An Overview of Presence Service Architecture and Functionalities

Milan Jankovic , Borislav Odadzic[1]

Abstract – **This paper in the first half describes Presence service Architecture and Functional description, and Presence service specific functionalities of existing network elements and possible new network element. The Session Intitiation Protocol (SIP) extension for Presence and future work based on studying the available literature and documents is presented too.**

Keywords – **Presence service, Architecture, SIP extension for Presence**

I. Introduction

Presence and availability technologies provide the ability to determine the event in which a mobile user is present in a certain location and available for certain events. Presence gives the user the ability to show their availability to communicate, as well as their current location, to the ones they chose. The user will be able to move between different networks and devices, while maintaining their Presence status. This gives Service Providers the opportunity to offer new and existing services to their users.

The ability to determine the presence of mobile user may be accomplished via both ”push” and ”pull” technologies. Push technologies include the ability to set static triggers in network nodes such as Visitor Location Registers (VLR-s) to report the presence of mobile user at a serving switch location based on GSM MAP (Mobile Application Part) messaging. Pull technologies include the ability for the network to proactively poll for presence information about selected mobile users.

The ability to determine availability is based on the possibility of the system to state (on/off) of the mobile device; it’s capabilities and is the mobile user willing to engage in certain activities. All of these network abilities depend on the existence of programs residing on applications that provide this information once the presence status is determine. There are too many applications to adequately introduce them. The application include location based services, mobile commerce, mobile advertising, and many more.

II. Presence Service Reference Architecture

This section describes the reference architecture, the reference points and interfaces used for Presence Service. The generic reference architecture for providing presence service is depicted in Figure 1 [1].

[1]M. Jankovic and B. Odadzic are with Community of Yugoslav PTT, Palmoticeva 2, 11000 Belgrade, Serbia and Montenegro, e-mail: miljankovic@zj.ptt.yu

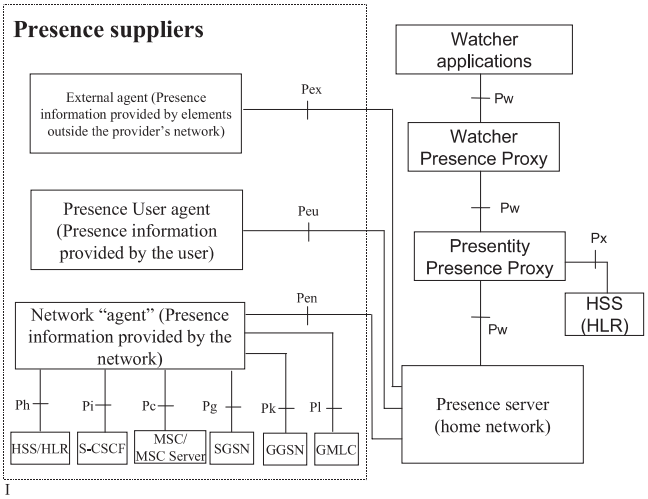

Fig. 1. Reference architecture to support a presence service

A. *Functional description of network elements*

This section describes the Presence Service specific functionalities of existing network elements and possible new network elements.

Presence Server – The Presence Server resides in the presentity’s home network. The Presence Server shall manage presence information that is uploaded by the Presence User/Network/External agents, and is responsible for combining the presence-related information for a certain presentity from the information it receives from multiple sources into a single presence document.

The mechanisms of combining the presence related information will be defined based on presence attributes, and according to certain policy defined in the Presence Server. The Presence Server is not required to interpret all information, the information that the Presence Server is not able to interpret shall be handled in a transparent manner. The Presence Server shall also allow users to fetch and subscribe for receiving presence information.

The Presence Server shall support internetwork operability mechanisms to allow for an interoperable Presence Service across multiple operators’ networks and domains (e.g. external Internet). Mechanisms for locating the Presence Server shall be developed, especially with respect to these internetwork operability aspects.

The Presence Server shall support SIP (Session Initiation Protocol)-based communications with the Presentity Presence Proxy. In the IP Multimedia Sub-system (IMS) the Presence Server is seen as a SIP Application Server, and is lo-

cated using SIP URLs, standard SIP and existing IMS mechanisms – (SIP routing, HSS query, ISC filtering, etc.).

