Presence SIMPLE Specification

Historic Version 1.0.1 – 28 Nov 2006

Open Mobile Alliance

OMA-TS-Presence_SIMPLE-V1_0_1-20061128-H

Contents

Figures

1. Scope

This document provides the specifications for the Presence Service enabler based on the IETF SIMPLE (SIP Instant Messaging and Presence Leveraging Extensions) technology. This enabler is specified such that it is available to be used by other service enablers.

This release of the specification utilizes a SIP/IP core based on the 3GPP IMS and 3GPP2 MMD network capabilities.

2. References

2.1 Normative References

OMA

[PRESREQ] "Presence Requirements", Version 1.0, Open Mobile Alliance™, OMA-RD-Presence_SIMPLE-V1_0, URL:http://www.openmobilealliance.org/

[PRESAD] "Presence using SIMPLE", Version 1.0.1, Open Mobile Alliance™, OMA-AD-Presence_ SIMPLE-V1_0_1, URL:http://www.openmobilealliance.org/

[PRESAC] "Presence Application Characteristics file of Presence V1.0.1", Version 1.0, Open Mobile Alliance™, OMA-SUP-AC_ap0002_presence-V1_0_1, URL:http://www.openmobilealliance.org/

[PRESMO] "OMA Management Object for SIMPLE Presence", Version 1.0.1, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE_MO-V1_0_1, URL:http://www.openmobilealliance.org/

[XDMSPEC] "XML Document Management Specification", Version 1.0.1, Open Mobile Alliance™, OMA-TS-XDM_Core-V1_0_1, URL:http://www.openmobilealliance.org/

[PRESXDM] "Presence XDM Specification", Version 1.0.1, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE_XDM- V1_0_1, URL:http://www.openmobilealliance.org/

[SharedXDM] "Shared XDM Specification", Version 1.0.1, Open Mobile Alliance™, OMA-TS-XDM_Shared-V1_0_1, URL:http://www.openmobilealliance.org/

[RLSXDM] "RLS XDM Specification", Version 1.0.1, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE_RLS_XDM- V1_0_1, URL:http://www.openmobilealliance.org/

[Provisioning Content] OMA – Provisioning Content V1.1, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[OMA–DM-v1-1-2] OMA Device Management, V1.1.2, Open Mobile Alliance™, (based on SyncML DM), OMA-DM-V1_1_2. URL:http://www.openmobilealliance.org/

[OMA-DM-v1-2] OMA Device Management, V1.2, Open Mobile Alliance™, (based on SyncML DM), OMA-DM-V1_2, URL:http://www.openmobilealliance.org/

IETF

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, (:http://www.ietf.org/rfc/rfc2119.txt)

[RFC2234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. November 1997, (http://www.ietf.org/rfc/rfc2234.txt)

[RFC2246] "The TLS Protocol Version 1.0", T. Dierks et al., January 1999, RFC 2246, (http://www.ietf.org/rfc/rfc2246.txt)

[RFC2387] "The MIME Multipart/Related Content-type", E. Levinson, Aug. 1998, RFC 2387, (http://www.ietf.org/rfc/rfc2387.txt)

[RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", T.Berners-Lee et al., Aug. 1998, RFC 2396, (http://www.ietf.org/rfc/rfc2396.txt)

[RFC2616] "Hypertext Transfer Protocol -- HTTP/1.1", URL:http://www.ietf.org/rfc/rfc2616.txt

[RFC2778] "A Model for Presence and Instant Messaging", M. Day et al., Feb. 2000, RFC 2778, (http://www.ietf.org/rfc/rfc2778.txt)

[RFC2779] "Instant Messaging / Presence Protocol Requirements ", M.Day et al., Feb 2000, RFC 2779, (http://www.ietf.org/rfc/rfc2779.txt)

[RFC2818] "HTTP Over TLS", E. Rescorla, May 2000, RFC 2818, (http://www.ietf.org/rfc/rfc2818.txt)

[RFC3261] "Session Initiaton Protocol (SIP)", Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler,June 2002, RFC 3261,

(http://www.ietf.org/rfc/rfc3261.txt)

[RFC3265] "Session Initiation Protocol (SIP)-Specific Event Notification", A.B.Roach, June 2002, RFC 3265, (http://www.ietf.org/rfc/rfc3265.txt)

[RFC3323] "A Privacy Mechanism for the Session Initiation Protocol (SIP)", Peterson, J., Nov. 2002, RFC 3323, (http://www.ietf.org/rfc/rfc3323.txt)

[RFC3325] "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", Jennings, C., et al, Nov. 2002, RFC 3325, (http://www.ietf.org/rfc/rfc3325.txt)

[RFC3320] "Signaling Compression (SigComp)", Price, R., et al., Jan. 2003, RFC 3320, (http://www.ietf.org/rfc/rfc3320.txt)

[RFC3485] "The Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Static Dictionary for Signaling Compression (SigComp)", Garcia-Martin, M., et al.,Feb. 2003, RFC 3485, (http://www.ietf.org/rfc/rfc3485.txt)

[RFC3486] "Compressing the Session Initiation Protocol (SIP)", Camarillo, G., Feb. 2003, RFC 3486, (http://www.ietf.org/rfc/rfc3486.txt)

[RFC3856] "A Presence Event Package for the Session Initiation Protocol (SIP)", J.Rosenberg, Jan. 2003, RFC 3856, (http://www.ietf.org/rfc/rfc3856.txt)

[RFC3857] "A watcher Information Event Template-Package for the Session Initiation Protocol (SIP)", J.Rosenberg, Aug. 2004, RFC 3857, (http://www.ietf.org/rfc/rfc3857.txt)

[RFC3858] "An Extensible Markup Language (XML) Based Format for Watcher Information", J.Rosenberg, Aug. 2004, RFC 3858, (http://www.ietf.org/rfc/rfc3858.txt)

[RFC3859] "Common Profile for Presence (CPP)", J.Peterson, Aug. 2004, RFC 3859, (http://www.ietf.org/rfc/rfc3859.txt)

[RFC3863] "Presence Information Data Format (PIDF)", H.Sugano et al., Aug 2004 (http://www.ietf.org/rfc/rfc3863.txt)

[RFC3903] "An Event State Publication Extension to the Session Initiation Protocol (SIP) ", A. Niemi, Oct. 2004, (http://www.ietf.org/rfc/rfc3903.txt)

[RFC4119] "Presence-based GEOPRIV Location Object Format", J. Peterson, Dec. 2005, (http://www.ietf.org/rfc/rfc4119.txt)

[RFC4122] "A Universally Unique IDentifier (UUID) URN Namespace", P.Leach et al., July 2005, (http://www.ietf.org/rfc/rfc4122.txt)

[RFC4479] "A Data Model for Presence", J. Rosenberg, Jul 2006 (http://www.ietf.org/rf/rfc4479.txt)

[RFC4480] "RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)", H. Schulzrinne et al., July 2006, (http://www.ietf.org/rfc/rfc4480.txt)

[RFC4483] "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages", E. Burger, Ed, May 2006 ,URL: http://www.ietf.org/rfc/rfc4483.txt

[RFC 4589] "Location Types Registry", H. Schulzrinne, July 2006, (http://www.ietf.org/rfc/rfc4589.txt)

[RFC4660] "Functional Description of Event Notification Filtering", H.Khartabil et al, Sep 2006 (http://www.ietf.org/rfc/rfc4660.txt)

[RFC4661] "An Extensible Markup Language (XML) Based Format for Event Notification Filtering", H.Khartabil et al, Sep 2006, (http://www.ietf.org/rfc/rfc4661.txt)

[RFC4662] "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists", A. B. Roach et al., August 2006, (http://www.ietf.org/rfc/rfc4662.txt)

[PARNOT] Session Initiation Protocol (SIP) extension for Partial Notification of Presence Information", M.Lonnfors et al., July 6. 2006, (http://www.ietf.org/internet-drafts/draft-ietf-simple-partial-notify-08.txt)

Note: IETF Draft work in progress

[PARFORMAT] "Presence Information Data format (PIDF) Extension for Partial Presence", M. Lonnfors et al., July

21, 2006, (http://www.ietf.org/internet-drafts/draft-ietf-simple-partial-pidf-format-07.txt)

Note: IETF Draft work in progress

[PARPUBLISH] " Publication of Partial Presence Information", M.LonnforsA. Niemi et al., July 20, 2006, (http://www.ietf.org/internet-drafts/draft-ietf-simple-partial-publish-05.txt)

Note: IETF Draft work in progress

[PRESRULES] "Presence Authorization Rules", J. Rosenberg, October 22, 2006, (http://www.ietf.org/internet-drafts/draft-ietf-simple-presence-rules-08.txt)

Note: IETF Draft work in progress

[XCAP_Diff] "An Extensible Markup Language (XML) Document Format for Indicating Changes in XML Configuration Access Protocol (XCAP) Resources", J. Rosenberg, October 17, 2006, (http://www.ietf.org/internet-drafts/draft-ietf-simple-xcap-diff-04.txt)

Note: IETF Draft work in progress

[XSD_PRSPIDF] "OMA-defined PIDF extensions", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_prs_pidf_omapres -V1_0, URL: http://www.openmobilealliance.org/

3GPP/3GPP2

[3GPP TS 23.228] "IP Multimedia Subsystem (IMS); Stage 2", 3GPP TS 23.228, Release 6, 2005

[3GPP2 X.S0013-002-A] "All-IP Core Network Multimedia Domain: IP Multimedia Subsystem - Stage 2", Revision A, Version 1.0, 3GPP2, 2005

[3GPP TS 24.229] "Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3", 3GPP TS 24.229, Release 6, 2005

[3GPP2 X.S0013-004-A] "All-IP Core Network Multimedia Domain: IP Multimedia Call Control Protocol Based on SIP and SDP Stage 3", Revision A, Version 1.0, 3GPP2, 2005

[3GPP TS 24.109] "Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details ; Stage 3", 3GPP TS 24.109, Release 6

[3GPP TS 33.203] "Access Security for IP-based services", 3GPP TS 33.203, Release 6, 2005

[3GPP2 S.R0086-A] "IMS Security Framework", Revision A, Version 1.0, 3GPP2, 2004

[3GPP TS 32.240] "Charging management; Charging architecture and principles",3GPP TS 32.240, Release 6, 2005

[3GPP2 X.S0013-007-0] "All-IP Core Network Multimedia Domain: IP Multimedia Subsystem - Charging Architecture", Revision A, Version 1.0, 3GPP2, 2005

[3GPP TS 22.228] "Service requirements for the Internet Protocol (IP) multimedia core network subsystem; Stage 1", 3GPP TS 22.228, Release 6, 2005

[3GPP TS 24.141] "Presence service using the IP Multimedia (IM) Core Network (CN) subsystem; Stage-3", 3GPP TR 24.141, Release 6, 2005

[3GPP2 X.P0027-003-0] "Presence Service using IP Multimedia Core Network Subsystem; Stage 3", Revision 0, Version 1.0, 3GPP2, 2005

Note: Work in progress, estimated availability January 2006.

[3GPP TS 32.260] "Charging Management; IP Multimedia Subsystem (IMS) Charging", 3GPP TS 32.260, Release 6, 2005

[3GPP2 X.S0013-008-A] "All-IP Core Network Multimedia Domain: IP Multimedia Subsystem – Offline Accounting, Information Flows and Protocol", Revision A, Version 1.0, 3GPP2, 2005

[3GPP TS 33.141] "Presence Service; Security", 3GPP TS 33.141, Release 6, 2004

[3GPP2 X.P0027-002-0] "Presence Security", Revision 0, Version 1.0, 3GPP2, 2005

Note: Work in progress, estimated availability January 2006.

[3GPP TS 26.141] "IP Multimedia System (IMS) Messaging and Presence; Media formats and codecs (Release 6)", 3GPP, 2005

[3GPP2 C.P0071-0] "IP Multimedia Domain(MMD) Codecs and Transport Protocols", Revision 0, Version 1.0, 3GPP2, 2005

2.2 Informative References

[RFC4474] "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", J.Peterson et al., August 2006, (http://www.ietf.org/rfc/rfc4474.txt)

[3GPP TS 22.141] "Presence Service; Stage 1", 3GPP TS 22.141 Release 6, 2005

[3GPP2 S.R0062-0] "Presence for Wireless Systems Stage 1 Requirements", Revision 0, Version 1.0, 2002

[3GPP TS 23.141] "Presence Service; Architecture and functional description", 3GPP TS 23.141, Release 6, 2005

[3GPP2 X.S0027-001-0] "Presence Service; Architecture and functional description", Revision 0, Version 1.0, 3GPP2, 2004

[3GPP2 X.P0027-004-0] "Network Presence", Revision 0, Version 1.0, 3GPP2, 2005

Note: Work in progress, estimated availability January 2006.

[3GPP TS 23.218] "IP Multimedia (IM) session handling; IM call model; Stage 2", 3GPP TS 23.218 Release 6, 2005

[3GPP2 X.S0013-003-A] "All-IP Core Network Multimedia Domain: IP Multimedia (IMS) session handling; IP Multimedia (IM) call model; Stage 2", Revision A, Version 1.0, 2005

[3GPP TS 29.228] "IP Multimedia (IM) Subsystem Cx and Dx Interfaces; Signalling flows and message contents", 3GPP TS 29.228 Release 6 2005

[3GPP2 X.S0013-006-A] "All-IP Core Network Multimedia Domain: Cx Interface based on the Diameter Protocol; Protocol Details", Revision A, 2005

[OMNA] Open Mobile Naming Authority, Open Mobile Alliance™
URL: http://www.openmobilealliance.org/tech/omna/

[OMNA_pidfSvcDesc] Open Mobile Naming Authority Presence <service-description> Registry, Open Mobile Alliance™,
URL: http://www.openmobilealliance.org/tech/omna/OMNA-prs-pidfSvcDesc-registry.htm

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Composition The function of the PS to combine the “views” of the various Presence Sources in one single raw presence document for a particular presentity.

Content Server The Content Server is the functional entity that is capable of managing MIME objects for Presence, allowing the Presence Sources to store MIME objects within, and support retrieval of those objects by watchers.

Source: [PRESAD]

Event Package Event Package: An event package is an additional specification, which defines a set of state information to be reported by a notifier to a subscriber. Event packages also define further syntax and semantics based on the framework defined by this document required to convey such state information.

Source: [RFC3265]

Event Publication Agent (EPA) The User Agent Client (UAC) that issues PUBLISH requests to publish event state.

Source: [RFC3903]

Event State Compositor (ESC) The User Agent Server (UAS) that processes PUBLISH requests, and is responsible for compositing event state into a complete, composite event state of a resource.

Source: [RFC3903]

Presence Content Rules Rules that determine the content of Presence information sent to the watchers.

Presence Information Dynamic set of information pertaining to a Presentity that may include Presence Information Elements such as the status, reachability, willingness, and capabilities of that Presentity.

Note: This definition is compatible with the 3GPP/3GPP2 definitions, as well as the IETF definition, though the latter is quite generic.

Source: [PRESREQ]

Presence Information Element A basic unit of Presence Information.

Source: [PRESREQ]

Presence External Agent (PEA) Presence source element that is located outside of the provider's network.

Source: [3GPP TS 23.141]/ [3GPP2 X.S0027-003-0]

Presence Network Agent (PNA) Network located element that collects and sends network related presence information on behalf of the presentity to a presence server

Source: [3GPP TS 24.141]/ [3GPP2 X.S0027-003-0]

Presence Source	A logical entity that provides *Presence Information* pertaining to exactly one or more *Presentities* to the *Presence Server*. Presence User Agents, Presence Network Agents, and Presence External Agents are examples of *Presence Sources*. Note: In [RFC3856], Presence Sources are referred to as Presence User Agents. In [RFC2778], they are referred to as Presentities. Source: [PRESREQ]
Presence User Agent (PUA)	A terminal or network located element that collects and sends user related presence information to a presence server on behalf of a Principal Source: [3GPP TS 24.141]/ [3GPP2 X.S0027-003-0]
Presentity	A logical entity that has *Presence Information* (see definition below) associated with it. This *Presence Information* may be composed from a multitude of *Presence Sources*. A *Presentity* is most commonly a reference for a person, although it may represent a role such as "help desk" or a resource such as "conference room #27". The presentity is identified by a SIP URI (as defined in [RFC3261]), and may additionally be identified by a pres URI (as defined in [RFC3859]). Note: This definition maps better to the [RFC2778] definition of a Principal, rather than that of [RFC2778] Presentity. This definition is compatible with the[RFC3856].
Resource List Server (RLS)	A functional entity that accepts and manages subscriptions to presence lists, which enables a Watcher application to subscribe to the presence information of multiple presentities using a single subscription transaction. Source: [PRESAD]
Subscription Authorisation Rules	Rules that determine the handling of an incoming Presence Subscription by the PS
User Equipment (UE)	A device allowing a user access to network services. For the purpose of 3GPP specifications the interface between the UE and the network is the radio interface. A User Equipment can be subdivided into a number of domains, the domains being separated by reference points. Currently defined domains are the USIM and ME Domains. The ME Domain can further be subdivided into several components showing the connectivity between multiple functional groups. These groups can be implemented in one or more hardware devices. An example of such a connectivity is the TE – MT interface. Further, an occurrence of a User Equipment is an MS for GSM as defined in GSM TS 04.02. Source: [3GPP TR 21.905]
Watcher	Any uniquely identifiable entity that requests presence information about a presentity, from the presence service. Special types of watcher are fetcher, poller, and subscribed-watcher. (Differs slightly from [RFC2778] and [3GPP2 X.S0027-003-0] definitions). Source: [PRESREQ]
Watcher information subscriber	Any uniquely identifiable entity that requests watcher information about a watcher, from the presence service.

3.3 Abbreviations

3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AD	Architecture Document
AS	Application Server
CID	Content ID
DM	Device Management
EPA	Event Publication Agent
ESC	Event State Compositor
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extensions
MWG	Messaging Workin Group
MWS	Mobile Web services
OMA	Open Mobile Alliance
OMNA	Open Mobile Naming Authority
OTAP	Over the Air Provisioning
PIDF	Presence Information Data Format
PoC	Push-to-talk over Cellular
PEA	Presence External Agent
PUA	Presence User Agent
PNA	Presence Network Agent
PS	Presence Server
RD	Requireemnt Document
RFC	Request For Comments
RPID	Rich Presence Information Data
RLS	Resource List Server
SIMPLE	SIP Instant Message and Presence Leveraging Extensions
SIP	Session Initiaion Protocol
TLS	Transport Layer Security
UA	User Agent
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
WLAN	Wireless LAN
WG	Working Group
XCAP	XML Configuration Access Protocol

XDMS	XML Document Manipulation Server
XML	Extensible Markup Language
XUI	XCAP User Identifier

4. Introduction

The document defines an application level specification for the OMA SIP/SIMPLE-based Presence Service. It defines the presence information semantics for presence information conveyed using the Presence Information Data Format (PIDF) the Rich Presence Information Data Format (RPID) and geographical information conveyed in a GEOPRIV location object (see [RFC4119]) specified by the IETF in conjunction with the overall Presence Data Model defined in [RFC4479].

This specification makes use of the implementations of the SIP protocol in the 3GPP IMS (IP Multimedia Subsystem) and 3GPP2 MMD (Multimedia Domain) for collecting and disseminating presence information between the various Presence Sources and their watchers as described in the Presence architecture document ([PRESAD]).

In addition to the SIP methods for subscription, publication, and notification of presence state based on [RFC3265], [RFC3856] and [RFC3903], this specification also addresses:

- The content of presence information, based on [RFC3863], [RFC4119], [RFC4479], [RFC4480] etc.
- The partial publication of (only the changed) presence information, based on [PARFORMAT]
- Triggers for the generation of notifications when specific events take place
- The handling of large presence information content, based on support of [RFC 2387] and [RFC4483]
- The control of the content of the notification sent to a watcher, based on [RFC4660] and [RFC4661]
- Back-end subscriptions to a presence list, based on [RFC4662]
- Subscription authorization rules for watchers, based on [PRESXDM], and
- Presence content rules for watchers, based on [PRESXDM].

The Presence Services makes use of various data repositories in the network that store information related to presentities and watchers, specifically:

- The Presence XDMS (see [PRESXDM]) for storage of documents related to a presentity, such as subscription authorization rules and presence content rules for watchers;
- The Shared XDMS (see [SharedXDM]) for URL Lists which may be referenced from other documents;
- The RLS XDMS (see [RLSXDM]) for storing a watcher's presence list; and
- The Content Server (see [PRESAD]) for managing MIME objects.

5. Presence Functional Entities

5.1 Presence Source

The Presence Source is an entity that provides presence information to a Presence Service. The Presence Source MAY be located in the user's terminal or within a network entity.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the Presence Source MAY be implemented in a UE or an AS as defined in [3GPP TS 23.228] and [3GPP2 X.S0013-002-A] respectively.

5.1.1 Publication of presence information

A Presence Source SHALL implement the Event Publication Agent (EPA) function and support the PUBLISH method according to the procedures described in [RFC3903].

A Presence Source SHALL support the 'application/pidf+xml' content type, according to [RFC3863].

The presentity is identified by a SIP URI (as defined in [RFC3261]), and may additionally be identified by a pres URI (as defined in [RFC3859]). If the presentity is identified by both a pres URI and a SIP URI and the Presence Source is aware of both of them, the Presence Source SHOULD insert the SIP URI in the Request-URI of the PUBLISH request. The Presence Source SHALL insert the same URI in the "entity" attribute of the <presence> element as in the Request-URI of the PUBLISH request.

A Presence Source SHALL use the elements listed in section 10.4 when it has to publish presence information with semantics identical to those elements.

When the SIP/IP Core network corresponds to 3GPP IMS or 3GPP2 MMD networks, and the Presence Source is a UE, it SHALL set the entity attribute of the <presence> element of the Presence Information document, defined in [RFC3863], to its registered public user identity, as defined in [3GPP TS 24.229] and [3GPP2 X.S0013-004-A]. If more than one registered public user identity is available,

- the Presence Source SHALL set the value of the "entity" attribute of the <presence> element in the Presence Information document with the value of the P-Preferred-Identity header field used in the SIP PUBLISH request, if present, as defined in [3GPP TS 24.229] and [3GPP2 X.S0013-004-A].
- If there is no P-Preferred-Identity header field included in the SIP PUBLISH request, the Presence Source SHALL include its default public user identity, as defined in [3GPP TS 24.229] and [3GPP2 X.S0013-004-A], in the "entity" attribute of the <presence> element of the Presence Information document,

When the SIP/IP Core network corresponds to 3GPP IMS or 3GPP2 MMD networks and if the Presence Source is an AS, it SHALL set the value of the "entity" attribute of the <presence> element in the Presence Information document with the value of the P-Asserted-Identity header field used in the SIP PUBLISH request as defined in [3GPP TS 24.229] and [3GPP2 X.S0013-004-A].

The Presence Source MAY support other PIDF extensions to publish elements whose semantics do not match with those defined in section 10.4, as long as, if a Watcher that does not understand those extensions can ignore them without changing the meaning of the presence elements that are understood.

The Presence Source SHALL be free to provide any value of the instance identifier attributes (id) for <tuple>, <person> and <device> (as defined in [RFC4479]) as this is being used only to syntactically differentiate between the elements and is not linked with any composition actions in the PS or resolution of conflicts in watcher.

For a given presentity, the information published by each presence source is composed into a single raw presence document as described in Section 5.4.3.1.

5.1.1.1 Partial publication

Partial publication is a mechanism such that a given presence source can publish only those parts of the presence information that have changed since its last publication, rather than the full presence state.

A Presence Source MAY support partial publication. A Presence Source performing partial publication SHALL support the following:

- Partial publication procedure, according to [PARPUBLISH], and
- Partial presence extension to PIDF, according to [PARFORMAT].

5.1.1.2 Handling of large objects

The Presence Source MAY implement the 'multipart/related' content type as described in [RFC2387], in order to aggregate other MIME objects with the 'application/pidf-diff+xml' content type.

If a presence attribute has a value of a refernce to a MIME object, the Presence Source can either:

- Use the content indirection mechanism as defined in [RFC4483] and upload the content to the Content Server; or
- Send the MIME object directly together with the presence document by utilising the 'multipart/related' content-type in the PUBLISH request.

The MIME object format SHALL conform to [3GPP TS 26.141] and [3GPP2 C.P0071-0].

5.1.1.2.1 Performing content indirection

If the Presence Source decides to use the content indirection mechanism for publishing an initial or modified value of a presence attribute, the Presence Source SHALL follow the following procedures:

1. Store the MIME object.

   NOTE: The procedure for storing MIME objects is not defined by this specification.

   The Presence Source MAY be provisioned with the HTTP or optionally HTTPS URI of the content server where the MIME objects will be stored. This can be done with OTA Provisioning or local configuration. In case it is performed with OTA Provisioning it SHALL use the value of the CONTENT-SERVER-URI defined in Appendix B.1.

2. Construct an HTTP URI or optionally an HTTPS URI referencing the stored MIME object.
3. Use the 'multipart/related' content type as described in [RFC2387] with the content indirection mechanism as specified in [RFC4483] for the publication of presence information format as follows:

   a) Set a CID URI referencing to other MIME multipart body which contains the content indirection information as the value of the XML element whose value is delivered as an indirect content;

   b) Include the presence document of the format 'application/pidf+xml' or 'application/pidf- diff+xml' in the root of the body of the 'multipart/related' content;

   c) Specify the part having information about the MIME object by using the 'message/external-body' content type, defining the HTTP or HTTPS URI, versioning information and other information about the MIME object as described in [RFC4483]. The versioning information is used for determining whether or not the MIME object indirectly referenced by a URI has changed or not.

5.1.1.2.2 Handling of direct content

When the Presence Source decides to publish the MIME object as a direct content inside the presence document, the Presence Source SHALL utilise the 'multipart/related' content type as described in [RFC2387] in the PUBLISH request with the following procedures:

- Set a CID URI referencing to other multipart body which contains the MIME object;
- Include the presence document of the format 'application/pidf+xml' or 'application/pidf- diff+xml' in the root of the body of the 'multipart/related' content.

If the Presence Source supports OTA Provisioning, the size limit for MIME data direct content in a PUBLISH request as set via OTA Provisioning SHALL NOT be exceeded.

In case it is performed with OTA Provisioning, it SHALL use the value of CLIENT-OBJ-DATA-LIMIT parameter is defined in Appendix B.1.

If the Presence Source does not support OTA Provisioning, the size limit for MIME data direct content in a PUBLISH request SHOULD be set by other means at the Presence Source and its value SHALL be the same as defined for OTA Provisioning compliant Presence Sources.

5.1.1.3 Limiting the rate of publications

The Presence Source MAY be configured with a rate (the shortest time period between two PUBLISH requests) at which PUBLISH requests are generated. This can be done with OTA Provisioning or local configuration. In case it is performed with OTA Provisioning it SHALL use the value of SOURCE-THROTTLE-PUBLISH defined in Appendix B.1.

