A New SIP Event Package For Group Membership Management in Advanced Communications

Vishal Kumar Singh, Henning Schulzrinne
Department of Computer Science, Columbia University
New York, NY, USA
{vs2140, hgs}@cs.columbia.edu

Piotr Boni
Verizon Communications
Waltham, MA, USA
piotr.boni@verizon.com

Abstract— **Group management is a trendy concept in the Next Generation Network environments. There are many software implementations such as Group List Management Server (GLMS) and in a broader scope: XML Document Management Server (XDMS). These entities are part of the OMA, 3GPP, and IMS specifications. However, in today's dynamic and mobile world, with a multitude of the ad-hoc and ephemeral groupings, there is a need to provide an event-based group membership information for any type of grouping and in a standard manner, so that communications needs of the group could be met. This paper proposes and describes a new SIP Membership Event Package that allows tracking changes in membership in groups. Groups can represent entities contained within a physical space, such as a room or vehicle, or a logical group of entities, such as a call center team. Each member of a group can support a different set of event packages. Tracking membership of entities with different event packages allows deriving presence information from these event packages. IMS, with its SIP foundation and SIMPLE-based Presence as an enabler seems to be the ideal hosting environment for such functionality.**

Keywords: IMS, watcher, presentity, event-package, membership protocol, social networking, vehicle-info.

I. INTRODUCTION

In the recent years, we have witnessed tremendous growth of presence-aware communication applications such as IM and Internet communications both in corporate and consumer world. Also, the social networking applications are using more and more real time updates such as user's current location, current activity of user and user's interests. A user's presence information traditionally included person's availability and willingness for communication. This notion has evolved to take into account user's location and his rich presence information, which includes but is not limited to his current activity, mood and interests. These interests can be varied and can include sports, multimedia files, and books.

Social networks phenomenon in the consumer space and the collaboration needs in the enterprise created motivation for the group management functionality whether this is a web, audio and video conferencing or a multiparty chat. Many types of groups are in use, for example a contact list a.k.a. address book, a buddy list in the IM world, Push To Talk over Cellular (PoC) lists, diverse types of groupings on the social network websites. Different types of OMA XDM specifications, [16], [17] and related XCAP protocol, [11], support group management functions. However, these specifications handle generally static groups like contact list, white lists and black lists and have limited or no support for tracking changes in the group memberships.

With applications providing richer experience than ever before, there are many cases where dynamically tracking groups and their memberships are of interest. In this paper, we propose a new SIP [4] event package called as Membership Event Package [1]. We consider a group used by a Membership Event Package to be a general type of group used for diverse purposes, for example collaboration services, common interests/hobbies, social events, process-defined activities. From such groups and (SIP) event packages supported by group members, we can infer how the communications may be established to the group member. We can also extend the information about a member by getting the data from each SIP event package that the member is associated with.

This event package allows watchers [5] to be notified when group membership changes. In presence-related applications, we encounter groups defined by physical and logical properties. Groups defined by physical properties include all members located in a vehicle, room or building. Groups defined by logical properties include teams in call centers or other groups of personnel where each member can reasonably respond to a request for assistance. For example, the group "sysadmin@example.com" may consist of all on-call system administrators. There can also be groups created and based on objects of interests e.g., users liking a particular genre of music or movies. These are usually based on logical relationship between participating entities and are more common in social networks applications.

Membership event package is also applicable in applications where the entities, which need to be watched are not known in advance, thus the event package provides an additional level of indirection. As an example, consider that an outside user may want to communicate with whoever is occupying a meeting room at the moment. The group of users in a meeting room is an example of group defined by physical property. With the help of the membership event package, the user would subscribe to the membership events for that room and thus obtain presence information for whoever happens to be in the room. Another example is a Call Center supervisor, interested in tracking the status of call center agents without knowing beforehand which agents to subscribe to. He first

subscribes to the group 'active-agents' and then individually subscribes to presence of each agent.

