users from one domain to another rather than extending services and identities across disassociated domains.

Skype

Skype currently has a presence-enabled web-link product called SkypeMe that allows a user "A" to publish "A's" SkypeID, and allows another user "B" to click on "A's" "SkypeMe" button to initiate a Skype session from "B" to "A". For this to work, both "A" and "B" have to be Skype users.

Jajah

Jajah has a web-based offering that allows a user to enter a user's own phone number and a destination phone number. This would then initiate a voice call between the user and the specified destination phone number. It can be envisaged that this can be extended to include "click to call" or "click to call back".

Nothing in this background section is an admission of prior art

Summary of the Disclosure

In accordance with an embodiment of the invention, a user "A" with one or more service identities, such as "a", can publish or send communications session invites called "EQO Me" to user "B" with service identity "b" which can progress through various session permissions stages and then communicate through a number modes comprising message, SMS, IM chat, and voice. The service identity "a" can be any identity which user "A" has exposed to user or users "B". A particularly useful scenario occurs when service identity "a" is a service identity that is defined within an online community, such as MySpace.

An embodiment of the invention allows user "A" to initiate a session request to user "B" and provides user "B" with user "A's" profile, call context, call options such as "IM chat", and "voice call", and a number of dynamic or statically configured session disposition options comprising "accept", "reject", "queue", and "redirect". Based on the disposition option selected by user "B", a session with media is then initiated from user "A" with service identity "a" via a communication service with identity such as "aa" to user "B" with service identity "b" via a communication service with identity such as "bb".

The session identities "aa" and "bb" are re-aliased back to the original user service identity "a" and "b" via a centralized gateway. Alternatively, if user "B" with service identity "b" accepts a session from user "A" with service identity "a", the centralized gateway initiates a session to user "A" on communications service identity "aa" and also to user "B" on communications service identity "bb" in such a way that the sessions appear to be associated with service identities "a" for user "A" and service identities "b" for user "B".

The solution as described in an embodiment of this invention also has user clients that can be deployed in a myriad of configurations comprising mobile phone, PC client, and software object that can be integrated with messaging clients such as MSN and Yahoo or ICQ messenger, or embedded on IP appliances.

Advantages

An embodiment of this invention enables a number of core benefits for end users such as:

1. Transient-mode spontaneous communications
Two users can engage in spontaneous communications such as IM chat without being on each other's buddy list.
2. Communications with native identity and context
User can communicate using the identity of the online community, and can do so with rich context such as the profile of the users and the intent of the communications session. It also helps to keep the user's communications services such as phone numbers anonymous thereby protecting the user's privacy and eliminating the exchange of less-disposable identities.
3. Multi-modal communications with session permission control
End users can dynamically choose to accept, reject, queue, and otherwise dispose the communications request and can dynamically switch communications modes such as messaging, IM chat, and voice. It allows users to separate the call intent from the actual communications tool used for the actual communications session.
4. Extend service reach
By publishing or sending the communications intent, users can reach a much broader audience or circle of friends

For service providers, the above end user value propositions can translate into a number of potential revenue opportunities comprising increased stickiness on community applications, new revenue opportunities in extending online communities to mobile and across disassociated networks.

Description of Specific Embodiments

A user context and use case scenarios are described below.

Case 1 – Published “EQO Me”

Colin's personal page such as on a dating site or online community publishes "EQO Me" which is viewed by Jane say from a web browser. Jane can see the "EQO Me" icon if Jane met Colin's profile filter criteria or if Colin had originally expressed interest in communicating with Jane. If Jane clicks on "EQO Me", then voice calls can be setup between Colin and Jane, or messages such as in the form of an SMS, call, or session requests which can include Jane's profile and presence status, can be sent to Colin. If Colin accepts the session request, then a session such as a voice call would be set up between Colin and Jane. In this scenario, it can be a setup such that it is Colin that pays for parts or all of the transaction. A variant of the payment model is that Jane can pay for parts or all of the transaction. In a caller (Jane) model, there is less generally less economic incentive for Jane to call Colin unless Jane really wants to communicate with Colin. However, in applications such as dating sites where it is generally the male that pays, the payment party can be gender related.

Case 2 – Pushed “EQO Me”

Colin found Jane online and wishes to communicate with Jane and hence pushes Jane an "EQO Me". The "EQO Me" message can be delivered as an email, flower, smile, or other forms of asynchronous or synchronous messaging mechanisms. Suppose Jane opens the message, potentially views Colin's profile and message, and decides to accept the communications session request. If it is a voice call, a direct call can be made between Colin and Jane. If the response is a message such as a SMS or MMS or flower, the message would be delivered to Colin. If Jane simply accepts the communications request, the response can be delivered back to Colin who then can initiate a number of modes of communications with Jane such as messaging, IM chat, and voice. Similar to the "case 1" above, the normal paying party can be Colin but in an alternate payment scheme Jane can also pay for part or all of the transaction.

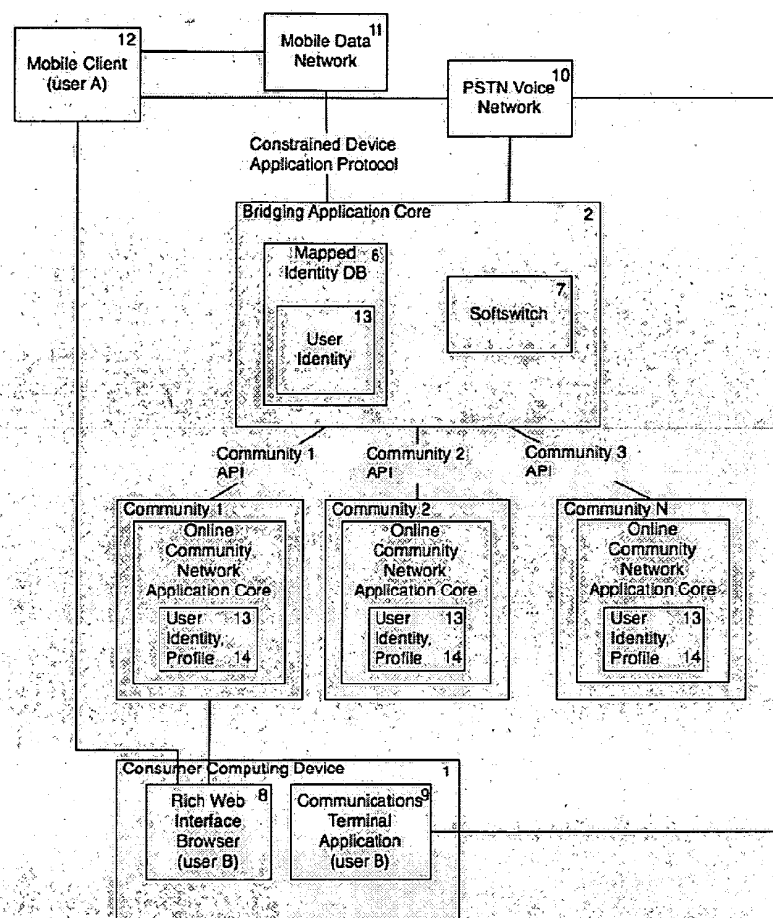
Solution Architecture

One of the design principles of EQO Me and multi-identity aliasing is that the call intent or expression of interest to communicate is decoupled from the actual communication session and media. The call intent can be expressed via an EQO Me button and transported as a generic EQO Message over disassociated networks. For example, the call intent on a web site can traverse through the EQO service network to a user's EQO enabled mobile client. For example, if user A is busy, user A's presence status is shown as busy to user B. User B can then initiate a call request to user A. Based on user B's call request, user A can then establish communications with user B through a number of multi-modal methods, such as messaging, IM chat, and voice calling.

Another design principle is that the call signaling can be asynchronous as well as synchronous. That is, an EQO message initiated by the caller can be delivered to the caller in either online or offline mode. The EQO message itself can have a time-to-live period in which communications intent is valid between the callers and callees. In addition, the EQO Me concept can be deployed as an asymmetric service (where one

party has the EQO capability) and as a symmetric service (where both parties are equipped with the EQO Me capabilities).

The solution architecture is depicted in the diagram below. In accordance with an embodiment of the invention, a set of functional elements are introduced and integrated into an existing telecommunications network and the Internet to provide the identity aliasing and session invitation management.



Element 1 – Consumer Computing Device

Element 1 is consumer computing device. This device can be a home PC, a laptop, a mobile telephone, or an internet-enabled television.

Element 2 - Bridging Application Core

Element 2 is the core application element providing bridging functionality between the Online Community and the Bridged Communications Network.

Element 3 – Online Community Application Core

Element 3 is the application element providing access to the Online Community. This element will usually take the form of a set of web application servers, application databases, and user account databases. This element provides service to the user via a web browser connected to Element 3, or other rich content rendering component which has connected to Element 3.

Element 4 – Bridged Communications Network Core

Element 4 is a Communications Network Core which is to be bridged by Element 2. This element can take the form of a voice-enabled communications network, an instant messaging network, SMS networks, an email network, or other type of communications network. The prerequisites are: (a) that the network makes use of well-defined addressing parameters by which communications endpoints can be reached. Examples of voice-enabled network technologies suitably fulfilling this prerequisite are SIP-based termination networks, H.323-based termination networks, ISUP/SS7-based termination networks, and XMPP/Jingle-based VoIP networks.

Element 5 – Browser Plug-in

Element 5 is a browser plug-in. This element of the system is optional. It is used in the case where the calling user's call indication is executed via a URL handler in a web browser or a MIME type handler in a web browser.

Element 5 – Online Community Rich-Content User Interface

Element 5 is a user interface providing, at least, a text interface. This element will usually be an HTML document-based interface provided for display on a web browser. The user interface can take other forms as well, such as Flash, Shockwave, DHTML / AJAX or other rich-content based user interface technology. The prerequisites are (a) that this user interface have the ability to display text or graphics to a first user, the viewer, representing the identity of a second user, and (b) that this user interface provide a mechanism for the first user to signal their intent to communicate with the second user; this mechanism, for example, can take the form of a selectable hyperlink.

Element 6 – Mapped Identity Database

Element 6 is a database storing the mapping of Online Community identities to Bridged Communications Network identities. This database can take several forms. Examples of typical forms of the database include: (a) a relational database management system, (b) an object-relational database management system, (c) an XML document database, and (d) a DNS-based database such as ENUM. The interface to this database can likewise take several forms. A SQL-based query interface, XPath/XQuery-based query

interface, or Web Services-based interface to the Mapped Identity Database would be typical.

Element 7 – Softswitch Application

Element 7 is a softswitch with capabilities to connect voice, video, or text sessions between 2 or more participants. These sessions can be initiated in two patterns:

(1) The session is initiated by a communications endpoint via a signaling protocol to the softswitch (leg 1), and the softswitch initiates a second communications leg to another communications endpoint (leg 2). This session initiation pattern will be referred to as “edge originated”.

(2) Two legs are initiated at the softswitch: one leg to communications endpoint A, and another leg to communications endpoint B. This session initiation pattern will be referred to as “core originated”.

The softswitch has capabilities to host conference-based communications sessions in which more than two communications endpoints are simultaneously connected in a shared media context.

The softswitch has capabilities to communicate with communications endpoints via a wide array of signaling protocols, such as SIP, ISUP, FXO, and Constrained Device Application Protocol.

For the purposes of this element definition, “communication endpoint” can be taken to mean either an end-user communication terminal, another communications switch, or a signaling proxy.

Element 8 – Web Browser

Element 8 is a web browser hosted on a computing device with rich graphics capability.

Element 9 – Communications Terminal

Element 9 is a communications terminal belonging to User “B”. This terminal can take the form of a PSTN phone, or a communications terminal application such as a SIP softphone, Skype client, or other voice enabled IM client.

Element 10 – PSTN Voice Network

The Public Switched Telephone Network which can be based on NGN technologies but the concept can be equally applied to other media such as push-to-talk or video.

Element 11 – Mobile Data Network

A packet switched, Internet Protocol-based data network supporting HTTP, TCP, or UDP data transport between the Mobile Device and the Internet.

Element 12 – Mobile Device

A Mobile Device supporting both voice communications (via PSTN or VoIP) and a software application environment (such as J2ME, BREW, Windows Mobile, etc.), such as a cell phone or a PDA.

Element 13 – User Identity

A User Identity is a unique identifier “a” for a user “A” within a particular network or community. The identifier can be made globally unique by creating a related identifier “c-a” by concatenating a unique identifier for the community “c” and the original identifier “a”.

Element 14 – User Identity Profile

A User Identity Profile for a user “A” with User Identity “a” is the set of user attributes and descriptive data within the particular network or community hosting identity “a”. This profile will typically include one or more pictures of the user, and a text description readable by other members of the community. Another common attribute is a feedback measure defined by the accumulated set of feedback provided by other community members. For example, other community members can provide their rating of User “A”’s friendliness. In this case, the feedback measure can be calculated as a simple average of the contributed ratings. Many other methods of calculation can be useful. Many other types of user attributes can also be included in the User Identity Profile.

One example of a session flow is shown below. Many variants of this session flow are supported. In this particular flow, user “B” sees user “A” published “EQO Me”. This “EQO Me” button can well be pushed to user “B” via email, SMS, IM application such as Skype, etc. rather than be accessible from a website.

The value of a dating event can be understood in two parts: The value to the sending or receiving user, and the value to the community operator.

The value to the sending user is that by invoking the dating event, they are able to express a social meaning which is derived from the context carried by the dating event by virtue of the real-world action for which it is a metaphor.

Here is an example: Dating Event: "Smile"

Social meaning:

The sender is flirting or expressing an interest to communicate further with the recipient.

Social context:

The sender and the recipient are interested in meeting people. The sender and the recipient do not know each other yet, except for what has been exposed in the recipient's profile to the sender. The sender wishes to establish contact with the recipient, while minimizing both the monetary cost within the system and the cost to the sender's self-esteem if the desire for contact is not reciprocated.

Social value:

The value to the recipient can take multiple forms. If the content of received dating events or statistics concerning received dating events are exposed publicly, a kind of dating capital or popularity measure is created.

Operator value:

The event has value to the community operator. The receiving user's popularity measure can be considered a new source of user metadata, which can be used for sort searches.

New opportunities for charging arise, from charging users a premium for the ability to filter searches via popularity metadata. In dating and matchmaking community applications this is a particularly valuable function, because popularity metadata derived as described is a very strong proxy for how appealing the recipient's profile appears to other users in the given community context. That can be different charging communications events using this mechanism. For example, to "send a smile" can cost a user 1 credit, to "send flowers" can cost a user 2 credits, while to send a longer message can cost a user 5 credits.

Specific Embodiment: Session Intent Signaling over Third Party Backplane

One embodiment of the invention involves its application to send the "EQO Me" message over a third party backplane such as Skype, Google, and MSN.

In this specific embodiment, a user "A" on a computing device such as a mobile phone or PC web browser and publish or send/push an "EQO Me" message to express context

and communications session intent and call handles through a backplane such as Skype. The receiving party user "B" can then click or invoke the "EQO Me" on a computing device such as a browser. This allows user "B" to accept the session request or to re-initiate the session request, and append or enter session context. This exchange of session intent can be initially carried through the third party backplane but subsequently carried directly between the interacting parties without necessarily involving the backplane.

For example, user "A" can send an "EQO Me" call invite via user "A" Skype to user "B" Skype via the Skype IM chat tool. User "B" can click on the "EQO Me" URL which then opens up a screen for the EQO Message. This screen can show profile information and call context information of user "A". User B can enter or respond with new call context, enter "B" phone number, and send the response back to user "A". At this point, the call can proceed or can be deferred until user "A" decides to initiate the call session. Then, a call session is made and the session media can be phone-to-phone which is independent of the third party backplane such as Skype.

Theory of Operation

We describe here the phases the system executes to accomplish the functionality. In some phases, there are multiple methods that can be employed to accomplish the outcome of that particular phase. We describe these variant methods for each applicable phase in the section "Detailed Description of Operational Phases".

Operational Phases

1. Initiating User Logs in to Online Community
2. Presentation to the Initiating User of Other User's Online Community Identity
3. Initiating User Requests Communication Session with Other User
4. Other user can have option to view profile and assert permission on communications
5. User Service Availability and Address Resolution
6. First Communications Leg Setup
7. Further Communications Leg Setup(s)
8. Leg Connection

Detailed Description of Operational Phases

Initiating User Logs in to Online Community

The Initiating User visits the login web page of the Online Community using the web browser on his/her Consumer Computing Device. The user executes the login sequence provided by the Online Community to achieve a logged-in state in the web browser. Depending on the particular login functionality implemented by the Online Community, the login process can be automatically executed on behalf of the Initiating User, based on saved username / password data or other form of saved authentication token.