In this case, the Presence Source SHALL NOT generate PUBLISH requests more often as it is instructed by the local rate limitation configuration.

5.1.2 Example realizations of a Presence Source (Informative)

5.1.2.1 Presence User Agent

The Presence Source MAY be implemented as a Presence User Agent (PUA) as defined by 3GPP/3GPP2 in [3GPP TS 23.141] and [3GPP2 X.S0027-001-0] respectively. The PUA is a Presence Source realization residing in the terminal or network. The PUA collects user related presence information from its corresponding presentity and sends it to the PS.

5.1.2.2 Presence Network Agent

The Presence Source MAY be implemented as a Presence Network Agent (PNA) as defined by 3GPP/3GPP2 in [3GPP TS 23.141] and [3GPP2 X.S0027-001-0] respectively. The PNA collects the network related presence information from the various network elements and send it to the PS.

The PNA may also notify the PS when the terminal is disconnected. The interfaces between the PNA and the various elements are defined in 3GPP/3GPP2 (see Figure 1and Figure 2) and are out of scope of the current specification.

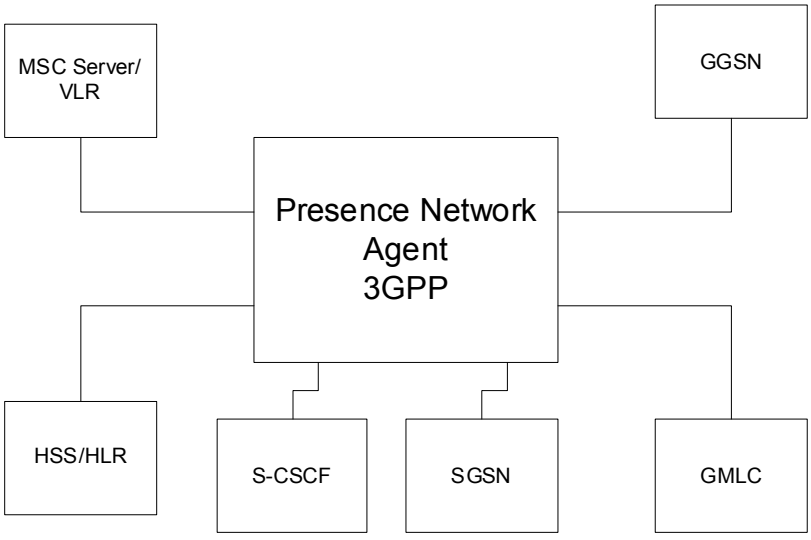

Figure 1-PNA in 3GPP

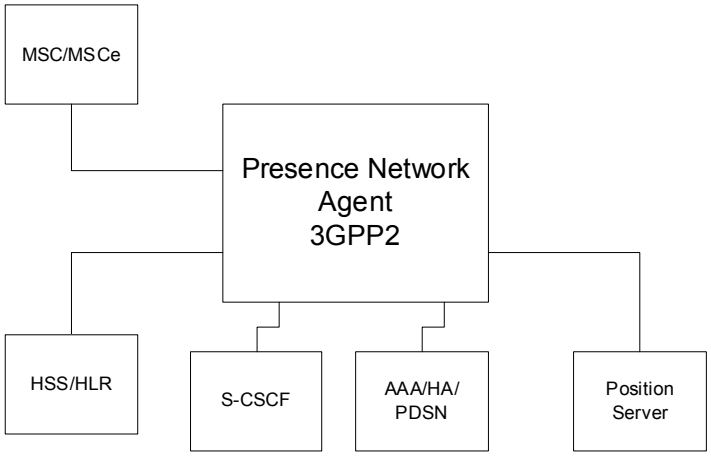

Figure 2-PNA in 3GPP2

The options of using a PNA in a non-3GPP/3GPP2 environment is shown on Figure 3:

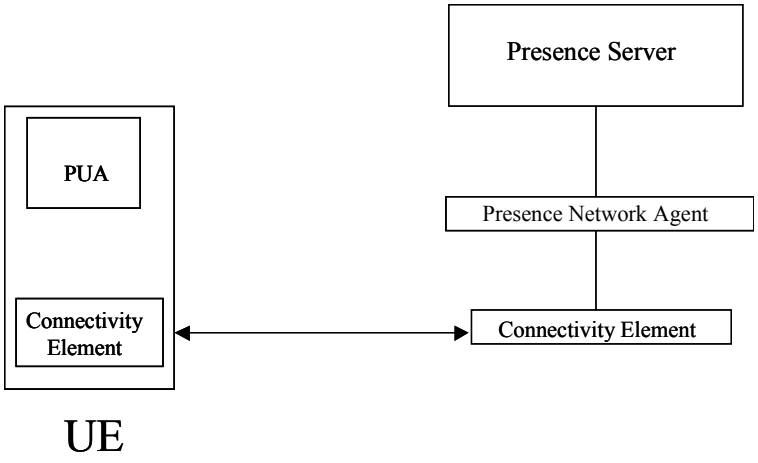

Figure 3-PNA in a non-3GPP/3GPP2 architecture. Presence information can be aggregated either directly to the PS or via a PNA.

5.1.2.3 Presence External Agent

The Presence Source MAY be implemented as a Presence External Agent (PEA) as defined by 3GPP/3GPP2 in [3GPP TS 23.141] and [3GPP2 X.S0027-001-0] respectively. The PEA performs the following functions:

- Supply presence information from external networks.
- Handle the interworking and security issues involved in interfacing to external networks.
- Resolve the location of the PS associated with the presentity.

Examples of presence information that the PEA may supply, include:

- Third party services (e.g. calendar applications, corporate systems)

- Internet Presence Services
- Non SIMPLE-based Presence Services
- Services that use Presence (e.g. PoC, IM).

5.2 Watcher

The watcher is an entity that subscribes to presence information about a presentity or list of presentities (i.e. presence list).

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the watcher MAY be implemented in a UE or an AS as defined in [3GPP TS 23.228] and [3GPP2 X.S0013-002-A] respectively.

5.2.1 General

A watcher SHALL support the 'application/pidf+xml' content type, according to [RFC3863].

5.2.2 Subscription to presence information

A watcher SHALL support subscription and notification of presence information, according to the subscriber procedures described in [RFC3265] and [RFC3856].

If the watcher knows both the pres URI (as defined in [RFC3859]) and the SIP URI (as defined in [RFC3261]) of the presentity, it SHOULD use the SIP URI in the Request-URI of the SUBSCRIBE request.

If the watcher only knows the pres URI of the presentity, the pres URI may get translated to a SIP URI by the SIP/IP core network. In this case, the watcher MAY learn the translated URI from the "entity" attribute of the <presence> element included in the NOTIFY request and use it for future subscriptions.

5.2.2.1 Limiting the number of subscriptions

The service provider MAY configure a watcher with a total maximum number of subscriptions to the presence event package. This can be done with OTA Provisioning or local configuration. In case of OTA Provisioning, the watcher SHALL use the value of MAX-NUMBER-OF-PRESENCE-SUBSCRIPTIONS (defined in Appendix B.1) as a value for total maximum number of subscriptions.

If such configuration is present for the watcher, the watcher SHALL NOT generate more subscriptions to the presence event package as it is instructed by the limitation configuration.

5.2.2.2 Subscription to a Presence List

Presence lists enable a watcher to subscribe to multiple presentities using a single subscription.

A watcher MAY subscribe to a presence list. If a watcher subscribes to a presence list, it SHALL support the SIP event notification extension for resource lists, according to the subscriber procedures described in [RFC4662].

5.2.2.2.1 Limiting the number of entries in presencelist document

The service provider MAY configure a watcher with a total maximum number of back-end subscriptions allowed for presencelists. This can be done with OTA Provisioning or local configuration. In case of OTA Provisioning, the watcher SHALL use the value of MAX-NUMBER-OF-SUBSCRIPTIONS-IN-PRESENCE-LIST (defined in Appendix B.1) as a value for total maximum number of back-end subscriptions allowed for presencelists.

NOTE: If such configuration is present for the watcher, the watcher will not receive notifications for those presentities that could not be subscribed by the RLS due to the maximum number of back-end subscriptions' limitation configuration. Therefore care has to be taken when creating presencelist documents taking into account this configuration parameter.

5.2.3 Presence information processing

This section describes how watchers SHALL interpret the received presence information.

If the watcher receives more than one <tuple> elements in the presence document including:

- <contact> elements (defined in [RFC3863]) with the same values; and
- <service-description> elements (defined in section 10.5.1), if present, with identical <service-id> and <version> elements;
- other conflicting child elements (i.e. elements with same names but different values or attributes),

Then the watcher SHALL select the child element with the latest <timestamp> element (defined in [RFC3863]) from the conflicting elements and SHALL ignore the remainder of the conflicting child elements from <tuple> elements. Note, that particular <tuple> child elements might specify a different behaviour (see presence information definitions such as section 10.4).

If the watcher recognizes more than one "person" components in the presence document with conflicting child elements (i.e. elements with same names but different values or attributes), the watcher SHALL select the conflicting child element from the <person> element with the latest <timestamp> element as defined in [RFC4479] and SHALL ignore the remainder of the conflicting child elements from <person> elements. Note, that particular <person> child elements might specify a different behaviour (see presence information definitions such as section 10.4).

A watcher SHALL be able to interpret any application-specific subset of the elements listed in section 10.4 using the semantics described therein. The Watcher MAY support other PIDF extensions to interpret elements whose semantics do not match with those defined in section 10.4, as long as, if a watcher that does not understand those extensions can ignore them without changing the meaning of the presence elements that are understood.

5.2.4 Partial Notifications

Partial notification is a mechanism for receiving only those parts of the presence information that have changed since the last notification received by the watcher, rather than the full presence state.

A watcher subscribing to presence information MAY request partial notifications. A watcher requesting partial notifications SHALL support the following:

- SIP extension for partial notifications, according to the watcher procedures described in [PARNOT], and
- partial presence extension to PIDF and its Content-type 'application/pidf-diff+xml', according to [PARFORMAT].

5.2.5 Event Notification Filtering

Event notification filtering is a mechanism for the watcher to control the content and triggers of notifications.

A watcher subscribing to presence information MAY request event notification filtering. A watcher requesting event notification filtering SHALL support the following:

- Event notification filtering, according to the subscriber procedures described in [RFC4660], and
- Content-type 'application/simple-filter+xml', according to [RFC4661].

5.2.6 Handling of large objects

A watcher MAY implement the 'multipart/related' content type as described in [RFC2387], in order to aggregate other MIME objects with the 'application/pidf+xml' content type. In this case, the watcher SHALL indicate the support for the 'multipart/related' content type by using the "Accept" header field in the SUBSCRIBE request.

5.2.6.1 Fetching indirect content

A watcher MAY support the content indirection mechanism [RFC4483]. If supported, the watcher SHALL indicate the support for the 'message/external-body' content type by using the "Accept" header field in the SUBSCRIBE request.

If the watcher receives an indirect content in a NOTIFY request, the watcher SHALL fetch the content from the Content Server as defined in [RFC4483].

If the URI received as indirect content in the NOTIFY request is an HTTPS URI the watcher SHALL perform according to [RFC2818].

5.3 Watcher information subscriber

The watcher information subscriber is an entity that subscribes to the dynamically changing set of (presence) watchers defined in section 5.2 and state of these (presence) subscriptions.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the watcher information subscriber MAY be implemented in a UE or an AS as defined in [3GPP TS 23.228] and [3GPP2 X.S0013-002] respectively.

5.3.1 Subscription to Watcher Information

A watcher information subscriber MAY be co-located with a Presence Source or a watcher and SHALL support subscription and notification of watcher information, according to the subscriber procedures described in [RFC3265] and [RFC3857].

A watcher information subscriber SHALL support the 'application/watcherinfo+xml' content type, according to [RFC3858].

A presentity for which the presence service is activated SHOULD have a corresponding watcher information subscriber, e.g. to support reactive authorization. The mechanism how this is achieved is outside the scope of this document.

If subscription to watcher information event package is required, the presentity SHALL subscribe to the watcher information event package with one of its corresponding watcher information subscribers upon activation of the service. This information can be used, for instance, for reactive authorization.

5.3.1.1 Event notification filtering

Event notification filtering is a mechanism for the watcher information subscriber to control the content of notifications sent to it.

A watcher information subscriber subscribing to watcher information MAY request event notification filtering. A watcher information subscriber requesting event notification filtering SHALL support the following:

- Event notification filtering, according to the subscriber procedures described in [RFC4660], and
- Content-type 'application/simple-filter+xml', according to [RFC4661].

5.4 Presence Server

The Presence Server (PS) is an entity that accepts, stores and distributes presence information. The PS performs the following functions:

- Handles publications from one or multiple Presence Source(s) of a certain presentity. This includes refreshing presence information, replacing existing presence information with newly published information, or removing presence information, for a given Presence Source (see section 5.4.1)
- Composes the presence information received from one or multiple Presence Source(s) into a single presence document (see section 5.4.3.1).
- Handles subscriptions from watchers to presence information and generates notifications about the presence information state changes (see section 5.4.2).

- Handles subscriptions from watcher information subscribers to watcher information and generates notifications about the watcher information state changes (see section 5.4.4).
- Authorizes the watcher's subscription to the presentity's presence information and applies policies (see section 5.4.3.2).
- Applies the watcher's event notification filtering preferences, as appropriate (see section 5.4.4.1).
- Applies rate control mechanisms to the notifications, as appropriate (see section 5.4.3.4).

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the PS SHALL be implemented in an AS as defined in [3GPP TS 23.228] and [3GPP2 X.S0013-002-A] respectively.

5.4.1 Presence information publication acceptance from Presence Sources

A PS SHALL implement the Event State Compositor (ESC) function and support the PUBLISH method according to the procedures described in [RFC3903].

A PS SHALL support the 'application/pidf+xml' content type, according to [RFC3863].

5.4.1.1 Applying Presence Publication

As part of the publication process, the Presence Server may need to replace existing presence information with new incoming information received by Presence Sources (see Section 4.4 of [RFC3903]).

The PS SHALL handle incoming publications as defined in [RFC3903].

5.4.1.2 Presence publication authorisation

Before accepting a PUBLISH request, the PS SHALL perform identity verification and authorization of the publication attempt of the Presence Source, per local policy.

When the SIP/IP Core corresponds to 3GPP IMS or 3GPP2 MMD then the PS SHALL

- verify the identity of the Presence Source of the PUBLISH request as described in [3GPP TS 24.229]/ [3GPP2 X.S0013-004-A] sub-clause 5.7.1.4

The PS publication authorisation policy SHALL authorize the publication for the presentity, and SHOULD reject the publication for all other users.

The PS SHALL perform authorisation of the publication by verifying that the identity of the source of the PUBLISH request matches against the value of the "entity" attribute of the <presence> element in the Presence information document as described in [RFC3863]. If the presentity is identified by both a pres URI and a SIP URI, they SHALL be considered equivalent for the purposes of publication and publication authorization.

In case of successful authorization, the PS accepts the PUBLISH request and SHALL process the PUBLISH request in accordance with [RFC3903].

If a <timestamp> element exists in a <tuple> element, <person> element or <device> element, the PS SHALL overwrite its value with the time the PUBLISH request was received. If a <timestamp> element does not exist in a <tuple> element, <person> element or <device> element, the PS SHALL add a <timestamp> element respectively. The PS SHALL NOT update <timestamp> element value on publication refreshes.

The PS SHALL ensure that consecutive publications never are assigned the same time-stamp, such that in the case of conflicts watchers are always able to differentiate between elements by looking at the time of their publication.

5.4.1.3 Handling of partial publications

The PS MAY support partial publication.

If the Presence Source generates a partial publication request as described in chapter 5.1.1.1 using the 'application/pidf-diff+xml' content-type defined in [PARFORMAT] the PS SHALL process the PUBLISH request in accordance with [RFC3903] and [PARPUBLISH].

5.4.1.4 Handling of large objects

The PS MAY support the 'multipart/related' content type in accordance with [RFC2387]:

If supported, the PS SHALL process a presence document represented as 'multipart/related' content type as follows:

- If the 'multipart/related' content type contains direct MIME object data, the PS SHALL check the size of the direct MIME object data.

    a. If the size exceeds the upper limit as defined by Presence Server policies the Presence Server SHALL stop processing and return the SIP response "413 Request Entity Too Large". The upper limit used by the Presence Server SHALL be at least equal to or greater than the respective limit defined for the Presence Source.

    b. If the size of the direct MIME object data is within the PS's upper limit, the PS SHALL either store the MIME object in case of initial publication or replace an existing content in case of modify operation.

- If the 'multipart/related' content type contains an indirect MIME object included in a 'message/external-body' content type and the content indirection [RFC4483] mechanism is supported by the PS, the PS SHALL associate the value of the relevant presence attribute with the external content.

If the PS does not support the 'multipart/related' content type, then the PS shall send a 415 (Unsupported Media Type) response and indicate the supported content types in the "Accept" header field.

5.4.2 Presence state event package

The PS SHALL support subscriptions for the presence event package, according to the procedures described in [RFC3265] and [RFC3856].

Before accepting a SUBSCRIBE request for the presence event package, the PS SHALL perform authorization of the subscription attempt of the watcher, per Presentity policy. The policies to authorize the watcher's subscription request are described in section 5.4.3.2. If the PS accepts the SUBSCRIBE request, the PS SHALL process the SUBCRIBE request in accordance with [RFC3265] and [RFC3856].

The PS SHALL support notification of changes to the presence event package, according to the procedures described in [RFC3265] and [RFC3856], to authorized watchers after applying the following:

- Composition policy,
- Content rules
- Event notification filtering.
- Partial notification processing
- Event throttling

5.4.2.1 Handling of large objects

The PS MAY generate notifications using the 'multipart/related' content type in accordance with [RFC2387], if:

- the presence information formatted as 'application/pidf+xml' includes references to other MIME objects; and

- the watcher indicates support for the 'multipart/related' content type using the "Accept" header field in the SUBSCRIBE request.

If the watcher does not indicate support for the 'multipart/related' content type or a MIME object cannot be accessed by the PS, the PS SHOULD exclude the MIME object from the notification.

If the size of the MIME object data in the NOTIFY request exceeds the limit defined for the Watcher the PS SHALL handle the MIME object data as indirect content, i.e. store the MIME object data in the Content Server and include an HTTP or optionally HTTPS URI in the notification pointing to the stored MIME object.

If the reference to the MIME object is an HTTPor optionally HTTPS URI, the PS SHALL either:

- fetch the content using the HTTP GET method defined in [RFC 2616] and include as direct content in the notification; or
- include an HTTP or optionally HTTPS URI as indirect content in the notification pointing to the MIME object.

Access to indirect content SHALL be restricted to the watcher. Any appropriate mechanism may be used, given it does not impose any requirements to the watcher other than having to issue an HTTP GET to fetch the indirect content from the provided URI.

In the case of sending the MIME object as direct content, the PS SHALL modify the value of the relevant presence attribute in the presence document to refer to the MIME object included in the 'multipart/related' content type.

5.4.2.2 Generating partial notifications

The PS SHALL support partial notifications. If the watcher indicates preference for partial notifications in the SUBSCRIBE request for the presence event package, the PS SHALL generate partial notifications in accordance with [PARNOT] and [PARFORMAT].

5.4.3 Presence information processing

The PS SHALL process the Presence Information published by the Presence Sources before delivering it to the watchers by applying the following (see Figure 4):

- Composition policy
- Content rules
- Event notification filtering
- Partial notification processing
- Event throttling

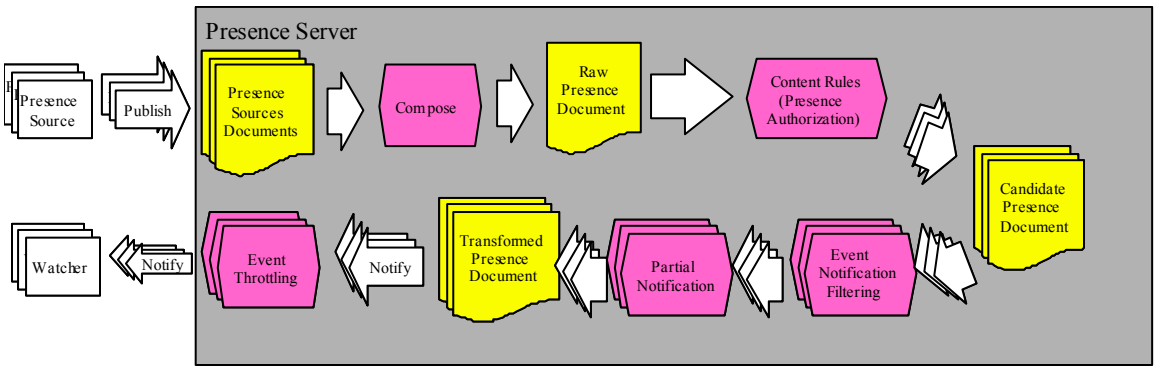

Figure 4 -Presence Information Processing Stages

5.4.3.1 Applying Composition Policy

The function of the PS to combine the "views" of the various Presence Sources in one single raw presence document for a particular presentity is called composition. The presence data model for the presence information is described in section 10.1.

The PS SHALL handle incoming publications as per [RFC3903] before applying the Composition Policy.

The PS SHALL apply the following Composition Policy.

Note: Local policy can augment this composition policy in which case implementations have to ensure that the semantics of this enabler are not violated.

5.4.3.1.1 Composition Policy

The PS SHALL compose the information from the different Presence Sources according to the following rules:

- Service elements (defined in section 10.1.2)

  If the following conditions all apply:

  a. If one <tuple> element includes a <contact> element, as defined in [RFC3863], other <tuple> elements include an identical <contact> element; and

  b. If one <tuple> element includes a <service-description> element, as defined in section 10.5.1, other <tuple> elements include an identical <service-description> element. Two <service-description> elements are identical if they contain identical <service-id> and <version> elements; and

  c. If one <tuple> element includes a <class> element, as defined in section 10.5.1, other <tuple> elements include an identical <class> element; and

  d. There are no conflicting elements (i.e. same elements with different values) or attributes under the <tuple> elements. Different <timestamp> values are not considered as a conflict.

  then the PS SHALL:

  a. Aggregate elements within a <tuple> element that are published from different Presence Sources into one <tuple> element. Identical elements with the same value and attributes SHALL not be duplicated; and

  b. Set the "priority" attribute of the <contact> element in the aggregated <tuple> element to the highest one among those in the input <tuple> elements, if any "priority" attribute is present; and

  c. Set the <timestamp> of the aggregated <tuple> to the most recent one among the ones that contribute to the aggregation; and

  d. Keep no more than one <description> element from the <service-description> elements of the aggregated <tuple> element when there are different values of the <description> elements.

  In any other case, the PS SHALL keep <tuple> elements from different Presence Sources separate.

- Device elements (defined in section 10.1.3)

  If the <deviceID> of the <device> elements that are published from different Presence Sources match, the PS SHALL

  a. Aggregate the non-conflicting elements within one <device> element.

The <timestamp> of the aggregated <device> element SHALL be the most recent one among the ones that contribute to the aggregation; and

b. Use the element from the most recent publication for conflicting elements.

- Person element (defined in section 10.1.1)
  If the following conditions all apply:

  a. If one <person> element includes a <class> element, as defined in section 10.5.1, other <person> elements include an identical <class> element;

  b. if there are no conflicting elements (same elements with different values or attributes) under the <person> elements. Different <timestamp> values are not considered as a conflict.

  then the PS SHALL:

  a. Aggregate elements within a <person> element that are published from different Presence Sources into one <person> element. Identical elements with the same value SHALL not be duplicated. Different <timestamp> values are not considered as a conflict.

  b. Set the <timestamp> of the aggregated <person> element to the most recent one among the ones that contribute to the aggregation.

  The PS SHALL keep <person> elements from different Presence Sources separate if there are conflicting elements under the <person> elements.

The PS SHALL ignore the values of instance identifier attributes (id) of <tuple>, <person> and <device> instances in presence documents published by Presence Sources. The <timestamp> of the aggregated <person> element SHALL be the most recent one among the ones that contribute to the aggregation.

The PS MAY change the values of instance identifier attributes (id) of <tuple>, <person> and <device> instances in presence documents that have been published by Presence Sources.

5.4.3.2 Applying Presence Authorisation Rules

Presence information is considered very sensitive personal information; therefore an authorisation mechanism SHALL be supported.

The PS SHALL apply the Presence Authorisation Rules to all authenticated SUBSCRIBE requests and outgoing notifications for the presence event package.

When the presentity changes the Presence Authorization Rules, the PS SHALL ensure it applies the Presence Authorization Rules with those most recent changes (see section 5.4.5).

As defined in [PRESXDM] the Presence Authorisation Rules has two parts defined by the presentity:

- Subscription Authorisation Rules, which determine if a watcher is allowed to subscribe to the presentity's presence information;
- Presence Content Rules, which determine the subset of the presentity's presence information the watcher is allowed to watch.

When a SUBSCRIBE request is received for the presence event package, the PS SHALL fetch the presentity's Presence Authorisation Rules document stored in the Presence XDMS according to the procedures defined in [XDMSPEC] section 6.1.1. When fetching the document, the PS SHALL construct the HTTP URI as follows:

- Set the XCAP Root URI as described in [XDMSPEC];
- Set the AUID to "org.openmobilealliance.pres-rules" as defined in [PRESXDM]; and

- Set the XUI to the SIP URI or TEL URI of the presentity.

For example, the HTTP URI of the Presence Authorisation Rules document for a presentity with a SIP URI of sip:user@domain.com would be http://xcap.example.com/services/org.openmobilealliance.pres-rules/users/sip:user@domain.com/presrules, if the XCAP Root URI is http://xcap.example.com/services.

The PS SHALL determine which rules in the Presence Authorisation Rules document are applicable and evaluate the combined permissions according to the procedures described in [XDMSPEC] section 6.6.2.3, with the following clarifications:

- When realized in 3GPP IMS or 3GPP2 MMD networks, the PS SHALL use the received P-Asserted-Identity header (as defined in [3GPP TS 24.229] and [3GPP2 X.S0013-004-A]) in the SUBSCRIBE request to determine the URI value used for matching against a conditions element.
- If a presence subscription is identified as anonymous (see section 7.1), the PS SHALL always evaluate the rule with the <anonymous-request> condition element as defined in [XDMSPEC].
- If an attempt to resolve an <external-list> condition element fails, the PS SHALL regard the Presence Authorization Policy Rules document as invalid and act according to the default policy of the PS. If there is no matching rule then the PS SHALL further handle the subscription according to the default policy of the PS. The default policy SHALL apply one of the <sub-handling> actions defined below. However, it is out of scope of the present specification to define how the default policy is configured.

