Presence management and merging presence information for NGN services

Sebastian Schumann[1] and Eugen Mikoczy[1] and Pavol Podhradsky[1] and
Feliciano Muruchi[2] and Michael Maruschke[2]

[1] Slovak Technical University, Ilkovicova 3, 812 19 Bratislava, Slovak Republic
{schumann,mikoczy,podhrad}@ktl.elf.stuba.sk
[2] Deutsche Telekom AG, Hochschule fuer Telekommunikation Leipzig,
Gustav-Freytag-Str. 43-45, 04277 Leipzig, Germany
{feliciano.muruchi,maruschke}@hft-leipzig.de

Abstract. This paper describes an approach for interworking scenarios between Session Initiation Protocol (SIP) based and non SIP based frameworks (e.g. web services) in case of the presence management service. The characteristics of the concept of a centralized presence management will be introduced.
Based on the Parlay X web service framework a presence management service solution has been developed, deployed and tested. The re-usability of the developed presence management service solution within an IP Television (IPTV) framework and the future potential of the presence service in various IPTV applications are analyzed.

1 Introduction

The presence service is one of the key services that next-generation architectures like the IP Multimedia Subsystem (IMS) can provide already.

Based on a simple SIP infrastructure (non-IMS), this paper will show the usefulness of interconnecting frameworks around the presence service. The authors deployed the SIP based presence management service interconnected with web services successfully. An existing testbed has been extended with a web service framework as first step. After the web service framework has been deployed in a stable state, standards have been discussed and solutions proposed. Moreover, prototypical applications for both presence client scenarios (presentity and watcher) have been defined, deployed and tested.

The paper will also discuss the re-usability of a once deployed presence service from the IPTV framework. The concepts are similar and proof again that service enablers and framework interconnections have a great future. Standardization is currently on-going within this area, the authors contributed here as well.

2 Presence architecture within the SIP framework

In a SIP framework, the proxy server passes all presence related messages (SIP SUBSCRIBE and SIP PUBLISH) to the presence server. It handles the publications and stores the presence information of the publishing user [1–3].

In contrast to the presence server, which is usually reachable for the users only through the SIP core, the Extensible Configuration Access Protocol (XCAP) server advertises its interface directly towards the User Equipment (UE). The XCAP protocol allows clients to read, write and modify XML data on the server.

In combination with a presence server, the XCAP server manages data like buddy lists, presence policy and presence information and makes it accessible through the XCAP interface to the presence clients.

2.1 The role of presence

Aggregation of presence information results in many presence states from many sources altogether collected by a single source in the network. To organize the presence states and also to create useful presence information for each service, a presence state model that prioritized the information will be proposed. It is up to the presence server to create a final state from all this information. Some major characteristics of this status model are listed below.

The state model should group all states that cannot be or have not been mapped into service specific groups. Attributes (see Tab. 2) that can be mapped and their respective communication means should be aggregated and prioritized according the users preferences.

The following list is an extract of possible service presence states:

– Voice presence (wired UE): online, present at phone, busy, on the phone
– Voice presence (wireless UE): online, profile (silent/on meeting - normal - work - at home - on road), out of reach, do not disturb, on the phone, current location
– Calendar presence: activity, time frame, location
– GPS/RFID enabled devices: location

Common attributes have been outlined already. An automatized privatization should take part acc. the following presentities:

– User set state preferred over automatic state. Example: mobile state, e.g. on the road, overwritten by active call, e.g. on the phone
– Real location information preferred over static assumption. Example: GPS information modifies calendar presence state
– Communications ranking.

All attribute ranking are proposals that the user should be able to modify, if required.

2.2 Merging presence sources for real-world scenarios

Nowadays, operators do not only operate telco domains (e.g. plain telephony using VoIP or some NGN), but offer various communication services through different domains (e.g. web 2.0 applications, web services, IPTV). The challenge appears now when different domains should be interconnected and share presence

information. Concretly, different frameworks will be taken into account to explain the merger of handling the presence information in detail: The SIP domain, which will be the presence management domain, and the web service and IPTV domains respectively, which will access the presence information.

One presence management service within different domains requires two conditions to be full-filled:

- The methods that SIP uses to publish/subscribe to presence information must be mapped to methods from the other domain. This ensures that the presence information can be accessed from the other domain.
- The actual presence information must be mapped towards the other domain. This ensures that the other domain understands the information it exchanges.