Presentation to the Initiating User of Other User's Online Community Identity

The Initiating User "B", possibly after "surfing" through several web pages on the Community site, arrives at a web page on the Community Site which presents a visual object representing the ability to initiate a communications session with the Other User "A" via the Other User A's Online Community identity. In another variant, the same object can be pushed or sent from user "A" to user "B" such as user B's message inbox.

The visual object can take the form of:

- Hyperlinked text
- A hyperlinked button
- A hyperlinked image
- An embedded rich object such as a Flash object, ActiveX control, or Java Applet

The visual object incorporates an activation mechanism by which the Initiating User can indicate their intent to begin a communications session with the Other User.

The activation mechanism can take many forms. Typical forms of the activation mechanism include:

- A hyperlink property
- A javascript action
- A two step action. For example, highlighting the visual object by using the user interface environment's selection mechanism, then providing activation by softkey. This type of activation mechanism is common on mobile Consumer Computing Devices.

In the preferred embodiment, the visual object is a combination of a hyperlinked image and textual representation of the Other User(s) Online Community identity in an HTML document, and the activation mechanism is the "href" hyperlink attribute of the HTML "img" and "a" tags.

Initiating User Requests Communication Session with Other User

The Initiating User indicates their intent to begin a communications session with the Other User(s) by making use of the visual object's activation mechanism.

The web browser then creates a communication session request. This communication session request is then transmitted to the Bridging Application Core.

The particular method of transmission of the communication session request can take several forms:

(1) Transmission of communication session request by browser URI handler:

In this transmission method, the activation mechanism is a hyperlink property wherein the target URI's scheme component is one for which a URI handler has been installed as a web browser plug-in. Some typical URI scheme values which can be used are "callto", "mailto", "tel", "sip", "aim", "h323", and "http". Other schemes can be used.

The browser's URI handler mechanism then transfers the URI to the Bridging Application Browser Plug-in which has been registered as the handler for the particular URI scheme in use.

The Bridging Application Browser Plug-in then connects to the Bridging Application Core, and sends a request containing the Initiating User's Online Community identity, and the Other User's (or Users') Online Community identity, and the type of communications session to be initiated.

(2) Transmission of communication session request by browser MIME-type handler:

In this transmission method, the activation mechanism is a hyperlink property wherein the target URI points to a location on a web server which upon being retrieved via HTTP GET, returns content of a particular MIME-type, say "application/vnd.ego.comm_initiation", for which a MIME-type handler has been installed as a web browser plug-in.

The browser's MIME-type handler mechanism then transfers the content retrieved via the HTTP GET on the target URI to the Bridging Application Browser Plug-in which has been registered as the handler for the particular MIME-type in use.

The Bridging Application Browser Plug-in then connects to the Bridging Application Core, and sends a request containing the Initiating User's Online Community identity, and the Other User's (or Users') Online Community identity, and the type of communications session to be initiated.

(3) Transmission of communication session request by submission to a web application interface in the Bridging Application Core:

In this transmission method, the activation mechanism is a hyperlink property wherein the target URI points to a location on a web server, which is deployed in combination with the Bridging Application Core. The hyperlink, upon being activated, submits the request data via HTTP GET or POST such as GET or POST parameters the Initiating User's Online Community identity, the Other User's (or Users') Online Community identity, and the type of communications session to be initiated,

(4) Transmission of communication session request by forwarding via Online Community Core:

In this transmission method, the activation mechanism is a hyperlink property wherein the target URI points to a location on a web server, which is deployed in combination with the Online Community Core. The hyperlink, upon being activated, submits the request data via HTTP GET or POST such as GET or POST parameters the Initiating User's Online Community identity, the Other User's (or Users') Online Community identity, and the type of communications session to be initiated, Once received by the Online Community Core, the request data is forwarded to the Bridging Application Core via an Internet connection technology.

In the preferred embodiment, transmission method (3), "Transmission of communication session request by submission to a web application interface in the Bridging Application Core", is used.

As stated in the transmission method descriptions above, the communication session request specifies the type of communications session to be initiated. The set of communications session types comprise at least one of the following:

- Voice
- Video
- Email
- Instant Message
- SMS
- Specific Voice Network: Skype
- Specific Voice Network: Jingle
- Specific Voice Network: Yahoo
- Specific Voice Network: Skype
- Specific Voice Network: AIM
- Specific Voice Network: MSN
- Specific Instant Message Network: Skype
- Specific Instant Message Network: XMPP
- Specific Instant Message Network: Yahoo
- Specific Instant Message Network: AIM
- Specific Instant Message Network: MSN
- Virtual Communications Session Type: Most Immersive Available
- Virtual Communications Session Type: Least Immersive Available
- Virtual Communications Session Type: Least Expensive Available

In general, any communications session type which has a clearly defined party or set of parties to which the session initiation is directed can be used in combination with the described system.

User Service Availability and Address Resolution

Use of the Virtual Communications Session Types indicates that the Bridging Application Core chooses the best matching communications type based on the best matching communications services available to the participants.

For example, say the requested communications session type is "Virtual Communications Session Type: Most Immersive Available". Given the service availabilities as listed in the table below,

	Initiating User	Other User
Email	available	available
Voice	available	available
Video	available	not available

the Bridging Application Core would choose "Voice" to satisfy the request for the most immersive communications type available.

First Communications Leg Setup

The first communications leg is setup between the Initiating User and the Bridging Application Core. This setup can proceed either from the Initiating User to the Bridging Application Core (Edge Originated), or from the Bridging Application Core to the Initiating User (Core Originated). Both of these types of originations are supported in the preferred embodiment.

By supporting both origination types, the service can be tailored to specific telecom markets. For instance, in some markets termination to particular numbers such as mobile phone numbers are tariffed at a high price, paid by the call initiator.

However, other constraining factors influence the choice of mechanism as well. For instance, for call sessions initiated from a software client on some mobile devices, technical limitations make edge initiation less desirable. In particular, some mobile devices initiate calls via a software client running on the device, but the software client immediately shuts down in order for the mobile device to successfully complete the call. This is undesirable from a usability point of view, so in this case the Core Originated call initiation can be preferred.

Further Communications Leg Setup(s)

When the Other User accepts the session at a particular address, the leg to that address proceeds to a connected state, while other communications legs initiated as a result of the First Communications Leg Setup are cancelled.

Conclusion

All of the methods and processes described above may be embodied in, and fully automated via, software code modules executed by one or more general purpose computers, and where applicable, by mobile computing devices. The code modules may be stored in any type of computer-readable medium or other computer storage device. Some or all of the methods may alternatively be embodied in specialized computer hardware.

Appendix

NELTEC.001PR

PATENT

Dynamic and context driven call control and service bridging

Field of the Invention

This invention relates generally to the field of dynamically facilitating end user communications. The invention relates specifically to computer-implemented methods for allowing an end user from a mobile device to dynamically set the presence and availability of the caller, and for the recipient to dynamically control the call base on caller driven call disposition options.

Background

In today's communications environment, a typical user has several communications services ranging from home landline, office landline, mobile cellular, to online services such as Instant Messaging, VoIP, and push-to-talk services. Each of services is usually associated with different communications devices, and the services may be provided by multiple service providers. As a result, the service complexity is increasing, and the user has limited control of the services without a simple means to harness the collective capabilities of the services or the power of the user devices. The differences and inconsistencies in user interactions across the services also add service complexity and some times create annoying end user experiences.

For inbound communications, a typical user is faced with constant interruptions from people using traditional telephony and SMS services as well as new IP services such as instant messaging, email, and VoIP. The user has to constantly filter and interact within socially acceptable boundaries with callers whom the user likes to hear from as well as those that the user would rather avoid or interact with lower priority. Unanswered calls can lead to voice mail tag and annoyance to the caller. Though a user may be able to replace his or her entire telephony communications needs with mobile service, a user may continue to keep different phone numbers such as landline, home phone line, and mobile service to partition callers a user wants to hear from and callers that a user wants to filter. In many cases, a user retains a landline phone service at home so that the line can be shared as a family group or an access line that provides lifeline service or a constant contact point for extended family and friends.

For outbound communications, a typical user is faced with annoying and time wasting telephone tag trying to reach the callee through a number of communications services and associated devices such as office line or mobile phone. The callee may elect to avoid the call due to various states of availability, would prefer to use an alternate service, or simply can not access the service such as while away from the office. For

instance, while the user is driving, it is difficult to engage with the caller in an instant message session. Likewise, while the user is at home or in the office, the user might have preferred the caller to use the landline due to cost or voice quality preferences.

Another problem associated with outbound communications is that each service may have different contact lists and network service profiles. For instance, a user may elect to use the mobile phone instead of the landline service while at home simply because of the contact list available on the mobile device. For the user, this specific scenario can mean lower voice quality or higher cost.

A number of solutions exist today that attempts to link a single telephony access service to multiple services, or link multiple services into a single telephony service. For instance, services such as RingCentral provides the user with a separate phone number which can be mapped to the user's various telephony services. In such a service, the caller calls the access phone number, and the service then signals the user's services such as office line, home line, and mobile service concurrently or in some predefined sequence. While this service reduces the number of call attempts the caller has to try to get hold of the callee, it is a static configuration that is usually only useful to situations where the callee always wants the caller to reach all of the callee's services.

Other network based solutions include concurrent ringing services offered by traditional telcos as well as basic call features such as call forwarding and call transfer. VoIP services such as Vonage allow concurrent ringing of all of the user's associated IP end points as well as generic ATA and user agent devices that can be accessed via any IP connection such as broadband IP and WiFi. While these services allow the user to be reached from a remote location, it falls short of providing dynamic call control to allow the user to filter the calls and to link calls based on services that are most convenient for the caller and the callee.

Conventional Call Forwarding Services

Basic call forwarding and follow-me cellular and PSTN services are offered by all leading cellular and wireline operators such as AT&T Wireless and Telus. Carrier VoIP services that can call forward or concurrently ring multiple IP-enabled phones include Vonage and Primus. Internet patching solutions include those from Pulver Innovations. IP-phone-to-IP-phone free calling services include FreeWorld Dialup, and PC-to-PC free calling services include Skype. Centralized number translation services include ENUM.

Skype

A peer-to-peer VoIP solution that uses a "skype name" to identify users. Incoming calls can signal multiple Skype enabled devices signed-on with the same skype user identifier.

RingCentral and CallWave

Services such as RingCentral provide the user with a separate service access number. The user can configure a number of services such as office landline, home landline, and mobile phone to be associated with the RingCentral access number. When a caller calls

the user's RingCentral service access number, the service can ring some or all of the user's configured services and devices concurrently or in sequence.

Vonage

A packet telephony service which uses centrally deployed SIP proxy and redirect services . Both incoming and outgoing PSTN connectivity is provided by PSTN gateways which are operated by the service provider as a part of the network infrastructure. A user may subscribe to multiple numbers from Vonage. For an incoming call, the user may configured the service to concurrently ring some or all of the user's Vonage service end points.

NeuStar

NeuStar is one of the reference implementation of ENUM and a leading provider of mission-critical clearinghouse services for communications. It enables the routing of telephone calls in North America, manages online digital identity efforts and number translation services, and operates the authoritative database of all North American telephone numbers. However, NeuStar is a service for service providers rather than a service for end user devices where the user is effectively a virtual service provider of the communications services on the user's own devices.

Summary

One aspect of the invention is a computer-implemented call control and bridging service for reducing or minimize interruption to allow the caller and callee to established call sessions out-of-band from the actual call, and to link disassociated services such as PSTN / PLMN calls outside the control domain of a service provider, or online services such as third party online applications such as Skype. The call control and bridging service also provides a simple method to filter calls and provides both the caller and callee a set of options to pick the most appropriate communications service for communications interaction. The service provides the dynamic call control to bridge the user's various services so that the call or session request can be initiated out-of-band so that the most appropriate service can be used for the actual communications interaction.

In one embodiment, the call control and bridging service provides a simple means to synchronize and leverage the contact lists available on the user devices. For the service provider, this solution offers a new application to deliver a compelling user experience beyond the communications access pipe. It enables a service convergence application that helps to bridge the user's various telephony and online services.

In one embodiment, a user may have multiple telecommunications services from one or more network service providers. These services can be traditional telecom services such as home phone POTS line or intelligent user devices with broadband IP connectivity. The call control and bridging service allows the user to bridge the user's landline, mobile, and online services such as presence, push-to-nudge, push-to-talk, and push-to-message. The call control and bridging service provides the dynamic call

control to bridge the user's various services so that the call or session request can be initiated out-of-band so that the most appropriate service can be used for the actual communications interaction.

The user may be equipped with an IP-enabled mobile device capable of supporting call control software that may be invoked on J2ME, Symbian, BREW, Palm, Windows CE, embedded Linux, and other operating systems. This mobile client software is linked with a service provider server application which aggregates and propagates user presence and availability, user contact and profile, as well as providing dynamic call control of user call sessions. This gateway application interfaces to network provider core infrastructure to control call sessions via VoIP network elements such as MGCF, telephony service platforms, and online services such as Yahoo and MSN; as well as IP end point entities such as IP phones, IP user agents, and messenger clients.

The solution as described in this document also has user clients that can be deployed in a myriad of configurations including PC client, and software object that can be integrated with messaging clients such as MSN and Yahoo or ICQ messenger that can initiate call sessions including "nudge", push-to-talk, push-to-message, VoIP, and other multimedia sessions such as video. The user clients can also be IP appliances or computer that can operate the same without a dedicated computer, or provide a PSTN gateway capability that can link disassociated PSTN services such as POTS lines and PBX services.

In accordance with one aspect of the invention, the user's landline and mobile telephony services are linked and bridged with the user's online services such as presence, push-to-talk, call initiation signaling, and push-to-message to provide a converged service application. This converged service application provides the user a means to dynamically control the calls with a set of caller and callee context driven call disposition options. For the end user, the converged service application reduces interruption from unwanted callers, and reduces frustration for the caller, and provides ways to link the most appropriate service between the caller and the callee. It reduces telephone tag, and provides a convenient means for the user to filter and control calls.

For the service provider, the call control and bridging service provides a new revenue stream associated with selling the mobile service client and the wireless data package that enables the connectivity for such an application. It also allows the service provider to capture incremental service revenue for routing disassociated and out-of-network calls onto the mobile service. For instance, a call to the user's home line can now be directly linked to the user's mobile service to capture incremental service revenue. The service application also allows the service provider to differentiate the service to provide a user centric fixed mobile service convergence application.

Description of Specific Embodiments

The following description is intended to illustrate specific embodiments and features of the invention, and is not intended to limit the scope of the invention.

The following user context and use case scenario will be used to describe specific features and embodiments of the invention. This specific context makes specific reference to certain platforms and operating systems. The invention can be applied and ported to different platforms with several variants of the call disposition options.

Context

1. Roger hates unwanted interruption and also dislikes telephone tag, and keeps changing between services just to get hold of people he wants to talk to
2. Roger has a Neltura-enabled J2ME client on his 1X enabled mobile phone

Roger has a Neltura-enabled PSTN gateway connected to an alternate provide PSTN line

Roger has a Neltura-enabled client running on his IPTV enabled television

3. Colin has a Neltura-enabled J2ME client on his 1X enabled Blackberry

Colin also has a Neltura-enabled PC client or Neltura-enabled Yahoo/MSN/etc. messenger

Use Case 1 - Service Nudge (only 1 variant of the use case call flow is shown)

1. Colin using his Neltura-enabled J2ME Blackberry sees Roger as "busy" or even "available". Trying not to be too interruptive, Colin sends a "nudge" to Roger's mobile phone J2ME client from his Neltura-enabled Blackberry device. The "nudge" from Colin comes with a set of call disposition options (the hardest nudge is an actual direct call - anything other than that is out-of-band signaling via 1X data network):
 - a) link call to my mobile phone
 - b) link call to my office line
 - d) link call to my IP phone, ...
2. Roger sees the "nudge" from Colin on Roger's mobile phone J2ME client with disposition options
 - a) link call (to Colin's mobile phone, office line, IP phone)
 - b) queue call (call place in queue, and Colin receives acknowledge that Roger will call back later)
 - c) voice mail, ...