Membership in logical and physical groups can change over time. For example, a meeting room is typically used by multiple different sets of people during the day, while a repair truck may be used by different repair crews.

Currently, in the pre-IMS environments, deployed applications or services are built vertically, often in the form of so-called stove-pipes. Service data, e.g., user's contact list, is often suitable for a specific service only. IMS infrastructure will enforce the horizontal service architecture, where such contact list will be shared by multiple applications. Since the membership package allows aggregation of data across SIP event packages that the members subscribe to, the IMS matches the requirements of our proposed package very well.

The remainder of the paper is organized as follows: Section II explains SIP event architecture and the related work. Section III presents our schema and protocol. Section IV discusses the membership event usage architecture and explains how presence derivation can be achieved. Section V presents the applications, which could benefit from the membership package. Section VI presents security considerations. We conclude the paper in Section VII.

II. Related Work

We briefly explain how SIP based presence system works and then we discuss some existing group management systems.

In a SIP-based presence system, watchers subscribe to presence information of presentities using SIP SUBSCRIBE [5] messages and are notified about the changes in state of the presentities by SIP NOTIFY [5] messages. Diverse sources of presence information update presence information of a presentity to the presence server using the SIP PUBLISH [20] message. Fig. 1 shows a basic block diagram of a presence system.

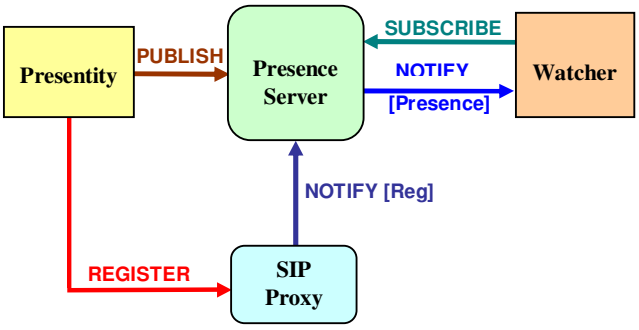

Figure 1 Presence Overview

The work that most closely relates to our work is a set of specifications for centralized conferencing, conference event package by XCON working group in IETF [14][15]. However, the membership and authorization mechanism are tightly coupled with conference application which makes the data structures heavy weight for other purposes. Moreover, the objective of this protocol was interoperability between various conferencing platforms rather than use of protocol for multiple types of group membership and collaborative applications.

There has been work in OMA on XDMS [16] specification, GLMS is used in Push-to-talk for Cellular (PoC), XCAP [11] and other Group Membership Protocol [21]. All of these are used to store and retrieve group membership by the same entity, typically have static groups and do not provide dynamic update support.

Other group membership update and tracking protocols were created at layer 3 for multicast management [18] or peer-to-peer (P2P) group management [19] and not for membership management and tracking, at layer-7, for collaboration services.

III. Membership Event Package Description

In this section, we describe the proposed event package called as membership package that allows tracking changes in membership in groups. We explain the SIP protocol extensions required for membership event package, we also present schema considerations while designing the schema for a group membership tracking protocol and give an example schema. As specified, groups can represent entities contained within a physical space or a logical group of entities. Each member of a group can support a different set of event packages, for example, presence package [6], conference package [14], vehicle-info package [3].

A. *Message Headers*

The membership package is identified by SIP "Event" header and "Allow" event header fields. The NOTIFY request will indicate the body type for the request; it will be "application/membership+xml". An example is given:

```
Event: membership
Content-Type: application/membership+xml
```

B. *Schema Considerations*

The schema proposed is simple and extensible. It can be used not only for updating changes in group memberships using SIP event protocol, but also to query the group memberships using HTTP/XCAP protocols.

The root element is "memberships" and contains the following attributes: entity, version and state. The child element <member> represents the actual members. Each child element has an "id" and "entity" attributes and lists the supported event packages. Since, the membership schema is designed to represent group memberships outside the scope of SIP event and presence framework, the <member> elements may not have any supported package and are just represented by their URI using the "entity" attribute.