If the requirements from above are implemented, the domains can communicate with each other and also understand the content. The non-SIP domain can request presence information, process it and also publish states.

3 Parlay X web service framework

The Parlay X framework has been defined by the standardization bodies ETSI, 3GPP and Parlay Group. Its specifications describe, how application developers can use different telecommunication services through an open standardized interface to generate new, innovative applications. The used technology is the web service architecture, which is based on SOAP[3] and Web Service Definition Language (WSDL).

3.1 Presence service within the Parlay X framework

The main functionalities of the presence service within the Parlay X framework are specified through logical interfaces that are standardized in [4]. This specification defines namespaces, sequence diagrams, data definitions, fault definitions, a detailed description of these logical interfaces and its corresponding WSDL descriptions. The specified interfaces are:

- The PresenceConsumer interface
- The PresenceNotification interface
- The PresenceSupplier interface.

The PresenceConsumer interface and the PresenceNotification interface define the watcher side of the presence architecture. The latter interface is a call back interface, which allows the presence web service to send notification requests to the web service client. Hence, the watcher does not only act as client, but also as server in the web service context by offering this interface. This approach is

---

[3] Originally defined as Simple Object Access Protocol but acronym was dropped with Version 1.2

not common or practical for a web service architecture. It is however the only possibility to receive information in a subscription mode. This is essential for the presence service.

The PresenceSupplier interface is used for the presentity side in order to publish presence information and to manage the authorization of its presence information.

3.2 Interconnecting presence web services with the SIP framework

Each interface supports different methods in order to meet its function. These methods have to be mapped to the corresponding methods, which are used in SIP for the presence service. [5] proposes presence mapping scenarios between the Parlay X web service framework and the SIP/IMS framework.

The analysis of [5] revealed some issues regarding the exact sequences of protocol messages. These sequence diagrams do not appear standard conform regarding [1–3] particularly the use of the SIP message ACK and the lack of the final SIP 200 OK responses. Furthermore, this dialog is not practical in terms of using the startPresenceNotification method.

Fig. 1 shows, how the mapping for the watcher side of the Parlay X specification could work successfully with an existing SIP presence core. Therefore, the PresenceConsumer interface and the Presence Notification interface are considered. Finally, the proposed diagram is oriented on [5] and is modified for practical reasons.

For the sake of completeness, all response messages (SIP and SOAP) were added. The main differences towards [5], sec. 5.1 are the following:

1. SIP ACK message is not implemented.
2. Implicit SIP NOTIFY after 200 class response for SUBSCRIBE is necessary.
3. SOAP startPresenceNotificationRequest requires a new SIP subscription.

The authors explain those changes with the following reasons:

1. The SIP ACK method is not valid within the SIP Event framework. It is only part of the SIP three-way handshake for session set-up. The ACK(nowledgement) described in [5], sec. 6.1.1 that changes state (S1) to (S2) is the 200 class reply for SUBSCRIBE.
2. The SIP NOTIFY is omitted in [5], sec. 5.1, however mentioned in sec. 6.1.1. As it is implictly required after the 200 class message ([3], sec. 3.1.4.1), it has been added to the figure.
3. The startPresenceNotificationRequest issues a new subscription on the SIP side. This major change towards [5] is caused by the fact that the first SUBSCRIBE (whose NOTIFY response could be used acc. [5], sec. 6.1.3) could be a one-time polling. StartPresenceNotification expects however a SIP 'subscriber-watcher' behavior, i.e. expiry not equal zero and hence ongoing notification. This is not explicitly excluded for the subscribePresence event. Limiting subscribePresence for expiry not equal zero does not make sense, as the notifySubsciption is not used to actually elaborate the presence information and only indicates a successful subscription.
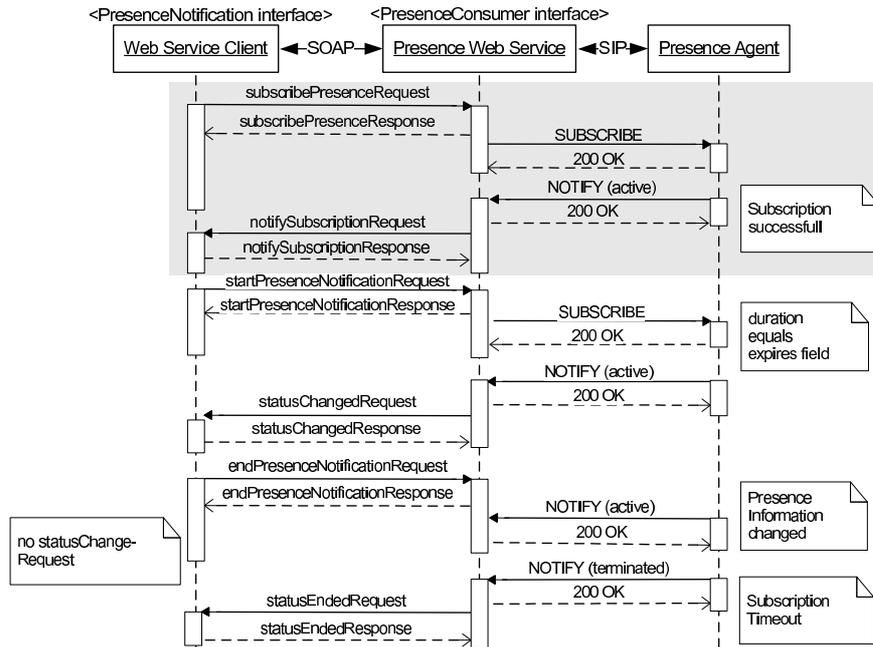
Fig. 1. Signaling flow overview of watcher side (as web service client)

For clarity, the getUserPresence fetching from [5], sec. 5.1 has been omitted in Fig. 1. The initial presence subscription (SOAP differentiates between subscription and actual presence information exchange) has been highlighted in light grey in Fig. 1. The following procedure shows the exchange of the actual presence information.

[5] does also not yet consider the mapping of attributes, which are used within the methods of both sides. Tab. 1 shows the proposed approach for the attribute mapping as a possible solution.

This proposal was implemented and tested successfully and will be described in section 3.3.

In addition to the previous steps, the presence status information needs to be mapped from the Presence Information Data Format (PIDF) and its extension Rich Presence Extension to PIDF (RPID), which is used by the SIP domain, to the defined structure within the Parlay X specification. Tab. 2 shows the mapping of basic presence information.

This mapping shows that all of the activity values, which are specified in the Parlay X specification, can be mapped clearly to PIDF/RPID. As not all activity elements of the RPID standard can be found within the Parlay X specification, some information will still be lost in case two entities of different domains (SIP and web service) would interact with each other.

Table 1. Parlay X attributes/SIP parameter mapping table for watcher side

SOAP methods	Parlay X attributes	SIP parameters
subscribePresence	presentity	TO header field
	attributes	PIDF[a] document
	reference (endpoint, interfaceName, corre-lator)	TO header field, FROM, CONTACT, watcher identifier
notifySubscription	presentity	FROM header field
	decisions	PIDF document
startPresenceNotification	presentity	TO header field
	attributes	PIDF document
	reference (endpoint, interfaceName, corre-lator)	TO header field, FROM, CONTACT, watcher identifier
	duration	Expires header field
endPresenceNotification	reference (endpoint, interfaceName, corre-lator)	TO header field, FROM, CONTACT, watcher identifier
subscriptionEnded	presentity	FROM header field
	reason	optional REASON header field
statusEnded	correlator	Take correlator from reference from startPresenceNotification
statusChanged	correlator	Take correlator from reference from startPresenceNotification
	presentity	TO header field
	changedAttributes	PIDF document

[a] PIDF: Presence Information Data Format

Table 2. Presence status information mapping table for watcher and presentity side

PIDF + RPID	Parlay X
status=open	AVAILABLE
RPID = busy	BUSY
RPID = on-the-phone	ON_THE_PHONE
RPID = away OR status=closed	AWAY
RPID = meeting	MEETING
RPID = lunch	MEAL
RPID = holiday	HOLIDAY
RPID = steering	STEERING
RPID = in-transit	INTRANSIT
RPID = travel	TRAVEL
RPID = sleepingv	SLEEPING
RPID = permanent-absence	PERMANENTABSENCE