3. Instead of on the road, Roger happens to be at home watching his IPTV. With the nudge, Roger sees an indication via the IPTV box to Roger's TV. Roger uses his IPTV box remote control to handle the call disposition (same as step 2 above)

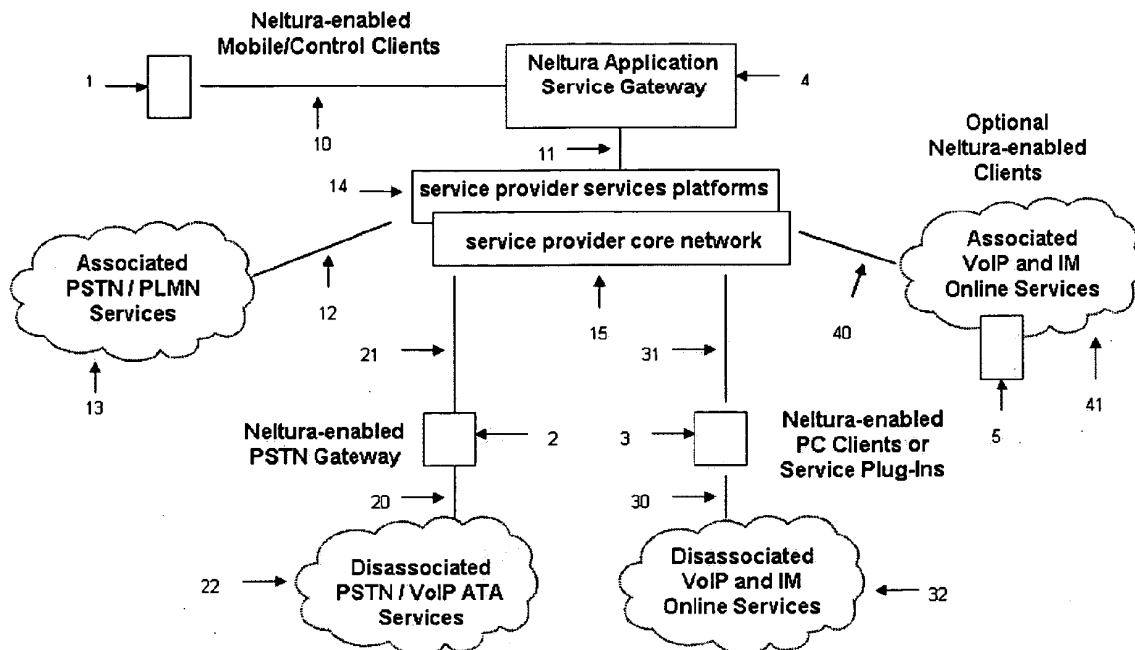
Use Case 2 - PSTN Service Bridging (1 variant of the use case call flow is shown)

1. Jim called Roger's alternate network home phone service. Through Roger's Neltura-enabled PSTN gateway, Roger gets a notification on Roger's mobile phone Neltura-enabled J2ME client with disposition options
 - a) link call to mobile (call would be delivered via IP to Roger's mobile phone via the PLMN network)
 - b) link call to IP phone, office phone, ...
 - c) play message (plays a message to the caller – e.g. to call back later, or if Jim leaves a voice mail, the message can be directly retrieved)

Use Case 3 - VoIP / IM Service Bridging (1 variant of the use case call flow is shown)

1. Colin called Roger from Colin's Neltura-enabled PC client (can also be push-to-talk or message)
2. Roger sees the call indication on his Neltura-enabled mobile phone J2ME client with disposition
 - a) answer call (if call is linked directly via PSTN or VoIP to his mobile phone)
 - b) queue call (Colin gets acknowledgement message that Roger will call back later)
 - c) voice mail

The solution architecture is depicted in the diagram below. In one embodiment, a set of functional elements are introduced and integrated into an existing telecommunications network and the Internet to provide the dynamic and context driven call control. These key functional elements (1), (2), (3), (4), and (5) are described below.



Element 1 – Neltura-enabled Mobile/Control client

Element 1 is a thin mobile client implemented in J2ME or other variants suitable for mobile devices or IP enabled devices such as PDA or IPTV. This client requires IP connectivity such as GPRS, Wi-Fi, CDMA1X, WCDMA, Bluetooth, WiMAX, TD-SCDMA, or landline variants such as LAN, ADSL, and packet cable. This client implements a call control client based on presence information, contact profile, and call disposition context via interface 10 between the Neltura-enabled mobile client and the Neltura service gateway. Optionally, element 1 may be integrated with a VoIP user agent capable of directly initiating or terminating a call session via IP networking.

Element 1 uses the contact profile available on the mobile device and synchronizes the contact list profile with the user's other devices (2, 3, 5) via (4). It displays the user's own aggregated resource status such as line availability and service set such as various line services available. It also displays the presence and availability status of the user's contacts.

For an incoming event such as a call session to an associated or disassociated service (such as call to the user's Skype account or to the user's home phone line, office phone line, or VoIP service) or a "nudge" from the user's associated or disassociated online service (such as messenger client), the service gateway (4) provides element 1 with the call disposition context to link or handle the call event. The call context may simply request the user to call the event initiator. This is done by taking the call digits in the call disposition context and placing the call out the user's mobile device. The call context can also allow the user to link the event initiator so that the call can be initiated from the

event initiator to the user. In such a case, the user's link request back to the event initiator cause the event initiator to place a call to the user. Such a call can be placed directly from the event initiator's device, or can be facilitated via the service provider network to setup the call. For instance, the service gateway (4) takes the link request context and instructs the service platform (14) and core network controller (15) to initiate call legs between the event initiator and the user.

The call disposition context also allows the user to "queue" the event. In this case, an acknowledgement message along with option response message string is returned to the event initiator, and the call is queued so that the user can link the call event at a later time. The call disposition context can also be a set of calling numbers and services such as voice mail. In such cases, the mobile device will initiate calls out to those numbers, or place the inbound session into voice mail.

The user can also modify the outgoing call delivery context so that instead of linking the mobile device with the various services of the event initiator, the user can link one of the user's alternate services such as VoIP phone or PSTN land line. The call control for linking these services is delivered via service gateway (4) to the core network (14) (15) and as appropriate to disassociates services via (2), (3), or (5).

The client (1) also sends the user's client (1) presence and resource status to the service gateway (4) so that it can be propagated to the user client (2), (3), (5), and to the user's contacts with similarly equipped clients (1) as well as on networks (32) and (41).

Element 2 – Neltura-enabled IP Appliance and gateway for disassociated PSTN service

Disassociated PSTN services (22) include third party TDM POTS services or services in which a service provider does not wish to invest to integrate into a dynamic call control application can be integrated via interface (20) into a Neltura enabled appliance (2). The bridging can be done only at the signaling layer or can also involve media services.

Element (2) is a PC or an IP appliance that implements a disaggregated softswitch application which can route and link disassociated PSTN services such as a POTS line or a PBX service. The PSTN line can be linked into a PC via peripherals such as a voice modem or dedicated IP appliance with support for linking FXO line interfaces (22). Enterprise services such as PRI services can be linked via a PBX service interface into (2) via variants of interface (22).

The Neltura-enabled PSTN gateway (2) provides the call control and bridging to link the disassociated PSTN service (22) into the service provider network via interface (21). Interface (21) can be integrated through the service provider core network (15) such as IMS infrastructure or it can be integrated via a variant of interface (21) directly into the Neltura service gateway (4).

The client (2) sends the user's client (2) resource status to the service gateway (4) so that it can be propagated to the user client (1).

Element 3 – Neltura-enabled client for disassociated online services

Disassociated online services (32) include VoIP services and IM services (for example, Skype) that are outside the management and control domain of a service provider. The bridging can be done only at the signaling layer or can also involve media services.

The disassociated online services (41) can be integrated via service API (30) into a Neltura-enabled client (3). Such a Neltura-enabled client (3) routes and bridges the services of (32) into the service provider network via interface (31). Interface (31) can be integrated through the service provider core network (15) such as IMS infrastructure or it can be integrated via a variant of interface (31) directly into the Neltura service gateway (4).

The client (3) sends the user's client (3) resource status and availability, as well as the user's contacts presence information in the disassociated online service (32) to the service gateway (4) so that it can be propagated to the user client (1).

Element 4 – Neltura application service gateway

Element 4 is a service gateway that proxy the service status and contact presence as well as call signaling events and call disposition options of the user's associated and disassociated services onto the user's mobile client (1).

Element (4) uses interface (11) to receive call triggers, session events, contact profiles, and call resource states from the core network service platforms (14) and service control elements (15). Based on the call control and disposition options selected the user on device / client (1), element (4) delivers the call and media control to service platforms (14) and core network control elements (15) to establish the call session or complete the handling of the call event such as voice mail. If the session involves linking the call with disassociated services outside the control of the service provider (13), (14), (15), (41), element (4) will also direct one or more call leg controls to (2) and (3).

The core network service platform (14) and core network control elements integrates with associated PSTN and PLMN services (13) via interface (12). It also can interface to associated and online IM service (41) via interface (40). In a deployment variant, the associated services (41) can be directly or integrated in parallel with the service gateway (4) via a variant of interface (40).

Element 5 – Neltura-enabled client for associated online services

Associated online services (41) such as VoIP and instant messaging may be already integrated into the service provider core network (15) via service interface (40) or it can be integrated directly via the proxy service of the service gateway (4) through a variant of interface (40).

A user on an associated online service (41) can initiate a call event or a session such as “nudge” or push-to-talk with the user on (1). In such a case, the call event is delivered to (1) via (4). The user on (1) can also initiate similar set of events to the user’s contacts on associated online services (41), associated PSTN and PLMN services (13) via services platform (14) or core control elements (15), and to disassociated online services (32) via (3) and disassociated PSTN services (22) via (2).

The client (5) sends the user’s client (5) resource status and availability, as well as the user’s contacts presence information in the associated service (41) to the service gateway (4) so that it can be propagated to the user client (1).

Conclusion

The various features described above may be implemented in code modules executed by general-purpose computing devices, including but not limited to PCs, Personal Digital Assistants, and mobile phones. The code modules may be stored in any type or types of computer storage device or memory.

Appendix

The attached appendix, which forms part of the disclosure of the present application, describes a particular implementation of a system that embodies some of the foregoing features as well as other features. The appendix is intended to illustrate how certain aspects of the invention may be implemented and used, and does not limit the scope of the invention.

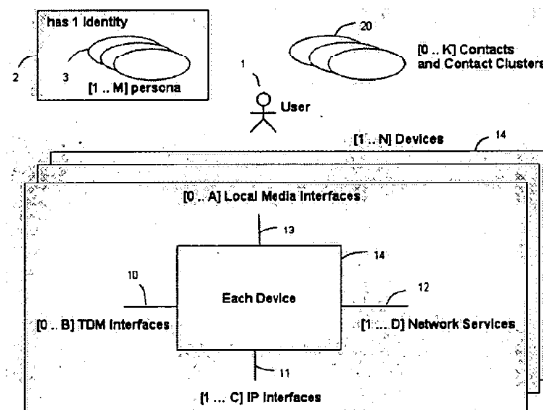
Appendix

1 The Personal Network Service Convergence Application

The Personal Network Service Convergence Application (PNSCA) is software that bridges the user's landline and mobile telephony services with the user's online services such as presence, push-to-talk, and instant messaging. The software seamlessly integrates the traditional telephone network with VoIP multimedia services to give the users a converged communications service that offers greater convenience and better ease of use. This gives the user more control of communications services and allows the user to better leverage his or her communications devices and services to make and receive calls at lower cost. It also offers a set of user-demand driven advanced services such as push-to-talk, push-to-message, ad-hoc conferencing, concurrent ringing / follow-me, and ring-back tones. Using web services and P2P technology to leverage existing telephone network infrastructure and the Internet, Neltura delivers a user experience that mimics the traditional PSTN and cellular experience while expanding its possibilities.

SYSTEM CONTEXT

A typical user communicates with different terminal devices and services depending on location and other factors such as time of day, cost, who is being called, and who is calling. The communications devices, services, user context, and user contacts of a typical user can be modeled as illustrated in the diagram below.



Each user (1) has 1 identity (2) and several [1..M] service interaction persona (3) such as co-worker, father, brother, son, and friend, each of which can be associated with specific communications identities such as office number, home number, cellular number.

Each user (1) has [0..K] social contacts (20) and each of these contacts can be grouped by persona, affiliation, and other logical groupings.

Each user (1) also has [1..N] devices (14) that are used for communications services.

1. Each device (14) can have [0..A] local media interfaces (13) such as microphone and speaker, handset, as well as visual display such as video and GUI.
2. Each device (14) also has [0..B] TDM or air interfaces (10) such as RJ11, PRI, GSM and CDMA for interface to the PLMN and PSTN. The TDM interfaces (10) implements [0..B] PSTN and PLMN routable services such as E.164 directory numbers. These interfaces and the associated services on these interfaces for the user's [1..N] devices can be clustered, as well as associated and clustered with the user's contacts devices and clusters (20).
3. Each device (14) also has [1..C] IP-based application interfaces (11) for IP trunks or trunk groups, links to IP end points such as SIP-based IP phones and ATAs, and IP-based network and web services, as well as interfaces between the users other IP-connected devices and the IP interface to the service provider for user authentication, peer-to-peer network bootstrapping, and user administration and control.

The IP interfaces (11) include a set of service interfaces for service discovery, binding, and transaction for service clustering, service access, and service integration. These service interfaces include:

a. Intra-User Service Interface

Service interface which enables each of the user's PNSCA-enabled devices to advertise to any of the user's other devices the services that it offers as well as the device resources. It also supports the exchange and propagation of presence information between the user devices.

b. Intra-User Service Proxy Service Interface

Service interface which enables a PNSCA-enabled device to proxy services to any of the user's other PNSCA-enabled devices.

c. Intra-Device Service Interface

Service interface which enables the PNSCA-enabled client on the user device to offer and invoke services with another client on the same user device. The implementation of this service interface is platform specific. For instance, for the Windows platform, it can be implemented as a COM object and linked in the form of a DLL.

d. Network Service Provider Service Interface

Service interface which enables the PNSCA-enabled client on the user device to discover, access, and invoke services with a network service provider including client registration and authentication and bootstrapping.

e. Inter-User Contact Service Interface

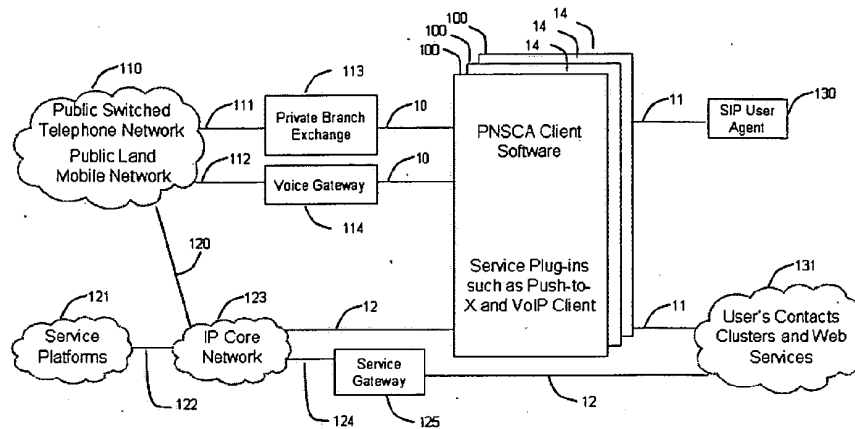
Service interface which enables the PNSCA-enabled client on any one of the user's device to discover, access, and invoke services with the user's contacts. It also supports the exchange and propagation of presence status and personal network events.

4. Each device (14) also has [1..D] network services (12) such as POTS, long distance dialing plan, cell phone service, as well as physical access network interfaces such as RJ45 LAN or other IP oriented physical interfaces such as GPRS, CDMA1X, and Wi-Fi or WiMAX.

Using the single user identity and profile (2, 3), the user can deploy any number of PNSCA software client instances on [1..N] user devices (14), and sign-in concurrently on all the user's devices. Each instance of the user client is uniquely identified by its peer end-point identifier, and is optionally given a time-bounded service such as through a granting ticket from the network service provider. The persistent private user data is stored on one or more of the user devices (14), and the public user data is stored in the distributed data-store and user indexed within the peer-to-peer services network.