The root element and child element <membership> and <member> both have "entity" attribute as both are represented using URI. These URI's can belong to same or different functional domains and can have different association relationships. For example, user (sip:alice@exp1.com can be a member of a room entity (sip:room1@dublin.ietf72.com).

C. *Sample XML for Membership Package Schema*

```
<?xml version="1.0" encoding="utf-8" ?>
 <membership xmlns="urn:ietf:params:xml:ns:membership"
     entity="sip:ur351f@nj.cars.gov"
     state="full"
     version="1" >
   <member id=87553EBFA0D4
        entity="sip:ur351f@nj.cars.gov">
                presence vehicle-info
   </member>
   <member id=8572770D69C3
        entity="sip:alice@example.com">presence
   </member>
   <member id=F547D3C519A0
        entity="sip:bob@example.com"> presence   foo
   </member>
   <member id=FB280604CEAF
         entity="pres:piotr@anotherexample.com"> presence
   </member>
  <member id=0C6247D54256
         entity="sip:carol@example.com">presence foobar
   </member>
 </membership>
```

IV. MEMBERSHIP EVENT USAGE ARCHITECTURE

In this section, we describe the use of Membership event package with the help of a SIP-based message flow. Fig. 2 shows typical message flow in a deployment which uses membership event package.

Membership Event Server (MES) is a SIP-based event server which maintains subscription information and sends notification upon change in state of the subscribed entity and supports membership event package.

In the message flow diagram below, the application subscribes to membership events of the specific group (e.g., a vehicle) on the membership event server (MES). The MES sends a NOTIFY request with an XML body according to membership package schema. The XML body indicates all the current members (entities) and the event packages supported by each entity. The application, such as a watcher or a presence server, may then choose to obtain information on each or some entities contained in the membership list obtained from MES. It will send one or more SUBSCRIBE requests to appropriate event servers handling the specific event package for the entity. In fig. 2, the application sends a SUBSCRIBE request with event package p1 to ES1 and receives a corresponding NOTIFY request. Similarly, the application generates a subscription to ES2 for the p2 event package and receives back notification.

The application may aggregate event information it has obtained in many different ways. Subscriptions with event package p1 and p2 may relate to different entities or to the same entity using different event packages.

An example where a user's information is obtained by subscribing to two different entities is described: a user with association with a room (sip:room1@example.com) and a video-conference (sip:conf@xyz.com) occurring in the room can have information composed by subscribing to information about room and video conference. Subscribing to room using a hypothetical event package "room" gives physical conditions of the room like temperature, number of people in room. Subscribing to video-conference which is a logical group gives information about the conference status.