3.3 Testbed solution

By offering a presence web service, which is described in chapter 3.2, any possible system or service can be extended directly with a web service client or through a proxy in order to communicate its presence information towards the presence service. Within the laboratories of STU, an example communication scenario has been deployed (Fig. 2).
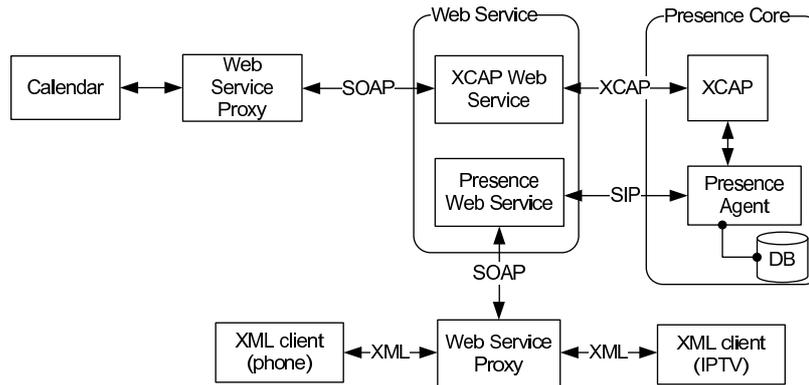

Fig. 2. Test platform presence web service

The tested scenario shows the usability of the discussed interconnection between the web service framework and the SIP framework. Part of this scenario are two non-SIP based entities. One entity is an IP phone with an XML interface (for provisioning or service integration) and acts as a watcher over a proxy towards the presence web service. The other entity is represented by an application, which retrieves calendar data from a calendar service and acts as presentity by publishing it. Since the pure publication of presence information can be done through pidf-manipulation over XCAP as well, the web service access offered by the XCAP server is used. Only the PresenceConsumer interface was implemented within the testbed presence web service.

The example sequence diagram in Fig. 3 shows the signaling flow for the watcher side, which receives presence information in a subscription mode.

1. The IP phone triggers the subscription mode with an HTTP GET message, which executes a script on the web service proxy.
2. The web service proxy creates a startPresenceNotificationRequest SOAP message, which is send to the presence web service.
3. The presence web service subscribes the presence information.
4. The notification is received by the presence web service.
5. In case the notification is active and contains PIDF data, this information is sent to the web service proxy using the statusChangedRequest SOAP message.
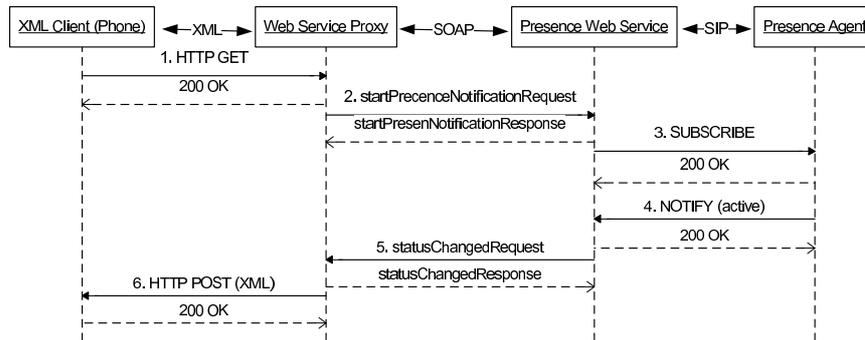
Fig. 3. Sequence diagram of subscription mode

6. As a result the presence information is revealed on the phone over a XML document.

   Note: The 200 OK messages and the SOAP response messages acknowledge the previous requests (like HTTP GET) on the application layer.

4   NGN based IPTV frameworks and use cases for presence

The following section describes IPTV services and related aspects for the usage of presence for NGN based IPTV, which have been defined within ETSI TISPAN. Both NGN based IPTV concepts were specified by TISPAN: NGN dedicated IPTV and IMS based IPTV [6] (where the IMS presence framework from 3GPP and OMA could be reused) will be explained, focussing on the role of presence in particular.