SOLUTION DEPLOYMENT ARCHITECTURE

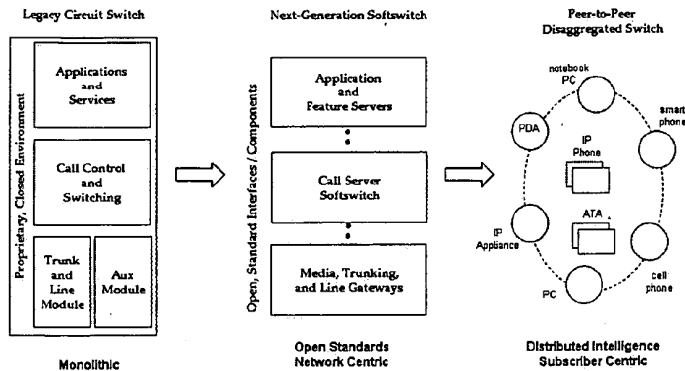
The PNSCA software can be deployed on a myriad of computing platforms including laptop and desktop computers, PDAs and smart phones, IP-connected embedded devices, and communications silicon chipsets (14). The software (100) allows the user's devices (14) to auto-discover each other and cluster together to form ad-hoc networks such that the communications services and resources (10, 11, 12) on any one of the user's devices (14) is accessible from the other devices (14). The software's web services (11, 12) capability also allows users to share and extend services amongst their social contacts, and to transact telecom network services (11, 12) with the core network as a web service application.



The PNSCA software (100) links the user's IP-enabled devices (14) with a single user identity (2) under the control of the service provider core network (12, 123), and clusters the user devices (14) and services (10 - 110, 11, 12) into an ad-hoc personal communications service network. The solution aggregates the user's presence and availability across the user's devices (14) and services (10, 11, 12), and allows the user to share this information with selected contacts on the user's buddy list (11, 131). A sophisticated call manager in the PNSCA software (100) routes calls amongst the user's devices as well as through shared resources within the user's personal network (11, 131). The solution (100) also offers a set of user centric services such as concurrent ringing, conferencing, push-to-talk, push-to-message, and ring-back tones with very little incremental core network infrastructure. A service gateway (125) bridges peer-to-peer services such as push-to-talk between the user's contacts (131) and the service provider network (121, 123) so that a contact on a PC (131) can interact with a user's mobile push-to-talk service (121).

DISAGGREGATED SOFT-SWITCHING

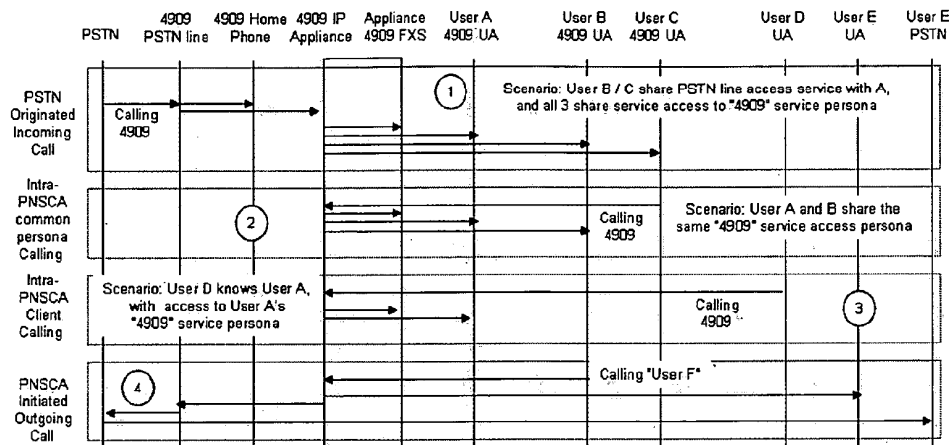
With the proliferation of powerful and relatively inexpensive IP-connected user devices, the value creation of telecommunications services has shifted significantly from the network core to the network edge. The PNSCA is taking the next generation network one step further by disaggregating some of the intelligence in the centralized soft-switch into software components on to network edge devices that are clustered together using peer-to-peer networking. This solution approach dramatically reduces the cost of infrastructure, leverages the powers of the Internet and inexpensive user devices to simplify complex system designs such as manageability, high availability, and scalability, and leverages the network intelligence at the edge devices to offer more personalized communications services.



The PNSCA core technology is software for peer-to-peer clustering, presence aggregation, disaggregated call management, subscriber centric call features, and telecom web services for end user devices. The PNSCA enables the end user IP-connected devices equipped with the PNSCA client software to self organize into ad-hoc networks and cluster together via web services so that the resources and services on any one of the user's devices is available to the user's other devices. For instance, the home PSTN service connected to a PNSCA-enabled PSTN gateway can be extended to the user's other devices such as IP phones so that the home PSTN service can be accessed via an IP connection from anywhere in the world. This particular service scenario allows the user to leverage the best of all services such as location-bounded lifeline services while enabling service mobility so that the home phone line can be accessed outside of the home. It also enables virtual networks with trusted friends and family, as well integration and access to network provider services.

Call Switching – Remote Line Service Context

The PNSCA disaggregated softswitch supports sophisticated service translation, routing, and call switching at the network edge. This software can be implemented on an IP appliance or other IP connected devices such as a PC. In the example below, an IP appliance is integrated with the PNSCA software and the device is associated with a PSTN line labeled as "4909". This particular device has an IP connection, has an RJ11 FXO connection to the PSTN line interface, and another RJ11 FXS connection to a POTS phone handset. In practice, the PNSCA software can be implemented and deployed on a variety of network devices such as PC and network service proxy, and have different device peripheral configurations.



In this sample call delivery context, a PNSCA-enabled IP appliance has the PSTN interface with an E.164 DN of “4909” and is shared by 3 users (user A, user B, and user C).

1. An incoming PSTN call to “4909” causes the “4909” home phones to ring as well as incoming call indication to the FXO interface of the “4909 IP Appliance”. The “4909 IP Appliance” relays the ring signal to the phones connected to its FXS interface, as well as all active registered SIP user agents belonging to Users A, B, and C that are associated with the “4909” service access persona. In this context, the “4909” service access persona is shared by user A, user B, and user C. A variant implementation can apply caller ID discrimination so that the incoming call is filtered by caller and deliver only to callee known to the caller.

If any of the phones (“4909 home phone”, “Appliance 4909 FXS”, and all applicable users A, B, and C user agents) picks up the call, the call is immediately seized, and the call ringing stops. Variant implementations of the control signaling control can allow the signaling to continue to ring on the other still on-hook devices for additional ringing cycles even after the call is seized. All users who pick up the call are placed into a virtual conference. In alternate implementation variants, it is also possible for the user who seized the call to enter a key code such as “*” on the phone numeric digit dial so that the call is placed back into the ring queue.

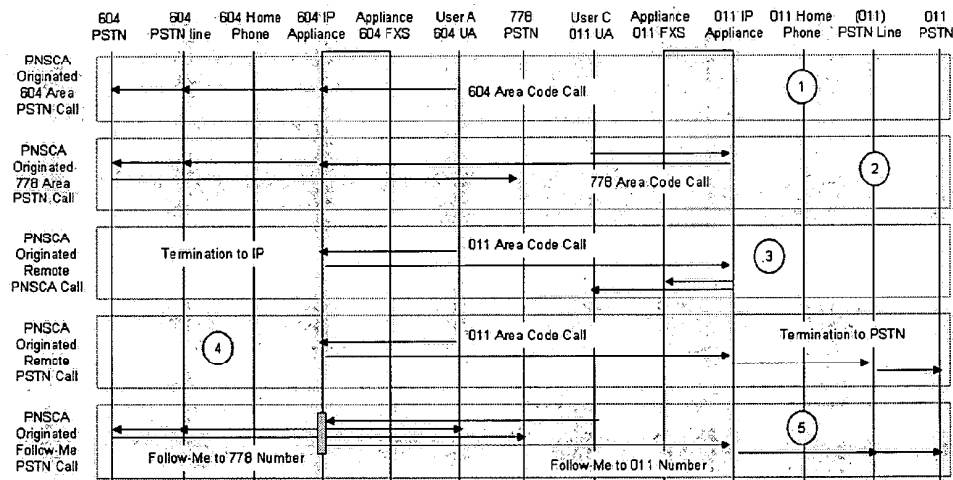
2. If any of the users who share the “4909” service access persona calls the “4909” service access number, then the FXS line of the IP Appliance as well as all other user agents belonging to users who share the same “4909” service access persona will ring. In this case, user C calling the home “known as” number causes the “4909” home FXS line as well as SIP user agents for user A and user B to ring.
3. If user D only has user A associated with the “4909” service access persona, and user D calls the “4909” service access persona number of user A, then only the

IP appliance FXS and the user agent for user A rings. Since users B and user C have the same "4909" service access persona number as user A, but users B and C are not known to user D, the SIP user agents belonging to users B and C will not ring.

4. User B calls the PSTN number of user E. In the initial case, if user E is online, then the call is routed the "4909" IP appliance directly to user E. Otherwise, the PNSCA attempts to find an alternate route to complete the call from user B to user E. If the PNSCA finds user B's "4909" PSTN access line as the alternate least cost path, then the call is placed from the IP appliance, out the "4909" PSTN access line to user E's PSTN access line.

Call Switching - Resource Pooling Context

The PNSCA disaggregated softswitch supports sophisticated pooling of services similar in context to the grouping of individual trunks into trunk groups, and allow the routing of specific calls through selected trunks or trunk groups. In the example below, an IP appliance is integrated with the PNSCA software and the device is associated with a PSTN line with local area code service access labeled as "604". Another similar IP appliance is associated with a PSTN line with local area code service access labeled as "011". It also has a user with a service access persona with service area code of "778". In this system context use case scenario, the "604" IP appliance is configured to originate "604" and "778" service area PSTN calls, and the "011" IP appliance is configured to originate "011" service area PSTN calls.

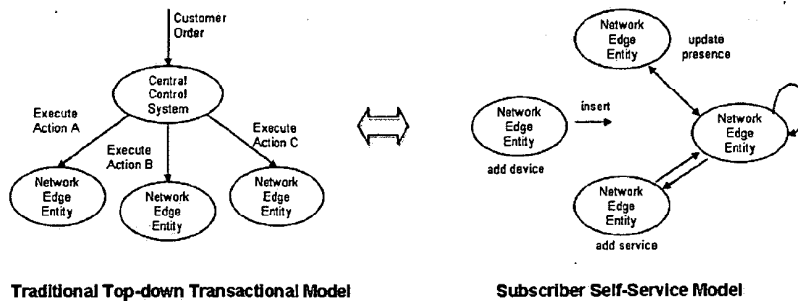


1. Assume in this case User "A" calls a "604" area code number from User "A's" user agent but the called number can not be resolved to an online IP end point. In this case, since user "A" owns the service line resource on the "604" IP Appliance, the call is placed as a PSTN call out of the "604" PSTN line. In this scenario, the user agent of User "A" can be anywhere in the world.

2. Assume in this case User "C" calls a "778" area code number from User "C's" user agent but the called number can not be resolved to an online IP end point. In this case, since User "A" extends the service line resources on the "604" IP Appliance to User "B", the "778" call is placed as a PSTN call out of the "604" PSTN line. In this scenario, the user agent of User "C" is proxied by the "011" PNSCA-enabled IP Appliance. In practice, the user agent for User "C" can be proxied or hosted by any number of User "C's" PNSCA-enabled devices.
3. Assume in this case User "A" calls a "011" area code number from User "A's" user agent and the called number resolved to the online IP end point of User "C". In this case, both User "C's" UA as well as the FXS line on User "C's" IP Appliance would ring. User "C" can seize the call from either User "C's" UA or from the FXS port of User "C's" "011" IP Appliance. In this case, the call is completed as an IP-to-IP call.
4. Assume in this case User "A" calls a "011" area code number from User "A's" user agent but the called number can not be resolved to an online IP end point. In this case, since User "C" extends the service line resources on the "011" IP Appliance to User "A", the "011" call is placed as a PSTN call out of the "011" PSTN line. Note that this is a generic long distance use case scenario. For instance, if one user has an unlimited PSTN calling plan or a Vonage line, several users can piggy-back on the free calling service of this calling line.
5. Assume in this case User "C" calls user "A", and user "A" configures a follow-me service that concurrently rings user "A's" "604" user agent, a "778" PSTN line that can not resolve to an IP end point, and a "011" PSTN line that can not resolve to an IP end point. When User "A" answers the call from any one of the 3 devices in this scenario, the call is connected. In this scenario, the call to "778" PSTN line uses the service line resources of "604" IP Appliance, and the call to the "011" PSTN line uses the service line resources of "011" IP Appliance. This is also a generic use case scenario in that the follow-me call sequence can well be a conference call, not just a person to person call.

SERVICE PROVISIONING MODEL

One of the fundamental paradigm shifts in peer-to-peer based solutions is that service delivery and provisioning is initiated by the user from the network bottom-up. In contrast, traditional telecom applications are provisioned and delivered from the network top-down. In a centralized telecom model, the network intelligence is mostly in the core network and the services tend to be network centric. The top-down transactional driven model is usually more costly to implement and operate, and slower to provision and activate new services.

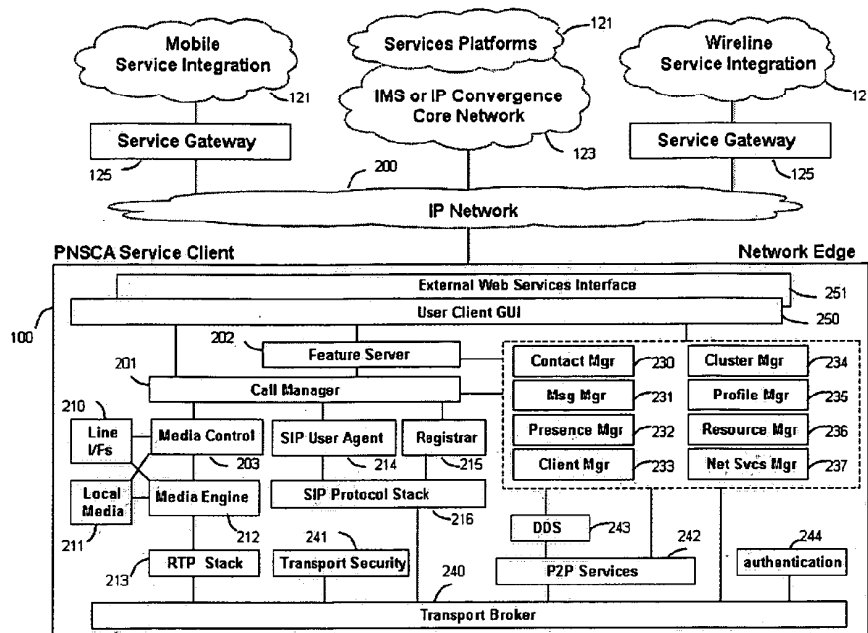


On the other hand, the PNSCA decentralized peer-to-peer model pushes more of the intelligence closer to the network edge to offer subscriber centric services that is presence and user context aware. Such a peer-to-peer and disaggregated switching solution can better adapt to network changes and can automatically reconfigure services based on user and network events. It enables subscriber self-service model that supports edge driven service discovery, provisioning and activation asynchronously. The PNSCA solution enables next generation services that are more self-managing with auto-diagnostics, making them more convenient for users to use, and more cost effective for service providers to deliver.

SOFTWARE ARCHITECTURE

PNSCA is an overlay communications service application that integrates with existing carrier communications and portal services. Unlike other peer-to-peer solutions such as Skype which use proprietary technologies, the PNSCA uses industry open standards such as Session Initiation Protocol (SIP) for signaling, Real-time Transfer Protocol (RTP) for media traffic, and web services. The high level software architecture of PNSCA is shown in the diagram below. The core PNSCA software is unified code that can be ported to all the major operating systems including Windows, Mac, and Linux, and is optimized for Windows CE, Symbian, embedded Linux, and Java Smartphone type devices. The only software component that is platform specific is the client GUI.

The solution is optimal for devices with broadband IP connectivity over any of the landline services such as ADSL and packet cable, and wireless services such as CDMA2000, WiFi, and WiMAX. Unlike typical VoIP clients, the PNSCA client can be deployed as both a server and an integrated telephony client. The server component of the Neltura client can host and proxy commercial off-the-shelf IP phones and soft SIP clients.