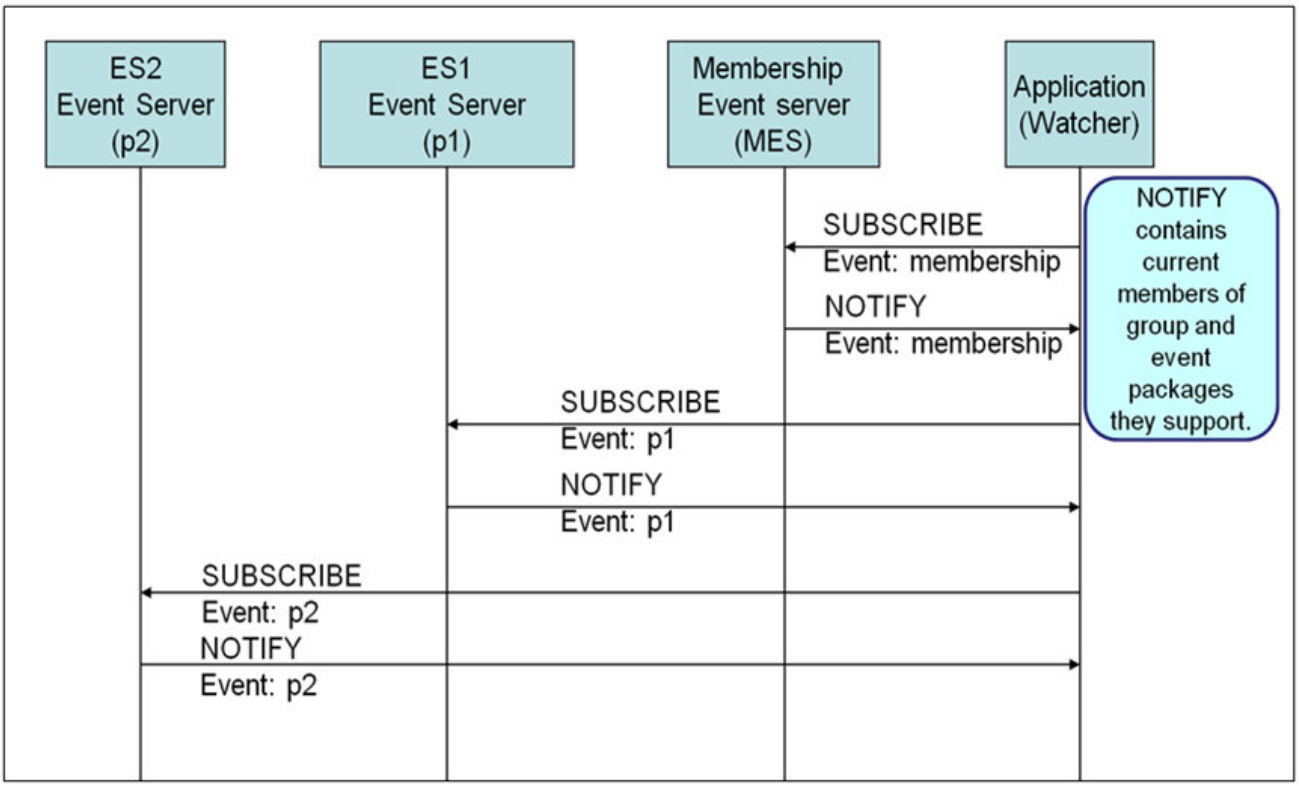

Figure 2 Membership Event Package Usage Example: Message Flow

Another example where a user's information is obtained by subscribing to two different event packages for the same entitys is described: When a user Alice gets into her GPS-equipped car, a sensor in the car discovers her identity, for example, by recognizing the Bluetooth identifier of her cell phone or the code on her smart key. The car electronics then updates the car's membership list, for example by using SIP PUBLISH or XCAP. The process of updating the car's membership list updates the membership XML for the car's entity on the MES. Alternatively, if Alice is authorized to obtain and update car's group membership information, she can obtain the current membership list and publish an updated version with her identity added. In this example, membership in a group representing a vehicle may include the vehicle itself, as well as the driver and passengers. The membership change results in the watcher applications receiving a NOTIFY request with Alice's and the vehicle's entities in the membership list. Let's assume that the application already subscribes to Alice's presentity, but now starts subscription to two vehicle-related event packages, one for the telematics ("vehicle-info" event package) and other for presence information (for the GPS location data ("presence" event package, PIDF-LO [8]).

The application aggregates incoming data across multiple event packages and across multiple entities's to render Alice's extended presence information to authorized users. The above described example elaborates how membership event package is used to derive presence information (presence composition [23]).

In some situations, an application may know the individual entity, but may not know the names of the groups the entity currently belongs to. However, that information can be published as part of the presentity's presence information and then lead the application to other members of her group, such as fellow passengers in a vehicle or fellow team members. This however, overloads presence event package with information, which is not necessarily user's presence or a direct indication of user's willingness to communicate. Hence, it would be better in such circumstances to record the groups that a person belongs to using the membership event package.

Below we give a SIP-based message flow for membership event package using the example we just described.