4.1   Services within the IPTV frameworks

Basic IPTV services and their implementation within the NGN framework can communicate with the presence framework to update the presence states:

– Broadcast TV (BC with or without trick modes) – delivery of linearly broadcasted TV channels. Trick Modes enable control playback and pause, forward, rewind content. Presence helps service control to follow user actions.
– Pay Per View (PPV) – users pay e.g. only for particular show or time period, not whole TV channel or package. Presence could be used to trigger PPV usage.
– Content on Demand (CoD) – user requests content consumption on demand.
– Personal Video Recording (PVR) – user can record content in network (network or n-PVR) or locally in STB (client or c-PVR). Presence could be used for decision, if content is recorded locally or in network (if User Equipment (UE) is offline, only n-PVR is used).

Basic services usually use presence to observe the actual state of the service (e.g. channel actually watched, state of VoD preview).

The hierarchical presence model can be used to aggregate information not only by UE updates, but also directly from IPTV functional elements (its internal state, service state or service action data). The utilization of web based presence aggregation concepts and relevant correlations of data allow the combination of presence with IPTV services. Perfect examples are combinational services, where IPTV services are combined with presence to improve the user experience (through enhanced interaction, profiling, personalization or targeting). More advanced IPTV services require also a more complex presence state model and correlation of presence states with the service behavior:

- Interactive TV (iTV) – service providing interactivity between provider/broadcaster and end user or between several users. Presence helps to identify available users for such an interactivity.
- User Generated Content (UGC) – content produced by the end user with the intention to share it with other users. Distribution of UGC makes sense only to the available (online) party, that is willing to receive content (user can set presence to busy to avoid any push services like UGC or PushCOD).
- Content Recommendation (CR) – service advisory for favorite shows based on user's preferences and behavior (here presence is crucial to define right recommendation because if system fails in identification of correct presence state for exact user, it will dismiss correct recommendation which will be in contradiction with user expectation and real preferences).
- Personalized channel (PCh) – user specific list of programs that are scheduled as play list for personalized preview. When the user pauses a show (e.g. moves out of the room with STB), it can be picked up on other device based on presence.
- Targeted advertising (Ad) – advertising mechanism which is targeted to specified group of users based on their user profiles. Together with user profiling and presence, the operator (or third party advertiser) can identify user interests and target type/form/place of advertising.
- Profiling and personalization (P&P) – feature which enables personalized IPTV services based on user preferences and user profile. The provider can also use the information about the user behavior and content consumption.
- IPTV and NGN Service Interaction (e.g. presence based game, incoming call notification, sharing the remote control).

4.2 Presence concept in NGN integrated IPTV

In the TISPAN NGN integrated IPTV subsystem (formerly NGN dedicated IPTV), any IPTV services may be combined with the presence service capability. Presence is provided via a presence server, which is presented as NGN Application Server Function (NGN ASF) or the NGN ASF could be used as gateway to an external presence server (previously described web service presence for example). More information is available in [7], where the authors contributed details to the particular IPTV procedures towards TISPAN already.

The following specific IPTV attributes may be fetched:

- Service currently accessed/activated.
- Content currently accessed/watched.
- Presence filtering for buddy list members and buddy management.

4.3 Presence concept in NGN IMS based IPTV

TISPAN specified presence [8] via an endorsement from the main presence standard from 3GPP (which in fact uses [9]).

The publication of IPTV specific information in presence documents depends on user-configurable data stored in the UE.

If the presence service is used, the UE shall implement the role of a Presence User Agent (PUA) as specified in TS 283 030. The UE may send a SIP PUBLISH request to update the presence status based on reception of a final SIP 200 OK concerning any service session initiation procedure (after any IPTV service, e.g. VoD, n-PVR, UGC, etc.). Additionally, the UE may also send a PUBLISH request during a BC session after having performed a channel-change (i.e. sending request for a particular BC service) or just after a timer expires send the presence information about really watched channel (e.g. when zapping, immediate change to next channel is not send). Fig. 4 shows the message exchange in detail.
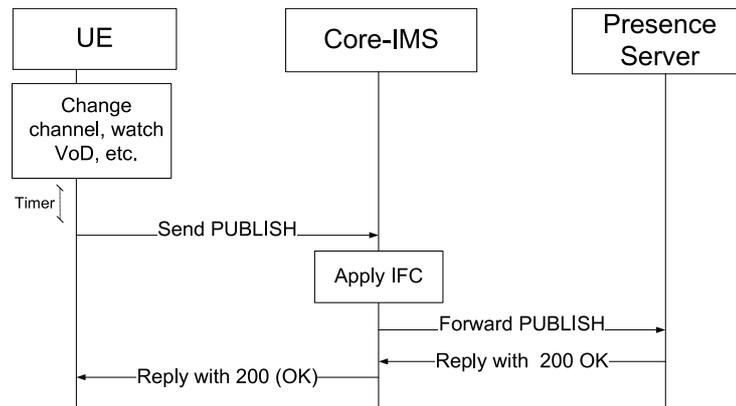