After evaluating the combined permissions the PS SHALL handle the subscription for this watcher based on the value of the <sub-handling> action as follows:

- if the value is “block” or there is no value, then the PS SHALL reject the subscription by responding to the SUBSCRIBE request to rules and procedures of [PRESRULES], 3.2.
- if the value is “polite-block”, then the PS SHALL politely block the subscription following the procedures defined in section 5.4.3.2.1.
- if the value is “confirm”, then the PS SHALL place the subscription in “pending” state according to rules and procedures of [PRESRULES], 3.2. The further treatment of the subscription will depend on the local policy of the PS, a typical example of such a local policy is the request for “reactive authorisation” from the presentity.
- if the value is “allow”, then the PS SHALL place the subscription in the “accepted” state according to rules and procedures of [PRESRULES], 3.2 and apply the Presence Content Rules defined under the “transformations” element of the matched rules as specified in [PRESXDM].

While watcher subscriptions are active, a presentity may update its Subscription Authorization Rules. The PS SHALL re-evaluate the subscription state for each watcher based on the new Subscription Authorization Rules. For example, a presentity may decide to block subscriptions from a watcher. If the watcher has active subscriptions to the presentity, the PS terminates these subscriptions and blocks any future subscription requests from this watcher.

Furthermore, while watcher subscriptions are active a presentity may update its Presence Content Rules. The PS SHALL re-determine the subset of the presentity’s presence information the watcher is allowed to watch. For example, a presentity may decide to stop disseminating specific presence elements to its watchers. In such a case the PS will generate presence notifications that will omit those specific presence elements.

The PS MAY determine that the Subscription Authorization and/or Presence Content Rules have been updated by subscribing to changes made to XML documents stored in the Presence XDMS and Shared XDMS.

5.4.3.2.1 Polite blocking

Polite blocking is a mechanism to deny providing presence information updates , while indicating to the watcher that the subscription is accepted.

If the result of applying Subscription Authorisation Rules is to perform polite blocking (see section 5.4.3.2), the PS SHALL perform the following:

- The PS SHALL respond to the SUBSCRIBE request according to rules and procedures of [PRESRULES].
- The PS SHALL then send only one NOTIFY request the PS with the following content:
    - provide only the <tuple> elements of the "raw presence document" of the presentity indicating that the presentity is "unwilling" and "un-available" for communication (see section 10.4 of the exact details of how these states are mapped to relevant presence information elements). If further child elements are contained in the "raw presence document" within the <tuple> elements apart from "willingness" and "availability", they SHALL be omitted by the PS.
    - not provide the <device> and <person> elements if existent in the presentity's "raw presence document"
    - perform all the next stages in the Presence Information processing framework, as they are listed in section 5.4.3 and detailed in relevant sub-sections (e.g. apply filtering, partial notifications, throttling etc)

5.4.3.3 Applying event notification filtering

The PS MAY support event notification filtering according to the following procedures:

- Event notification filtering, according to the procedures described in [RFC4660], and
- Content type 'application/simple-filter+xml', according to [RFC4661].

If the PS supports event notification filtering; and

- understands the particular filter included in the payload of the SUBSCRIBE request using the content type 'application/simple-filter+xml', the PS SHALL apply the requested filter. As a result, the authorized watchers are notified of the actual presence information after first applying the privacy filtering procedures as described in section 5.4.3.2 then the event notification filtering procedures described in this section.
- does not understand the particular filter included in the payload of the SUBSCRIBE request as requested by the watcher, the PS SHALL indicate it to the watcher as specified in [RFC4660] and [RFC4661].

5.4.3.4 Applying partial notification

The PS SHALL support partial notification procedures as described in section 5.4.2.2.

5.4.3.5 Applying event throttling

The PS MAY have local configuration settings that limit the rate at which notifications are generated (i.e. the shortest time period between two NOTIFY messages for a given watcher). In this case, the PS does not generate NOTIFY messages more often than the configuration dictates.

5.4.3.6 Generation of Notifications

At the last stage of the presence information processing the Presence Server SHALL generate new notifications for each watcher and transmit each of those to the respective watcher when the content of the new notification is different from the last one that was transmitted.

The presentity is identified by a SIP URI (as defined in [RFC3261]), and may additionally be identified by a pres URI (as defined in [RFC3859]). If the presentity is identified by both a pres URI and a SIP URI, the Presence Server SHALL set the "entity" attribute of the <presence> element included in the NOTIFY request to the same URI as the one used in the Request-URI of the received SUBSCRIBE request.

5.4.4 Watcher information event package

Before accepting a SUBSCRIBE request for the watcher information event package, the PS SHALL perform authorization of the subscription attempt of the watcher information subscriber, per local policy. The default policy SHALL be to authorize the subscription if the watcher information subscriber is the presentity, and to reject the subscription for all other users. If the PS accepts the SUBSCRIBE request, the PS SHALL process the SUBSCRIBE request in accordance with [RFC3265], [RFC3857], [RFC3858].

5.4.4.1 Applying event notification filtering

The PS MAY support event notification filtering according to the following procedures:

- Event notification filtering, according to the procedures described in [RFC4660], and
- Content type 'application/simple-filter+xml', according to [RFC4661].

If the PS supports event notification filtering; and

- understands the particular filter included in the payload of the SUBSCRIBE request, the PS SHALL apply the requested filter.
- does not understand the particular filter included in the payload of the SUBSCRIBE request, the PS SHALL indicate it to the subscriber as specified in [RFC4660] and [RFC4661].

5.4.5 XDM Functions

Certain PS functionality depends on particular policy documents stored in the Presence and Shared XDMSs. In order to provide this functionality the PS SHALL support the following :

- Retrieval of XML documents stored in the Presence XDMS and Shared XDMS, according to [XDMSPEC] section 6.1.1 (via the PRS-8 and PRS-5 reference points, respectively).
- XCAP application usages specified in [PRESXDM] and [SharedXDM].

The PS MAY subscribe to changes made to XML documents stored in the Presence XDMS and Shared XDMS, If so, the PS SHALL follow the procedure defined in [XDMSPEC] section 6.1.2 (via the PRS-3 reference point).

5.5 Resource List Server

The Resource List Server (RLS) performs the following functions:

- Accepts subscriptions to presence lists.
- Authorizes the watcher's usage of the presence list.
- Creates and manages back-end subscriptions to all presentities in the presence list, on behalf of the watcher.
- Sends notifications to the watcher, based on information received from the back-end subscriptions.
- Applies aggregation and rate control mechanisms to the notifications, as appropriate.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the RLS SHALL be implemented in an AS as defined in [3GPP TS 23.228] and [3GPP2 X.S0013-002-A] respectively.

5.5.1 General

The RLS SHALL support list subscriptions to the presence event package, according to the RLS procedures described in [RFC4662].

Before accepting a list subscription, the RLS SHALL perform authorization of the usage of a presence list by the watcher, per local policy.

If the list subscription is authorized, the RLS SHALL resolve the presence list into individual presentities according to section 5.5.4.

When sending a list notification, the RLS SHALL set the "uri" attribute of each <resource> element included in the RLMI document to the URI for the presentity that is stored in the presence list.

NOTE: If a presentity is identified by a pres URI in the presence list, the pres URI is included in the RLMI document even if the RLS has knowledge of an equivalent SIP URI.

5.5.2 Back-end Subscriptions

For list subscriptions to the presence event package, the RLS SHALL generate back-end subscriptions to learn the presence information of presentities in the list.

For back-end subscriptions using SIP, the RLS SHALL support subscription and notification of presence information, according to the procedures described in sections 5.2.1, 5.2.2, 5.2.4, 5.2.5 and 5.2.6.

When the SIP/IP Core network corresponds to 3GPP IMS or 3GPP2 MMD networks, the RLS SHALL follow the procedures described in section 5.7.3 of [3GPP TS 24.229] and [3GPP2 X.P0013-004-A] and insert the SIP URI of the P-Asserted-Identity header of the incoming SIP SUBSCRIBE request (as defined in [3GPP TS 24.229] and [3GPP2 X.P0013-004-A]) to the SIP SUBSCRIBE request of the back-end subscription, as opposed to acting as an authentication service ([RFC4474]) required by the [RFC4662].

If the OTA Provisioning parameter MAX-NUMBER-OF-SUBSCRIPTIONS-IN-PRESENCE-LIST or local policy instructs, the RLS SHALL limit the number of back-end subscriptions. The RLS SHALL:

- initiate no more back-end subscriptions as instructed by the provisioning parameter or local policy; and
- return no <instance> element for those <resource> elements that could not be subscribed from the presence list document due to this limitation. The <instance> and <resource> elements are part of the Resource List Meta-Information (RLMI) document as defined in [RFC4662].

When the watcher adds presentities to the presence list while the list subscription is active, the RLS SHALL generate back-end subscriptions for the newly added presentities, and SHALL include the newly added presentities in the next list notification. This procedure SHALL NOT require the watcher to re-subscribe to the presence list.

When the watcher removes presentities from the presence list while the list subscription is active, the RLS SHALL terminate back-end subscriptions to the recently removed presentities, and SHALL indicate that the back-end subscriptions have been terminated in the next list notification. This procedure SHALL NOT require the watcher to re-subscribe to the presence list.

5.5.3 Event Notification Filtering

The RLS MAY support event notification filtering according to the following procedures:

- Event notification filtering, according to the RLS and notifier procedures described in [RFC4660], and
- Content type 'application/simple-filter+xml', according to [RFC4661].

If the RLS supports event notification filtering; and

- understands the particular filter included in the payload of the SUBSCRIBE request, the RLS SHALL apply the requested filter.
- does not understand the particular filter included in the payload of the SUBSCRIBE request, the RLS SHALL indicate it to the subscriber as specified in [RFC4660] and [RFC4661].

5.5.4 XDM Functions

In order to resolve presence lists into individual presentities, the RLS SHALL support the following :

- Retrieval of XML documents stored in the RLS XDMS and Shared XDMS, according to [XDMSPEC] section 6.1.1 (via the PRS-10 and PRS-9 reference points, respectively).
- XCAP application usages specified in [RLSXDM] and [SharedXDM].

On receiving a SIP SUBSCRIBE request directed at a presence list identified by a Request-URI, the RLS SHALL access the global "index" document described in [RLSXDM] using the XCAP path
[XCAP Root URI]/rls-services/global/index.

The RLS SHALL retrieve the presence list from the contents of the <service> element within the index document whose "uri" attribute value matches the Request-URI of the received SUBSCRIBE request. If the RLS is unable to retrieve the presence list from the RLS XDMS, the RLS SHALL reject the SUBSCRIBE request with a 404 (Not Found) response.

The presence list can contain references to URI Lists stored in the Shared XDMS. If the RLS is unable to retrieve a URI List from the Shared XDMS, then that URI List SHOULD be ignored; if so, the watcher is made aware of this when the URIs which could not be de-referenced are omitted from the list notification.

The RLS MAY subscribe to changes made to XML documents stored in the RLS XDMS and Shared XDMS. If so, the RLS SHALL follow the procedures defined in [XDMSPEC] section 6.1.2 (via the PRS-4 reference point).

When realized in 3GPP IMS or 3GPP2 MMD networks, the RLS SHALL insert the SIP URI of the received P-Asserted-Identity header (as defined in [3GPP TS 24.229] and [3GPP2 X.P0013-004-A]) from the SIP SUBSCRIBE request in the X-3GPP-Asserted-Identity header, as defined in [3GPP TS 24.109] or the "X-XCAP-Asserted-Identity" header as defined in [XDMSPEC], of the HTTP GET request.

5.5.5 Rate control and Aggregation

Subject to rate limitations described below, the RLS SHALL generate notifications when it receives updated information from back-end subscriptions.

The RLS MAY have configuration settings that limit the rate at which notification are generated (i.e. the shortest time period between two NOTIFY message). In this case, the RLS does not generate NOTIFY messages more often than the configuration dictates.

If multiple back-end notifications arrive while rate control restrictions apply, the RLS MAY aggregate those notifications (i.e. combine the presence content into a single NOTIFY message) and transmit them when those restrictions expire. The mechanism by which multiple notifications are aggregated is described in [RFC4662].

5.6 XDM Client

The XDM Client SHALL support the XDM Client procedures described in [XDMSPEC] section 6.1, and the XCAP application usages described in [PRESXDM], [RLSXDM], and [SharedXDM].

5.7 Presence XDMS

The Presence XDMS SHALL support the XDM Server procedures described in [XDMSPEC] section 6.2, and the XCAP application usages described in [PRESXDM].

5.8 RLS XDMS

The RLS XDMS SHALL support the XDM Server procedures described in [XDMSPEC] section 6.2, and the XCAP application usages described in [RLSXDM].

5.9 Content Server

The Content Server SHALL support HTTP GET and PUT methods [RFC2616], and the procedures defined in [RFC4483].

The Content Server SHALL store a MIME object when receiving it in an HTTP PUT request behind the HTTP URI therein. .

The Content Server SHALL return a MIME object in a 200 OK response to an HTTP GET request. The Content Server SHALL fetch the MIME object from the Request URI of the HTTP GET request.

The Content Server can be used by Presence Sources as described in section 5.1.1.2, Watchers as described in 5.2.6 and the Presence Server as described in sections 5.4.1.4 and 5.4.2.1.

NOTE: The procedure for storing MIME objects is not defined by this specification.

5.10 Shared XDMS

The Shared XDMS is described in [XDMSPEC] section 5.2.

6. Description of the Presence Reference Points

The following sections give a description of each of the reference points utilised by this specification, and provides references to several external specifications. Note, that the remainder of this document should be consulted in combination with this section, as it further specifies how some of those external specifications are utilised.

An overview of the reference points can be found in the [PRESAD].

6.1.1 Reference Point PRS-1: Presence Source – SIP/IP Core network

The PRS-1 reference point supports the communication between the Presence Source and the SIP/IP Core network. The protocol for the PRS-1 reference point SHALL be SIP [RFC 3261] and the traffic is routed to the PS via the SIP/IP Core.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the PRS-1 reference point SHALL conform with the following reference points: Pep, Pex, Pen [3GPP TS 23.141], [3GPP2 X.S0027-001-0] depending on the instantiation of the Presence Source (e.g. PUA, PNA, PEA).

In a non-3GPP/3GPP2 system, this reference point is a network connection between the Presence Source and the PS via the SIP/IP core.

The Presence Source SHOULD compress the SIP signaling according to [RFC 3320], [RFC 3485] and [RFC 3486] to reduce the transmission delays.

If the Presence Source initiate the signaling compression according to [RFC 3320], [RFC 3485] and [RFC 3486], then the SIP/IP Core SHALL compress the SIP signaling according to [RFC 3320], [RFC 3485] and [RFC 3486].

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks the signalling compression procedures as defined [3GPP TS 24.229] / [3GPP2 X.S0013-004-A] SHALL be used.

6.1.2 Reference Point PRS-2: Watcher – SIP/IP Core network

The PRS-2 reference point supports the communication between the watcher and SIP/IP Core network. The protocol for the PRS-2 reference point SHALL be SIP [RFC 3261] and the traffic is routed to the PS or RLS via the SIP/IP Core.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the PRS-2 reference point SHALL conform with the Pw reference point [3GPP TS 23.141], [3GPP2 X.S0027-001-0].

In a non-3GPP/3GPP2 system, this reference point is a network connection between the watcher and the PS via the SIP/IP core.

The Watcher SHOULD compress the SIP signaling according to [RFC 3320], [RFC 3485] and [RFC 3486] to reduce the transmission delays.

If the Watcher initiates the signaling compression according to [RFC 3320], [RFC 3485] and [RFC 3486], then the SIP/IP Core SHALL compress the SIP signaling according to [RFC 3320], [RFC 3485] and [RFC 3486].

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks the signalling compression procedures as defined [3GPP TS 24.229] / [3GPP2 X.S0013-004-A] SHALL be used.

6.1.3 Reference Point PRS-3: SIP/IP Core – Presence Server

The PRS-3 reference point supports the communication between the SIP/IP Core network and the PS. The protocol for the PRS-3 reference point SHALL be SIP [RFC 3261].

When SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the PRS-3 reference point SHALL conform with the Pwp reference point [3GPP TS 23.141], [3GPP2 X.S0027-001-0].

6.1.4 Reference Point PRS-4: SIP/IP Core – Resource List Server

The PRS-4 reference point supports the communication between the SIP/IP Core network and the Resource List Server. The protocol for the PRS-4 reference point SHALL be SIP [RFC 3261].

When SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the PRS-4 reference point SHALL conform with the Pwp reference point [3GPP TS 23.141], [3GPP2 X.S0027-001-0].

6.1.5 Reference Point IP-1: SIP/IP Core network – External Presence Network (based on a SIP/IP Core)

The IP-1 reference point supports the communication between the SIP/IP Core network and an External Presence Network based on a SIP/IP Core network. The protocol for the IP-1 reference point SHALL SIP [RFC 3261].

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the IP-1 reference point SHALL conform with the Pw reference point as it is defined in [3GPP 23.141] and [3GPP2X.S0027-001-0].

It is for interconnecting two “trusted” domains through their respective SIP/IP Core networks.

7. Security

The security mechanism provides the protection to the Presence Service environment.

7.1 Privacy

7.1.1 Watcher privacy

If the watcher desires subscription privacy, it SHALL set the From header field of the SIP SUBSCRIBE request to anonymous value as defined in [RFC3261].

The watcher MAY indicate further privacy preferences in accordance with [RFC 3323].

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the watcher SHALL include a Privacy header value set to “id” as described in [RFC3325].

7.1.2 Watcher Information Subscriber Privacy

A subscription to watcher information SHALL be authorized if the watcher information subscriber is the presentity and SHALL be rejected for all other users.

Anonymous watcher information subscription SHALL be rejected.

7.1.3 Presentity Privacy

Privacy of the presentity, i.e. who receives which of the presentity’s presence information is ensured by the presence authorization mechanism described in section 5.4.3.2.

7.1.4 Handling of anonymous presence subscriptions in Presence Server

The PS SHALL consider a subscription as anonymous if any of the following conditions are true:

- When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, and the SIP SUBSCRIBE request contains a Privacy header value set to "id" or "user" as described in [3GPP TS 24.229]/[3GPP2 X.S0013-004-A] section 5.7.1.4.
- The SIP SUBSCRIBE request contains a From header indicating an anonymous value as described in [RFC3261].

Authorization of anonymous subscriptions SHALL be according to the presentity's Subscription Authorization Rules for anonymous subscriptions (see section 5.4.3.2).

7.2 Authentication of SIP requests

The PS or RLS SHALL authenticate all incoming SIP requests. The PS or RLS SHOULD rely on the authentication mechanisms provided by the underlying SIP/IP Core network to accomplish user identity verification.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks:

- The authentication mechanism is specified in [3GPP TS 33.203]/[3GPP2 S.R0086-A].
- The PS or RLS SHALL authenticate the SIP request originator as specified in [3GPP TS 24.229]/[3GPP2 X.S0013-004-A] section 5.7.1.4.
- The PS or RLS acting on behalf of the Presence Source or the Watcher SHALL populate security related SIP header fields according to the procedures given in [3GPP TS 24.229]/[3GPP2 X.S0013-004-A] section 5.7.3.

An AS acting as originating UA SHALL follow the authentication procedures given in [3GPP TS 24.229]/[3GPP2 X.S0013-004-A] section 5.7.3.

7.3 Integrity and confidentiality protection

The access level security mechanism SHALL be provided by the SIP/IP Core network to support integrity and confidentiality protection of SIP signalling.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the integrity and confidentiality protection mechanism is specified in [3GPP TS 33.203]/[3GPP2 S.R0086-A].

8. Charging

8.1 Charging Architecture

Since both online and offline charging SHOULD be supported according to [PRESREQ], there are two different charging architectures, which can be simplified as following:

8.1.1 Offline Charging Architecture

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks the offline charging SHOULD be performed according to [3GPP TS 32.240] [3GPP TS 32.260] for 3GPP and [3GPP2 X.S0013-007-A] [3GPP2 X.S0013-008-A] for 3GPP2.

In the context of other realisations of the SIP/IP Core network similar charging functions SHOULD be provided.

8.1.2 Online Charging Architecture

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks the online charging SHOULD be performed according to [3GPP TS 32.240] [3GPP TS 32.260] for 3GPP and [3GPP2 X.S0013-007-A] [3GPP2 X.S0013-008-A] for 3GPP2.

In the context of other realisations of the SIP/IP Core network similar charging functions SHOULD be provided.

9. Registration

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the Presence Source and the Watcher implemented by a UE SHALL use the 3GPP IMS or 3GPP2 MMD networks registration mechanisms as defined in [3GPP TS 24.229] and [3GPP2 X.S0013-004-A].

In a non-3GPP/3GPP2 network, this document has no requirement regarding the SIP registration procedures.

10.Content of the Presence Document

10.1 Presence data model

The Presence Data Model as defined in IETF [RFC4479] is categorized in four key components: the Presentity URI, the Person, the Service and the Device:

- The Presentity's URI component indicating the Presentity's identifier (e.g. SIP URI, tel. URI)
- The Person components model the information about the Presentity.
- The Service components model the forms of communication used by the Presentity.
- The Device components model the physical pieces of equipment used by the Presentity.

The relationship between the data elements is according to the following scheme:

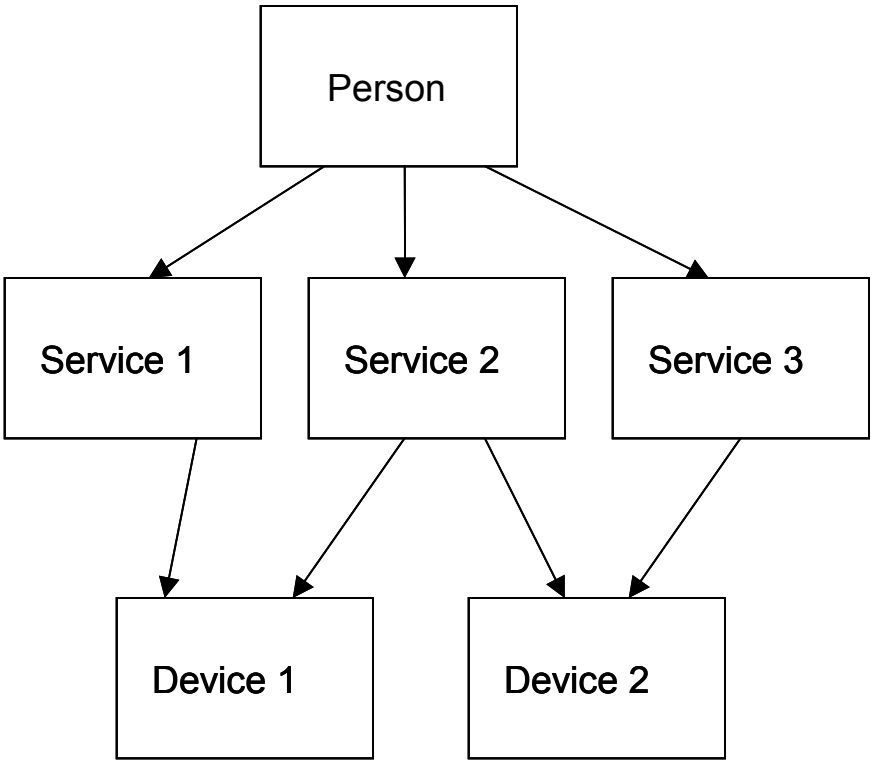

Figure 5: Relationship between the elements of the Presence Data Model

Each of these data elements models Presence information (i.e. Presence attributes) that provides a description about a form of communication, a Presentity, or piece of equipment.

10.1.1 Person

The "person" component models information about the Presentity whom the presence data is trying to describe. Examples of Presence information that can be represented by the "person" component are the activity that the Presentity is involved in, his/her overall willingness for any kind of communication, his/her physical appearance and mood.

The model supports only one "person" component per presentity; nevertheless this does not preclude representing a group which appears to the watcher as a single Presentity. However there may be cases where more than one "person" component instances exist in the Presence document, in cases where composition policy in the PS cannot clearly semantically

differentiate between the multiple instances of the same component. In that case the conflict is resolved according as defined in section 5.2.3.

The “person” component SHALL be mapped to the <person> element. The <person> element is specified in [RFC4479].

10.1.2 Service

The “service” components model the forms of communication that the Presentity has potentially access to. Examples Presence information that can be represented by the “service” components are the Presentity’s willingness to communicate with PoC or IM, the availability of SMS service in his/her terminal.

One other important characteristic of each “service” might be the devices on which that service executes. Each device is uniquely identified by the device identifier <deviceID> defined in [RFC4479]. A service may contain zero or more <deviceID> elements to indicate which devices that service is available on. The Presence document may contain information on each device, but this is a separate part of the document modeled by the “device” component described in the next section.

The “service” component (defined in [RFC4479]) SHALL be mapped to the <tuple> element. The <tuple> element is specified in [RFC3863].

10.1.3 Device

The “device” components model the physical piece of equipment in which services execute. Examples of Presence information that can be represented by “device” elements include mobile phones, PCs and PDAs. As the same services may execute in multiple devices (e.g. IM running in the home PC and the mobile phone) the mapping of services to devices are many to many. Devices are uniquely identified with a device identifier. The model supports only one “device” component per device identifier, however the Presence Sources publish their own “device” component instances. The PS composes the multiple instances into one component and resolves conflicts among the Presence Sources according to the section 5.2.1.1.1.

The “device” component SHALL be mapped to the <device> element. The <device> element is specified in [RFC4479].

For a given presentity, the value of the <deviceID> element of the <device> element SHALL be unique for each device used by the presentity. In case that multiple presence sources exist on a device, the Presence Sources SHALL ensure that irrespective of how many network access means are available in the device only one unique device identifier is used for presence publication.

A version 4 UUID as defined in [RFC4122] SHALL be used for <deviceID> to uniquely identify the device . This is a purely random identifier, providing uniqueness. As this pseudo-random used for <deviceID> is supposed to uniquely identify the particular device it SHALL not change over the lifetime of the device and has to be stored in a non-volatile memory. It SHALL be used in all the Presence publications requiring the use of <deviceID>.

10.2 Void

10.3 Presence Document Overview (Informative)

Information structured according to the OMA presence data model is exchanged in an XML document that conforms to the basic Presence Information Data Format as defined in [RFC3863], and extended in other documents for the purpose of interworking.

The scheme below provides a high level overview of the data elements that may comprise an OMA presence XML document (<presence>).