A. Message Flow Example

SUBSCRIBE sip:ur351f@nj.cars.gov SIP/2.0
Via: SIP/2.0/TCP app.example.com;branch=z9hG4bKnashds7
To: <sip:ur351f@nj.cars.gov>
From: <sip:app.example.com>
Call-ID: 1234@app.example.com
CSeq: 1001 SUBSCRIBE
Max-Forwards: 70
Event: membership
Accept: application/membership+xml
Contact: <sip:app.example.com>
Expires: 86400
Content-Length: 0
……...

Membership event server, which maintains the membership list for the group ur351f@nj.cars.gov, responds with a NOTIFY request.

NOTIFY sip:app.example.com SIP/2.0
Via: SIP/2.0/TCP membership.example.com;branch=
z9hG4bKnashds7
From: <sip: ur351f@nj.cars.gov>
To: <sip:app.example.com>
Call-ID: 1234@app.example.com
Event: membership
Subscription-State: active;expires=6660
Max-Forwards: 70
CSeq: 8775 NOTIFY
Contact: <sip:membership.example.com>
Content-Type: application/membership+xml
Content-Length: ...
…..Membership body….

The NOTIFY request body indicates that the user Alice is in the car with some other passengers. Alice's presentity can be extended by information from vehicle entity. This requires subscription to vehicle-info (vehicle specific information, e.g., telematics) and presence (GPS location) event packages of the ur351f@nj.cars.gov vehicle entity. The application sends SUBSCRIBE requests to a vehicle using event=vehicle-info and event=presence and then processes the obtained information to derive user's presence information.

Using event=vehicle-info:
SUBSCRIBE sip:ur351f@nj.cars.gov SIP/2.0
Via: SIP/2.0/TCP app.example.com;branch=
z9hG4bKnashds7
To: <sip:ur351f@nj.cars.gov>
From: <sip:app.example.com>
Call-ID: 12345@app.example.com
CSeq: 1004 SUBSCRIBE
Max-Forwards: 70
Event: vehicle-info
Accept: application/vehicle-info+xml
Contact: <sip:app.example.com>
Expires: 86400
Content-Length: 0

The vehicle-info event server sends back a NOTIFY request with the vehicle information.

NOTIFY sip:app.example.com SIP/2.0
Via: SIP/2.0/TCP es1.avis.com;branch=z9hG4bKna998sk
From: <sip:ur351f@nj.cars.gov>;tag=ffff
To: <sip:app.example.com>;tag=ght5
Call-ID: 12345@app.example.com
Event: vehicle-info
Subscription-State: active;expires=86660
Max-Forwards: 70
CSeq: 1104 NOTIFY
Contact: sip:es1.avis.com
Content-Type: application/vehicle-info+xml
Content-Length: ...

<vehicle-info body>

Using event=presence:

SUBSCRIBE sip:ur351f@nj.cars.gov SIP/2.0
Via: SIP/2.0/TCP app.example.com;branch=z9hG4bKnashds7
To: <sip:ur351f@nj.cars.gov>
From: <sip:app.example.com>
Call-ID: 123456@example.com
CSeq: 1005 SUBSCRIBE
Max-Forwards: 70
Event: presence
Accept: application/pidf+xml
Contact: <sip:app.example.com>
Expires: 86400
Content-Length: 0

The presence event server sends back a NOTIFY request with vehicle location information.

NOTIFY sip:app.example.com SIP/2.0
Via: SIP/2.0/TCP es2.avis.com;branch=z9hG4bKna998sk
From: <sip:ur351f@nj.cars.gov>;tag=ffff
To: <sip:app.example.com>;tag=ght5
Call-ID: 123456@app.example.com
Event: presence
Subscription-State: active;expires=86660
Max-Forwards: 70
CSeq: 1105 NOTIFY
Contact: sip:es2.avis.com
Content-Type: application/pidf+xml
Content-Length: ...

<PIDF body>

The application (e.g., presence server) may use the information from the last two NOTIFY requests to compose user's presence state and to send expanded PIDF to requesting watchers. For example, the PIDF/RPID status (the activity tag) could be set to 'driving' if the car is moving [22]. The vehicle location information, if present, will be included in user's expanded PIDF.