The Presence Server shall support authorization and security mechanisms, at least the following levels of authorization are foreseen: providing presence information to any Watcher application that requests it and provide presence information to only those Watcher applications in an "allowed" list.

The Presence Server may also support authorization and security mechanisms that is based on asking permission from the Presence User agent on a case-by-case basis.

The Presence Server may support rate-limiting or filtering of the presence notifications based on local policy in order to minimize network load.

The Presence Server could be extended to a generic State Agent, supporting subscriptions and notifications regarding other types of events than presence as well. An example for such event is the combined presence of a whole buddy list.

Watcher and Presentity Presence Proxy – When a Watcher application intends to access some presence information of a presentity, it first needs to find the Presence Server containing this information. The Watcher Presence Proxy provides the following functionality: address resolution and identification of target networks associated with a presentity, authentication of watchers, interworking between presence protocols for watcher requests and generation of accounting information for watcher requests.

The Presentity Presence Proxy provides the following functionality: determination of the identity of the presence server associated with a particular presentity and generation of accounting information for updates to presence information.

The Presentity and or the Watcher Presence Proxies may also be responsible for providing network configuration hiding.

Communications between the Presentity Presence Proxy and the Watcher Presence Proxy shall be based on SIP as shown in Figure 2. Other IP-based mechanisms may also be needed to support the delivery of large amount of presence information. Support for non-SIP based Watchers may be provided by the use of an interworking functions located at the Watcher Presence Proxy.

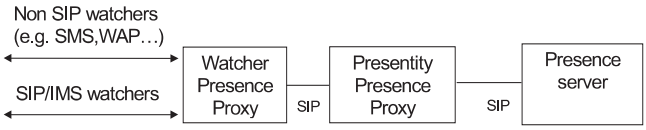

Fig. 2. Communications between the Presentity Presence Proxy and the Watcher Presence Proxy for Watchers

External Agent – The Agent elements in the Presence Architecture are functionally distinct from the Presence Server functional element. The generic function of the Agent elements is to make presence information available to the Presence Server element in standardized formats across standardized interfaces.

The External Agent element provides the following functionality: the External Agent supplies Presence information from external networks, the External Agent sends the Presence information across the Pex interface according to the format standardized for the Pex interface and the External Agent handles the interworking and security issues involved in interfacing to external networks.

Examples of Presence Information that the External Agent may supply, include: third party services (e.g. calendar applications, corporate systems), Internet Presence Services and other Presence Services.

Presence User Agent – The Agent elements in the Presence Architecture are functionally distinct from the Presence Server functional element. The generic function of the Agent elements is to make presence information available to the Presence Server element in standardized formats across standardized interfaces.

The Presence User Agent (PUA) element provides the following functionality: the Presence User Agent collects Presence information associated with a Presentity representing a Principal, the Presence User Agent assembles the Presence information in the format defined for the Peu interface, the Presence User Agent sends the Presence information to the Presence Server element over the Peu interface, the Presence User Agent shall be capable of managing the Access Rules and the Presence User Agent shall handle any necessary interworking required to support terminals that do not support the Peu reference point.

From a conceptual view, the PUA element resides between the presence server and the user's equipment as illustrated in the reference architecture in Figure 1. In reality, a Presence User Agent may be located in the user's terminal or within a network entity.

Where the PUA is located in a terminal, the terminal shall support the Peu interface to the presence server as illustrated in Figure 3.

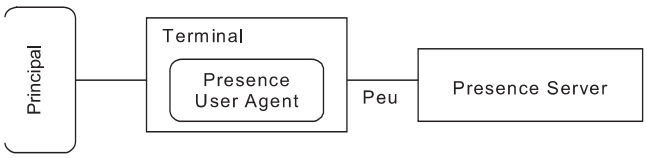

Fig. 3. Terminal based Presence User Agent

Where the PUA is located within the network, the particular network entity shall support the Peu interface to the presence server as illustrated in Figure 4. In such a case an additional functionality may be required to resolve the location of the presence server associated with the presentity. In this case, the interface between the terminal and the Presence User agent is outside of the scope of standardisation of the presence service.

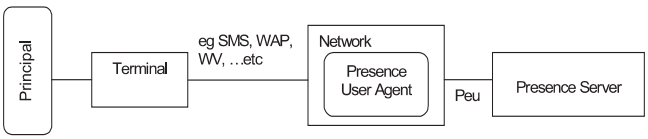

Fig. 4. Network based Presence User Agent

Network Agent – The Agent elements in the Presence Architecture are functionally distinct from the Presence Server functional element. The generic function of the Agent elements is to make presence information available to the Presence Server element in standardized formats across standardized interfaces.