At the network service access layer, the PNSCA software (100) implements the transport broker (240) for firewall and NAT traversal, transport layer security (241) for signaling and data encryption and security, client authentication (244) for access control to the services network, and P2P services (242) and distributed data store (243) for the peer-to-peer services networking. Collectively, this network service access layer provides a set of mechanisms for end user devices to act as peers on a virtual network, and together with the profile manager (235) and cluster manager (234) allow users to setup ad-hoc networks which enable IP-to-IP communications.

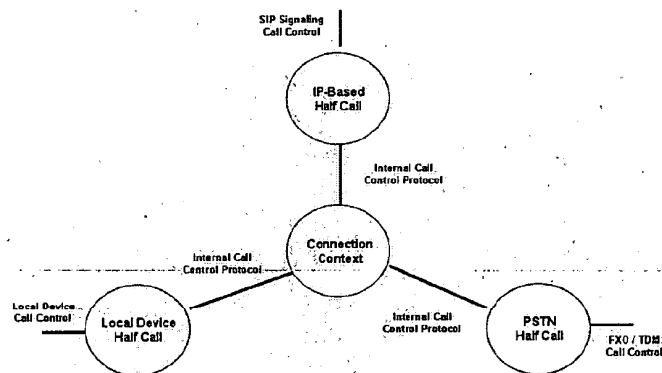
A number of service managers can be integrated above the PNSCA network service access layer. One of the core service managers is a disaggregated softswitch consisting of the call manager (201), feature server (202), SIP user agent (214), SIP registrar and proxy (215), SIP protocol stack (216) including support for header compression and protocol mediation, media control manager (203), media engine (212) which includes media processing, media conferencing, codecs, and media QoS management, TDM line interfaces (210) which includes drivers and line processing software such as caller ID demodulation, and local media (211) which includes drivers and control interface for media input such as microphone and media output such as speaker. It also includes the media transport RTP stack (103) with RTCP support.

The disaggregated softswitch provides sophisticated call routing and network translations, and supports a broad range of a telecom features including PSTN gateway, conferencing, and voice mail. The PNSCA disaggregated softswitch also implements a SIP server function which allows the user to use many commercial off-the-shelf SIP phones, analog-telephone-adapters (ATA), and soft clients. The resolver within the call manager (201) provides the mapping of IP end points and user contacts,

including the support for resolving user known-as PSTN virtual numbers and service persona to actual IP service end points. The user agent (214) provides an integrated user agent and back-to-back user agent allowing the PNSCA software (100) to register into the core network such as IMS or a centralized SIP server.

CALL MANAGER AND FEATURE SERVER

The PNSCA software (100) implements a sophisticated call manager (201) with extensible call features (202). Call Manager is a disaggregated, decentralized softswitch component based on a half-call architecture, including address resolution, signaling and media functions. The call model implements the solution context by providing the routing, translations, and call control for SIP-based IP call-legs, PSTN trunk call-legs, and local device call-legs.



The call manager supports the translation, resolution, and routing of calls between any half-call leg to another half-call leg. Features such as concurrent or follow-me ringing, and conferencing make use of the half-call models to construct sophisticated call objects and call features.

In the first product iteration, Call Manager has the capability to route call legs to and from standard SIP devices, PNSCA-based peer endpoints, POTS lines, and the Local Terminal. Support for routing of call legs over other types of networks is envisioned, the specific network type depending on the application. Such network types may include PRI in the case of a customer premises deployment, or SS7 signaling in the case of integration with the PSTN/PLMN core network. The network types of half calls are abstracted from the switching logic. The only exception to this is the dialed number resolver, which integrates knowledge of the network type of the incoming leg to determine the outgoing address set.

Feature Logic

The Call Manager's design supports segregated feature logic. Examples of supported features include Concurrent Ringing, Delayed Ringing, Conferencing and Push-to-Talk. Support for further calling features is envisioned. Such features include CLASS features, PBX features, voicemail and features integrating presence information.

Resolver

The Call Manager (201) features a novel dialed number resolver that enables subscriber-centric routing, user identity and least-cost routing. In practice, this means that a caller may dial any one of several telephone numbers belonging to the recipient such as the recipient's home phone number, office phone number, or mobile phone number. During call setup, the resolver takes the dialed number and type of network of the incoming call leg and executes a search to determine a set of addresses to which the call should be delivered. This set of addresses is returned to the Call object and the call is delivered to each address in the result set, forking the call path in the case of multiple addresses. The search encompasses the user's list of contacts, other owners of the PNSCA peer on which the resolution is executing, and endpoints on the PNSCA P2P network.

The resolver will discover the network location of all the recipient's communications devices that are enabled with similar device discovery software. The number resolver provides a search infrastructure by which general searches can be propagated. The resolver uses the propagation capability of the underlying peer-to-peer network to perform the search as well as the resolution of the search to the URL of the recipient devices. Having done so, the resolver provides the user's call manager with the Internet addresses needed to initiate calls to one or more of the recipient's devices. In effect, the resolver enable a call in which the caller dialed any one of the recipient's known-as E.164 PSTN numbers instead of a series of individual IP end point addresses such as SIP URI (Session Initiation Protocol User Resource Identifier) or IP address such as 26.127.5.1:2060 associated with the recipient's communications devices.

The resolver algorithm is designed such that the resolution process for a complete call will always pass through resolvers in at most two (2) PNSCA peers. This invariant does not limit Call Manager's generality with respect to conference calling scenarios, as each conference call is set up individually as a call to each participant, who is then joined to another call or conference call already set up. That is, a call itself may pass through more than two PNSCA peers but the resolution phase for any given call will only occur on at most two PNSCA peers.

The resolver invariant behavior can change to support future services. In general, this behavior may change in any case where the second resolver in the call path has more knowledge than the first resolver. Two such potential examples follow:

1. PNSCA Integration with an IXC or with CLECs

Should the PNSCA application be deployed with a centralized PSTN call delivery service through integration with an IXC or with CLECs, the supporting architecture may involve resolution at more than two resolver points. The caller may use a standard SIP phone registered on their home PC's PNSCA Client (A). The resolver at (A) may then determine that based on the information available, the least cost route for the call is through the centralized PSTN call delivery service. The call may then proceed from (A)

to a PNSCA PSTN call delivery service border gateway (B). This gateway (B) may then execute its own resolution, searching for the best PSTN gateway (C) from which to enter the PSTN.

2. Neltura Deployment of an Address Database

Should the PNSCA application be deployed integrated with an Address Database, perhaps supporting 800-service-like number database scenarios, the supporting architecture may involve resolution at more than two resolver points.

The resolver algorithm is motivated by the following design goals:

- Least-cost routing based on available network paths
- Unchanged PSTN-like dialing experience for the caller
- Subscriber-centric virtual identity
- Correct routing
- Deterministic resolution time

A resolution is performed by executing the following steps:

1. The telephone number digits are entered by the user via a user interface or delivered from a constrained device.
2. The telephone number to be resolved is normalized into canonical form. The canonical form can be considered to be a “fully-qualified form” of the telephone number. For example, a local North American telephone number such as “555-1212” owned by someone living in Toronto may be normalized to “14165551212”. This allows unambiguous comparisons to determine if a particular queried telephone number “matches”. Note that the example normalization here is only one example of a normalization that will be performed. Depending on the form of the non-normalized number and the home location of the owner of the number, different patterns of normalization will be necessary.
3. A resolution query request is generated and issued to the local P2P network interface for transmission into the P2P network.
4. If the incoming call leg is a regular IP (such as SIP) end point or a FXO (or local end point)

```
CollectDialedDigits()
CanonicalDialedDigits = CanonicalizeDigitString(DialedDigits)
CalledUserIdList = QueryUserIdByDialedDigits()
if (IncomingLeg.Type is not Fxo)
{
    foreach Contact in CallingDeviceOwner.ContactList
    where Contact.UserId matches any CalledUserId in CalledUserIdList
    {
        CalledLocationIndex = QueryLocationIndexByUserId(Contact.UserId)
        foreach NodeId in CalledLocationIndex
        {
            append NodeId to AddressList
        }
    }
}

foreach Owner in PcClientDevice.OwnerList
{
```

```

foreach KnownAsNumber in Owner.KnownAsNumberList
{
    if CanonicalDialedDigits match KnownAsNumber
    {
        foreach NodeId in Owner.OwnedDevices
        {
            append NodeId to AddressList
        }
        foreach SipUA registered on this PCClient
        {
            append SIP address to AddressList
        }
        If IsLoggedInToTerminal(Owner)
        {
            append LocalTerminal Address to AddressList
        }
    }
}

If AddressList is Empty
{
    if IsLongDistance(CanonicalDialedDigits)
    {
        foreach SharedLineLease in SharedLineLeaseList
        {
            if InDialingArea(CanonicalDialedDigits)
            {
                append NelturaSipShared Address to AddressList
                terminate foreach over SharedLineLeaseList
            }
        }
        if AddressList is empty
        {
            append Fxo Address to AddressList
        }
    }
    if not IsLongDistance(CanonicalDialedDigits)
    {
        append Fxo Address to AddressList
    }
}

// Filter Addresses:
if IncomingLeg.Type is Fxo
{
    remove all NelturaSipShared Addresses from AddressList
    remove all NelturaSip address from AddressList
}

remove IncomingLeg.Address (Sip or Fxo or Local) from AddressList

```

5. If the incoming call leg is a PNSCA-enabled IP end point (NelturaSip)

```

foreach Owner in PcClientDevice.OwnerList
{
    foreach KnownAsNumber in Owner.KnownAsNumberList
    {
        if CanonicalDialedDigits match KnownAsNumber
        {
            foreach SipUA registered on this PCClient
            {
                append SIP address to AddressList
            }
        }
        if IsLoggedInToTerminal(Owner)
        {
            append LocalTerminal Address to AddressList
        }
    }
}

```

```

    }
}

assert(AddressList does not contain Fxo)
assert(AddressList does not contain NelturaSip)
assert(AddressList does not contain NelturaSipShared)

```

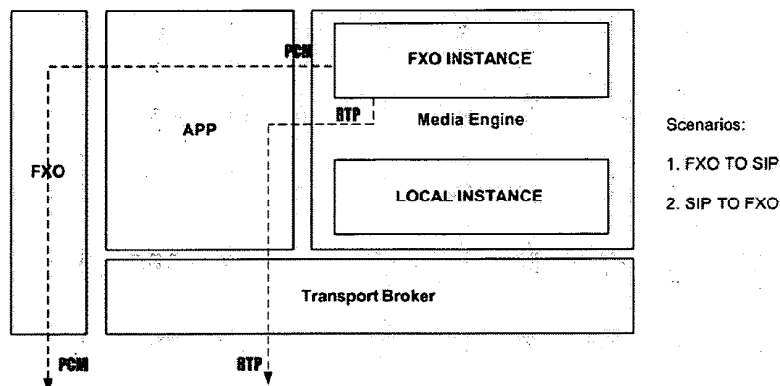
6. If the incoming call leg is a PNSCA-enabled shared resource end point (NelturaSipShared)

Append Fxo Address to AddressList

Media Control

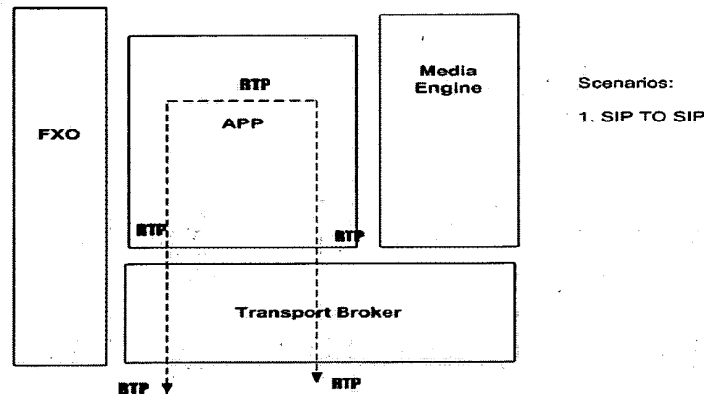
The Media Control Subsystem (203) of the PNSCA client is based on the "Mediator" design pattern. There are three different objects, along with the "Mediator", that interact together to set up and tear down media connections. The objects are: FXO object, SIP Object and Local object. Other variants of the design can include other PSTN objects such as PRI instead of FXO and other variants of IP instead of SIP.

The Mediator is a single entry point for the CallManager to interact with the SIP, FXO and Local Objects. The CallManager initializes the Mediator upon startup of the application and receives a "handle" to the Mediator. Any further communication with the Mediator is done using that "handle". The CallManager interacts with the Mediator to setup and teardown media streams. Once the Mediator receives a setup or teardown request from the CallManger, it will initiate the correct sequence of operations in the Local/FXO/Sip objects, in order to service the request from the CallManager.

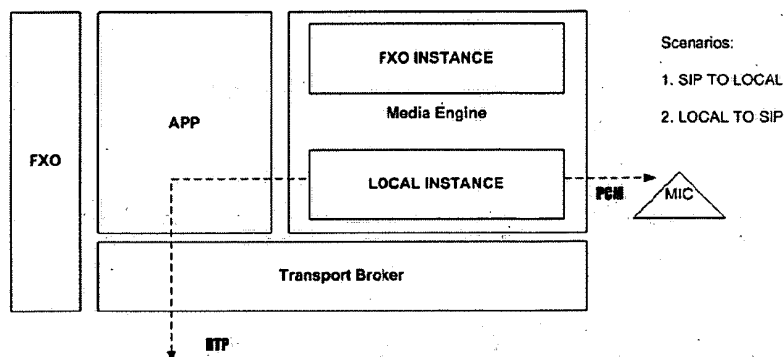


The FXO object is responsible for sending/receiving media from the PSTN. The media coming from the PSTN is received by the Line Interface (210), passed to the FXO object and then fed into the Media Engine (212) through the Mediator. If the user is not in a conference, the Media Engine (212) will play the media to the user's local media (211) or other device as configured by the user. If the user is in a conference, the Media Engine (212) will play the media to the user's local media (211) if the user is not on hold, and also to all the other parties that are in a conference. The other parties maybe SIP hardphones or other PNSCA-enabled clients. The media generated by the user (i.e. by

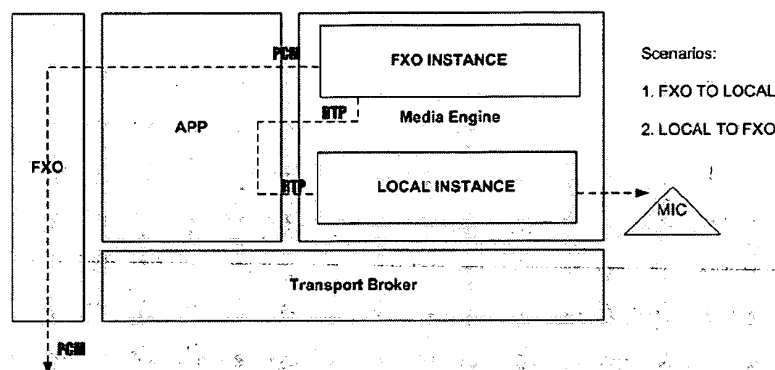
speaking into the microphone or other external recording device) is captured by the Media Engine (212), passed to the Mediator and then to the FXO object, which is then relayed to the PSTN using the Line Interface (210). If the user is in a conference, it is possible that the user might not speak anything at all into the microphone or other external recording device. In this case, media being received from other parties is mixed by the Media Engine (212), and then passed to the FXO object through the Mediator, which is then relayed to the PSTN using the Line Interface (210).



The SIP object is used to send and receive media from SIP hardphones or other PNSCA-enabled clients. The media is received by the SIP object, and passed to the Media Engine (212) through the Mediator, to be played out to the user's local media (211) or other playout device as configured by the user. If the user is in a conference, the received media is mixed by the Media Engine (212), and sent to all the other parties as well as gets played out locally to the user's local media (211) such as sound card. The local media generated by the user (such as from microphone or other external recording device) is captured by the Media Engine (212), passed to the SIP object through the Mediator, which determines where the media needs to be sent (i.e. IP Address and port), and then gets relayed to the appropriate SIP client. Also, if the user is in a conference, the media is mixed by the Media Engine (212), and sent to all the other parties in a conference. The media is sent using RTP (Real Time protocol) over the UDP transport protocol. The CallManager interacts with this object directly only when it needs to set remote SDP or retrieve local SDP.