Column 1: presence information (as defined in this TS)

Column 2: document where the associated **<**element**>** schema is defined

Column 3: location of the **<**element**>** within the <presence> document;

- data elements defined in [RFC3863] are written in *italic*
- data elements defined in this document are written in **bold**

Person	schema	<person> ([RFC4479])
Overriding Willingness	[XSD_PRSPIDF]§	**<overriding-willingness>** **→*<basic>* *open/closed***
Activity	[RFC4480]	<activities>
Location	[RFC4480]	<place-type>
Time-zone	[RFC4480]	<time-offset>
Mood	[RFC4480]	<mood>
Icon	[RFC4480]	<status-icon>
Class	[RFC4480]	<class>
Geographical Location	[RFC4119]	<geopriv> →<location-info> <geopriv> →<usage-rules>
Note	[RFC4479]	<note>
Timestamp	[RFC4479]	<timestamp>

Note that according to the definition of the <person> element in [RFC4479], all child elements outside of [RFC4479] namespace MUST be placed before the <note> element.

Service	schema	*<tuple>* ([RFC3863])
Application-specific Availability	[RFC3863]	*<status>* →*<basic> open/closed*
	[XSD_PRSPIDF]	**<registration-state>**
	[XSD_PRSPIDF]	**<barring-state>**
Application-specific Willingness	[XSD_PRSPIDF]	**<willingness>** **→*<basic>* *open/closed***
Icon	[RFC4480]	**<**status-icon**>**
Session Participation	[XSD_PRSPIDF]	**<session-participation>** **→*<basic>* *open/closed***
Service Description	[XSD_PRSPIDF]	**<service-description>**
Class	[RFC4480]	<class>
Per service device identifier	[RFC4479]	<deviceID>
Communication Address	[RFC3863]	***<contact>***
Timestamp	[RFC3863]	<timestamp>

Note that according to the definition of the <tuple> element in [RFC3863], all child elements outside of [RFC3863] namespace MUST be placed between the <status> and the <contact> element.

Device	schema	<device> ([RFC4479])
Network Availability	[XSD_PRSPIDF]	**<network-availability>** **→<network>**
Geographical Location	[RFC4119]	**<**geopriv**>** →**<**location-info**>** **<**geopriv**>** →**<**usage-rules**>**
Device identifier	[RFC4479]	<deviceID>
Timestamp	[RFC4479]	<timestamp>

Note that according to the definition of the <device> element in [RFC4479], all child elements outside of [RFC4479] namespace MUST be placed before the <deviceID> element.

The following is an example of a raw OMA presence XML document.

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:pdm=" urn:ietf:params:xml:ns:pidf:data-model"
    xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
    xmlns:op="urn:oma:xml:prs:pidf:oma-pres"
    xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
    xmlns:cl="urn:ietf:params:xml:ns:pidf:geopriv10:civicLoc"
    xmlns:gml="urn:opengis:specification:gml:schema-xsd:feature:v3.0"
    xmlns:lt="urn:ietf:params:xml:ns:location-type"
    entity="sip:someone@example.com">

    <tuple id="a1231">
      <status>
        <basic>open</basic>
      </status>
      <op:willingness>
        <op:basic>open</op:basic>
      </op:willingness>
      <op:session-participation>
        <op:basic>open</op:basic>
      </op:session-participation>
      <rpid:status-icon> http://example.com/~my-icons/PoC-Session</rpid:status-icon>
      <op:registration-state>active</op:registration-state>
      <op:barring-state>terminated</op:barring-state>
      <rpid:class>forfriends</rpid:class>
      <op:service-description>
        <op:service-id>org.openmobilealliance:PoC-session</op:service-id>
        <op:version> 1.0 </op:version>
        <op:description>This is the OMA PoC-Session service</op:description>
      </op:service-description>
      <pdm:deviceID>urn:uuid:d27459b7-8213-4395-aa77-ed859a3e5b3a</pdm:deviceID>
      <contact>sip:my_name@example.com</contact>
      <timestamp>2005-02-22T20:07:07Z</timestamp>
    </tuple>
    <tuple id="a1232">

      <status>
        <basic>closed</basic>
      </status>
      <op:willingness>
        <op:basic>closed</op:basic>
      </op:willingness>
      <rpid:status-icon>http://example.com/~my-icons/PoC-Alert </rpid:status-icon>
      <op:registration-state>active</op:registration-state>
      <op:barring-state>active</op:barring-state>
      <rpid:class>forfriends</rpid:class>
      <op:service-description>
        <op:service-id>org.openmobilealliance:PoC-alert</op:service-id>
        <op:version>1.0</op:version>
        <op:description>This is the OMA PoC-Alert service</op:description>
      </op:service-description>
```

```
      <contact>sip:my_name@example.com</contact>
      <timestamp>2005-02-22T20:07:07Z</timestamp>
    </tuple>

    <pdm:person id="a1233">
      <op:overriding-willingness>
         <op:basic>open</op:basic>
      </op:overriding-willingness>
      <rpid:activities>
          <rpid:meeting/>
      </rpid:activities>
      <rpid:place-type> <lt:office/> </rpid:place-type>
      <rpid:mood> <rpid:happy/> </rpid:mood>
      <rpid:status-icon>http://example.com/~my-icons/busy</rpid:status-icon>
      <rpid:time-offset>120</rpid:time-offset>
      <gp:geopriv>
        <gp:location-info>
          <cl:civicAddress>
            <cl:country>US</cl:country>
            <cl:A1>New York</cl:A1>
            <cl:A3>New York</cl:A3>
            <cl:A6>Broadway</cl:A6>
            <cl:HNO>123</cl:HNO>
            <cl:LOC>Suite 75</cl:LOC>
            <cl:PC>10027-0401</cl:PC>
          </cl:civicAddress>
        </gp:location-info>
      </gp:geopriv>
      <rpid:class>forfriends</rpid:class>
      <pdm:note xml:lang="en">I'm in a boring meeting!!</pdm:note>
      <pdm:timestamp>2005-02-22T20:07:07Z</pdm:timestamp>
    </pdm:person>

    <pdm:device id="a1234">
      <op:network-availability>
        <op:network id="IMS">
           <op:active/>
         </op:network>
      </op:network-availability>
      <gp:geopriv>
          <gp:location-info>
             <gml:location>
                <gml:Point gml:id="point1" srsName="epsg:4326">
                  <gml:coordinates>37:46:30N 122:25:10W</gml:coordinates>
                </gml:Point>
             </gml:location>
          </gp:location-info>
           <gp:usage-rules>
             <gp:retransmission-allowed>no</gp:retransmission-allowed>
             <gp:retention-expiry>2003-06-23T04:57:29Z</gp:retention-expiry>
           </gp:usage-rules>
        </gp:geopriv>
      <pdm:deviceID>urn:uuid:d27459b7-8213-4395-aa77-ed859a3e5b3a</pdm:deviceID>
      <pdm:timestamp>2005-02-22T20:07:07Z</pdm:timestamp>
    </pdm:device>
</presence>
```

10.4 Presence Information Elements semantics

OMA Presence RD [PRESREQ] specifies a set of building blocks of presence information that need to be supported by the Presence enabler and their semantics.

The following sections describe the mapping of those presence information building blocks initially to some presence data model components and then to some element of PIDF [RFC3863], or one of its extensions (e.g. RPID [RFC4480], Location Types [RFC 4589], geographical, location object [RFC4119], and Presence Data Model [RFC4479]). In case such a mapping is not possible because elements with similar semantics have not been defined so far in IETF, then OMA-specific extensions to PIDF are performed.

The OMNA maintains a registry of presence information packages to permit easy registration of new PIDF extensions to this enabler. The complete list of the OMA-specific PIDF extensions is available from [OMNA].

The mandatory instance identifier "id" attribute in the <person>, <tuple> and <device> elements of a presence document serves no other purpose than to syntactically distinguish between the elements.

10.4.1 Application-specific Willingness

10.4.1.1 Description

The "Application-specific Willingness" indicates whether the user of the specified communication service desires to receive incoming communication requests for the specified application and device (if specified).

10.4.1.2 Mapping to presence data model

The "Application-specific Willingness" is a part of "service" information according to the presence data model.

10.4.1.3 Mapping to PIDF

The "Application-specific Willingness" building block SHALL be mapped to PIDF as following: <tuple>→ <willingness>→ <basic>→ open/closed and <service-description>.

The <willingness> element is defined is section 10.5.1.2.

The <service-description> element is defined in section 10.5.1.

10.4.1.4 Watcher Processing

Watcher Processing SHALL be performed as described in section 5.2.3.

If the "Overriding Willingness" element exists, then the value of that element SHALL be used and the value of the "Application-specific Willingness" SHALL be ignored.

If none of the two elements exist, then it should be concluded that it is not known whether the user of this communication service desires or not to receive incoming requests.

The semantics of the deduced "willingness" for a watcher are the same, regardless if "application-specific" or "overriding" willingness was used by the presentity.

10.4.1.5 Limitations

None.

10.4.2 Overriding Willingness

10.4.2.1 Description

The "Overriding Willingness" provides an indication, set by the presentity that takes precedence over "Application-specific willingness" settings. For example, when an "Overriding Willingness" element is present, a positive setting indicates that the user is willing to accept communications for all available communications types, while a negative setting indicates that the user is not willing to accept any communications.

10.4.2.2 Mapping to presence data model

The "Overriding Willingness" is part of the "person" component according to the presence data model.

10.4.2.3 Mapping to PIDF

The "Overriding Willingness" building block SHALL be mapped to PIDF as following: <person>→ <overriding-willingness>→ <basic>→ open/closed.

The <overriding-willingness> element is defined is section 10.5.1.3.

10.4.2.4 Watcher Processing

Watcher Processing SHALL be performed as described in section 5.2.3.

The semantics of the deduced "willingness" for a watcher are the same, regardless if "application-specific" or "overriding" willingness was used by the presentity.

10.4.2.5 Limitations

None.

10.4.3 Application-specific Availability

10.4.3.1 Description

The "Application-specific Availability" indicates whether it is possible to receive an incoming communication request using the specified service and device (if specified). For example, if a presentity is provisioned with the PoC Service, within coverage, has an appropriate handset, etc., he would be available for PoC, whereas if any of those were not true, he would be "Not Available".

10.4.3.2 Mapping to presence data model

The "Application-specific Availability" is part of the "service" component according to the presence data model.

10.4.3.3 Mapping to PIDF

The "Application-specific Availability" building block SHALL be mapped to PIDF as following: <tuple>→ <status>→ <basic>→open/closed and <service-description>. The "Application-specific Availability" building block MAY also be mapped to <registration-state> and <barring-state>, if the information for creating these elements is available.

The <service-description> element, <registration-state> element and <barring-state> element are defined in section 10.5.1.

NOTE: The semantics of the <registration-state> and <barring-state> elements are service specific. A particular service should further define the meaning of these elements in the scope of the service.

10.4.3.4 Watcher Processing

Watcher Processing SHALL be performed as described in section 5.2.3.

10.4.3.5 Limitations

None.

10.4.4 Network Availability

10.4.4.1 Description

A device may be "connected" to one or more networks, such as a GSM, CDMA, GPRS, 802.11x, IMS, etc. However, connectivity to a network cannot be defined in a generic manner, as different states may exist for different networks. As such, the <network-availability> element is defined in a generic, extensible way. Each network that needs to be supported needs to extend this specification in order to stipulate the details.

The <network-availability> element SHALL include one or more <network> child elements. Each <network> element SHALL contain an "id" attribute indicating the network type. This value needs to be registered with OMNA, such that it is unique for that type of network. Additionally, each network type will need to define the meaning of "connected", as well any additional information that is relevant for that type of network. The OMNA network-availability registry is available from [OMNA].

10.4.4.2 Mapping to presence data model

The "Network Availability" is part of the "device" component according to the presence data model.

10.4.4.3 Mapping to PIDF

The "Network Availability" building block SHALL be mapped to PIDF as following: <device>→ <network-availability>.

The <network-availability> element is defined is section 10.5.1.4.

10.4.4.4 Watcher Processing

Watcher Processing SHALL be performed as described in Section 5.2.3.

10.4.4.5 Limitations

None.

10.4.5 Communication address

10.4.5.1 Description

The value of this element is the URI used to contact invoke the specific service of the presentity (e.g. SIP URI for a PoC service). When defining a new "service description type" for a new service, the precise semantics of what it means to "invoke the service" SHALL be defined.

10.4.5.2 Mapping to presence data model

The "Communication Address" is part of the "service" component according to the presence data model.

10.4.5.3 Mapping to PIDF

The "Communication Address" building block SHALL be mapped to PIDF as following: <tuple>→ <contact>

10.4.5.4 Limitations

None.

10.4.6 Activity

10.4.6.1 Description

The "Activity" building block is an enumeration of activity-describing elements provided by the Presentity indicating his/her/its current activity(ies).

10.4.6.2 Mapping to presence data model

The "Activity" is part of the "person" component according to the presence data model.

10.4.6.3 Mapping to PIDF

The "Activity" building block SHALL be mapped to <activities> element defined in [RFC4480].

10.4.6.4 Watcher Processing

The watcher processing rules described in section 5.2.3 do not apply for this element.

Should more than one "activities" elements be present in different <person> elements within a presence document, watcher SHALL consider the activities of the presentity to be the aggregate of all activity-describing elements with all <activities> elements Duplicates SHALL be ignored.

10.4.6.5 Limitations

None.

10.4.7 Location Type

10.4.7.1 Description

The "Location-Type" building block indicates an enumerated or free text location value as provided by the presentity. The value of this element indicates the type of location where the presentity physically resides at that point in time. . Free text values of this element SHALL be equal to or less than 20 characters in length, such that they can be displayed on user interfaces.

10.4.7.2 Mapping to presence data model

The "Location-Type" is part of the "person" component according to the presence data model.

10.4.7.3 Mapping to PIDF

The "Location-Type" building block SHALL be mapped to <place-type> element defined in [RFC4480].

10.4.7.4 Watcher Processing

Watcher Processing SHALL be performed as described in section 5.2.3, , except for the case when the <place-type> element has a "from" or "until" attribute. In that case, should more than one "Location-type" element be present in different <person> elements within a presence document, the watcher SHALL consider the locations of the presentity to be the aggregate of all location-describing elements with all <place-type> elements. Duplicates SHALL be ignored..

10.4.7.5 Limitations

None.

10.4.8 Geographical Location

10.4.8.1 Description

The “Geographical Location” building block indicates the presentity’s or the device’s geographical location.

10.4.8.2 Mapping to presence data model

The “Geographical Location” is part of the “person” and/or “device” components according to the presence data model.

10.4.8.3 Mapping to PIDF

The “Geographical Location” building block SHALL be mapped to PIDF as following: <person> -> <geopriv> -> <location-info> and <person> -> <geopriv> -> <usage-rules>> and/or <device> -> <geopriv> -> <location-info> and <device> -> <geopriv> -> <usage-rules>. The <geopriv>, <location-info> and <usage-rules> elements are defined in [RFC4119].

10.4.8.4 Watcher Processing

Watcher Processing SHALL be performed as described in section 5.2.3.

10.4.8.5 Limitations

None.

10.4.9 Time-zone

10.4.9.1 Description

The value of this element indicates the difference in minutes between the time at the current location of the presentity and current UTC time in minutes. The value should be such that when added to UTC, the time at the current location of the presentity is obtained.

10.4.9.2 Mapping to presence data model

The “Time-zone” is a part of “person” component according to the presence data model.

10.4.9.3 Mapping to PIDF

The “Time-zone” building block SHALL be mapped to <time-offset> element defined in [RFC4480].

10.4.9.4 Watcher Processing

Watcher Processing SHALL be performed as described in section 5.2.3, except for the case when the <time-offset> element has a “from” or “until” attribute. In that case, should more than one “Time-zone” element be present in different <person> elements within a presence document, the watcher SHALL consider the time-zone of the presentity to be the aggregate of all time-zone describing elements with all <time-offset> elements. Duplicates SHALL be ignored.

10.4.9.5 Limitations

The ”Time-zone” SHALL follow the limitations described in [RFC4480].

10.4.10 Mood

10.4.10.1 Description

The “Mood” building block is an enumerated value indicating the presentity’s mood.

10.4.10.2 Mapping to presence data model

The “Mood” is a part of “person” component according to the presence data model.

10.4.10.3 Mapping to PIDF

The “Mood” building block SHALL be mapped to <mood> element defined in [RFC4480].

10.4.10.4 Watcher Processing

The watcher processing rules described in section 5.2.3 do not apply for this element.

Should more than one “mood” elements be present in different <person> elements within a presence document, watcher SHALL consider the moods of the presentity to be the aggregate of all “mood” elements. Duplicates SHALL be ignored.

10.4.10.5 Limitations

None.

10.4.11 Icon

10.4.11.1 Description

The “Icon” building block provides a small image that the presentity may chose, such that the watcher’s terminal can use this information to represent the presentity in a graphical user interface.

Presentities SHOULD provide images of sizes and aspect ratios that are appropriate for mobile devices.

The “Icon” SHALL be expressed in one of the following image formats: JPEG, PNG and GIF, as described in [3GPP TS 26.141] and [3GPP2 C.P0071-0].

10.4.11.2 Mapping to presence data model

The “Icon” is part of the“person” and/or “service” component according to the presence data model.

10.4.11.3 Mapping to PIDF

The “Icon” building block SHALL be mapped to <status-icon> element defined in [RFC4480].

10.4.11.4 Watcher Processing

Watcher Processing SHALL be performed as described in section 5.2.3, except for the case when the <status-icon> element has a “from” or “until” attribute. In that case, should more than one “Icon” element be present in different <person> or <tuple> elements within a presence document, the watcher SHALL consider the icon of the presentity to be the aggregate of all icon describing elements with all <status-icon> elements. Duplicates SHALL be ignored.

10.4.11.5 Limitations

None.

10.4.12 Session Participation

10.4.12.1 Description

The “Session Participation” building block indicates that the user is involved in at least one session of a specific service (e.g. PoC session). However definition of a “session” cannot be described in a “generic” manner, as it depends on the semantics of the specific enabler. As such the “session-participation” element is defined in a generic, extensible way. Each enabler that needs to support this element needs to extend this specification in order to stipulate the details. The participation in a session indicates to the watcher that the presentity may not be able to communicate with him/her even though it is possible technically.

10.4.12.2 Mapping to presence data model

The “Session Participation” is part of the “service” component according to the presence data model.

10.4.12.3 Mapping to PIDF

The “Session Participation” building block SHALL be mapped to PIDF as following: <tuple>→ <session-participation>→ <basic>→ open/closed, and <service-description>.

The <session-participation> element is defined is section 10.5.1.5.

The <service-description> element is defined in section 10.5.1.1.

10.4.12.4 Watcher Processing

Watcher Processing SHALL be performed as described in section 5.2.3.

10.4.12.5 Limitations

None.

10.4.13 Timestamp

10.4.13.1 Description

The “Timestamp” building block provides a timestamp specifying the time when the presence server received the most recent information pertaining to the data component instance that contributes to the data component instance’s aggregation. The watcher may use this information to compare information provided in data compont instances. A “Timestamp” building block supplied by a Presence Source on publication of presence information is ignored by the presence server when composing a presence document.

10.4.13.2 Mapping to presence data model

The “Timestamp” can be part of “service”, “device” or “person” components according to the presence data model.

10.4.13.3 Mapping to PIDF

The “Timestamp” building block SHALL be mapped to <timestamp> element defined in [RFC3863] for “service” and [RFC4479] for “device” and “person”.

10.4.13.4 Limitations

The <timestamp> SHALL follow the limitations as defined in [RFC3863] for “service”and [RFC4479] for “device” and “person”.

10.4.14 Class

10.4.14.1 Description

The “Class” element describes the class of the “service” element or “person” element. Multiple elements can have the same class name within a presence document. The naming of classes is left to the presentity. The presentity can use this information to group similar “services” or “person” elements or to convey information that the PS can use for filtering or authorization.

10.4.14.2 Mapping to presence data model

The “Class” is a part of “service” and/or “person” information according to the presence data model.

10.4.14.3 Mapping to PIDF

The “Class” element SHALL be mapped to <class> element defined in [RFC4480].

10.4.14.4 Watcher Processing

Watcher Processing SHALL be performed as described in section 5.2.3.

10.4.14.5 Limitations

None.

10.4.15 Note

10.4.15.1 Description

The “Note” building block is a free text value used to provides any type of written information to a potential watcher.

10.4.15.2 Mapping to presence data model

The “Note” element is part of the “person” component according to the presence data model.

10.4.15.3 Mapping to PIDF

The “Note” building block SHALL be mapped to the <note> element defined in [RFC4479].

10.4.15.4 Watcher Processing

The watcher processing rules described in Section 5.2.3 do not apply for this element.

Should more than one “Note” element be present in different <person> elements within a presence document, the watcher SHALL consider the notes of the presentity to be the aggregate of all <note> elements.

10.4.15.5 Limitations

The values of <note> elements SHALL be limited to 40 characters.

10.4.16 Per service device identifier

10.4.16.1 Description

The “Per service device identifier” building block identifies the device or devices where a particular “service” component executes.

10.4.16.2 Mapping to presence data model

The “Per service device identifier” is part of the “service” component according to the presence data model.

10.4.16.3 Mapping to PIDF

The “Per service device identifier” building block SHALL be mapped to the <deviceID> element defined in [RFC4479].

10.4.16.4 Limitations

The value of the “Per service device identifier” SHALL be following the methodology and restrictions of section 10.1.3.

10.5 OMA specific PIDF extensions

10.5.1 OMA PIDF elements

The complete list of the OMA-specific PIDF extensions is available from [OMNA].

10.5.1.1 <service-description>

The <service-description> element is an extension to PIDF that is used to describe OMA-specific services. The <service-description> element SHALL be used as a child element of the <tuple> element as defined in [PIDF].

Services utilizing this element SHALL register a unique value with OMNA. The OMNA Presence <service-description> Registry is available from [OMNA_pidfSvcDesc].

The <service-description> element SHALL contain the following child elements:

- <service-id> element: Uniquely identifies the service. This element is mandatory and it SHALL contain a string value.
- <version> element: Defines the version of the service. This element is mandatory and it SHALL contain a string value in the form of "x.y" where "x" is the major version and "y" is the minor version of the particular service.
- <description> element: This element is optional. If present, it SHALL contain a string value providing additional informative description of the service.

10.5.1.2 <willingness>

The <willingness> element is an extension to PIDF that is used to describe the "Application-specific willingness" building block. The <willingness> element SHALL be used as a child element of the <tuple> element as defined in [PIDF].

The <willingness> element SHALL include the <basic> element and have two values "open" and "closed" indicating willingness for communication.

10.5.1.3 <overriding-willingness>

The <overriding-willingness> element is an extension to PIDF that is used to describe the "Overriding willingness" building block. The <overriding-willingness> element SHALL be used as a child element of the <person> element defined in [RFC4479].

The <overriding-willingness> element SHALL include the <basic> element with the values "open" and "closed" indicating overriding willingness.

10.5.1.4 <network-availability>

The <network-availability> element is an extension to PIDF that is used to describe the "Network Availability" building block. The <network-availability> element SHALL be used as a child element of the <device> element as defined in [RFC4479].

Each <network-availability> element SHALL include one or more <network> child elements. Each <network> element SHALL contain an "id" attribute indicating the type of the network.

Each <network> element SHALL include at least one of the following elements:

- the <active> element indicating that the particular device is connected to the specific network; and
- the <terminated> element indicating that the particular device is not connected to the specific network.

10.5.1.5 <session-participation>

The <session-participation> element is an extension to PIDF that is used to describe the "Session Participation" building block. The <session-participation> element SHALL be used as a child element of the <tuple> element as defined in [PIDF].

The <session-participation> element SHALL include the <basic> element and have either

- the value "open" indicating that the particular presentity is participating in at least one session of a specific service; or
- the value "closed" indicating that the presentity is not participating in any session of the specific service.

10.5.1.6 <registration-state>

The <registration-state> element is an extension to PIDF that is used to contain the presentity's registration state pertaining to a particular service. The <registration-state> element, if present, SHALL be used as a child element of the <tuple> element defined in [PIDF].

The <registration-state> element SHALL include either

- the value "active" indicating that the particular presentity has an active registration with a specific service; or
- the value "terminated" indicating that the presentity does not have an active registration with a specific service.

10.5.1.7 <barring-state>

The <barring-state> element is an extension to PIDF that is used to contain the presentity's barring state pertaining to a particular service. The <barring-state> element, if present, SHALL be used as a child element of the <tuple> element defined in [PIDF].

NOTE: this element is only useful for those services, which have the option to block incoming or outgoing communication.

The <barring-state> element SHALL include either

- the value "active" indicating that the particular presentity has activated communication barring pertaining to a specific service; or
- the value "terminated" indicating that the presentity has deactivated communication barring pertaining to a specific service.

10.5.2 XML Schema definitions

The XML Schema for the OMA-defined PIDF extensions is given in [XSD_PRSPIDF].

10.6 Presence information examples (Informative)

Examples of how the Presence information semantics are described in a typical Presence Information XML schema are shown below:

Presence Document describing:

- **PoC-Session Specific Availability: Not Available/ Not Registered**

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:op="urn:oma:xml:prs:pidf:oma-pres"
    entity="sip:someone@example.com">
    <tuple id="a1232">
      <status>
        <basic>closed</basic>
      </status>
      <op:registration-state>terminated</op:registration-state>
      <op:barring-state>terminated</op:barring-state>
      <op:service-description>
        <op:service-id>org.openmobilealliance:PoC-session</op:service-id>
        <op:version>1.0</op:version>
      </op:service-description>
      <contact>sip:someone@example.com</contact>
      <timestamp>2005-02-22T10:25:01Z</timestamp>
    </tuple>
</presence>
```

Presence Document describing:

- **PoC-Session Specific Availability: Available/Registered/ISB not activated**
- **PoC-Session Specific Willingness: Willing**
- **Activity: Meal**
- **Geographical Location: Coord <X> and <Y>**

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:pdm="urn:ietf:params:xml:ns:pidf:data-model"
    xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
    xmlns:op="urn:oma:xml:prs:pidf:oma-pres"
    xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
    xmlns:gml="urn:opengis:specification:gml:schema-xsd:feature:v3.0"
    entity="sip:someone@example.com">

    <tuple id="a1232">
      <status>
        <basic>open</basic>
      </status>
      <op:willingness>
        <op:basic>open</op:basic>
      </op:willingness>
      <op:registration-state>active</op:registration-state>
      <op:barring-state>terminated</op:barring-state>
      <op:service-description>
        <op:service-id>org.openmobilealliance:PoC-session</op:service-id>
        <op:version>1.0</op:version>
      </op:service-description>
      <contact>sip:someone@example.com</contact>
      <timestamp>2005-02-23T12:14:56Z</timestamp>
    </tuple>

    <pdm:person id="a1233">
```

```
      <rpid:activities>
        <rpid:meal/>
      </rpid:activities>
      <gp:geopriv>
        <gp:location-info>
          <gml:location>
            <gml:Point gid="point1" srsName="epsg:4326">
              <gml:coordinates>
                <gml:X>30 16 28S</gml:X>
                <gml:Y>45 15 33W</gml:Y>
              </gml:coordinates>
            </gml:Point>
          </gml:location>
        </gp:location-info>
        <gp:usage-rules/>
      </gp:geopriv>
      <pdm:timestamp>2005-02-23T12:14:56Z</pdm:timestamp>
    </pdm:person>