V. APPLICATIONS OF MEMBERSHIP EVENT PACKAGE

A. *Group Communications*

There is an opportunity to synchronize the membership event package concepts with industry efforts aiming at group communications; for example the efforts of the CTIA in their Enhanced Messaging Initiative and in the 3GPP and other IMS-standards organizations, which are discussing the IMS real-time messaging functionality.

B. *Twitter or Other Apps for Real-Time Messaging*

Twitter [24] in its own words - is a service for friends, family, and co–workers to communicate and stay connected through the exchange of quick, frequent answers about person's activity. Applying the membership package to Twitter, would expand the information on what Twitter-ers are doing and where they are, using presence event packages. The services part of the presentity tuple may indicate the relative current service preference for delivery such as the use of IM over SMS when sending a Twitter update to a specific follower.

C. *Social Networks*

On a social network website, available to users on PC's and mobile handsets, a social event such as a multicast of a rock band concert will bring many people, from the same social group together. John, Monica and Jane decided to opt-in into this social event. They know each other from the social network environment but right now, as they participate in the live concert event, they may get current information about what each one of them is doing. Each of them uses some standards-based but different SIP-presence environment. Consequently, their member entries list the SIP presence event package [6]. That allows a social network (watcher) application to subscribe – with members' consent - to the presence information of John, Monica and Jane and render that information through the social network community. During the music concert they know that John is at home with his laptop and plans to stop watching half an hour before the concert ends. Monica is receiving this concert on her smart mobile phone, while traveling by train. Her location is frequently updated and others can see her position on the map [8]. Monica also engages in some phone conversations and her phone status is reflected using the SIP Event Dialog Package, [12] listed in her membership entry. Jane's may be watching the concert on her TV set or desktop computer; her typical presence is rendered to other two friends, but additionally her status is shown as downloading the rock band music/songs from a music sharing/selling site. This information may come from another proposed SIP event package, namely, file event package [13]. Once the concert is over, the ad-hoc group dissolves and membership information is not needed. The social network acts as an aggregator of the current, dynamic information of its users and facilitates the communications between them

D. *Presence Aware, Location-based Service*

Earlier on, we presented a paper on presence aware, location-based service - PALS, [2]. In the paper, we described a prototype system in which presence information of a user is combined with the location information (derived from a vehicle's location information) to achieve an integrated communication environment. This allows for building advanced domain-specific services, e.g., vehicle status monitoring, automatic communication set up based on triggers, for example, communicating with the user who is typically driving, only when his vehicle's status changes to "Stop" or vehicle ignition status is "Off". These concepts are very conducive to work well with monitoring group memberships in a dynamic way, for example if we want to manage fleet vehicles by groups based on location.

VI. SECURITY

Different kinds of group membership have different level of security requirements. The group membership data is privacy

sensitive information as it can be used to deduce more detailed presence information of the user from different entities or to obtain a list of users participating in common activities such as traveling, meetings and on-call duties. Hence, access to membership lists should be controlled and be unavailable to unauthorized entities.

For example, if a user and a vehicle are member of same group, the user's presence can be derived from vehicle's location. In such a case, it is important that relationship between user and vehicle is not exposed to authorized consumers of location of vehicle. In a more general sense, an entity should be able to see only his memberships and entity's information for a particular event package should be obtainable only to other authorized entity's present in the membership list.

There may be many consumers of information contained in the group membership lists and in the data received from event packages, which group member's support. For example, a vehicle management company may be authorized to obtain the vehicle information using vehicle-info event package. Conversely, a vehicle management server may allow vehicle-info data to be passed to user only if the user is a member of a group representing this vehicle. In the car rental scenario example, apart from car rental company, only the presentity associated with the car is authorized by the car rental company to get vehicle-info data for the car. The same applies to the vehicle location data.

In many cases, other users may get the membership data indirectly. The PA would send presence information based on presentity's privacy preferences [9].