The Local object maintains user side media and control. Any media coming from FXO or SIP object, is passed to the Local Object by the Mediator before it is passed to the Media Engine (212). Every call has a Local Object associated with it. The only time when a Local object might not participate in a call is when the user is in a conference and the user puts the call on hold. In this case, the media coming from FXO or SIP objects is not passed to the Local object, rather it is passed directly to the Media Engine (212) from those objects, which gets mixed and gets sent to all the parties in the conference. In the case of local media generated from user (such as from microphone) the Media Engine (212) passes the media to the Local Object first (if Local is not in conference), which sends it to the Mediator, which finds the appropriate receiver object (i.e. FXO or SIP) for that media. The CallManager interacts with this object directly only when the call needs to be put on hold.



PROFILE MANAGER

The Profile Manager (235) provides a user account profile configuration synchronization service. Various PNSCA Client components maintain configuration information properly owned not by any given PNSCA Client device or an instance of a user on such a device, but rather by the user's network wide persona or account. Profile Manager makes a best attempt to maintain synchronization between copies of the user's account profile on all devices owned by the user.

Typical User Account Profile Content

The user account configuration profile information is known as the user's PrivateProfile. Examples of information stored in the PrivateProfile include:

- Known-As Numbers
- Contacts / Buddy List
- Shared-Line Leases Given
- Shared-Line Leases Received
- Private Key
- Signing Key

Synchronization Procedures

a. Local Changes to PrivateProfile

When a consumer of ProfileManager's updates its configuration, and commits the change, Profile Manager will find and contact all other PNSCA Clients owned by the same user account and send update the PrivateProfile on those clients as well. If any of the PNSCA Clients owned by the user account happens to be offline when the synchronization attempt is made, then a PrivateProfileChangeTransaction will be inserted into the DDS. The PrivateProfileChangeTransaction will contain markers showing which clients owned by the user account have or have not synchronized the PrivateProfile change.

b. Remote Changes to PrivateProfile

When ProfileManager is contacted by another PNSCA Client providing a PrivateProfile update, the local ProfileManager will accept the update and notify ProfileManager service consumers of the update to their XML-serialized configuration data.

c. Client Log on to P2P Network

On PNSCA client startup, once the client has successfully logged on to the P2P network, Profile Manager initiates a search for any PrivateProfileChangeTransaction that may have been stored in the DDS. If a PrivateProfileChangeTransaction is found and it has not already been applied to this PNSCA Client, the client will first apply the PrivateProfileChangeTransaction. The client will then update the PrivateProfileChangeTransaction record, marking the record to indicate that the retrieving PNSCA Client has applied the PrivateProfileChangeTransaction.

Profile Manager Service Consumer Behavior

Consumers of ProfileManager's synchronized persistence service are responsible for:

- Registering their name with ProfileManager
- Accepting asynchronous updates to their XML-serialized configuration data at runtime
- Providing any XML-serialized configuration data they wish to reside in the PrivateProfile
- Minimizing the number of updates to their XML-serialized configuration data to prevent unnecessary synchronization operations

PRESENCE MANAGER

Presence Manager (232) is a software component of PNSCA Client which provides rich presence information to other components of PNSCA Client (e.g. GUI) or external

applications through Web Services. The Presence Manager of a PNSCA Client node propagates the presence information of the user or the user's devices collected from the node to the other nodes belongs to the same user (called cluster nodes) so that all cluster nodes have the same presence view of the user. It also propagates an aggregated view of the collected presence information to all nodes of the contacts on the user's buddies list which has registered for receiving the notification of presence status. The type of aggregated presence (e.g. aggregated person status, aggregated service status etc.) which the contact nodes can receive is determined by the policy specified by the user for privacy protection.

Algorithm

The Presence Manager of each PNSCA Client node is responsible for receiving presence status, performing aggregation and propagation of presence status to the other registered nodes. The logical steps of each of these tasks are described below:

1. Registration for Receiving Notification of Presence Status
 - When a user logs onto a PNSCA Client node, the Presence Manager of this node sends a REGISTRATION request to all nodes of the contacts' cluster. The Presence Manager will receive either an "accepted" or "denied" response from each of the contact nodes depending on whether the contact allows the requestor to view the contact's presence.
 - The Presence Manager of the nodes which received a REGISTRATION request and returned with an "accepted" response will add the requestor Id to the presence notification list.
2. Cancellation of Registration for Receiving Notification of Presence Status
 - When a user logs off from a PNSCA Client node and if this node is the only node in the user cluster, the Presence Manager of this node sends a CANCELLATION request to all nodes of the contacts' cluster.
 - The Presence Manager of the nodes which received a CANCELLATION request will remove the requestor Id from the presence notification list.
3. Propagation of Presence Status
 - When the Presence Manager of a node received a PRESENCE message from some component of the node (e.g. phone status from Call Manager, or person status from GUI), it retrieves the list of user cluster nodes from the local Cluster Manager.
 - The Presence Manager propagates the original PRESENCE message to each of the user cluster nodes.
 - For each of the contacts on the presence notification list, the Presence Manager retrieves the list of contact cluster nodes from the local Cluster Manager and the sends the aggregated presence to the contact node based on the filtering policy.
4. Aggregation of Presence Status

- When the Presence Manager of a node received a PRESENCE message from some component of the node or from the other node of the same cluster, it performs aggregation of the presence status. For example, the status of a voice service is changed from available to unavailable if all the phone devices associated with the voice service became all in-use.
- The Presence Manager then stores the aggregated presence status in its presence cache.
- The PRESENCE message which is received from any nodes of other clusters should be an aggregated presence. In this case, the Presence Manager does not need to perform any aggregation. It simply stores the received presence in its presence cache.

Communication Protocol

Presence Manager of a node communicates with the Presence Manager of another node by sending messages in XML format through Transport Broker. There are three types of messages supported, namely:

- REGISTRATION Request/Response message
- CANCELLATION Request Message
- PRESENCE Message (based on IETF's standards Rich Presence Extensions to the PIDF)

No acknowledgement response is expected from the recipients of the CANCELLATION Request Message or the PRESENCE messages. Examples of these messages are shown below:

1. REGISTRATION Request Message


```
<?xml version="1.0" encoding="UTF-8"?>
  <PresenceRegistrationRequest xmlns="urn:Neltura:PresenceManager:v1">
    <sender>francis@neltura.com</sender>
    <action>register</action>
  </PresenceRegistrationRequest>
```
2. REGISTRATION Response Message


```
<?xml version="1.0" encoding="UTF-8"?>
  <PresenceRegistrationResponse xmlns="urn:Neltura:PresenceManager:v1">
    <sender>jessica@yahoo.com</sender>
    <result>accepted</result>
  </PresenceRegistrationResponse>
```
3. CANCELLATION Request Message


```
<?xml version="1.0" encoding="UTF-8"?>
  <PresenceRegistrationRequest xmlns="urn:Neltura:PresenceManager:v1">
    <sender>francis@neltura.com</sender>
    <action>cancel</action>
  </PresenceRegistrationRequest>
```

4. PRESENCE Message

```
<?xml version="1.0" encoding="UTF-8"?>
  <presence xmlns="urn:ietf:params:xml:ns:pidf"
    entity="sip:francis@neltura.com">

    <!-- Service #1 -->
    <tuple id="sg89ae" testid="test123">
      <status>
        <basic>closed</basic>
      </status>
      <device-id>urn:esn:600b40c7</device-id>
      <servcaps>
        <methods>
          <method>INVITE</method>
          <method>OPTIONS</method>
          <method>BYE</method>
          <method>ACK</method>
          <method>CANCEL</method>
        </methods>
        <audio>true</audio>
        <duplex>half</duplex>
      </servcaps>
      <contact>sip:gruu-aa@example.com</contact>
    </tuple>

    <!-- Service #2 -->
    <tuple id="sg90ae">
      <status>
        <basic>open</basic>
      </status>
      <device-id>urn:esn:8889001bb</device-id>
      <contact>sms:1234567</contact>
    </tuple>

    <person>
      <status>
        <activities>
          <activity>on-the-phone</activity>
          <activity>busy</activity>
        </activities>
      </status>
    </person>

    <!-- Device #1 -->
    <device id="urn:esn:600b40c7">
      <status>
```

```

        <basic>closed</basic>
    </status>
    <devcaps>
        <mobility>fixed</mobility>
    </devcaps>
    <phone>true</phone>
</device>

<!-- Device #2 -->
<device id="urn:esn:8889001bb">
    <status>
        <basic>open</basic>
    </status>
    <devcaps>
        <mobility>mobile</mobility>
    </devcaps>
    <phone>true</phone>
</device>
</presence>

```

Application Programming Interface

Presence Manager provides services for other software components of PNSCA Client to perform certain tasks through API. These services include:

- Adding or removing a contact.
- Registering for receiving presence status.
- Retrieving presence status of a specified entity type (e.g. person, service, or device).
- Sending presence to Presence Manager.

Use Cases

The following is a set of specific use cases for presence events. Other design variants are possible to accommodate additional presence triggers and user events.

1. Registration for Receiving Notification of Presence Status

a) Triggered by User Logon

After the user logged on to the PNSCA-enabled network, the user's client software NC (100) will register for receiving notification of presence status event in each of the user's other NCs and in each of contacts on the buddy list who are online.

Normal Flow:

1. The user logs on to PNSCA-enabled network from a NC.

2. The user NC sends a presence registration request to each of the user's other NCs.
3. The user NC receives a presence registration response from each of the user's other NCs.
4. The user NC sends a presence registration request to each of the master NC of the contacts on the buddy list who are online.
5. The user NC receives a presence registration response from the master NC of the contacts on the buddy list who are online. The request is rejected if the user is NOT allowed to receive the presence status of the contact.
6. The master NC of the contacts notifies their slave NCs of the changes in the registration list. This enables switching over of the master role to any one of the slave NCs in case of master NC failure.

b) Triggered by Contact Logon

When a contact on the buddy list of a user logged on to the PNSCA-enabled network, the user's PNSCA-enabled Client will receive notification of the logon from the contact's PNSCA-enabled Client. The user's PNSCA-enabled Client will register for notification of presence status event in the contact's PNSCA-enabled Client.

Normal Flow:

1. A contact of the buddy list of the user logs on to the Neltura network.
2. The user NC receives notification of the contact logon event.
3. The user NC sends a presence registration request to the contact's master NC.
4. The user NC receives a presence registration response from the contact's master NC. The request is rejected if the user is NOT allowed to the presence status of the contact.
5. The master NC of the contacts notifies their slave NCs of the changes in the registration list

2. Cancellation of Registration for Receiving Notification of Presence Status

a) Triggered by User Logoff

When a user logged off from the PNSCA-enabled network, the user's NC will cancel the registration for receiving notification of presence status in each of the user's other NCs and in each of contacts on the buddy list who are online.

Normal Flow:

1. The user logs off from Neltura network.
2. The user NC sends a request to each of the user's other NCs for cancellation of the notification of presence status.
3. The user NC sends a cancel request to each of the master NC of the contacts on the buddy list who have been registered for receiving notification of presence status.
4. The master NC of the contacts notifies their slave NCs of the changes in the registration list.

3. Notification of Presence Status Change

a) Triggered by Device Status Change

The presence manager will send a presence message to the user NC and each master NC of the contacts which have registered for receiving notification of the presence status when there is a change in the device status (e.g. a phone line changed its status from on-hook to off-hook).

Normal Flow:

1. The status of a user device has changed.
2. The presence manager of the user NC associated with the device gets notified.
3. The presence manager constructs a presence message with the updated device status.
4. The presence manager sends the presence message to each of the user NCs and the master NC of the contacts which has registered for receiving notification of the presence status.
5. The master NC of the contacts updates their slave NCs of the changes in the presence status.

b) Triggered by User Preference Change

The presence manager will send a presence message to the user NC and each master NC of the contacts which have registered for receiving notification of the presence status when there is a change in the settings of user preference (e.g. the user changed the Do-Not-Disturbed setting from disabled to enabled).

Normal Flow:

1. The user changes the preference using NC GUI.

2. The presence manager of the user NC associated with the device gets notified by the NC GUI.
3. The presence manager constructs a presence message with the updated user preference.
4. The presence manager sends the presence message to each of the NCs which has registered for receiving notification of the presence status.
5. The master NC of the contacts updates their slave NCs of the changes in the presence status.

c) Triggered by Applications

The presence manager will send a presence message to the user NC and each master NC of the contacts which have registered for receiving notification of the presence status when there is a change in application status notified by applications (e.g. user log on/off the IM application).

Normal Flow:

1. The user performs a task which triggers the status change in an application.
2. The application notifies the presence manager of the user NC (where the application is running) the new status .
3. The presence manager constructs a presence message with the new presence status.
4. The presence manager sends the presence message to each of the NCs which has registered for receiving notification of the presence status.
5. The master NC of the contacts updates their slave NCs of the changes in the presence status.

4. Propagation of Presence Status to Applications

a) Propagation of Presence Status to Applications

When the presence manager receives a presence message from the other NCs, it will propagate the presence status to the applications which have registered for receiving the presence status (e.g. NC GUI).

Normal Flow:

1. The presence manager receives a presence message from another NC.

2. The presence manager extracts the presence information from the message and performs aggregation of the status as described in Use Case "aggregation of presence status".
3. The presence manager notifies the registered applications of the new presence status.

5. Aggregation of Presence Status

a) Aggregation of Presence Status

Consolidate the presence status of both the user and the associated contacts for easy access by other components of the node (e.g. GUI, web services).

Normal Flow:

1. The presence manager receives a presence message from another NC.
2. The presence manager extracts the presence information from the message.
3. The presence manager stores the presence information in its cache. The data model is designed to facilitate the easy retrieval of the data by other components.
4. The presence manager determines a status representation for the user based on the various presence data from the user's NCs.

CONTACT MANAGER

The Contact Manager (230) allows the adding, deleting, and updating of the contacts in the contact list and gives all the descriptive information for a contact such as the pictures, display name, email address et In addition, it handles the mechanism to search for a contact in the PNSCA network. Also, it performs the importing of contacts from email and instant messaging applications such as MS Outlook, Skype, and Yahoo Instant Messenger.

CLUSTER MANAGER

Each user in a PNSCA solution can have a set of computing devices on which PNSCA software (100) is running. Each of these devices is called a PNSCA node. The set of PNSCA nodes belongs to a user is referred to as a cluster. Cluster Manager (234) is responsible for keeping track of nodes in clusters and their location information such as node address and geographic location. The application modules (such as Presence Manager, Instant Messenger etc.) can make enquiry for the address of all nodes in a cluster specific to a PNSCA user through the Cluster Manager.

Algorithm

The location information of all nodes in both user's cluster and his/her contacts' clusters are cached in the Cluster Manager in each node. The algorithm for synchronizing the cache with the network is as follows:

1. Joining the user's cluster

After a node successfully joined the peer-to-peer network, the Cluster Manager will perform the following steps:

- Read the location index record of the user's cluster from Distributed Data Store.
- If the location index record does not exist, it means that this node is the first node joining the cluster. The Cluster Manager puts the node location information in a new location index record and stores it to the DDS.
- Else the Cluster Manager adds the node location information to the retrieved location index record and stores it back to the DDS. It also sends a JOIN notification (including the node location information) to the other nodes of the cluster.
- Sends a JOIN notification to each node of all the user's contacts.

2. Leaving the user's cluster

Before a node leaves the peer-to-peer network, the Cluster Manager will perform the following steps:

- Read the location index record of the user's cluster from Distributed Data Store.
- Remove itself from the location index record.
- Store the modified location index record back to DDS.
- Sends a LEAVE notification to the other nodes of the cluster.
- Sends a LEAVE notification to each node of all the user's contacts.