</presence>
```

Presence Document describing:

- **PoC-Session Specific Availability: Available/Registered/ISB not activated**
- **PoC-Session Specific Willingness: Willing**
- **PoC Specific Session Participation: Not Engaging**
- **Device Identifier: urn:uuid: 48662e19-5fbf-43fc-a2fd-d23002787599**

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
     xmlns:pdm="urn:ietf:params:xml:ns:pidf:data-model"
     xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
     xmlns:op="urn:oma:xml:prs:pidf:oma-pres"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          entity="sip:someone@example.com">

     <tuple id="a1232">
       <status>
         <basic>open</basic>
       </status>
       <op:willingness>
         <op:basic>open</op:basic>
       </op:willingness>
       <op:session-participation>
          <op:basic>closed</op:basic>
       </op:session-participation>
       <op:registration-state>active</op:registration-state>
       <op:barring-state>terminated</op:barring-state>
       <op:service-description>
         <op:service-id>org.openmobilealliance:PoC-Session</op:service-id>
         <op:version>1.0</op:version>
       </op:service-description>
        <pdm:deviceID>urn:uuid:48662e19-5fbf-43fc-a2fd-d23002787599</pdm:deviceID>
       <contact>sip:someone@example.com</contact>
       <timestamp>2005-02-21T16:25:56Z</timestamp>
     </tuple>
```

```
     <pdm:device id="a1233">
<pdm:deviceID>urn:uuid:48662e19-5fbf-43fc-a2fd-d23002787599</pdm:deviceID>
       <pdm:timestamp>2005-02-21T16:25:56Z</pdm:timestamp>
     </pdm:device>

</presence>
```

Presence Document describing:

- **PoC-Alert Specific Availability: Not Available/Registered/ISB activated**
- **PoC-Alert Specific Willingness: Not Willing**
- **Network-Availability: IMS-registered**
- **Mood: happy**
- **Location: mall public noisy**
- **Icon: http://example.com/~someone/myicon.gif**
- **the Device Identifier: urn:uuid:48662e19-5fbf-43fc-a2fd-d23002787599**

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:pdm="urn:ietf:params:xml:ns:pidf:data-model"
    xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
    xmlns:op="urn:oma:xml:prs:pidf:oma-pres"
    xmlns:lt="urn:ietf:params:xml:ns:location-type"
           entity="sip:someone@example.com">
    <tuple id="a1232">
      <status>
        <basic>closed</basic>
      </status>
      <op:willingness>
        <op:basic>closed</op:basic>
      </op:willingness>
      <op:registration-state>active</op:registration-state>
      <op:barring-state>active</op:barring-state>
      <op:service-description>
        <op:service-id>org.openmobilealliance:PoC-Alert</op:service-id>
        <op:version>1.0</op:version>
        <op:description>This is the OMA POC-Alert service</op:description>
      </op:service-description>
      <pdm:deviceID>urn:uuid:48662e19-5fbf-43fc-a2fd-d23002787599</pdm:deviceID>
      <contact>sip:someone@example.com</contact>
      <timestamp>2005-02-22T20:07:07Z</timestamp>
    </tuple>

    <pdm:person id="a1233">
          <rpid:place-type>
               <lt:shopping-area/>
               <lt:public/>
          </rpid:place-type>
          <rpid:mood>
             <rpid:happy/>
          </rpid:mood>
```

```
          <rpid:status-icon>http://example.com/~someone/myicon.gif</rpid:status-
icon>
       <pdm:timestamp>2005-02-22T20:07:07Z</pdm:timestamp>
    </pdm:person>