The Network Agent element provides the following functionality:

- The Network Agent receives Presence information from network elements within the Operator's network.
- The Network Agent associates Presence information with the appropriate Subscriber/Presentity combination.
- The Network Agent converts the Presence information into the format standardized for the Pen interface.
- The Network Agent sends the Presence information across the Pen interface.
- The Network Agent may Push Presence information to the Presence Server alternatively some network elements may be queried, signaled, or provisioned to deliver Presence information. For those elements that require querying or signaling, the Presence Server makes a request to the Network Agent directing it to acquire the Presence information. The Network Agent then issues the appropriate commands to the element.

III. Presence and Availability Management (PAM)

This section of the document specifies the Presence and Availability Management Service Capability Feature (SCF) aspects of the interface. All aspects of the Presence and Availability Management SCF are defined here, these being: Sequence Diagrams, Class Diagrams, Interface specification plus detailed method descriptions, State Transition diagrams, Data Definitions and IDL Description of the interfaces.

The process by which this task is accomplished is through the use of object modelling techniques described by the Unified Modelling Language (UML).

Consider the following simple but desirable scenario for a communication service: An end-user wishes to receive instant messages from her management at any time on her mobile phone, from co-workers only on her desktop computer, and in certain cases for the messages to be forwarded to e-mail or even a fax machine/printer. The senders may know her availability for various forms of communication in the way she chooses to reveal it or alternatively the senders may never know how she will be receiving their messages. This scenario spans over multiple services and protocols and can only be solved currently by a proprietary solution that maintains the required information in an ad-hoc fashion within the application.

PAM is not a replacement for the protocols being standardized for various communication and network services. PAM attempts to standardize the management and sharing of presence and availability information across multiple services and networks.

The PAM specification is motivated by the observations that

- The notions of Identity, Presence and Availability are common to but independent of the various communication technologies, protocols and applications that provide services using these technologies.
- Presence does not necessarily imply availability. End-users or organizations require greater control over making
- Themselves available through various communication devices.
- Presence based services need to address privacy concerns on who can access presence information and under what conditions.

Management of availability will span over multiple communication services and service providers

IV. Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying and terminating sessions with one or more participants. These sessions include Internet multimedia conferences, Internet telephone calls and multimedia distribution. Members in a session can communicate via multicast or via a mesh of unicast relations, or a combination of these. SIP invitations used to create sessions carry session descriptors which allow participants to agree on a set of compatible media types. SIP supports user mobility by proxying and redirecting requests to the user's current location. Users can register their current location. SIP is not tied to any particular conference control protocol. SIP is designed to be independent of the lower-layer transport protocol and can be extended with additional capabilities.

SIP is a request-response protocol in that clients send SIP requests to servers which send back responses to these requests. A typical application is generally made up of both client and server functionality. A User Agent is an intelligent end point. A User Agent initiates a session by creating and sending an INVITE request. This request can either be sent directly to another User Agent or one or more proxies can be traversed. Proxies forward requests based on local policyand information contained in the SIP request. A typical SIP call / session set up and tear down between two User Agents, traversing a SIP proxy can be seen in Figure 5.

In the Figure 5, a User Agent acting as a User Agent Client, initiates an INVITE request and sends it to a second User Agent, acting as a User Agent Server, via a proxy. The server returns an OK response. When the client receives the OK response, an ACK request is sent to acknowledge the receipt of the final response and the communication between the two user agents is set up; a session is now in progress. The ACK may not traverse the proxy as the network path has been established. The BYE request indicates that one side wants to terminate the session.