Communication Protocol

Cluster Manager (234) of a node communicates with the Cluster Manager of another node by sending messages in XML format through Transport Broker. There are two types of message, namely the JOIN Notification message and LEAVE Notification message. No acknowledgement response is expected from the recipients of these notification messages. Examples of these messages are shown below:

1. JOIN Notification Message

```
<?xml version="1.0" encoding="UTF-8"?>
<cluster xmlns="urn:Neltura:ClusterManager:v1">
  <NodeJoined>
    <UserId>Francis</UserId>
    <NodeId>28e4eb14</NodeId>
```

```

        <IPAddress>112.23.56.1</IPAddress>
    </NodeJoined>
</cluster>

```

2. LEAVE Notification Message

```

<?xml version="1.0" encoding="UTF-8"?>
<cluster xmlns="urn:Neltura:ClusterManager:v1">
    <NodeLeft>
        <UserId>Francis</UserId>
        <NodeId>28e4eb14</NodeId>
        <IPAddress>112.23.56.1</IPAddress>
    </NodeLeft>
</cluster>

```

Location Index Data Store

The location information of all nodes in each cluster is stored in Distributed Data Store (243). This piece of information is named Location Index, one record per cluster. The key of the Location Index is the hash value of the unique user id plus a suffix ".LocationIndex". An example of Location Index record is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<LocationIndex xmlns="urn:Neltura:LocationIndex:v1">
    <UserId>Francis</UserId>
    <Node>28e4eb14,112.23.56.1:8090</Node>
    <Node>3e0fc6d6,182.77.88.3:9093</Node>
</LocationIndex>

```

Use Cases

The following is a set of specific use cases for cluster manager. Other design variants are possible to accommodate additional user events and scenarios.

1. A new node joins a cluster

This use case describes the sequence of actions to be done in Cluster Manager when a new node joins a cluster.

Normal Flow:

1. The peer-to-peer layer notifies the Cluster Manager of the "Join Finished" event when the node has successfully joined the network.
2. The Cluster Manager of the node retrieves the Location Index record of the cluster with the hash value of the user id as the key from the DDS.
3. The Cluster Manager adds the node information to the Location Index record.

4. The Cluster Manager stores the updated Location Index record in its cache and DDS as well.
5. The Cluster Manager notifies the other nodes in the cluster of the new node information.
6. Each of the nodes which received the new node notification updates its own cache with the new node information.

2. A node leaves a cluster

This use case describes the sequence of actions to be done in Cluster Manager when a new node leaves a cluster.

Normal Flow:

1. The peer-to-peer layer notifies the Cluster Manager of the "Leave Finished" event when the node has successfully left the network.
2. The Cluster Manager of the node retrieves the Location Index record of the cluster with the hash value of the user id as the key from the DDS.
3. The Cluster Manager of the node removes the node information from the Location Index record.
4. The Cluster Manager stores the updated Location Index record in DDS.
5. The Cluster Manager notifies the other nodes in the cluster of the "Node Left" event.
6. Each of the nodes which received the "Node Left" notification updates its own cache (i.e. remove the node entry from the Location Index record).

3. A node goes down

This use case describes what actions will be taken by the Cluster Manager of all nodes in the cluster when any one of the node in the cluster goes down.

Normal Flow:

1. The Cluster Manager of a node monitors the heartbeat of the other nodes in the cluster through the Heartbeat subsystem.
2. The Cluster Manager is notified of the "Node Failure" event by the Heartbeat subsystem when a node in the cluster goes down.
3. Cluster Manager removes the failed node information from its own Location Index cache.
4. If there is at least one more node alive in the cluster (excluding itself), then the Cluster Manager notifies the other nodes in the cluster of the "Node Failure" event, and go to next step. Otherwise, it updates the Location Index record in DDS immediately and skips the remaining steps (job is done!).

5. Each of the nodes which received the "Node Failure" notification updates its own cache. To avoid duplication of updating the Location Index record in the DDS. The node whose information appears first on the Location Index record is responsible for updating the record in the DDS.

4. A node encountered network failure

This use case describes the steps taken by the Cluster Manager of node to recover from a network failure.

Normal Flow:

1. The Cluster Manager of a node monitors the heartbeat of the other nodes in the cluster through the Heartbeat subsystem.
2. If the Cluster Manager of a node detected no heartbeat of all the other nodes in the cluster, then it simply declares there is a problem with its network.
3. Once the network problem is resolved (i.e. the node can see the heartbeat of any one node in the cluster), the Cluster Manager of the node will retrieve the Location Index record from the DDS and updates its cache accordingly.

TRANSPORT BROKER

The Transport Broker (240) handles the details of sending messages among nodes in the PNSCA network and to other IP devices. It provides a higher-level interface so that other components operate at the level of sending and receiving messages, rather than at the level of IP stacks, network sockets, port numbers, firewalls, network address translation (NAT) devices, and so on.

The Transport Layer provides additional services such as fragmentation and reassembly of large messages, and guaranteed message delivery. Finally, the Transport Layer provides a routing system that allows messages to be delivered to particular components in the Neltura solution. This makes message processing much more simple and efficient.

In addition, the Transport Layer is stackable. That is, incoming and outgoing messages traverse a series of filters that can modify the message in many ways. This allows the solution, for example, to add compression, encryption, or logging of messages in a simple way that is transparent to other components.

Since the Transport Layer provides a high-level interface, the underlying implementation details can be changed without any changes to other components. In the initial production system will use UDP/IP as its primary message delivery medium, but for testing and specific deployment purposes the solution can create different Transport Layer implementations. For example, the solution can simulate networks that drop or delay messages in different ways, or we can log all messages passed between nodes.

DISTRIBUTED DATA STORE

The Distributed Data Store (DDS) (243) component provides a distributed, redundant database. Each user of the Neltura system becomes a member of this database, sharing the load of storing database records. The software interface of the DDS is very much like that of an asynchronous database which allows keys to be mapped to records. The keys are globally unique, and the records are arbitrary chunks of binary data. The key determines on which nodes the record will be stored. The DDS, per se, does not impose any structure on the records stored.

The DDS has automatic redundancy; if a node disappears from the network, a new node will be recruited to store all the records the previous node was storing. Furthermore, the nodes chosen to store a given record will typically be well separated geographically, so it is extremely unlikely that all the nodes storing a record will disappear simultaneously.

In order to allow application-specific processing of records, the DDS has the concept of Update Policies. Each record has associated with it a Record Type. When a node attempts to store a record, or update an existing record, the DDS finds the Update Policy (Policy) corresponding to the Record Type. The Policy determines whether the update is allowed to proceed, and what the result of the update will be. Policies could, for example, authenticate updates using a shared-secret or public-key cryptography system, or append the new record to a bounded queue of recent record updates.

PEER-TO-PEER SERVICES

A number of peer-to-peer internetworking techniques such as Sun's JXTA, gnutella, FastTrack, CHORD, and Pastry can be used to implement peer-to-peer services. Many of these design architectures and protocols are well suited for non-time sensitive applications such as file sharing networks. However, time sensitive applications such as communications and presence propagation require peer-to-peer services that are highly scalable, have network fault tolerant characteristics to support high service availability, and have deterministic behaviors such that the service application can be delivered with user predictable system response time.

For the initial instance of the PNSCA design implementation, the peer-to-peer services layer is based on Pastry. Pastry is a generic, scalable and efficient substrate for peer-to-peer applications. Pastry nodes form a decentralized, self-organizing and fault-tolerant overlay network within the Internet. Pastry provides efficient request routing, deterministic object location, and load balancing in an application-independent manner. Furthermore, Pastry provides mechanisms that support and facilitate application-specific object replication, caching, and fault recovery.

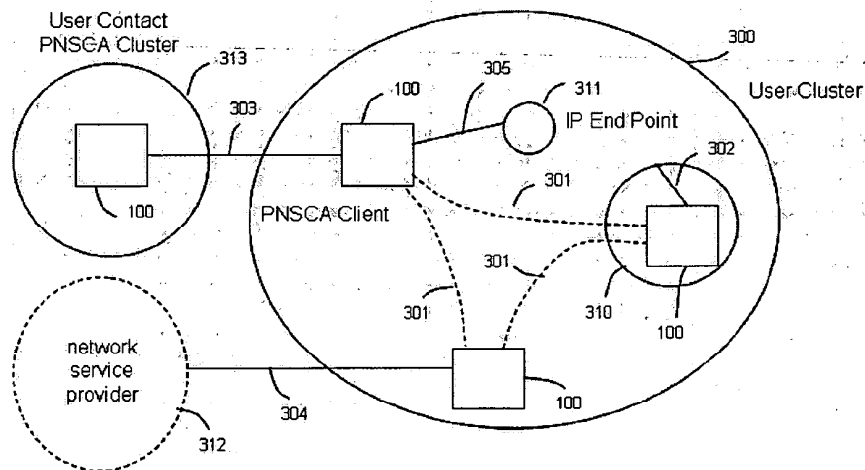
Per the designers of its algorithm, pastry provides the following capabilities. First, each node in the Pastry network has a unique, uniform random identifier (nodeId) in a circular 128-bit identifier space. When presented with a message and a numeric 128-bit key, a Pastry node efficiently routes the message to the node with a nodeId that is numerically closest to the key, among all currently live Pastry nodes. The expected number of forwarding steps in the Pastry overlay network is $O(\log N)$, while the size of the routing

table maintained in each Pastry node is only $O(\log N)$ in size (where N is the number of live Pastry nodes in the overlay network). At each Pastry node along the route that a message takes, the application is notified and may perform application-specific computations related to the message.

Second, each Pastry node keeps track of its L immediate neighbors in the nodeId space (called the *leaf set*), and notifies applications of new node arrivals, node failures and node recoveries within the leaf set. Third, Pastry takes into account locality (proximity) in the underlying Internet; it seeks to minimize the distance messages travel, according to a scalar proximity metric like the ping delay. Pastry is completely decentralized, scalable, and self-organizing; it automatically adapts to the arrival, departure and failure of nodes.

SERVICE INTERFACES

The PNSCA client (100) makes extensive use of web service interfaces for discovering, advertising, extending, aggregating, and accessing multimedia communications services, user presence, and resources amongst connected devices, and with network service providers.



In the above diagram, the user has several PNSCA software enabled devices each represented by (10) that are clustered together into a user device and service cluster (30). Such a software client may be integrated into another software application such as MSN or Yahoo Messenger (20).

The user devices cluster together via interface (1). Through (1), the software component enables each device to autonomously offer a set of services to the user's other devices so that the resources and services, and service presence available on the user's devices can be discovered, aggregated, and accessed. The software component enables certain user devices to autonomously offer a set of services such that the devices can proxy the discovery and invocation of services on behalf of another user device through interface (2). This interface is used in situation where a thin device such

as a mobile phone can leverage the resources of the user's thick device such as a PC via interface (2). The more capable devices such as a PC can also be used as a disaggregated switch and SIP server to host subtending IP devices such as IP phones (61), SIP software clients (61), and ATAs (61) via interface (60).

The software component enables certain user devices (10) to autonomously offer a set of services as a plug-in to another software application, and use this interface (3) to integrate into another software application or platform (20). The software component enables certain user devices to autonomously offer, discover, and access services from a network service provider (40) through a service interface with a service provider (4). This same interface (4) can be used as a client interface into a service provider core network (40) such as an IMS for service control. The software component also enables certain user devices to autonomously offer services and enable the discovery and invocation of services with the user contact's devices and device clusters (11) through interface (5).

Intra-User Service Interface

Service interface set (1) enables each of the user's Neltura-enabled devices to advertise to any of the user's other devices the services that it offers as well as the device resources. This service interface is implemented as a web service and described using XML-based structures.

A user may have a number of Neltura-enabled devices such as PDA, notebook computer, desktop computer, PSTN gateway IP appliance, and mobile phone. These devices may be connected together via an IP network, and may be discovered via a number of means including peer-to-peer device discovery, and clustered together under the same user identity. In the service interface, each device describes the resources and services that it offers through an XML document, and presents a set of service methods which can act on its resources.

As an example, the PSTN gateway may offer the following resource description and service methods to the user's other devices. In this example, the other device discovers the PSTN termination resource with services for terminating calls with specific outgoing dialing digit patterns. The other device can then invoke the method to utilize the service. The invocation method can be entirely done using the standard SIP methods such as INVITE.

Resource offer:

```
<resourceOffer>
  <resourceId>uuid:093a2da1-q345-739r-ba5d-
pqff98fe8j7d</resourceId>
  <leaseStart>Thu Sep 30 22:09:46 PDT 2004</leaseStart>
  <leaseDuration>86400</leaseDuration>
</resourceOffer>
```

PSTN Resource:

```
<pstnResource>
  <resourceId>uuid:093a2da1-q345-739r-ba5d-
pqff98fe8j7d</resourceId>
  <dn>16042751234</dn>
  <pattern>1604275XXXX</pattern>
  <pattern>1604271XXXX</pattern>
  <pattern>1778XXXXXXX</pattern>
</pstnResource>
```

Resource invoke:

```
<resourceInvoke type="pstnResource">
  <resourceId>uuid:093a2da1-q345-739r-ba5d-
pqff98fe8j7d</resourceId>
  <method>route-call</method>
  <args>
    <destination>16042719999</destination>
  </args>
</resourceInvoke>
```

The user's other devices discover the available resources and service methods based on the advertisement of the service offer from the user's PSTN gateway. In some cases, the advertisement is visually advertised to the user, and the user may have to manually provision the configuration parameters into devices such as SIP IP phones which do not have an embedded client that directly supports such a service interface.

The user device that discovers the service offer may act on the service offer immediately, or may defer any actions until the service is required. For instance, after discovery of the service, an IP phone may immediately register with the device. Alternatively, the device may not immediately act on the service offer, but may invoke the service only when the service is required.

Intra-User Service Proxy Service Interface

Service interface set (2) enables a Neltura-enabled device to proxy services to any of the user's other Neltura-enabled devices. This service interface is implemented as a web service and described using XML-based structures.

The service proxy interface enabled a Neltura-enabled device to act as a proxy for another Neltura-enabled device. For example, one particular Neltura-enabled device presents to the other Neltura-enabled devices that it offers resources for signaling header compression as well as wireless media codecs. In such a case, such a Neltura-enabled device offers a set of service interfaces on behalf of a Neltura-enabled device such as a mobile phone.

The resources such as the PSTN interface, mobile service interface, user agent clients, media CODECs, signaling compression, service proxy, media relay, peer end-point relay, and GUI interface of a Neltura client on any of the user's devices is advertised as resources to the user's other Neltura clients. Essentially, the resources of the user's devices are pooled so that the resources can be dynamically allocated to process the delivery of the service.

Depending on resource contention and availability and bandwidth availability, the functional roles of a user's Neltura client on any of the user's devices is dynamically reconfigured. For instance, in the event of the failure of the primary service proxy, any of the user's other Neltura clients can take over the role of the primary service proxy. If media conference capabilities are required on a Neltura client, the service capability can be delegated to any of the user's other Neltura clients. For calls to a user's mobile device, the signaling and media path may be pinned through the user's landline Neltura client so that signaling and media compression can be used to optimize bandwidth utilization in bandwidth and processing constrained devices.

Intra-Device Service Interface

Service interface set (3) enables the Neltura-enabled client on the user device to offer and invoke services with another client on the same user device. The implementation of this service interface is platform specific. For instance, for the Windows platform, it is implemented as a COM object and linked in the form of a DLL.

Through service interface set (1), the Neltura-enabled client on a device can leverage the communications resources of the user's other connected Neltura-enabled devices. On this same device, the user may have another software client such as Microsoft MSN Messenger. A service interface is provided between such an application and the Neltura-enabled client or software module. This Neltura-enabled client or software module may be a separate executable residing on the same computing device or on another computing device, or can be software library such as a DLL or software plug-in which can be loaded by an application such as the Microsoft MSN Messenger client.

For example, upon sign-on, the MSN Messenger client instantiates the Neltura-enabled software module. The Neltura-enabled software module then propagates the status of the subtending Neltura-enabled devices to the MSN Messenger. The user's contacts on MSN Messenger would then be notified of the status of the user using MSN Messenger's host servers. If the user's Neltura-enabled PSTN gateway detects the PSTN line in the off-hook state, the status is propagated to the Neltura-enabled software module, which notifies the MSN Messenger client application. The MSN Messenger client then propagates the "off-hook" status through the MSN host servers to the appropriate MSN Messenger client's on the user's contact list.

Network Service Provider Service Interface

Service interface set (4) enables the Neltura-enabled client on the user device to discover, access, and invoke services with a network service provider. This service interface is implemented as a web service and described using XML-based structures.

An example of such an interface is for Neltura-enabled user to send a short-message to or receive an SMS from users through a service provider network such as a cellular operator. For instance, the Neltura-enabled device invokes the following service to send an SMS to the service provider network.

```
<sendSms>
  <destAddressSet>tel:+16042444567</destAddressSet>
  <senderAddressSet>tel:+16048731234</senderAddressSet>
  <message>I am running late. Please start meeting without me</message>
</sendSms>
```

A second example below illustrates the use of the service interface to discover, subscribe, and initiate an ad-hoc network-based conference. The following assumes that the conference initiator wants to use network-based conference resources, was able to discover the service, and obtained authorization from the service provider to use the service.