    <pdm:device id="a1234">
        <op:network-availability>
          <op:network id="IMS">
            <op:active/>
           </op:network>
        </op:network-availability>
       <pdm:deviceID>urn:uuid:48662e19-5fbf-43fc-a2fd-d23002787599</pdm:deviceID>
       <pdm:timestamp>2005-02-22T20:07:07Z</pdm:timestamp>
    </pdm:device>
</presence>
```

11.SIP Methods

11.1 SUBSCRIBE Method

When SIP/IP Core is realised with 3GPP IMS or 3GPP2 MMD networks, the supported headers of the SUBCRIBE method and its responses SHALL correspond to those defined in [3GPP TS 24.229] and [3GPP2 X.S0013-004-A] respectively.

In the context of other realisations of the SIP/IP Core network the supported headers of the SUBSCRIBE method and its responses SHALL correspond to those defined in [RFC3265], [RFC3857] and [RFC3856].

11.2 PUBLISH Method

When SIP/IP Core is realised with 3GPP IMS or 3GPP2 MMD networks, the supported headers of the PUBLISH method and its responses SHALL correspond to those defined in [3GPP TS 24.229] and [3GPP2 X.S0013-004-A] respectively.

In the context of other realisations of the SIP/IP Core network the supported headers of the PUBLISH method and its responses SHALL correspond to those defined in [RFC3903].

11.3 NOTIFY Method

When SIP/IP Core is realised with 3GPP IMS or 3GPP2 MMD networks, the supported headers of the NOTIFY method and its responses SHALL correspond to those defined in [3GPP TS 24.229] and [3GPP2 X.S0013-004-A] respectively.

In the context of other realisations of the SIP/IP Core network the supported headers of the NOTIFY method and its responses SHALL correspond to those defined in [RFC3265], [RFC3857] and [RFC3856].

Appendix A. Static Conformance Requirements

The SCR's defined in the following tables include SCR for:

- Presence Source
- Presence Server
- RLS Server
- RLS Client
- Watcher
- XDM Client
- Presence XDMS
- RLS XDMS

Each SCR table indentifies a list of supported features as:

Item: Identifier for a feature.

Function: Short description of the feature.

Reference: Section(s) of this specification with more details on the feature.

Status: Whether support for the feature is mandatory or optional. MUST use "M" for mandatory support and "O" for optional support in this column.

Requirement: This column identifies other features required by this feature. If no other features are required, this column is left empty.

This section describes the dependency grammar notation to be used in the Requirement column of the SCR and CCR tables using ABNF [RFC2234].

TerminalExpression = ScrReference / NOT TerminalExpression / TerminalExpression LogicalOperator TerminalExpression / "(" TerminalExpression ")"

ScrReference = ScrItem / ScrGroup

ScrItem = SpecScrName "–" GroupType "–" DeviceType "–" NumericId / SpecScrName "–" DeviceType "–" NumericId

ScrGroup = SpecScrName ":" FeatureType / SpecScrName "– " GroupType "–" DeviceType "–" FeatureType

SpecScrName = 1*Character;

GroupType = 1*Character;

DeviceType = "C" / "S"; C – client, S – server

NumericId = Number Number Number

LogicalOperator = "AND" / "OR"; AND has higher precedence than OR and OR is inclusive

FeatureType = "MCF" / "OCF" / "MSF" / "OSF"; See Section A.1.6

Character = %x41-5A ; A-Z

Number = %x30-39 ; 0-9

A.1 Presence Source

Item	Function	Reference	Status	Requirement
SIMPLE-SRC-C-001	Support SIP PUBLISH to publish presence information according to [PUBLISH]	5.1.1	M	SIMPLE-SRC-C-006 OR SIMPLE-SRC-C-007
SIMPLE-SRC-C-002	application/pidf+xml PIDF	5.1.1	M	
SIMPLE-SRC-C-003	Presence Data Model	5.1.1, 10.1, 10.4	M	
SIMPLE-SRC-C-004	Partial Publication	5.1.1.1	O	SIMPLE-SRC-C-006 OR SIMPLE-SRC-C-007
SIMPLE-SRC-C-005	Content Indirection	5.1.1.2.1	O	
SIMPLE-SRC-C-006	IMS SIP PUBLISH Method	11.2	O	
SIMPLE-SRC-C-007	NON-IMS SIP PUBLISH Method	11.2	O	
SIMPLE-SRC-C-008	Direct Content	5.1.1.2.2	O	
SIMPLE-SRC-C-009	Source Throttle Publish	5.1.1.3	O	
SIMPLE-SRC-C-010	Presence entity attribute settings	5.1.1	M	SIMPLE-SRC-C-006
SIMPLE-SRC-C-011	Supporting other PIDF extensions	5.1.1, 10.1, 10.4	O	

A.2 Presence Server

Item	Function	Reference	Status	Requirement

Item	Function	Reference	Status	Requirement
SIMPLE-PS-S-001	Presence Data Model	10.1	M	
SIMPLE-PS-S-002	Publication of Presence Information	5.4.1	M	SIMPLE-PS-S-007 OR SIMPLE-PS-S-008
SIMPLE-PS-S-003	Presence information Subscriptions	5.4.2	M	SIMPLE-PS-S-005 OR SIMPLE-PS-S-006
SIMPLE-PS-S-004	Presence Information Notifications	5.4	M	SIMPLE-PS-S-009 OR SIMPLE-PS-S-011
SIMPLE-PS-S-005	IMS SIP SUBSCRIBE Method	11	O	
SIMPLE-PS-S-006	NON-IMS SIP SUBSCRIBE Method	11	O	
SIMPLE-PS-S-007	IMS SIP PUBLISH Method	11	O	
SIMPLE-PS-S-008	NON-IMS SIP PUBLISH Method	11	O	
SIMPLE-PS-S-009	IMS SIP NOTIFY Method	11	O	
SIMPLE-PS-S-011	NON-IMS SIP NOTIFY Method	11	O	
SIMPLE-PS-S-012	Fetch Presence XDMS content	5.4	M	Presence_XDM-AU-S-001, Presence_XDM-AU-S-002, Presence_XDM-AU-S-003, Presence_XDM-AU-S-004, Presence_XDM-AU-S-005, Presence_XDM-AU-S-006, Presence_XDM-AU-S-007

Item	Function	Reference	Status	Requirement
SIMPLE-PS-S-013	Subscribe for Presence XDMS change event package	5.4	O	Presence_XDM-AU-S-001, Presence_XDM-AU-S-002, Presence_XDM-AU-S-003, Presence_XDM-AU-S-004, Presence_XDM-AU-S-005, Presence_XDM-AU-S-006, Presence_XDM-AU-S-007
SIMPLE-PS-S-014	Fetch Shared XDMS content	5.4	M	Shared_XDM-AU-S-001, Shared_XDM-AU-S-002, Shared_XDM-AU-S-003, Shared_XDM-AU-S-004, Shared_XDM-AU-S-005, Shared_XDM-AU-S-006, Shared_XDM-AU-S-007
SIMPLE-PS-S-015	Subscribe for Shared XDMS change event package	5.4	O	Shared_XDM-AU-S-001, Shared_XDM-AU-S-002, Shared_XDM-AU-S-003, Shared_XDM-AU-S-004, Shared_XDM-AU-S-005, Shared_XDM-AU-S-006, Shared_XDM-AU-S-007

Item	Function	Reference	Status	Requirement
SIMPLE-PS-S-016	Content Indirection of Presence Notification	5.4	O	
SIMPLE-PS-S-017	Direct Content of Presence Notification	5.4	O	
SIMPLE-PS-S-018	Watcher Information Subscriptions	5.4.4	M	SIMPLE-PS-S-005 OR SIMPLE-PS-S-006
SIMPLE-PS-S-019	Watcher Information Notifications	5.4.4	M	SIMPLE-PS-S-009 OR SIMPLE-PS-S-011
SIMPLE-PS-S-021	Partial Notifications	5.4	M	
SIMPLE-PS-S-022	Polite Blocking	5.4	M	

A.3 Watcher Information Subscriber

Item	Function	Reference	Status	Requirement
SIMPLE-WIS-C-001	IMS SIP SUBSCRIBE Method	11	O	
SIMPLE-WIS-C-002	NON-IMS SIP SUBSCRIBE Method	11	O	
SIMPLE-WIS-C-003	IMS SIP NOTIFY Method	11	O	
SIMPLE-WIS-C-004	NON-IMS SIP NOTIFY Method	11	O	
SIMPLE-WIS-C-005	Subscription for the watcher information template package	5.3	O	

A.4 RLS Server

Item	Function	Reference	Status	Requirement
SIMPLE-RLS-S-001	RL Subscription	5.5	M	RLS_XDM-AU-S-001, RLS_XDM-AU-S-002, RLS_XDM-AU-S-003, RLS_XDM-AU-S-004, RLS_XDM-AU-S-007, SIMPLE-RLS-S-004 OR SIMPLE-RLS-S-005

Item	Function	Reference	Status	Requirement
SIMPLE-RLS-S-002	RL Notifications	5.5	M	RLS_XDM-AU-S-001, RLS_XDM-AU-S-002, RLS_XDM-AU-S-003, RLS_XDM-AU-S-004, RLS_XDM-AU-S-007, SIMPLE-RLS-S-006 OR SIMPLE-RLS-S-007
SIMPLE-RLS-S-004	IMS SIP SUBSCRIBE Method	11	O	
SIMPLE-RLS-S-005	NON-IMS SIP SUBSCRIBE Method	11	O	
SIMPLE-RLS-S-006	IMS SIP NOTIFY Method	11	O	
SIMPLE-RLS-S-007	NON-IMS SIP NOTIFY Method	11	O	
SIMPLE-RLS-S-008	Partial Notifications	5.5	M	SIMPLE-RLS-S-006 OR SIMPLE-RLS-S-007
SIMPLE-RLS-S-009	Subscribe for RLS XDMS change event package	5.5	O	RLS_XDM-AU-S-001, RLS_XDM-AU-S-002, RLS_XDM-AU-S-003, RLS_XDM-AU-S-004, RLS_XDM-AU-S-005, RLS_XDM-AU-S-006, RLS_XDM-AU-S-007, RLS_XDM-AU-S-008, RLS_XDM-AU-S-009
SIMPLE-RLS-S-010	Fetch RLS XDMS content	5.5	M	RLS_XDM-AU-S-001, RLS_XDM-AU-S-002, RLS_XDM-AU-S-003, RLS_XDM-AU-S-004, RLS_XDM-AU-S-005, RLS_XDM-AU-S-006, RLS_XDM-AU-S-007, RLS_XDM-AU-S-008, RLS_XDM-AU-S-009

Item	Function	Reference	Status	Requirement
SIMPLE-RLS-S-011	Subscribe for Shared XDMS change event package	5.5	O	Shared_XDM-AU-S-001, Shared_XDM-AU-S-002, Shared_XDM-AU-S-003, Shared_XDM-AU-S-004, Shared_XDM-AU-S-005, Shared_XDM-AU-S-006, Shared_XDM-AU-S-007
SIMPLE-RLS-S-012	Fetch Shared XDMS content	5.5	M	Shared_XDM-AU-S-001, Shared_XDM-AU-S-002, Shared_XDM-AU-S-003, Shared_XDM-AU-S-004, Shared_XDM-AU-S-005, Shared_XDM-AU-S-006, Shared_XDM-AU-S-007

A.5 RLS Client

Item	Function	Reference	Status	Requirement
SIMPLE-RLS-C-001	RL Subscription	5.5	M	SIMPLE-RLS-C-004 OR SIMPLE-RLS-C-005
SIMPLE-RLS-C-002	RL Notifications	5.5	M	SIMPLE-RLS-C-006 OR SIMPLE-RLS-C-007
SIMPLE-RLS-C-004	IMS SIP SUBSCRIBE Method	11	O	

Item	Function	Reference	Status	Requirement
SIMPLE-RLS-C-005	NON-IMS SIP SUBSCRIBE Method	11	O	
SIMPLE-RLS-C-006	IMS SIP NOTIFY Method	11	O	
SIMPLE-RLS-C-007	NON-IMS SIP NOTIFY Method	11	O	

A.6 Watcher

Item	Function	Reference	Status	Requirement
SIMPLE-WATCH-C-001	Presence Data Model	10.1	M	
SIMPLE-WATCH-C-002	Presence Subscription	5.2.1	M	SIMPLE-WATCH-C-009 OR SIMPLE-WATCH-C-010
SIMPLE-WATCH-C-003	RLS Subscription	5.2.1	O	SIMPLE-WATCH-C-009 OR SIMPLE-WATCH-C-010
SIMPLE-WATCH-C-004	Presence Notifications	5.2.1	M	SIMPLE-WATCH-C-011 OR SIMPLE-WATCH-C-012
SIMPLE-WATCH-C-005	Partial Notification	5.4.2.2	O	SIMPLE-WATCH-C-009 OR SIMPLE-WATCH-C-010
SIMPLE-WATCH-C-006	Content Indirection	5.1.1.2.1	O	
SIMPLE-WATCH-C-07	Rich Presence Information defined in [RFC4480]	5.1	O	
SIMPLE-WATCH-C-008	Presence-based Location Object [PIDF-LO]	5.1	O	
SIMPLE-WATCH-C-009	IMS SIP SUBSCRIBE Method	11	O	
SIMPLE-WATCH-C-010	NON-IMS SIP SUBSCRIBE Method	11	O	
SIMPLE-WATCH-C-011	IMS SIP NOTIFY Method	11	O	

Item	Function	Reference	Status	Requirement
SIMPLE-WATCH-C-012	NON-IMS SIP NOTIFY Method	11	O	

A.7 XDM Client

Item	Function	Reference	Status	Requirement
Presence_SIMPLE-XDMC-C-001	Mandatory XDMC functions	5.6	M	XDM_Core: MCF AND SHARED_XDM:MCF AND RLS_XDM:MCF AND PRESENCE_XDM:MCF
Presence_SIMPLE-XDMC-C-002	Optional XDMC functions	5.6	O	XDM_Core:OCF AND SHARED_XDM:OCF RLS_XDM:OCF AND PRESENCE_XDM:OCF

A.8 Presence XDMS

Item	Function	Reference	Status	Requirement
Presence_SIMPLE-PresenceXDMS-S-001	Mandatory Presence XDMS functions	5.7	M	XDM_Core:MSF AND PRESENCE_XDM:MSF
Presence_SIMPLE-PresenceXDMS-S-002	Optional Presence XDMS functions	5.7	O	XDM_Core:OSF AND PRESENCE_XDM:OSF

A.9 RLS XDMS

Item	Function	Reference	Status	Requirement
Presence_SIMPLE-RLSXDMS-S-001	Mandatory RLS XDMS functions	5.8	M	XDM_Core:MSF AND RLS_XDM:MSF
Presence_SIMPLE-RLSXDMS-S-002	Optional RLS XDMS functions	5.8	O	XDM_Core:OSF AND RLS_XDM:OSF

Appendix B. Presence Client Provisioning (Normative)

This Appendix specifies the parameters that are needed by the presence client. Existing parameters in [Provisioning Content] and [OMA-DM-v1-1-2] are re-used; those without corresponding parameters are defined and to be registered in OMNA through OMA official ergistration process.

The Management Object (MO) for the OMA SIMPLE Presence 1.0 enabler is defined in [PRESMO]. The MO MAY be used for initial provisioning of parameters when the DM Profile is to be used (as specified on [OMA-DM-v1-2]), and the MO SHOULD be used for continuous provisioning of parameters according to [OMA-DM-v1-1-2] or [OMA-DM-v1-2], if required by the service provider to update service configurations.

B.1 Presence Client provisioning parameters

The following table lists the parameters available in an instance of the Presence Application Characteristic

Parameter Name	Man / Opt	Instances	Default
Standard Application Characteristic fields as defined in [Provisioning Content]			
APPID	Mandatory	1	“ap0002”
PROVIDER-ID	Optional	0 or 1	none
APPREF	Mandatory	1	None
TO-APPREF	Mandatory	1 or more	none
NAME	Optional	0 or 1	none
TO-NAPID	Optional	0 or more	none
Application Characteristic fields specifically required for the Presence Enabler			
CLIENT-OBJ-DATA-LIMIT	Mandatory	1	none
CONTENT-SERVER-URI	Optional	0 or 1	none
SOURCE-THROTTLE-PUBLISH	Optional	0 or 1	none
MAX-NUMBER-OF-PRESENCE-SUBSCRIPTIONS	Optional	0 or 1	none
MAX-NUMBER-OF-SUBSCRIPTIONS-IN-PRESENCE-LIST	Optional	0 or 1	none
SERVICE-URI-TEMPLATE	Optional	0 or 1	none

The Presence Application Characteristics file for the OMA SIMPLE Presence 1.0 enabler is defined in [PRESAC].

Appendix C. Presence Signalling Flows (Informative)

The following signalling flows illustrate the implementation of the relevant use cases, derived from the [PRESREQ]. The supported headers of the SIP methods used in order to perform those functions are defined in section 11 and the body of the messages, when required, in section 11.

C.1 Subsystem Collaboration

This section presents message flow examples for the implementation of the basic mechanisms of the SIMPLE Presence Service.

C.1.1 Signalling flows for publishing presence information

C.1.1.1 Publishing Presence Information

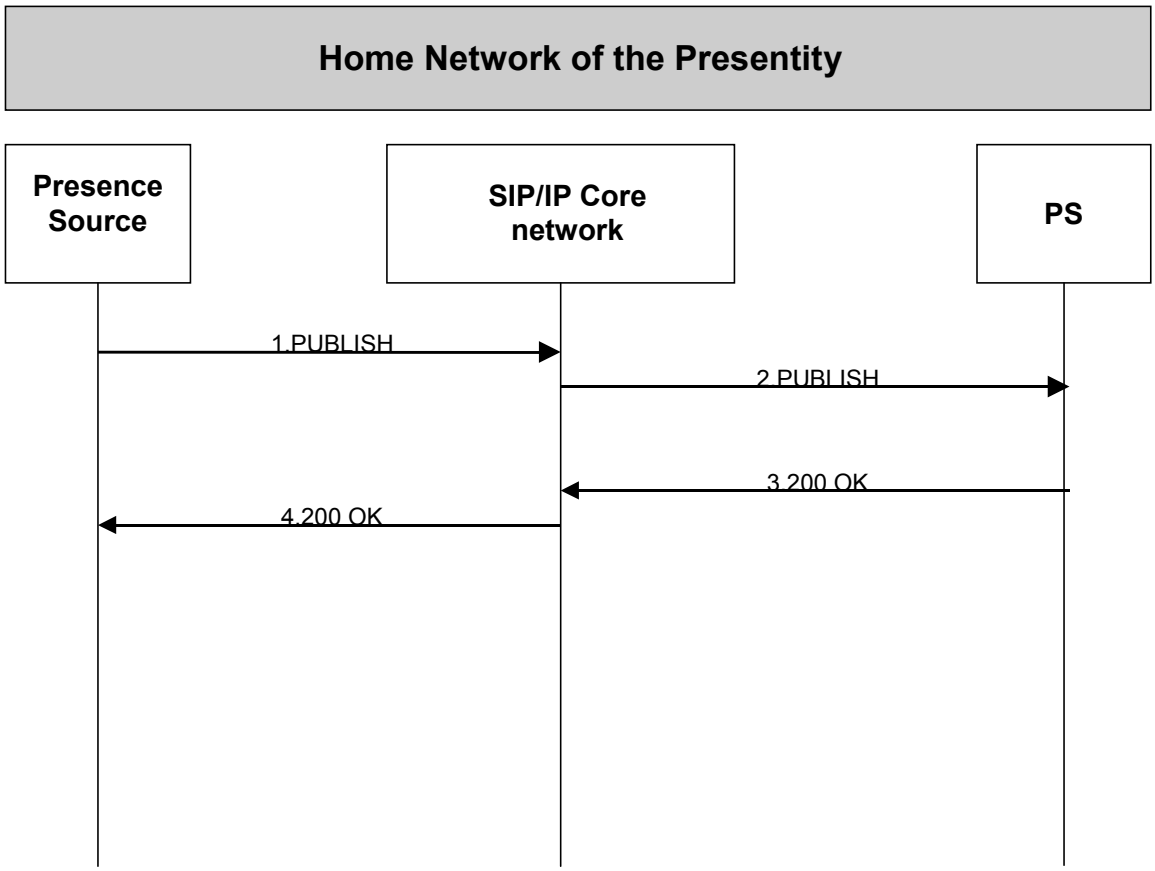

Figure 6- Publishing Presence Information

1. The Presence Source generates a SIP PUBLISH request, which contains a presence document. The means for the Presence Source to compose this presence document is outside the scope of this specification.
2. The SIP/IP Core network routes the request to the correct PS.
3. The PS authorises the presence publication, and checks the information the message contains. The PS then processes the presence information and sends a SIP 200 OK response back to Presence Source.
4. The SIP/IP Core network forwards the response back to the Presence Source.

C.1.1.2 Publishing presence information on behalf of another presentity

C.1.1.2.1 Successful attempt

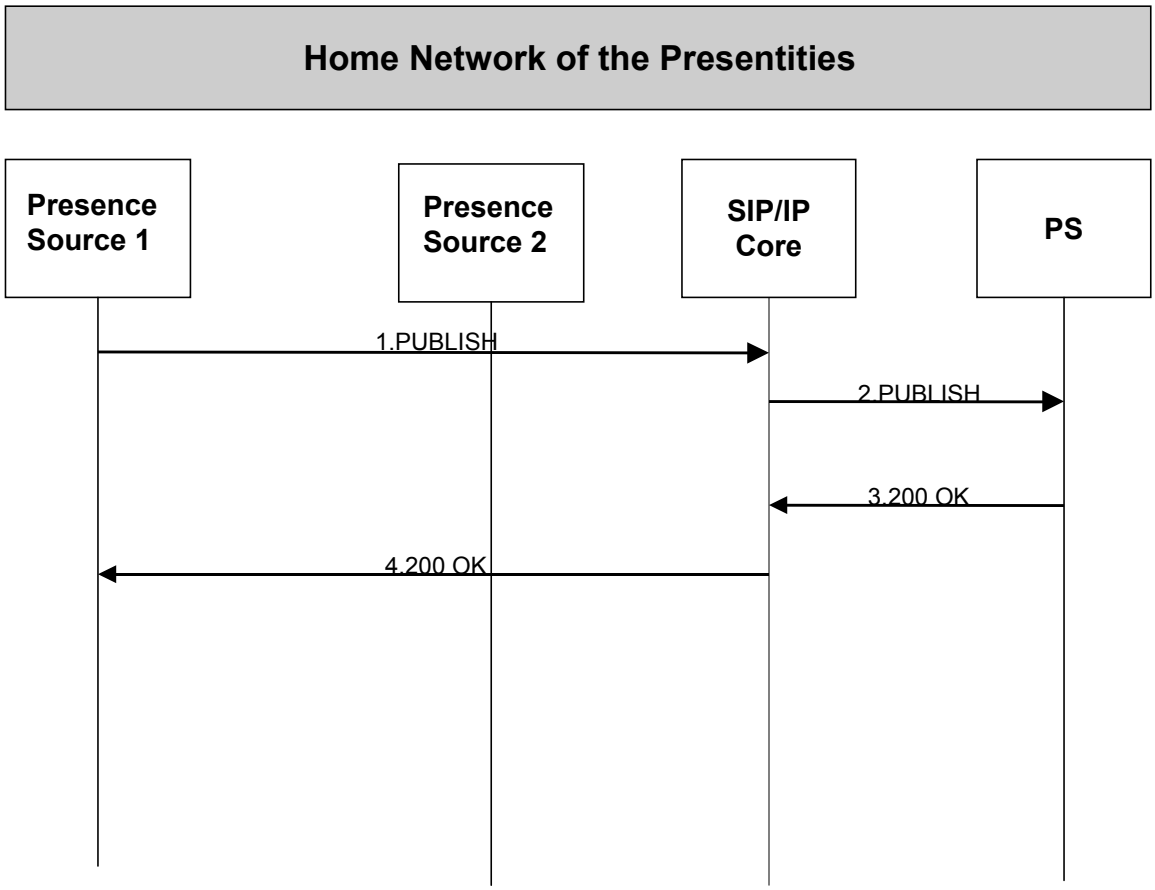

Figure 7 - Aggregating published presence information from multiple sources

1. Presence Source1 generates a SIP PUBLISH request, which contains presence information relating to Presence Source2's presentity. The means for the Presence Source1 to compose the presence information is outside the scope of this specification.

2. The SIP/IP Core network forwards the SIP PUBLISH request to the appropriate PS.

3. The PS authorises the publication attempt and checks the content of the request. The PS then composes the presence information to the presence document of Presence Source2's presentity. The PS sends a SIP 200 OK response back to the SIP/IP Core network.

4. The SIP/IP Core network forwards the SIP 200 OK response back to the Presence Source1.

C.1.1.2.2 Unsuccessful attempt

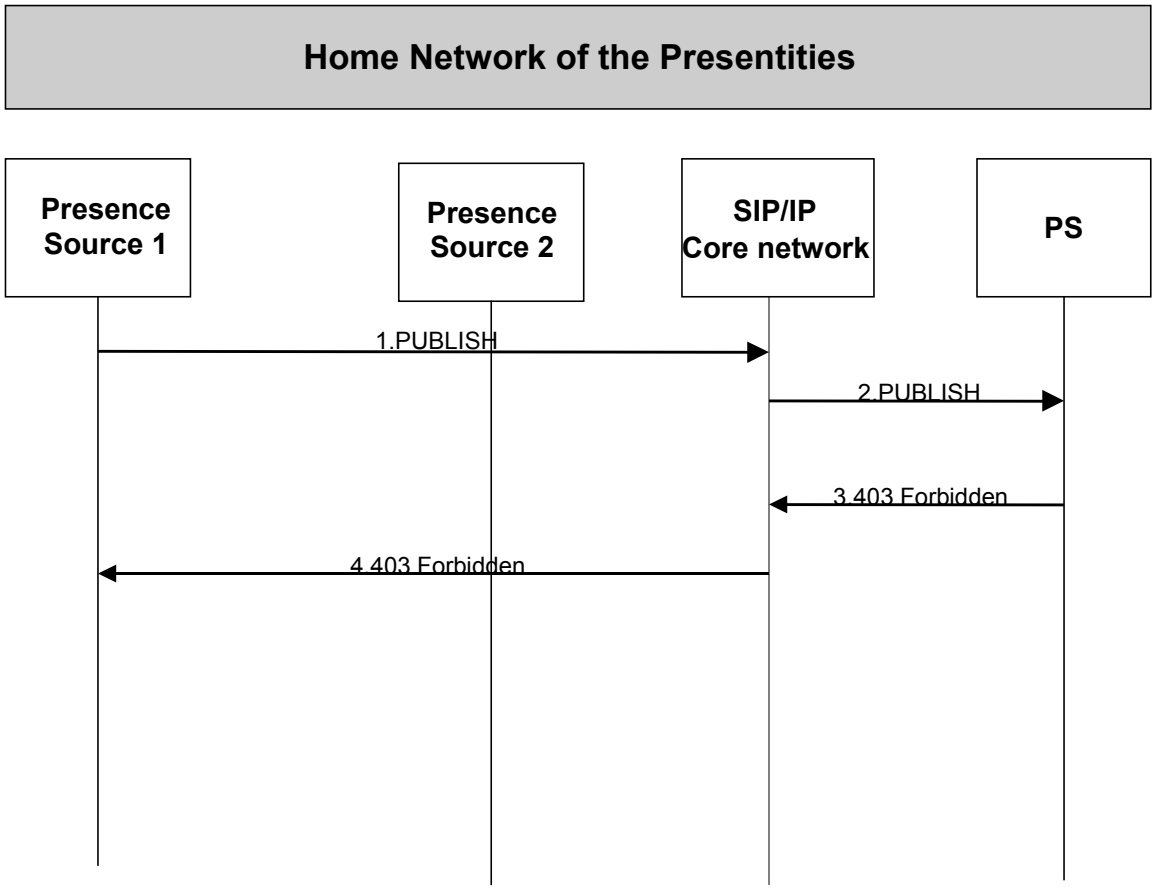

Figure 8 - Aggregating published presence information from multiple sources

1. Presence Source1 generates a SIP PUBLISH request, which contains presence information relating to Presence Source2's presentity. The means for the Presence Source1 to compose the presence information is outside the scope of this specification.
2. The SIP/IP Core network forwards the SIP PUBLISH request to the appropriate PS.