VII. Future Work

We plan to build these some of the services explained in this paper using the proposed framework. Applying privacy filters to data which can be derived from other entity's using same or different event packages is another area of future work. Another area of future work is creation of membership list specifically, who is authorized to create and update the membership list.

VIII. Conclusion

In this paper, we proposed a new event package to be used in conjunction with presence event package, other event packages and SIP event framework to track group membership changes. The proposed schema for group membership is simple and extensible and it can be used in a presence event framework or in a standalone way using web-technologies. Various applications and services are described which can benefit from the membership event package. An example of how membership event package can be used to compose presence of a presentity by deriving information from other event packages for the same, or other entity's with which the presentity is associated, is described.

We also presented a broader perspective on the usage of SIP as the tracking/updating protocol for reporting non-SIP events.

References

[1] Singh, V., Schulzrinne, H., and P. Boni, "Membership Event Package", draft-singh-simple-membership-01 (work in progress), July 2008.

[2] Singh, V., Schulzrinne, H., Boni, P., Elman, B., Kenneson, D., "Presence Aware Location-Based Service For Managing Mobile Communications", CCNC 2007, Las Vegas, January 2007

[3] Singh, V., Schulzrinne, H., and P. Boni, "Vehicle Info Event Package", draft-singh-simple-vehicle-info-01 (work in progress), July 2007.

[4] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[5] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[6] Rosenberg, J., "A Presence Event Package for the Session Initiation Protocol (SIP)", RFC 3856, August 2004.

[7] Sugano, H., Fujimoto, S., Klyne, G., Bateman, A., Carr, W., and J. Peterson, "Presence Information Data Format (PIDF)", RFC 3863, August 2004.

[8] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, December 2005.

[9] Schulzrinne, H., Tschofenig, H., Morris, J., Cuellar, J., Polk, J., and J. Rosenberg, "Common Policy: A Document Format for Expressing Privacy Preferences", RFC 4745, February 2007.

[10] Schulzrinne, H., Tschofenig, H., Morris, J., Cuellar, J., and J. Polk, "Geolocation Policy: A Document Format for Expressing Privacy Preferences for Location Information", draft-ietf-geopriv-policy-17 (work in progress), June 2008.

[11] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", RFC 4825, May 2007

[12] Rosenberg, J., Schulzrinne, H., and Mahy, R., "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", RFC 4235, November 2005.

[13] Garcia-Martin, M., and Matuszewski, M., "A Session Initiation Protocol (SIP) Event Package and Data Format for Describing Files", draft-garcia-sipping-file-event-package-01 (work-in-progress), November 2007.

[14] Barnes, M., Boulton, C., and Levin, O., "A Framework for Centralized Conferencing", RFC 5239 June 2008.

[15] Camarillo, G., Srinivasan, S., Even, R., and J. Urpalainen, "Conference Event Package Data Format Extension for Centralized Conferencing (XCON)", draft-ietf-xcon-event-package-00 (work in progress), Aug 2008.

[16] XML Document Management Architecture, OMA-AD-XDM-V1_1-20080128-C.pdf

[17] XML Document Management Specification, OMA-TS-XDM_Core-V1_1-20080128-C.pdf

[18] Cain, B., Deering, S., Fenner, B., Kouvelas, I., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[19] Ganesh, A.J., Kermarrec, A.-M., and Massoulie, L., "Peer-to-peer membership management for gossip-based protocols", IEEE Transactions on Computers, Feb. 2003

[20] A. Niemi, "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, Oct 2004

[21] http://wiki.openid.net/Group_Membership_Protocol

[22] Schulzrinne H., Gurbani V.,Kyzivat P. and Rosenberg J., "RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)", RFC 4480, July 2006.

[23] Schulzrinne, H., Shacham, R., Kellerer, W., and Thakolsri, S.,"draft-schulzrinne-simple-composition-02.txt", (work-in-progress), June 2006.

[24] http://twitter.com/