```
<setupConference>
  <sessionID>J129KL124UY1</sessionID>
  <conferenceChair>Colin Quon</conferenceChair>
  <conferenceChairContact>+16042444567</conferenceChairContact>
  <conferenceChairEmail>colin_quon@neltura.com</conferenceChairEmail>
  <numberOfPorts>15</numberOfPorts>
  <startTime>14:30</startTime>
  <endTime>15:30</endTime>
  <conferenceDate>02012004</conferenceDate>
</setupConference>
```

A third example shows the use of the service interface to determine the location of a user device.

```
<getLocation>
  <targetUserIdentifier>+16048721234</targetUserIdentifier>
  <requestUserIdentifier>+16042446789</requestUserIdentifier>
</getLocation>
```

Inter-User Contact Service Interface

Service interface set (5) enables the Neltura-enabled client on any one of the user's device to discover, access, and invoke services with the user's contacts. This service interface is implemented as a web service and described using XML-based structures.

A user may have a number of contacts such as family members, co-workers, and friends. Each of the user's contacts has a set of devices that offer specific communications resources and services. The user contact service interface enables the Neltura-enabled device to extend the user's communications resources and services to the user contact's Neltura-enabled devices. The user devices and the user contact devices may be discovered via a number of means including peer-to-peer device discovery, and clustered together under the user and user contact identities, and access permissions

The following is an example of a Neltura-enabled user's service offer to the user's Neltura-enabled contact. In this example, the user offers the contact a set of called-by or known-as phone numbers.

```
<extendCallingByKnownAsNumbers>
  <contactName>Colin Quon</contactName>
  <homeNumber>6042444567</homeNumber>
  <officeNumber>6048726789</officeNumber>
  <mobileNumber>778781234</mobileNumber>
</extendCallingByKnownAsNumbers>
```

A second example illustrates a Neltura-enabled user device offering the user contact a service to share PSTN local access services.

```
<extendPSTNTerminationService>
  <accessLineNumber>6042444567</accessLineNumber>
  <callingAreaRestrictions>604</callingAreaRestrictions>
  <maxPerCallUsageMinutes>15</maxPerCallUsageMinutes>
  <callWaitingPreemptable>TRUE</callWaitingPreemptable>
  <lineOffHookPreemptable>TRUE</lineOffHookPreemptable>
  <lineIDBlocking>TRUE</lineIDBlocking>
  <lineBlackOutStartTime>18:00</lineBlackOutStartTime>
  <lineBlackOutEndTime>21:00</lineBlackOutEndTime>
</extendPSTNTerminationService>
```

CLIENT SIGN-ON AND SECURITY

To become part of the services network, the PNSCA user software clients are first authenticated and authorized by the service gateway (110, 111) or IP core network services (130). Service gateways (110, 111) are also used to bridge network services so that users on the service provider network can interact with peer-to-peer domain users (100) who are not part of the service provider network. The user administration server (130) can be used to grant as well as to deny user access to the service provider controlled peer-to-peer network. It also enables the service provider to provision IP appliances (100), as well as the ability to delete, suspend, and block specific users and devices. In addition to the authentication servers, the PNSCA also has bootstrap servers to link PNSCA clients into the PNSCA peer-to-peer services network. The

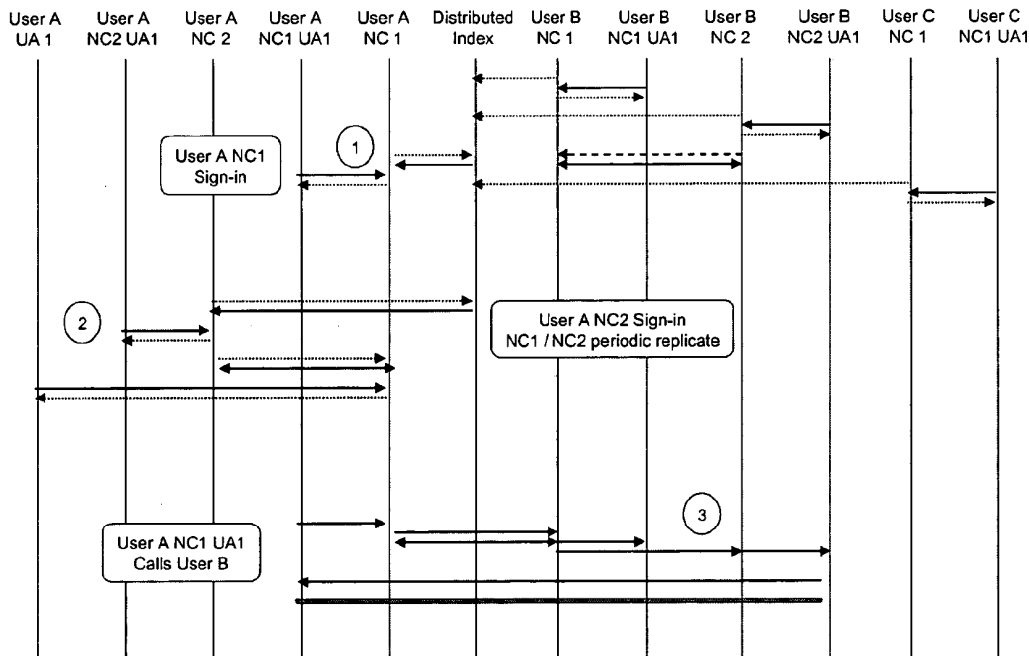
authentication and user administration servers, service gateways, and bootstrap servers are not involved in actual service delivery.

Neltura Client Interaction Context

Below is a summary of the Neltura client interaction context. This diagram is not intended as a call flow diagram other than to illustrate the basic behavior of Neltura clients within the Neltura P2P network.

The sample below has 3 users: user A, user B, and user C. In this case, user A has 2 Neltura clients: "User A NC 1" and "User A NC 2"; and user B also has 2 Neltura clients: "User B NC 1" and "User B NC2".

As noted in the system context above, each Neltura client on a given device always has at least 1 SIP user agent. In this sample, user A has 3 user agents: "User A NC1 UA1", "User A NC2 UA1", and "User A UA" (such as a standard IP phone which is not integrated with a Neltura client). User B has 2 user agents: "User B NC1 UA1", and "User B NC2 UA1". User C has only 1 Neltura client ("User C NC1") along with its integrated SIP user agent ("User C NC1 UA1").



In case 1 above, "User B NC1" sign-on to the distributed Neltura network index. Upon successful sign-on, SIP user agent "User B NC1 UA1" registers with the server agent "User B NC1". Suppose now "User B NC2" sign-on and registers with the distributed index. "User B NC2" found out that "User B" is already sign-on, and hence performs a scan and synchronization of known "User B" devices listed on the index.

After “User B NC2” found “User B NC1”, “User B NC2” then exchanges synchronization and delegation parameters with “User B NC1”. Suppose that “User B NC1” and “User B NC2” decide that “User B NC1” is the primary SIP proxy.

Suppose that “User C NC1” also sign-on to the distributed index, and that “User C NC1 UA1” registers with “User C NC1”.

Again suppose that “User A NC1” sign-on to the distributed index. Upon successful sign-on, “User A NC1” performs a status synchronization of devices in its contact list (“User B” and “User C”) and found that both “User B” and “User C” are signed on. With this status information, “User A NC1” updates the presence and availability status of “User B” and “User C” on its local cache.

In case 2 above, “User A NC2” sign-on and found that another “User A” device is already signed-on. Hence, it performs a scan and synchronization of known “User A” devices “User A NC1”. After “User A NC2” found “User A NC1”, “User A NC2” then exchange synchronization and delegation parameters with “User A NC1”. Suppose that “User A NC1” and “User A NC2” decide that “User A NC1” is the primary SIP proxy.

Suppose now that “User A UA” performs a standard SIP registration. Since “User A NC1” is the primary SIP proxy, “User A UA” is registered with “User A NC1”.

Note that a specific P2P transport pipe may be established between any 2 Neltura clients if any one of them is behind restrictive firewalls. In such cases, the SIP messages from the Neltura client UAs ride on the P2P transport pipe setup between “User A NC1 UA1” and “User A NC1 UA2”. On a periodic basis, “User A NC1” and “User A NC2” synchronizes status parameters and replicate the registrar information. In the event that “User A NC1” becomes inactive or failed, “User A NC2” becomes the new primary proxy server for “User A UA”.

In case 3 above, “user A NC1 UA1” wants to call “user B”. The session “INVITE” is first delivered from “User A NC1 UA1” to its SIP proxy “User A NC1”. “User A NC1” then performs a call delivery look up for “User B” primary client “User B NC1”. Upon finding “User B NC1”, “User A NC1” routes the call to “user B NC1”. “User B NC1” then forks the call to “User B NC1 UA1” and relays the call to “User B NC2”. “User B NC2” then relays the call signaling to “User B NC2 UA1”.

If “User B” answers the call using “User B NC2 UA1”, “User B NC1 UA1” communicates with “User A NC1 UA1” through the pipe established between “User B NC2” and “User A NC1”.

If a standard IP phone is required, the SIP session is first terminated between the IP phone and the Neltura user agent that hosts the IP phone.

* Note that on the distributed index, a Neltura user is uniquely identified by the user ID and password. Associated with this user may be 1 or more registered Neltura clients.

User Registration and Sign-on

Each PNSCA user can have multiple devices or software client instances, but all the software client instances are uniquely identified by a single set of user ID and password. The user ID and password is created when the user is first created on the PNSCA-enabled peer-to-peer network. Generally, the user ID and password is created using an instance of the PNSCA software client on a user device that supports a rich GUI interface such as a PC.

After the user ID, password, and user profile is created, the user can install any number of PNSCA software client instances on any number of user devices, and sign-in concurrently on all the user's devices. Each instance of the user client is uniquely identified by its peer end-point identifier. The persistent private user data is stored on one or more of the user devices, and the public user data is stored in the distributed data-store and user indexed within the PNSCA peer-to-peer network.

Device Registration and Authentication

Device registration and authentication involves PNSCA embedded clients such as the PSTN Gateway appliance and PNSCA enabled IP phones. Such embedded devices are network resources and hence do not have default user IDs. Each such PNSCA enabled embedded device has an embedded unique peer-point ID. These PNSCA embedded devices use this peer-point ID to make itself visible to the PNSCA enabled peer-to-peer network. PNSCA users use this pre-assigned peer-point ID to search and locate the device. This step for discovering PNSCA embedded clients can be omitted if the device is located within a subnet where discovery protocols such as universal plug-and-play and multi-cast DNS can be used. In such environment, the device can be automatically discovered.

A security code is assigned to each PNSCA enabled embedded device, that is stored in ROM or on a detachable media such as a SIM card. After the device is discovered using the pre-assigned peer-point ID or automatically discovered through a mechanism such as MDNS, the user uses the security code to bind and associate the device to the user's list of devices. If the security code entered by the user matches the provisioned security code of the device, the device is attached, and the PNSCA user becomes owner of the device.

The device peer-point ID and security code may also be centrally provisioned on a server such that the device can be located and then administrated by the central server. Through this central server, all user devices can be retrieved from the distributed index, and the user devices can be centrally policed.

Services Security

User contacts are added by first searching for the contact, and then added to the user's contact list. When such user contacts are added, there is no guaranteed that the person being added is really the actual intended individual. After each incoming or outgoing call between the user and the contact, a dialog box is displayed to prompt the user to confirm the authenticity of the contact. If the user indicates the contact is not the intended individual, the user will be prompted with 2 actions: (1) confirm delete of the contact, (2) report the individual as a fraudulent end-point or end-point with confusing or overlapping identity. The sharing of resources is only allowed amongst contacts with identities that have been validated by the user.

Transport layer security is implemented for sign-on and device discovery messaging between PNSCA peer-points, and between PNSCA peer-points and the service provider administration network. Transport layer security such as the use of AES encryption for session signaling and media traffic is implemented on an end-to-end basis, between the user's devices, and between the user's device and the user contact's devices.

Network provider services can be openly discovered by the user's PNSCA client, but a service can only be added if the service provider has been certified by the service providers as an authorized service application and telecom web service. The network service manager within the PNSCA client provides the mechanisms for service authentication and authorization.

SECURITY SERVICES

The Neltura security framework is partitioned into 5 layers: (1) user registration and sign-on, (2) device registration and authentication, (3) transport layer security and encryption for signaling and media, (4) user contact list validation, and (5) service access security.

User Registration and Sign-on

When the Neltura user is first created, the user ID and password is stored in the Kerberos security authentication server in the Neltura user and device manager. In order to sign-on to the Neltura peer-to-peer network, the user is first authenticated, and then obtains a time-bounded service granting ticket from the Neltura Kerberos security service.

Device Registration and Authentication

Device registration and authentication involves Neltura embedded clients such as the Neltura IP Appliance and Neltura enabled IP phones. We decided to follow the proven and familiar mobile device security model using in the cellular industry.

Neltura embedded devices are network resources and hence do not have default user IDs. Each Neltura enabled embedded device has an embedded unique peer-point ID. Neltura embedded devices use this peer-point ID to make itself visible to the Neltura network. Neltura users use this pre-assigned peer-point ID to search and locate the device.

A security code is also generated and assigned to each Neltura enabled embedded device at the time of shipment. This security code is provisioned and associated with the pre-assigned peer-point ID of the device in the Neltura controlled device management server.

For the Neltura embedded devices to be usable, the Neltura user (from a Neltura PC client) enters the security code of the device that is provisioned in the Neltura controlled server. If the security code entered by the user matches the provisioned security code of the device, the device is attached, and the Neltura user becomes owner of the device.

The Neltura controlled device management server has the ability to send commands to global index servers to suspend, block, as well as delete Neltura devices.

Transport Layer Security

Transport layer security is implemented for sign-on and device discovery messaging between Neltura peer-points, and between Neltura peer-points and super-peers (including the distributed index for sign-on). Transport layer security for SIP signaling and RTP media is also implemented using TLS.

User Contact List Security

User contacts are added by first searching for the contact, and then added to the user's contact list. When such user contacts are added, there is no guarantee that the person being added is really the actual intended individual.

After each incoming or outgoing call between the user and the contact, a dialog box will be prompted to the user until the user confirms whether the person is who the person claimed to be. If the user indicates the contact is not the intended individual, the user will be prompted with 2 actions: (1) confirm delete of the contact, (2) report the individual as a fake or has confusing identity.

There exist situations where voice call or invite spamming can occur. If an "invite" and "accept" mechanism is used for adding contacts (the authenticity of the contact is also not guaranteed in such method), then "invite" spamming can occur if the Neltura client code is cracked. Where contacts are directly added based on search results, it is possible for "voice call" spamming. Since Neltura has the ability to delete offending users and the Neltura client is proprietary software, such risks are minimized.

The sharing of PSTN access lines with contacts is provisioned by the Neltura user on a consolidated IP appliance resource sharing screen. The provisioning to shared IP appliance resources with a contact can only be performed after the contact has been authenticated (the user and contact have talked, and the user has accepted the contact as authentic).

Service Access Security

A service can only be added if the service provider has been certified by Neltura as a Neltura authorized service provider. The service plug-in manager provides all the mechanisms for service authorization and billing.

Solution Deployment and Platforms

The Neltura software can be deployed as a single integrated entity or separately as software modules that can be integrated with other software modules. Such software modules may include the Neltura call manager, the user associated number resolver, service interface manager, and software components that constitute the disaggregated telephone switch.

The software can be integrated and deployed as an application on PC, IP phone, IP appliance, PDA, etc. In addition, the software components such as the call manager may be embedded into the on-chip or off-chip application memory such as ROM and FLASH of networking processing and communications system-on-chip silicon devices.

Similarly, these software components can be embedded into the application memory such as ROM and FLASH of IP appliances such as routers, analog telephone adaptors, and IP phones. It can also be embedded, downloaded, or installed on consumer devices such as PDA and mobile phones.

The software components can also be packaged such that it can be linked into or work with other software applications including instant messenger applications. The software component can be packaged as code which can be integrated or as binary objects such as a dynamic linked library which can be linked. The functionality incorporated into these other software application can be manifested as an integrated software application such an enhanced messenger application or can be deployed using other IP networking methods such as through a web portal.

2 GLOSSARY

API	Application Programming Interface
GSM	Global System for Mobile Communications
iLBC	Internet Low Bitrate Codec
IP	Internet Protocol
JXTA	JXTA (Juxtaposition)
MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol
OS	Operating System
P2P	Peer-to-peer
RTP	Real Time Transport Protocol
RTCP	Real Time Transport Control Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
TCP	Transport Control Protocol
UDP	User Datagram Protocol
XML	Extensible Markup Language

2681436
061406