3. The PS does not authorise the request and sends a SIP 403 Forbidden response back to the SIP/IP Core network.
4. The SIP/IP Core network forwards the SIP 403 Forbidden response back to the Presence Source1.

C.1.1.2.3 Aggregating published presence information from multiple sources

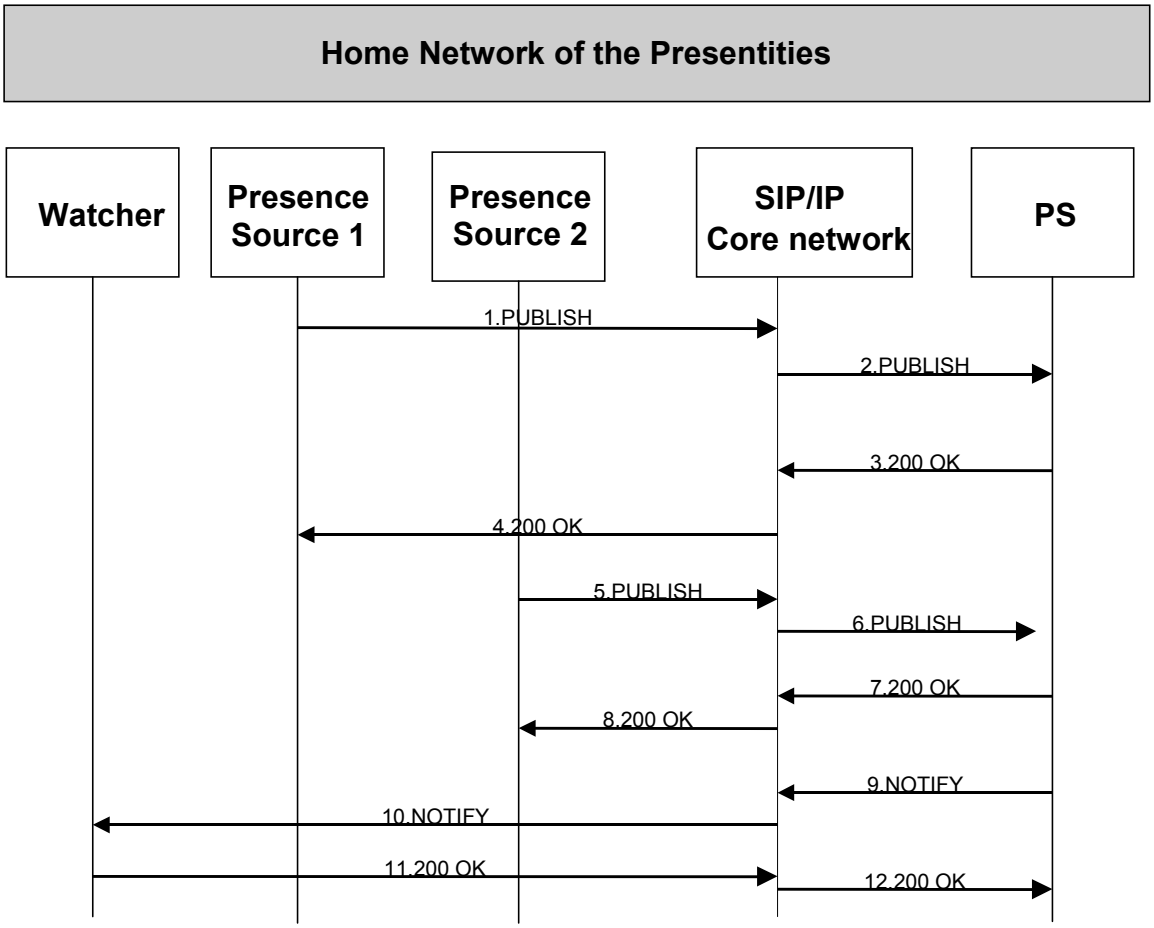

Figure 9- Aggregating published presence information from multiple sources

1. Presence Source1 generates a SIP PUBLISH request, which contains the presence information Presence Source1 wishes to publish on behalf of the presentity.
2. The SIP/IP Core network forwards the SIP PUBLISH request to the appropriate PS.
3. The PS authorises the publication attempt and checks the content of the request. The PS then composes the presence information to the presentity's presence document. The PS sends a SIP 200 OK response back to the SIP/IP Core.
4. The SIP/IP Core network forwards the SIP 200 OK response back to the Presence Source1.
5. Presence Source2 generates a SIP PUBLISH request, which contains the presence information Presence Source2 wishes to publish on behalf of the presentity.
6. The SIP/IP Core network forwards the SIP PUBLISH request to the appropriate PS.
7. The PS authorises the publication attempt and checks the content of the request. The PS then composes the presence information to the presentity's presence document aggregating with the information Presence Source1 has published. The PS sends a SIP 200 OK response back to the SIP/IP Core.
8. The SIP/IP Core network forwards the SIP 200 OK response back to the Presence Source2.

9. The PS determines which authorised watchers are entitled to receive the updates of the presence information for this presentity. For each appropriate watcher, the PS sends a SIP NOTIFY request that contains the aggregated presence information from Presence Source1 and Presence Source2. The SIP NOTIFY request is sent along the path of the SUBSCRIBE dialog to the SIP/IP Core network of the watcher.
10. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher.
11. The watcher acknowledges the SIP NOTIFY request with a SIP 200 OK response to its SIP/IP Core network.
12. The SIP/IP Core network of the watcher forwards the SIP 200 OK response to the PS.

C.1.2 Signalling flows for watchers subscribing to presence event notification

C.1.2.1 Subscribing to Presence Information state changes - Proactive Authorization

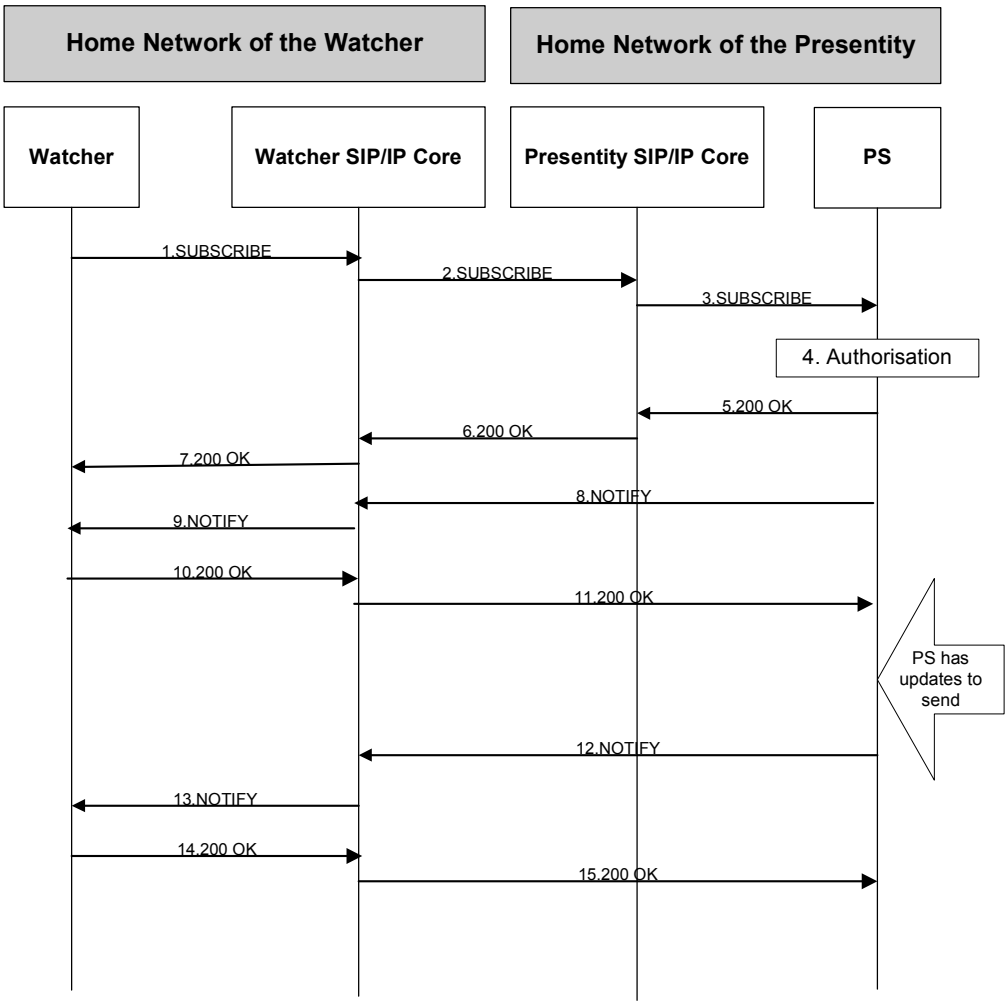

Figure 10 - Subscribing to presence information state changes (watcher and presentity are in different networks) – Proactive Authorization

1. A watcher wishes to watch a presentity's presence information, or certain parts of the presentity's presence information. To initiate a subscription, the watcher sends a SIP SUBSCRIBE request for the presence event package including an indication of the duration this subscription should last. The SIP SUBSCRIBE request may also include an indication of the watcher's capability to handle partial notifications.

2. The SIP/IP Core network of the watcher resolves the address of the presentity and forwards the request to the SIP/IP Core network of the presentity
3. The SIP/IP Core network of the presentity routes the SUBSCRIBE request to the correct PS.
4. The PS performs the necessary authorisation checks on the originator to ensure it is allowed to watch the presentity.

NOTE: In the case where the privacy/authorisation checks fail, then a negative acknowledgement is sent to the watcher.

5. Once all privacy conditions are met, the PS issues a SIP 200 OK to the SIP/IP Core.
6. The SIP/IP Core network of the presentity forwards the response to the SIP/IP Core network of the watcher.
7. The SIP/IP Core network of the watcher forwards the response to the watcher.
8. As soon as the PS sends a 200 OK response to accept the subscription, it sends a SIP NOTIFY request including the current full state of the presentity's tuples that the watcher has subscribed and been authorised to. The SIP NOTIFY request is sent to the watcher SIP/IP Core network. Further notifications sent by the PS may either contain the complete set of presence information, or only those tuples that have changed since the last notification if the watcher has indicated the capability to process partial notifications.
9. The SIP/IP Core network of the watcher forwards the SIP NOTIFY request to the watcher.
10. The watcher acknowledges the receipt of the SIP NOTIFY request with a SIP 200 OK response sent to its SIP/IP Core network.
11. The SIP/IP Core network of the watcher forwards the SIP 200 OK response to the PS.
12. Upon the presence information for the presentity changes (the means how the presence information changes are outside the scope for this use case), the PS determines which authorized watchers are entitled to receive notifications. For each appropriate watcher, the PS sends a SIP NOTIFY request that contains the full or partial updates to the presence information. The SIP NOTIFY request is sent along the path of the SUBSCRIBE dialog to the SIP/IP Core network of the watcher.
13. The watcher’s SIP/IP core forwards the SIP NOTIFY request to the watcher.
14. The watcher acknowledges the SIP NOTIFY response with a SIP 200 OK response to its SIP/IP core network.
15. The SIP/IP core network of the watcher forwards the SIP 200 OK response to the PS.

NOTE: Steps 2 and 3 as well as 5 and 6 are combined if the watcher is in the same domain as the presentity.

C.1.2.2 Fetching Presence Information state – Proactive authorization

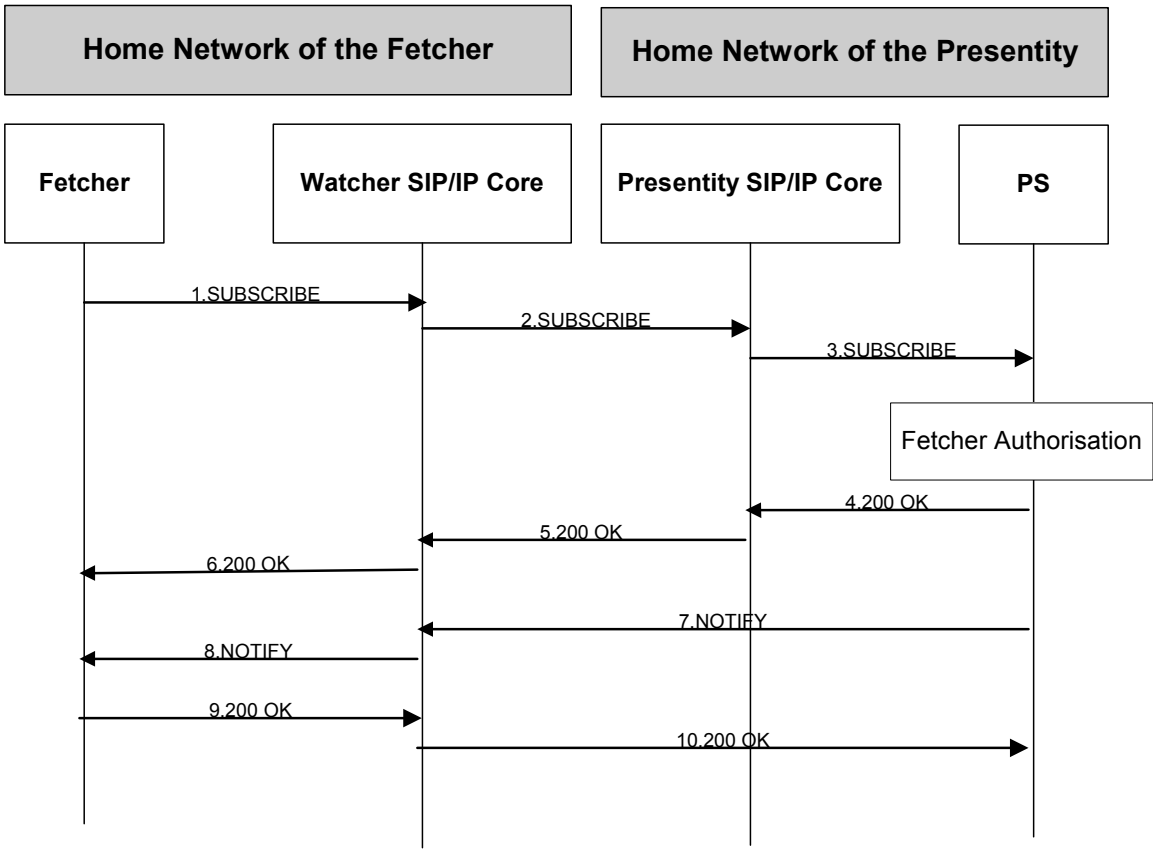

Figure 11 - Fetching presence information state (fetcher and presentity are in different networks)

A watcher requests presence information of a certain presentity from the PS, acting as a fetcher. For the remaining use case, watcher will be used uniformly.

1. The watcher requests presence information of the presentity using a SIP SUBSCRIBE request by setting the "Expires" header field to zero, as defined in [RFC3265].

2. The watcher's SIP/IP Core network resolves the address of the SIP/IP Core network of the presentity and forwards the request.

3. The SIP/IP Core network forwards the SIP SUBSCRIBE request to the appropriate PS.

4. The PS performs the necessary authorization checks on the originator to ensure it is allowed to request presence information of the presentity. Assuming all privacy conditions are met, the PS issues a SIP 200 OK response to the SIP/IP Core network of the presentity.

5. The SIP/IP Core network of the presentity forwards the response to the SIP/IP Core network of the watcher.

6. The SIP/IP Core network of the watcher forwards the SIP 200 OK response to the watcher

7. As soon as the PS sends a SIP 200 OK response to accept the request, it sends a SIP NOTIFY request with the current full state of the presentity's tuples that the watcher has requested and been authorized to. The SIP NOTIFY request is sent along the path of the SUBSCRIBE dialog to the SIP/IP Core network of the watcher.

8. The SIP/IP Core network of the watcher forwards the SIP NOTIFY request to the watcher.

9 . The watcher acknowledges the receipt of the SIP NOTIFY request with a SIP 200 OK responseto the SIP/IP Core network of the watcher.

10. The watcher's SIP/IP Core network forwards the SIP 200 OK response to the PS.

NOTE: Steps 2 and 3 as well as 5 and 6 are combined if the watcher is in the same domain as the presentity.

C.1.2.3 Subscribing to Presence Information state changes - Reactive Authorization

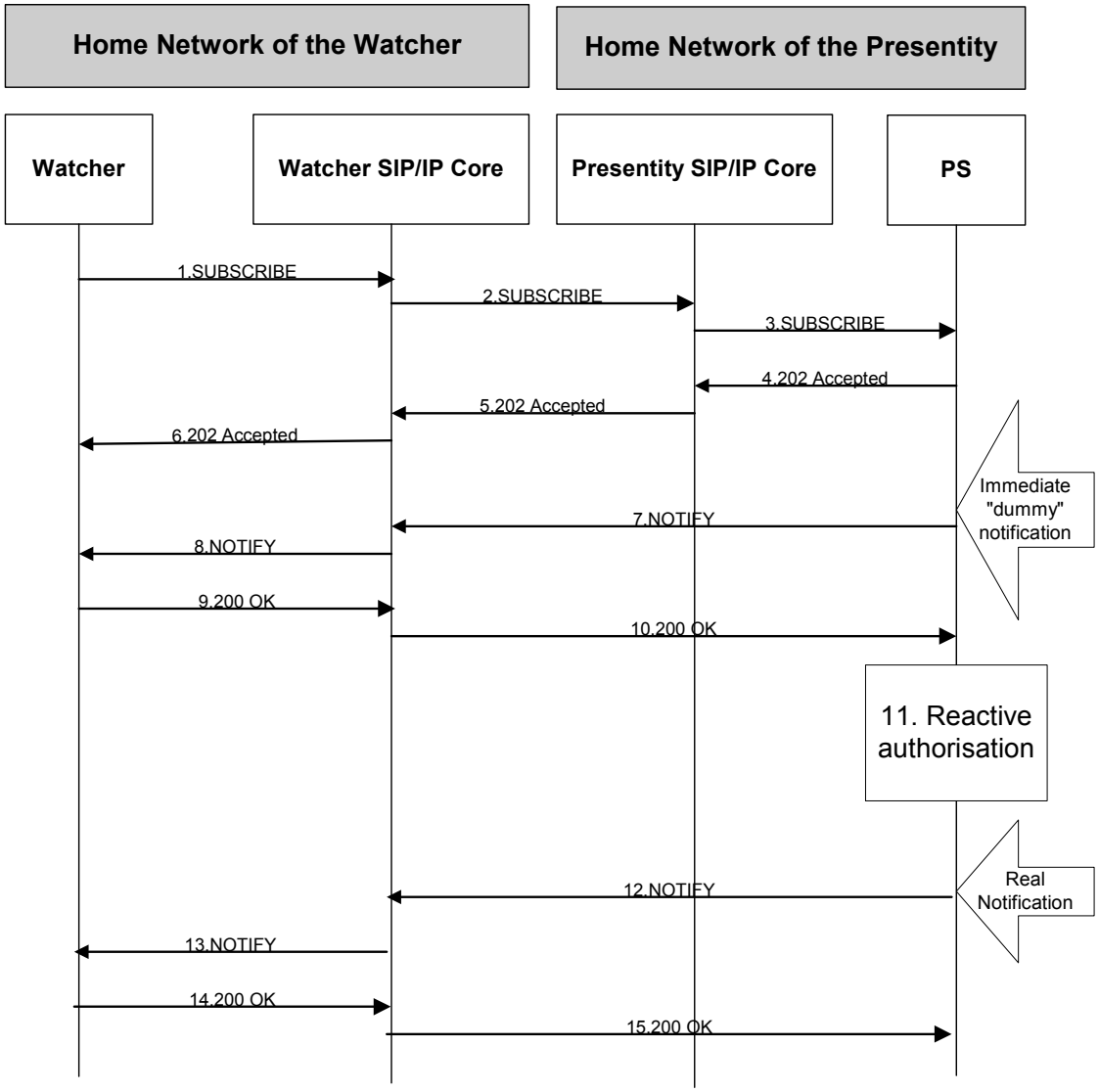

Figure 12 - Subscribing to presence information state changes (watcher and presentity are in different networks) - Reactive Authorisation

1. A watcher wishes to watch a presentity's presence information, or certain parts of the presentity's presence information. To initiate a subscription, the watcher sends a SIP SUBSCRIBE request for the presence event package including an indication of the duration this subscription should last. The SIP SUBSCRIBE request may also include an indication of the watcher's capability to handle partial notifications.

2. The SIP/IP Core network of the watcher resolves the address of the presentity and forwards the request to the SIP/IP Core network of the presentity.

3. The SIP/IP Core network of the presentity routes the SUBSCRIBE request to the correct PS
4. The PS acknowledges the request with a SIP 202 Accepted response sent to the SIP/IP Core network of the presentity.
5. The SIP/IP Core network of the presentity forwards the response to the SIP/IP Core network of the watcher
6. The SIP/IP Core network of the watcher forwards the response to the watcher..
7. As soon as the PS sends a SIP 202 Accepted response to accept the subscription, it sends a SIP NOTIFY request as mandated by [RFC3265]. At this time, the presence information may be inaccurate or not fully available for the presentity. However a "dummy" SIP NOTIFY request must be sent, with a valid neutral or empty presence information and a valid Subscription-State header field (set to "pending") for the time being.
8. The SIP/IP Core network of the watcher forwards the SIP NOTIFY request to the watcher
9. The watcher acknowledges the receipt of the SIP NOTIFY request with a SIP 200 OK response sent to its SIP/IP Core network.
10. The SIP/IP Core network of the watcher forwards the SIP 200 OK response to the PS.
11. The PS authorizes the watcher, after the presentity modifies the Subscription Authorization Policy (see section 5.4.3.2).
12. The PS issues another SIP NOTIFY request, to amend the neutral state known to the watcher with valid presence information.
13. The watcher's SIP/IP core forwards the SIP NOTIFY request to the watcher.
14. The watcher acknowledges the SIP NOTIFY response with a SIP 200 OK response to its SIP/IP core network
15. The SIP/IP core network of the watcher forwards the SIP 200 OK response to the PS.

NOTE 1: Steps 2 and 3 as well as 5 and 6 are combined if the watcher is in the same domain as the presentity.

NOTE 2: If the immediate presence information is accurate, then there is no need for another notification (shown in steps 12-15) until presence information state changes. In fact, the PS may choose to best describe the presence information as known in the immediate notification, and if upon completing the required steps to grant the real presence information, it matches the information previously sent, there is no need for the second SIP NOTIFY request.

C.1.2.4 Receiving a Presence Notification for an Existing Subscription

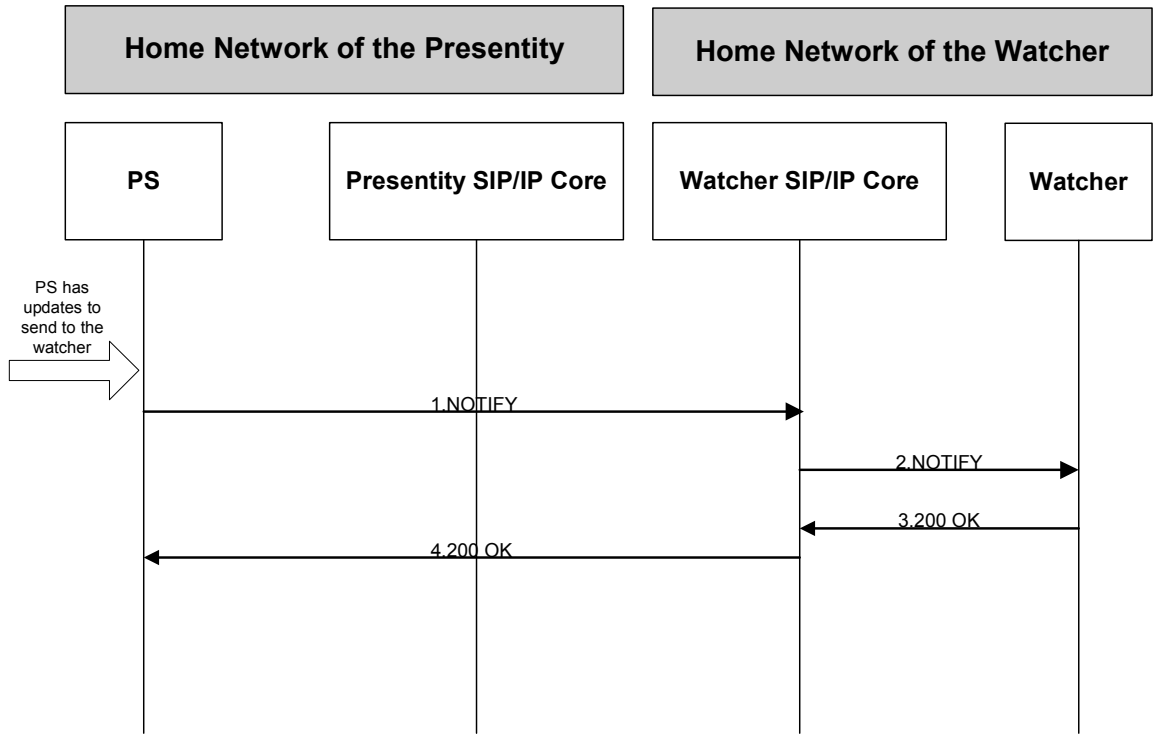

Figure 13- Receiving a presence notification

1. The PS determines which authorised watchers are entitled to receive the updates of the presence information for this presentity. For each appropriate watcher, the PS generates a SIP NOTIFY request that contains either the full or partial updates of the presence information. The SIP NOTIFY request is sent inside the existing dialog created by the SIP SUBSCRIBE request to the SIP/IP Core network of the watcher.
2. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher.
3. The watcher acknowledges the SIP NOTIFY request with a SIP 200 OK response to its SIP/IP Core network.
4. The SIP/IP Core network of the watcher forwards the SIP 200 OK response to the PS.

C.1.2.5 Partial Notifications

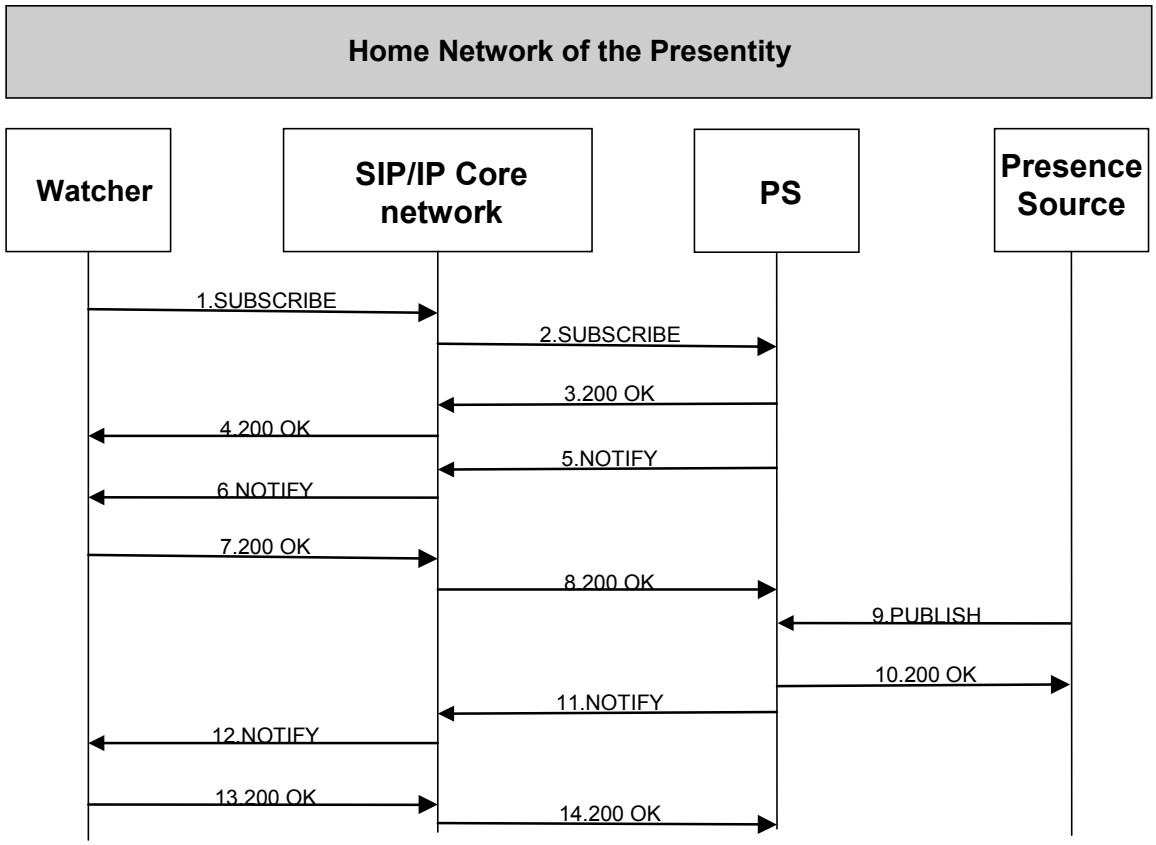

Figure 14 -Partial Notifications Information Flow

1. A watcher sends a SIP SUBSCRIBE request to the PS indicating the support for the default Presence Information Data Format defined in [PIDF] and the partial PIDF defined in [PARFORMAT]. The watcher also indicates the support for the partial notification mechanism according to [PARNOT].
2. The SIP/IP Core network forwards the SIP SUBSCRIBE request to the PS.
3. The PS authorizes the subscription and sends a SIP 200 OK response to the SIP/IP Core network.
4. The SIP/IP Core network forwards the SIP 200 OK response to the watcher.
5. The PS, based on the watcher's indication supporting partial notification mechanism, generates a SIP NOTIFY request, which includes a full state presence document formulated according to [PARNOT]. The SIP NOTIFY request is forwarded to the SIP/IP Core network.
6. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher.
7. The watcher sends a SIP 200 OK response to the SIP/IP Core network to acknowledge the SIP NOTIFY request.
8. The SIP/IP Core network forwards the SIP 200 OK response to the PS.
9. After some time the presentity's presence information changes (e.g. a tuple changes its <status>) so a Presence Source publishes the new state to the PS by generating a SIP PUBLISH request.
10. The PS acknowledges the SIP PUBLISH request with a SIP 200 OK response.

11. The PS generates a NOTIFY request which includes a partial presence document formulated according to [PARFORMAT] showing only the changed presence information.

12. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher.

13. The watcher acknowledges the SIP NOTIFY request with a SIP 200 OK response.

14. The SIP/IP Core network forwards the SIP 200 OK response to the PS.

NOTE: If the watcher and the presentity reside at different domains the SIP/IP core of the watcher will perform address resolution on the address of the presentity to foward the SUBSCRIBE request to the SIP/IP core of the presentity. Then the SIP/IP core of the presentity will route the SUBSCRIBE request to the PS. (See step 2 and 3 as well as 5 and 6 in Section 12.1.2.1.)

C.1.2.6 Expiry of published presence information

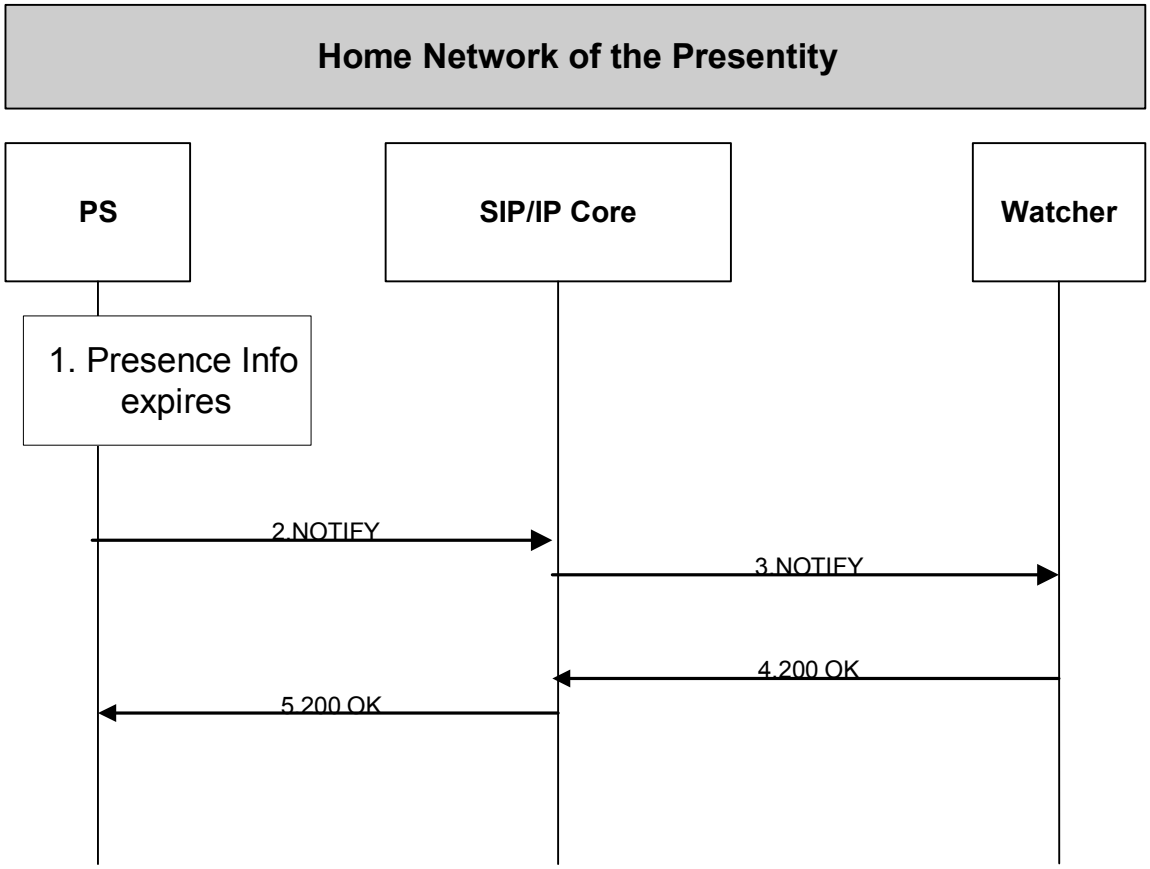

Figure 15- Expiry of published presence information

1. The lifetime of some presence information elapses and there is no refreshing transaction to update the lifetime of this presence information.

2. The PS issues a SIP NOTIFY request including the updated presence information.

3. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher.

4. The watcher sends a 200 OK response to the SIP/IP Core network to acknowledge the SIP NOTIFY request.

5. The SIP/IP Core network forwards the 200 OK response to the PS.

C.1.2.7 Subscription Authorization Failure

A presentity can deny a subscription request by either rejecting the request outright (so called “blocking”), or accepting the request but providing possibly inaccurate presence information (so called “polite blocking”).

C.1.2.7.1 Blocking

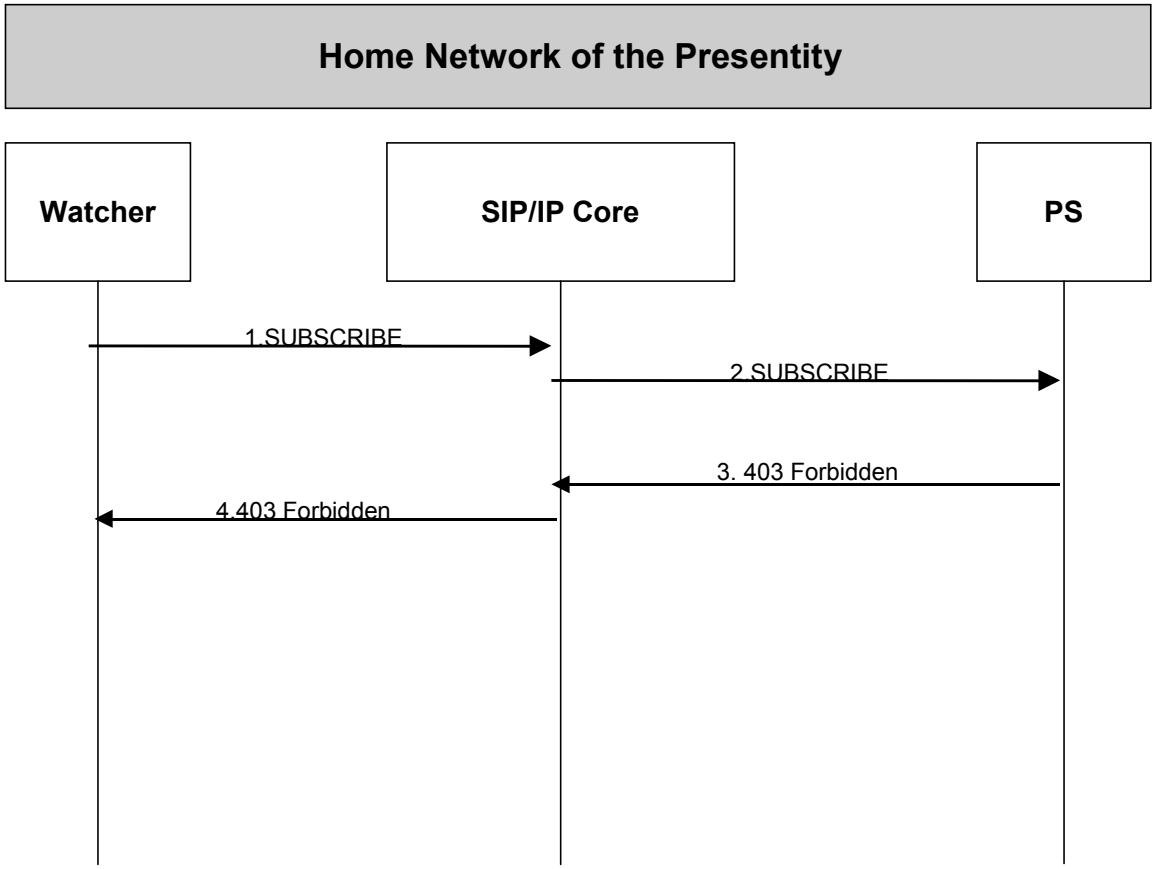

Figure 16- Blocking