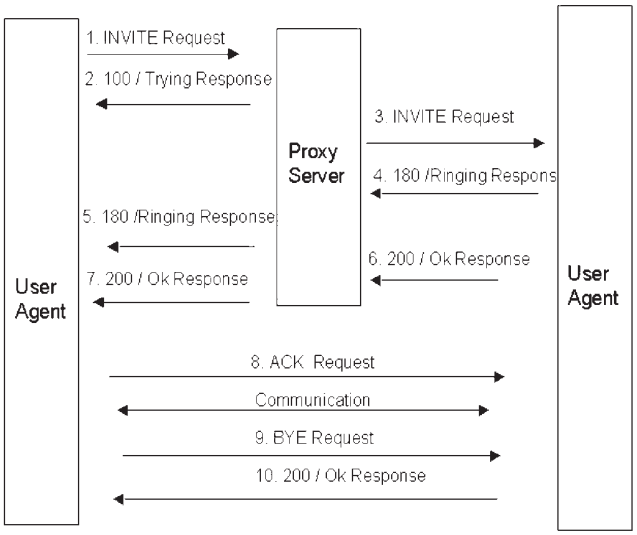

Fig. 5. A typical call set-up and tear down

V. SIP extension for Presence

Presence is defined, as subscription to and notification of changes in the communications state of a user. This communications state consists of the set of communications means, communications address, and status of that user. A presence protocol is a protocol for providing such a service over the Internet or any IP network.

This section proposes the usage of the Session Initiation Protocol for presence [9]. This is accomplished through a concrete instantiation of the general event notification framework defined for SIP, and as such, makes use of the SUBSCRIBE and NOTIFY methods defined there. User presence is particularly well suited for SIP. SIP registrars and location services already hold aspects of user presence information; it is uploaded to these devices through REGISTER messages, and used to route calls to those users. Furthermore,SIP networks already route INVITE messages from any user on the network to the proxy that holds the registration state for a user. As this state is user presence, those SIP networks can also allow SUBSCRIBE requests to be routed to the same proxy. This means that SIP networks can be reused to establish global connectivity for presence subscriptions and notifications.This event package is based on the concept of a presence agent, which is a new logical entity that is capable of accepting subscriptions, are changes in user presence. The entity is defined as a logical one, since it is generally co-resident with another entity. This event package is also compliant with the Common Presence and Instant Messaging (CPIM) framework that has been defined in [7]. This allows SIP for presence to easily interwork with other presence systems compliant to CPIM.

When an entity, the subscriber, wishes to learn about presence information from some user, it creates a SUBSCRIBE request. This request identifies the desired presentity in the request URI, using a SIP URI, SIPS URI or a presence URL [9,7]. The subscription is carried along SIP proxies as any other request would be. It eventually arrives at a presence server, which can either terminate the subscription (in which case it acts as the presence agent for the presentity), or proxy it on to a presence client. If the presence client handles the subscription, it is effectively acting as the presence agent for the presentity. The decision at a presence server about whether to proxy or terminate the SUBSCRIBE is a local matter; however, we describe one way to effect such a configuration, using REGISTER.

VI. Conclusion

The SIP and PAM study looked at a relatively new aspect of the interaction between PAM and SIP. This work is related to the problem that Parlay's generic nature makes the relationship with order technologies such as Session Initation Protocol (SIP) difficult to define. The issue arise when analysing how the functionality of new protocols, SIP in particular, can be mapped onto the current Parlay interfaces.

At first glance the proposed SIP Presence Extensions seem to map well onto the Parlay PAM Event Management Interfaces. More network is required to see how the order Parlay PAM Interfaces fit into a SIP enabled network. In particular one could look at how Parlay PAM could be used to manage a presentity's presence information, which would require looking at an application that actually registers its information with a presence server, rather than subscribing to another presentity's presence information.

References

[1] 3GPP: Technical Specification Group Services and System Aspects; Presence service, TR 23.841, v1.1.0 (2002-03), Release 6.

[2] 3GPP: TS 32.200, Charging Principles.

[3] 3GPP: TS 32.225, Charging Data Description for the IMS domain.

[4] 3GPP: TS 33.210, Network Domain Security.

[5] 3GPP: TS 23.228, IM Subsystem Stage-2.

[6] 3GPP: TS 23.218, IP Multimedia (IM) Session Handling; IP Multimedia (IM) call model.

[7] IETF: CPIM Presence Information Data Format, Internet Draft in IMPP WG http://www.ietf.org/internet-drafts/draft-ietf-impp-cpim-pidf-01.txt, October 2001

[8] http://www.pamforum

[9] http://www.sipforum

[10] Rosenberg et al.: SIP Extensions for Presence, Internet-Draft in SIMPLE WG, http://www.ietf.org/internet-drafts/draft-ietf-simple-presence-06.txt, April 2002.

[11] http://www.3gpp.org