Fig. 4. Example of presence information update after service status change (from UE)

Another more efficient option is the direct interaction of the Service Control Function (SCF) with the presence server to update the presence status for user[4]. This solution could minimize the need for presence signaling traffic and transmit SIP messages across the IMS for each service initiation a second time[5].

---

[4] The SCF serves in fact service initiation requests
[5] Additional to INVITE messages with service initiation

5   Future prospects

The described SIP presence framework and the Parlay X testbed and prototypes were deployed with NGNlab [10]. Between main services (IMS core, SIP AS), we count multimedia communication, collaborative tools, video streaming services, enhanced messaging and also plenty of different web services.

One part of the further work will concentrate on realizing the interaction with more services. Besides extending the platform with measurement tools, the following integration scenarios are planned:

- NGN web based applications (e.g. Presence Web 2.0 integration), web and e-learning portal integration;
- NGN multimedia streaming services (e.g. Web streaming, IPTV applications) will try to communicate with the presence service in described manners.

6   Conclusion

The centralized presence management concept presented in paper can reduce overhead within the provider network and anticipates redundancy. Users or services can use the presence management, and benefit from the information that is already contributed towards the system. Due to different meanings, i.e. mappings, it might not be able to interpret all attributes, but mapped states can be interpreted as well. The proposed concepts have been prototyped and tested in real environment of NGNlab. The integration cross different frameworks (SIP or Web based) enables, aggregates and merges more presence information from multiple sources, platforms or devices. Aggregated and correlated information then should be reused by end user, his buddies or operator service platform to improve user accessibility, communication or entertainment experience. Operator can with user profiling and collecting user presence information as well as service usage behavior improve the user Quality of Experience (QoE) via personalized or more target services provided to subscribers. Possible improvements of IPTV services by mentioned concept have been presented in paper together with existing role of presence in NGN based IPTV frameworks.

References

1. Rosenberg, J. et al., "SIP:Session Initiation Protocol", RFC 3261, 2002
2. Rosenberg, J. "A Presence Event Package for the Session Initiation Protocol (SIP)", RFC 3856, 2004
3. Roach, A. B. "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, 2002
4. ETSI ES 202 391-14 v1.2.1 (2006-12) Open Service Access (OSA); Parlay X Web Services; Part 14: Presence (Parlay X 2)
5. ETSI TR 102 397-14-1 v1.1.1 (2005-08) Open Service Access (OSA); Mapping of Parlay X2 Web Services to Parlay/OSA APIs; Part 14: Presence Mapping; Sub-part 2: Mapping to SIP/IMS Networks
6. ETSI TS 183 063 V2.3.0 (2008-11), TISPAN; IMS based IPTV, Stage 3, 2008
7. ETSI TS 183 064 V2.1.1 (2008-10),TISPAN; Dedicated IPTV subsystem, Stage 3, 2008
8. ETSI TS 182 008 V2.1.1 (2008-10), TISPAN; Presence Service; Architecture and functional description, (Endorsement of 3GPP TS 23.141 and OMA-AD-Presence_SIMPLE-V1_0)
9. OMA Presence SIMPLE, Architecture Document OMA-AD-Presence_SIMPLE-V1_0-20060725-A, Open Mobile Alliance, Jul 2006
10. NGNlab - NGN laboratory at STU in Bratislava, http://www.ngnlab.eu
11. NetLab - Use Cases for Interconnected Testbeds and Living Labs, http://www.celtic-initiative.org/Projects/NETLAB/
12. InCert Next Generation Network Protocols Professionals certification in InCert, International Certificates of Excellence in Selected Areas of ICT, http://incert.eu
13. Train2Cert, Vocational Training for Certification in ICT, http://train2cert.eu