1. A watcher wishing to subscribe to presence information about a presentity, sends a SUBSCRIBE request to the SIP/IP Core network.
2. The SIP/IP Core network forwards the SIP SUBSCRIBE request to the appropriate PS.
3. The PS performs a subscription authorization check on the watcher to verify whether it is allowed to watch the presentity. After applying the subscription authorization policies of the presentity, the PS determines to reject the subscription request. The PS sends either a SIP 403 Forbidden response to the SIP/IP Core network.
4. The SIP/IP Core network forwards the SIP 403 Forbidden response to the watcher.

NOTE: If the watcher and the presentity reside at different domains the SIP/IP core of the watcher will perform address resolution on the address of the presentity to foward the SUBSCRIBE request to the SIP/IP core of the presentity. Then the SIP/IP core of the presentity will route the SUBSCRIBE request to the PS. (See step 2 and 3 as well as 5 and 6 in Appendix D.1.2.1.)

C.1.2.7.2 Polite Blocking

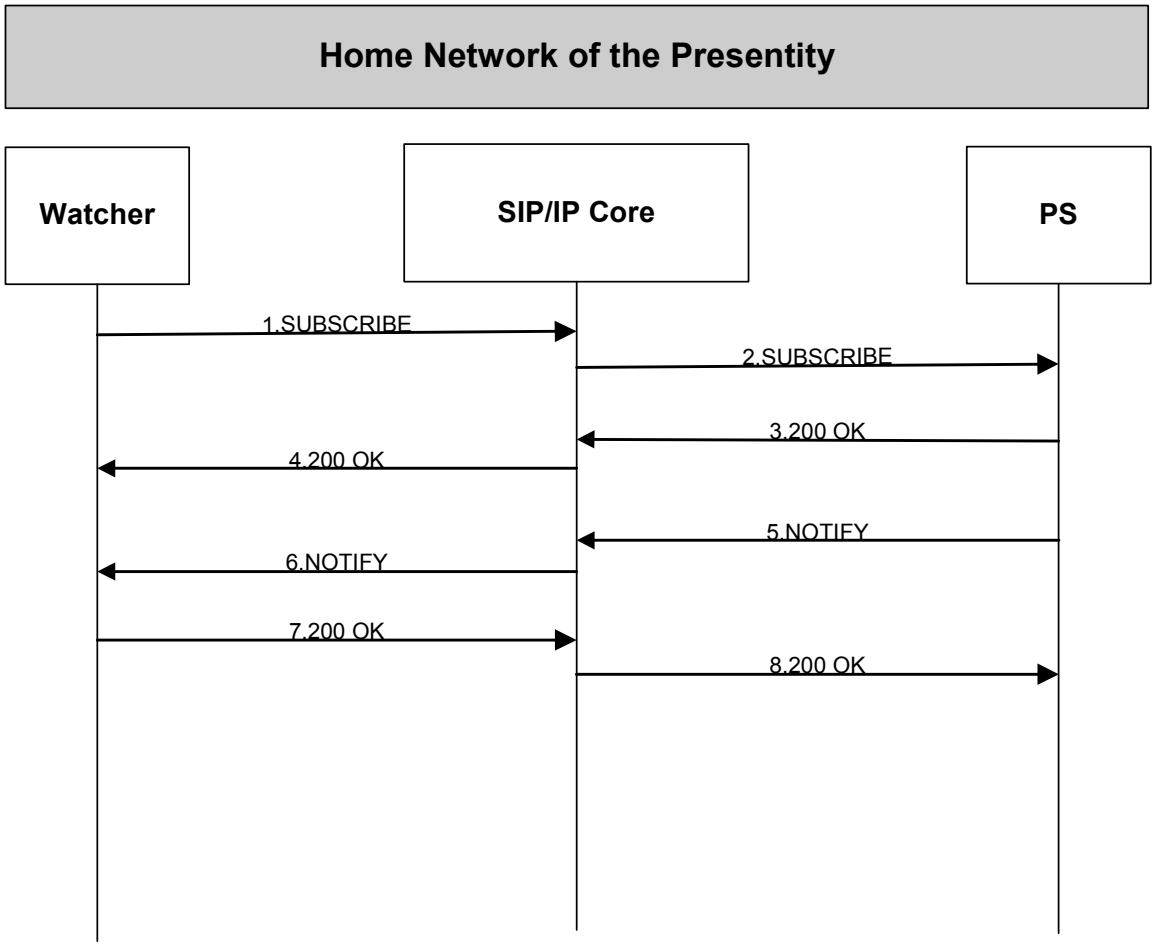

Figure 17- Polite Blocking

1. A watcher wishing to subscribe to presence information about a presentity, sends a SIP SUBSCRIBE request to the SIP/IP Core network.
2. The SIP/IP Core network forwards the SIP SUBSCRIBE request to the appropriate PS.
3. The PS performs a subscription authorization check on the watcher to verify whether it is allowed to watch the presentity. After applying the subscription authorization policies of the presentity, the PS determines to reject the subscription request but give the appearance that the request has been granted (so called "polite blocking") see section 5.4.3.2.1. The PS sends a 200 OK to the SIP/IP Core.
4. The SIP/IP Core network forwards the SIP 200 OK response to the watcher.
5. As soon as the PS sends the SIP 200 OK response, it sends a SIP NOTIFY request with the appropriate presence information as defined by the presence privacy policy.
6. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher.
7. The watcher acknowledges the SIP NOTIFY request with a SIP 200 OK response.
8. The SIP/IP Core network forwards the SIP 200 OK response to the appropriate PS.

NOTE: If the watcher and the presentity reside at different domains the SIP/IP core of the watcher will perform address resolution on the address of the presentity to foward the SUBSCRIBE request to the SIP/IP core of the presentity. Then

the SIP/IP core of the presentity will route the SUBSCRIBE request to the PS. (See step 2 and 3 as well as 5 and 6 in Section 12.1.2.1.)

C.1.2.8 Subscription Filters

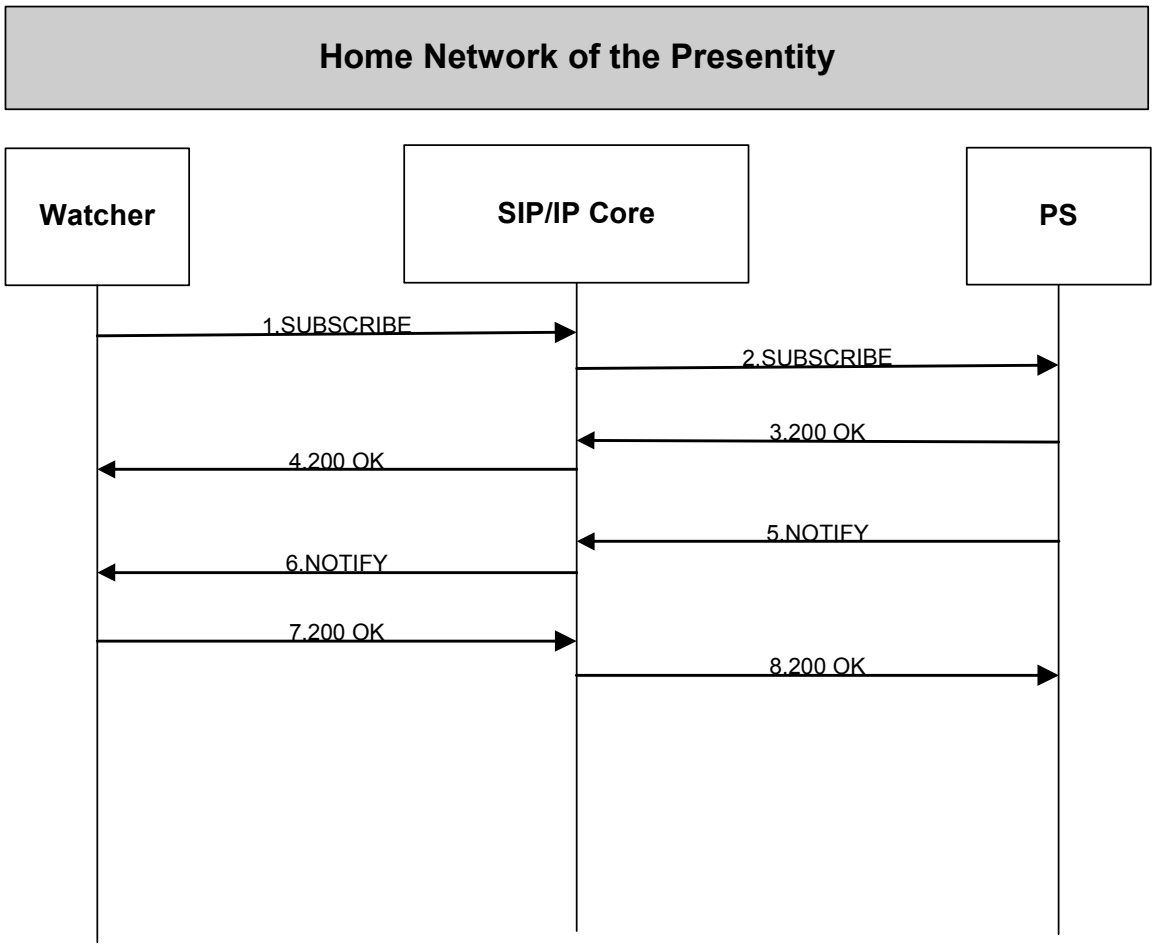

Figure 18 - Subscription Filters

In this example, a presentity has a presence document that includes two presence tuples: one for Instant Messaging (IM) and another for gaming services.

1. A watcher sends a SIP SUBSCRIBE request to the PS requesting the presence information related to all the messaging applications (e.g. MMS, SMS, IM) of the presentity. This is done by including a filter in the body of the SIP SUBSCRIBE request according to [RFC4660] and [RFC4661].

2. The SIP/IP Core network forwards the SIP SUBSCRIBE request to the PS.

3. The PS authorizes the subscription and interprets the subscription filter and sends a SIP 200 OK response to the SIP/IP Core network indicating that the subscription has been accepted and the subscription filter understood.

4. The SIP/IP Core network forwards the SIP 200 OK response to the watcher.

5. The PS sends a SIP NOTIFY request to the the SIP/IP Core network including only the Instant Messaging related tuple that was requested by the watcher's subscription filter.

6. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher.

7. The watcher acknowledges the SIP NOTIFY request with a SIP 200 OK response.

8. The SIP/IP Core network forwards the SIP 200 OK response to the PS.

NOTE: If the watcher and the presentity reside at different domains the SIP/IP core of the watcher will perform address resolution on the address of the presentity to foward the SUBSCRIBE request to the SIP/IP core of the presentity. Then the SIP/IP core of the presentity will route the SUBSCRIBE request to the PS. (See step 2 and 3 as well as 5 and 6 in Section 12.1.2.1.)

C.1.3 Signalling flows for watchers canceling a subscription

C.1.3.1 Watcher Initiated Canceling

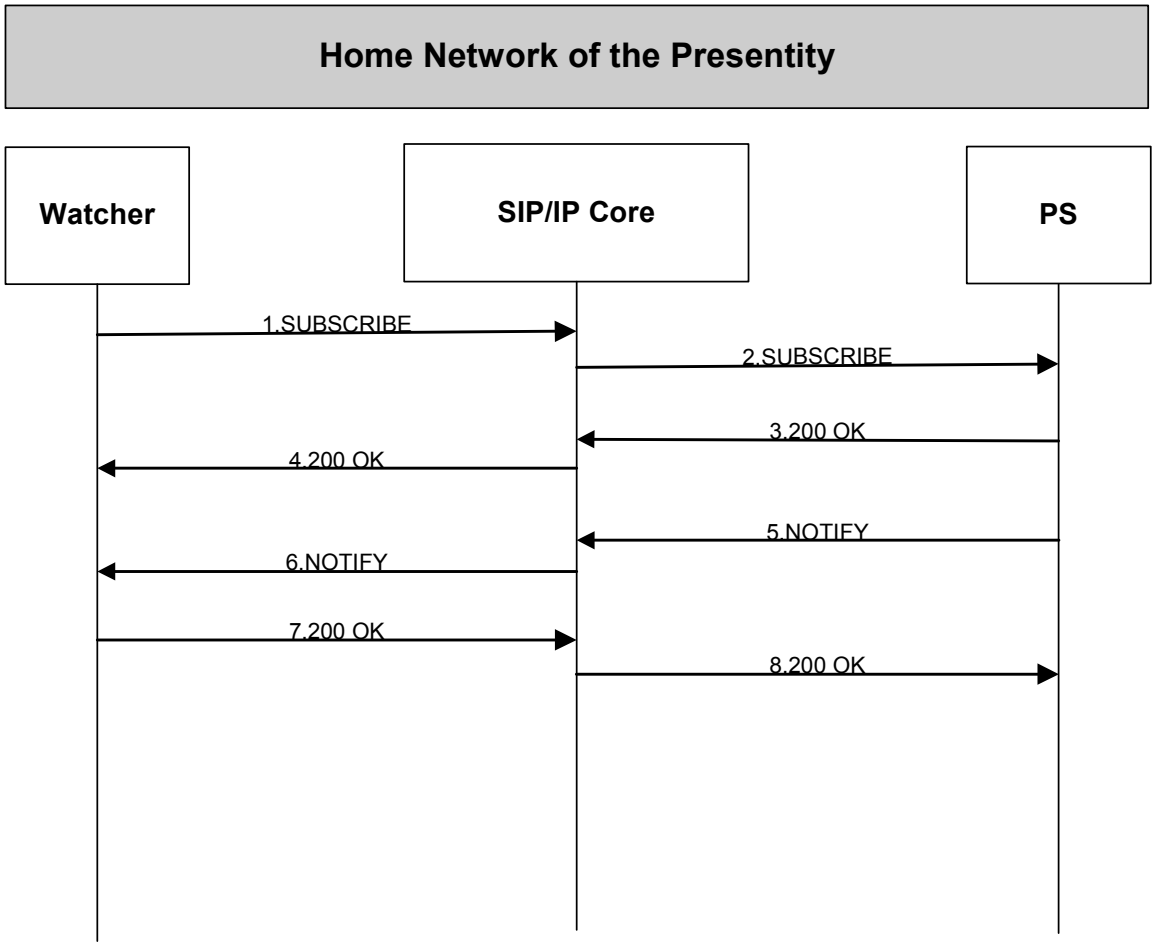

Figure 19 - Watcher Initiated cancelling

1. A watcher sends a SIP SUBSCRIBE request to the SIP/IP Core network with the “Expires” header field set to 0 indicating the cancelling of the subscription, according to [RFC3265].

2. The SIP/IP Core network forwards the SIP SUBSCRIBE request to the PS.NOTE: Even when the watcher and the presentity reside at different domains the SIP/IP core of the watcher will foward the SUBSCRIBE request directly to the PS since it has already performed the address resolution on the address of the presentity during the initial subscription.

3. The PS accepts the SUBSCRIBE message with the "Expires" header set to 0 indicating the canceling a subscription operation, and sends a 200 OK to the SIP/IP Core.
4. The SIP/IP Core forwards the 200 OK to the Watcher.
5. The PS sends a SIP NOTIFY request to the SIP/IP Core network with a "Subscription-State" header field set to "terminated" indicating that the subscription has been terminated, according to [RFC3265].
6. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher.
7. The watcher sends a SIP 200 OK response to the SIP/IP Core network to acknowledge the SIP NOTIFY request.
8. The SIP/IP Core network forwards the SIP 200 OK to the PS.

C.1.3.2 Presence Server Initiated Canceling

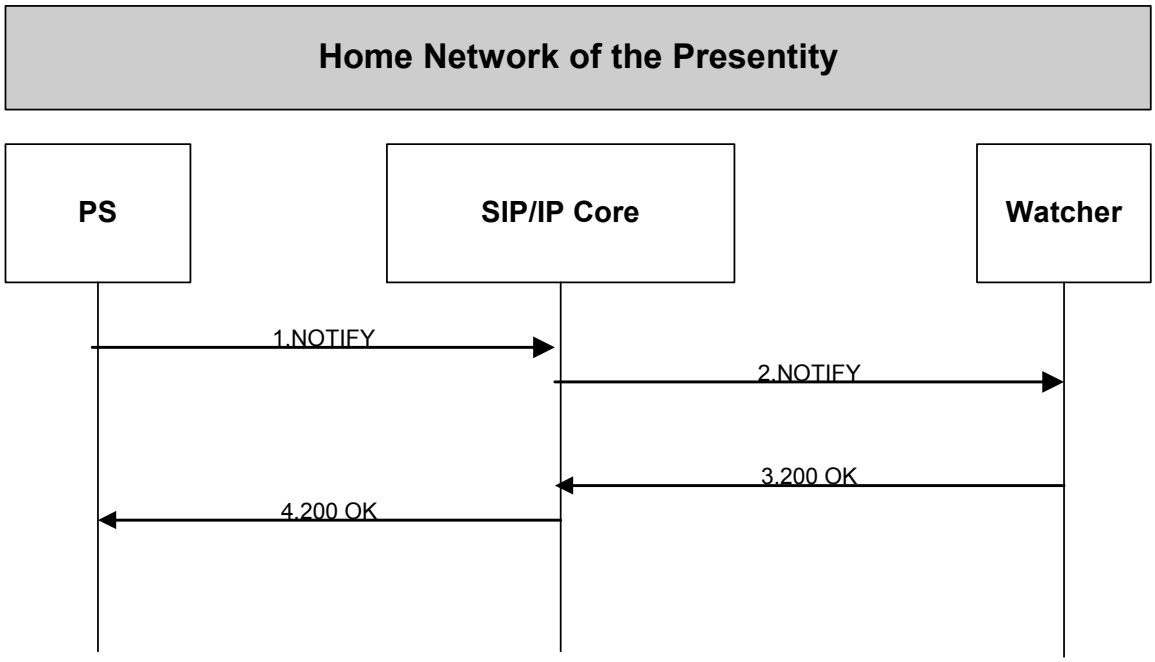

Figure 20 - Presence Server Initiated cancelling

1. The PS sends a SIP NOTIFY request with a "Subscription-State" header field set to "terminated" indicating that the PS wants to terminate a subscription, according to [RFC 3265].
2. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher.

NOTE: Even when the watcher and the presentity reside at different domains the SIP/IP core of the presentity will foward the NOTIFY request directly to the Watcher since it already has the address of the Watcher.

3. The watcher sends a SIP 200 OK response to the SIP/IP Core network to acknowledge the SIP NOTIFY request.
4. The SIP/IP Core network forwards the SIP 200 OK to the PS.

C.1.4 PS subscribing to changes of Presence authorisation policy

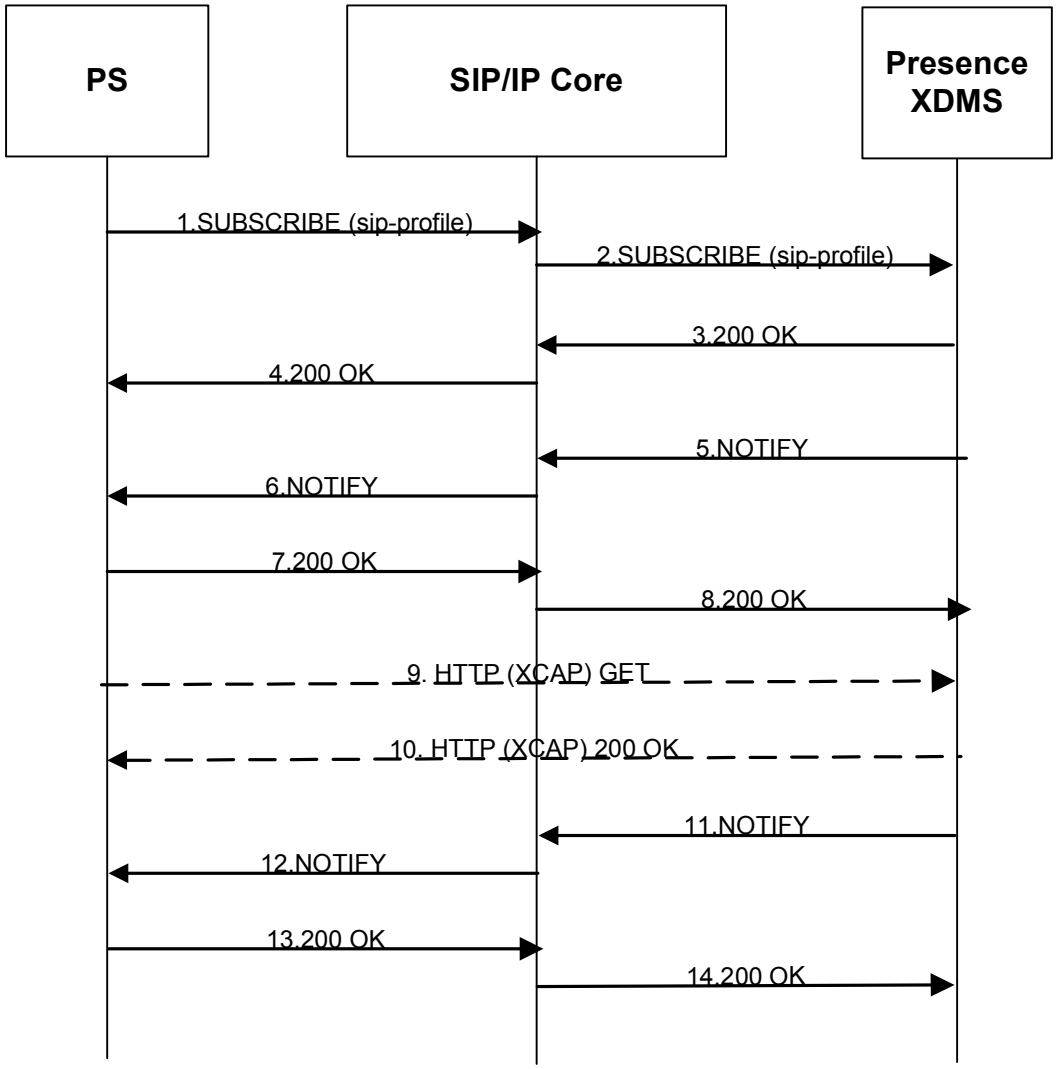

Figure 21 – PS subscribing to changes of a Presentity's authorisation policy

1. A PS that wishes to subscribe to changes of a Presentity's authorisation policy document, sends a SIP SUBSCRIBE request with the "Event" header field set to "sip-profile" as described in [XDMSPEC]. The Request URI of the SIP SUBSCRIBE request is set to the public user identity of the Presentity that wishes to subscribe for changes against its Presence authorisation document. The "document" header field is set to appropriate path for that Presentity as described in section 5.4.3.2 and [XDMSPEC].
2. The SIP/IP Core forwards the request to the appropriate Presence XDMS.
3. The Presence XDMS accepts the subscription and responds with a SIP 200 OK.
4. The SIP/IP Core forwards the response to the PS.
5. The Presence XDMS sends the first SIP NOTIFY request, this is used in order to synchronise the Presence XDMS and PS on a common "baseline" document as described in [XCAP_Diff].
6. The SIP/IP Core forwards the SIP NOTIFY request to the PS.
7. The PS accepts the SIP NOTIFY request with a SIP 200 OK response.
8. The SIP/IP Core forwards the SIP 200 OK response to the Presence XDMS.

9. The PS fetches using HTTP (XCAP) GET request the version of the document indicated (with the Etag) in the received SIP NOTIFY request, as defined in [XCAP_diff] and [XDMSPEC].
10. The version of the document requested is provided by the Presence XDMS.
11. When changes happen in the Presence authorisation rules document the Presence XDMS informs the PS about the changes with a SIP NOTIFY request with the changed data.
12. The SIP/IP Core forwards the SIP NOTIFY request to the PS.
13. The PS responds to the SIP NOTIFY with a 200 OK.
14. The SIP/IP Core forwards the 200 OK response to the Presence XDMS.

C.1.5 Subscribing to Watcher Information state changes

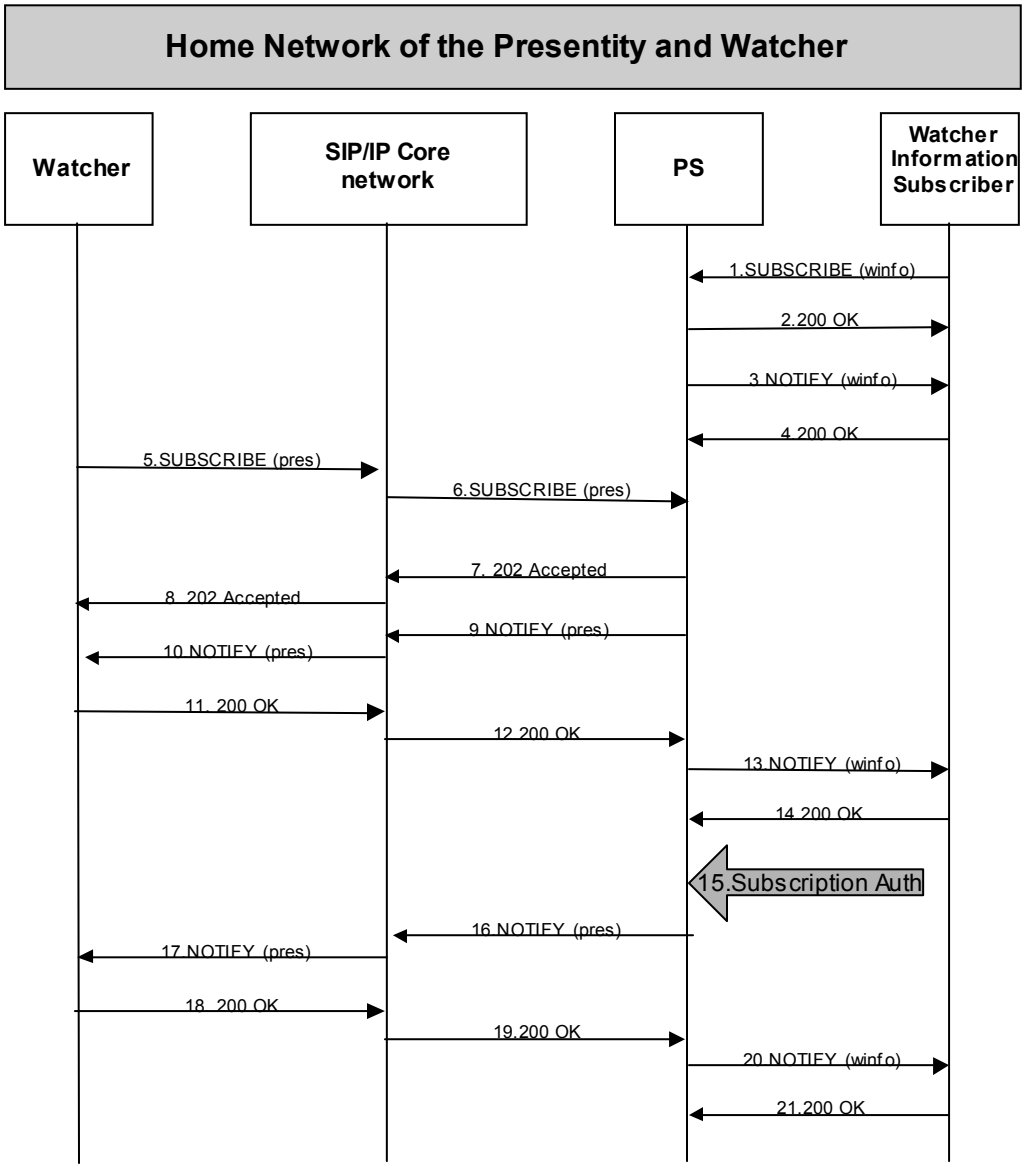

Figure 22- Watcher Information (Subscriptions/Notifications)

Note: The SIP/IP Core between the PS and the watcher information subscriber is not shown in the figure due to simplicity reasons.

In this use case we assume that the application of the presence subscription authorization rules for the watcher results in placing the subscription into the state "pending".

1. The watcher information subscriber subscribes to the watcher information (see section 5.3.1) of its own presentity in order to receive notifications about new, unauthorized watchers that subscribe to its presence information. This is performed by sending a SIP SUBSCRIBE request to the PS according to [RFC3857].
2. The PS after authorizing the subscription allows the watcher information subscriber to subscribe to the watcher information. The PS acknowledges the SIP SUBSCRIBE request by generating a SIP 200 OK response.
3. The PS generates a SIP NOTIFY request including the current state of the watcher information of the presentity.
4. The watcher information subscriber acknowledges the SIP NOTIFY request by sending a SIP 200 OK response.
5. After time elapses, a watcher attempts to subscribe to the presentity's presence information by sending a SIP SUBSCRIBE request according to [RFC3856].
6. The SIP/IP Core network forwards the SIP SUBSCRIBE request to the PS.
7. The PS acknowledges the SIP SUBSCRIBE request and returns a SIP 202 Accepted response.
8. The SIP/IP Core network forwards the SIP 202 Accepted response to the watcher.
9. The PS immediately sends a SIP NOTIFY request as mandated by [RFC3265], setting the "Subscription-State" header field to the value of "pending" indicating that the subscription has been received, but the Subscription Authorization Policy is insufficient to accept or deny the subscription at this time.
10. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher
11. The watcher acknowledges the SIP NOTIFY request by sending a SIP 200 OK response.
12. The SIP/IP Core network forwards the SIP 200 OK response to the PS.
13. As the watcher information state for the presentity changes (a watcher has requested to subscribe to the presence information), the PS sends a SIP NOTIFY request to indicate the change (a subscription for the presentity's presence information is pending) to the watcher information subscriber according to [RFC3857].
14. The watcher information subscriber acknowledges the SIP NOTIFY request with a SIP 200 OK response.
15. The presentity authorizes the subscription of the pending watcher .
16. As the subscription state for the presence event package changes, the PS sends a SIP NOTIFY request to the watcher indicating that the subscription is authorized. The SIP NOTIFY request also conveys the current presence information state of the presentity.
17. The SIP/IP Core network forwards the SIP NOTIFY request to the watcher
18. The watcher acknowledges the SIP NOTIFY request by sending a SIP 200 OK response.
19. The SIP/IP Core network forwards the SIP 200 OK response to the PS.
20. As the subscription state for the presence event package changes, at the same time of step 16, the PS sends a SIP NOTIFY request to the winfo template package to the watcher information subscriber indicating that the subscription is authorized.
21. The watcher information subscriber acknowledges the SIP NOTIFY request with a SIP 200 OK response.

C.1.6 Sending different presence information to different watchers

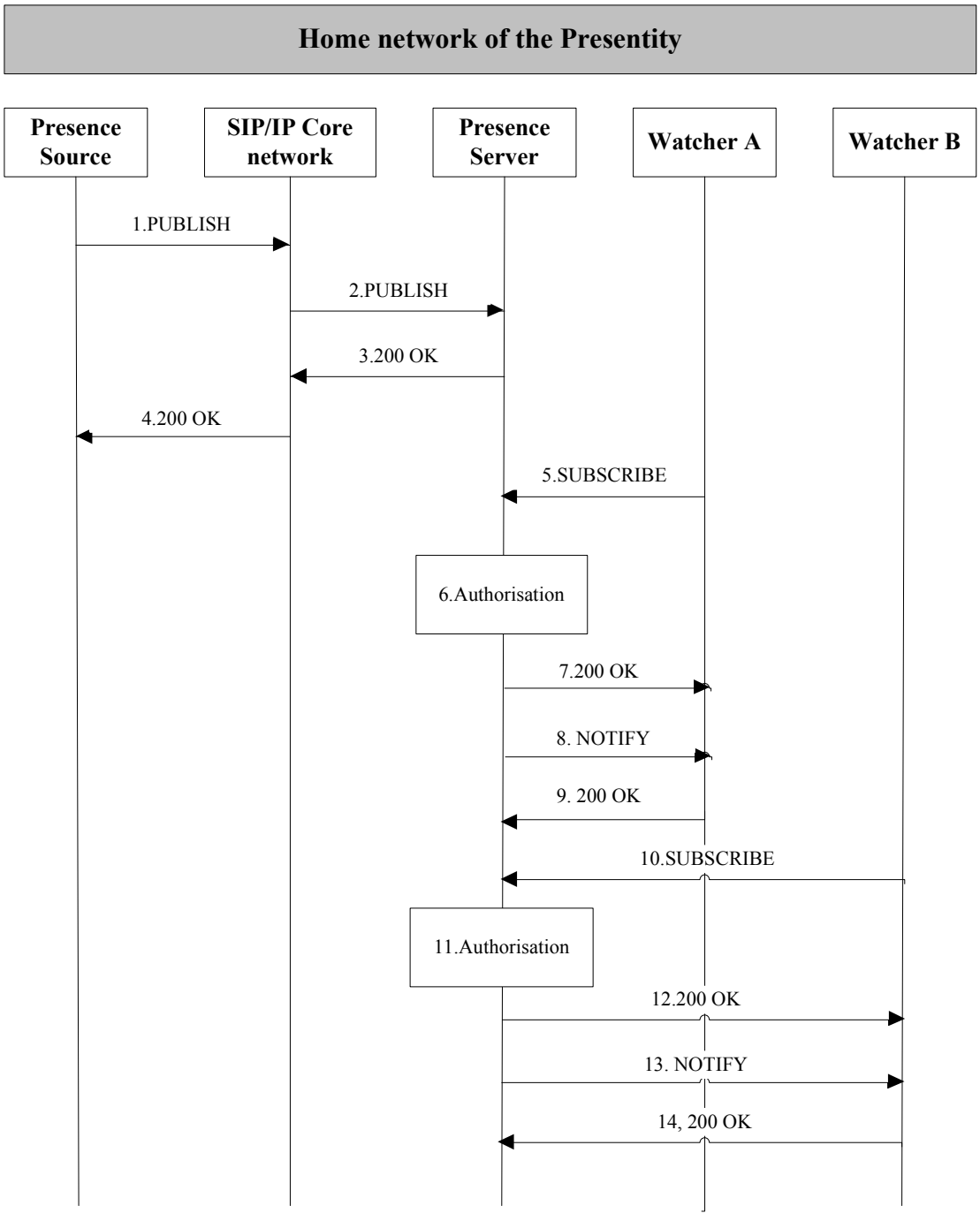

Figure 23 - Sending different presence information to different watchers

Note: The SIP/IP Core between the PS and the watchers is not shown in the figure due to simplicity reasons.

1.The Presence Source generates a SIP PUBLISH request, which contains a presence document. This document contains more than one tuple that contain the same element with different value. The association of tuples to different watchers and watcher groups is based on the Presence authorisation policies.

2.The SIP/IP Core network routes the request to the corresponding PS.

3.The PS authorises the presence publication, and checks the information the message contains. The PS then processes the presence information and sends a SIP 200 OK response back to Presence Source.

4.The SIP/IP Core network forwards the response back to the Presence Source.

5.Watcher A wishing to subscribe to presence information about a presentity, sends a SIP SUBSCRIBE request to the PS.

6.The PS performs the necessary authorisation checks on watcher A to ensure it is allowed to watch the presentity and to watch what specified tuples based on e.g. <class> element.

7.The PS sends a SIP 200 OK response back to watcher A.

8.The PS generates a NOTIFY request which contains a presence document for watcher A.

9. Watcher A sends a SIP 200 OK response to PS.

10.Watcher B wishing to subscribe to presence information about a presentity, sends a SIP SUBSCRIBE request to the PS.

11.The PS performs the necessary authorisation checks on watcher B to ensure it is allowed to watch the presentity and to watch what specified tuples based on e.g. <class> element.

12.The PS sends a SIP 200 OK response back to watcher B.

13.The PS generates a NOTIFY request which contains a presence document for watcher B. Watcher B MAY receive different presence information than Watcher A.

14. Watcher B sends a SIP 200 OK response to PS.

Appendix D. Change History (Informative)

D.1 Approved Version History

Reference	Date	Description
OMA-TS-Presence_SIMPLE-V1_0-20060725-A	25 Jul 2006	TP approved: OMA-TP-2006-0223R04-INP_Presence_SIMPLE_V1_0_for_final_approval
OMA-TS-Presence_SIMPLE-V1_0_1-20061128-A	28 Nov 2006	Incorporated CRs: OMA-PAG-2006-0392R02 OMA-PAG-2006-0412 OMA-PAG-2006-0414 OMA-PAG-2006-0416 OMA-PAG-2006-0481R01 OMA-PAG-2006-0496 OMA-PAG-2006-0497 OMA-PAG-2006-0508 OMA-PAG-2006-0512 OMA-PAG-2006-0514 OMA-PAG-2006-0541 OMA-PAG-2006-0542 OMA-PAG-2006-0580 OMA-PAG-2006-0662 OMA-PAG-2006-0690 OMA-PAG-2006-0691 OMA-PAG-2006-0717R01 OMA-PAG-2006-0733 OMA-PAG-2006-0749R01
	29 Feb 2008	Status changed to historic OMA-TP-2008-0100R03-INP_Presence_SIMPLE_1_0_Historical