Presence SIMPLE Specification

Candidate Version 2.0 – 23 Dec 2008

Open Mobile Alliance

OMA-TS-Presence_SIMPLE-V2_0-20081223-C

Contents

Figures

1. Scope

This document provides the specifications for the OMA Presence SIMPLE 2.0 enabler. This enabler is based on the IETF SIMPLE technology and utilizes the network capabilities of a SIP/IP Core (e.g. 3GPP IMS and 3GPP2 MMD). This enabler is specified such that it is available to be used by other service enablers.

This document is built upon and backward compatible with the specifications for the OMA Presence SIMPLE 1.1 enabler (see [PRS_ERELD-V1_1]).

2. References

2.1 Normative References

OMA

[CP_ProvCont] "Client Provisioning ProvBoot", Version 1.1, Open Mobile Alliance™, OMA-WAP-TS-ProvCont-V1_1, URL: http://www.openmobilealliance.org/

[DM_ERELD] "Device Management (based on SyncML DM)", Version 1.2, Open Mobile Alliance™, OMA-DM-V1_2, URL: http://www.openmobilealliance.org/

[DM_StdObj] "OMA Device Management Standardized Objects", Version 1.2, Open Mobile Alliance™, OMA-TS-DM_StdObj-V1_2, URL: http://www.openmobilealliance.org/

[PDE_DDS] "Presence SIMPLE Data Specification", Version 2.0, Open Mobile Alliance™, OMA-DDS-Presence_Data_Ext-V2_0, URL: http://www.openmobilealliance.org/

[PRS_AC] "Presence Application Characteristics file of Presence V2.0", Version 1.0, Open Mobile Alliance™, OMA-SUP-AC_ap0009_presence-V1_0, URL: http://www.openmobilealliance.org/

[PRS_AD] "Presence SIMPLE Architecture", Version 2.0, Open Mobile Alliance™, OMA-AD-Presence_ SIMPLE-V2_0, URL: http://www.openmobilealliance.org/

[PRS_ContXDM] "Presence Content XDM Specification" Version 1.0, Open Mobile Alliance™, OMA-TS-Presence-SIMPLE_Content_XDM- V1_0, URL: http://www.openmobilealliance.org/

[PRS_ERELD-V1_1] "Enabler Release Definition for Presence SIMPLE", Version 1.1, Open Mobile Alliance™, OMA-ERELD-Presence_ SIMPLE-V1_1, URL: http://www.openmobilealliance.org/

[PRS_MO] "OMA Management Object for SIMPLE Presence", Version 2.0, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE_MO-V2_0, URL: http://www.openmobilealliance.org/

[PRS_PresXDM] "Presence XDM Specification", Version 2.0, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE_XDM-V2_0, URL: http://www.openmobilealliance.org/

[PRS_RD] "Presence SIMPLE Requirements", Version 2.0, Open Mobile Alliance™, OMA-RD-Presence_SIMPLE-V2_0, URL: http://www.openmobilealliance.org/

[PRS_RLSXDM] "Resource List Server (RLS) XDM Specification", Version 2.0, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE_RLS_XDM-V2_0, URL: http://www.openmobilealliance.org/

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL: http://www.openmobilealliance.org/

[XDM_AD] "XML Document Management Architecture", Version 2.0, Open Mobile Alliance™, OMA-AD-XDM-V2_0, URL: http://www.openmobilealliance.org/

[XDM_Core] "XML Document Management Specification", Version 2.0, Open Mobile Alliance™, OMA-TS-XDM_Core-V2_0, URL: http://www.openmobilealliance.org/

[XDM_List] "Shared List XDM Specification", Version 2.0, Open Mobile Alliance™, OMA-TS-XDM_Shared-V2_0, URL: http://www.openmobilealliance.org/

[XSD_suppNot] "OMA-defined presence-based event notification suppression filter", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_prs_suppnotFilter-V1_0, URL: http://www.openmobilealliance.org/

IETF

[IETF-EventThrottle] IETF draft-niemi-sipping-event-throttle-07 “Session Initiation Protocol (SIP) Event Notification Extension for Notification Throttling”, A. Niemi et al., Oct 22, 2008,
URL: http://www.ietf.org/internet-drafts/draft-niemi-sipping-event-throttle-07.txt

Note: IETF Draft work in progress

[IETF-GRUU] IETF draft-ietf-sip-gruu-15 “Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)”, J. Rosenberg, Oct 11, 2007,
URL: http://www.ietf.org/internet-drafts/draft-ietf-sip-gruu-15.txt

Note: IETF Draft work in progress

[IETF-SessionPol] IETF draft-ietf-sip-session-policy-framework-05 “A Framework for Session Initiation Protocol (SIP) Session Policies”, V. Hilt et al., Nov 1, 2008,
URL: http://www.ietf.org/internet-drafts/draft-ietf-sip-session-policy-framework-05.txt

Note: IETF Draft work in progress

[IETF-SubNotEtag] IETF draft-ietf-sip-subnot-etags-03 “An Extension to Session Initiation Protocol (SIP) Events for Conditional Event Notification”, A. Niemi, Jul 14, 2008,
URL: http://www.ietf.org/internet-drafts/draft-ietf-sip-subnot-etags-03.txt

Note: IETF Draft work in progress

[IETF-ViewShare] IETF draft-ietf-simple-view-sharing-02 “Optimizing Federated Presence with View Sharing”, J. Rosenberg et al., Nov 3 2008,
URL: http://www.ietf.org/internet-drafts/draft-ietf-simple-view-sharing-02.txt

Note: IETF Draft work in progress

[RFC1952] IETF RFC 1952 “GZIP file format specification version 4.3”, P. Deutsch, May 1996,
URL: http://www.ietf.org/rfc/rfc1952.txt

[RFC2046] IETF RFC 2046 “Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types”, N. Freed et. al., Nov 1996,
URL: http://www.ietf.org/rfc/rfc2046.txt

[RFC2119] IETF RFC 2119 “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, Mar 1997,
URL: http://www.ietf.org/rfc/rfc2119.txt

[RFC2387] IETF RFC 2387 “The MIME Multipart/Related Content-type”, E. Levinson, Aug 1998,
URL: http://www.ietf.org/rfc/rfc2387.txt

[RFC2392] IETF RFC 2393 “Content-ID and Message-ID Uniform Resource Locators”, E. Levinson, Aug 1998,
URL: http://www.ietf.org/rfc/rfc2392.txt

[RFC2616] IETF RFC 2616 “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding et al., Jun 1999,
URL: http://www.ietf.org/rfc/rfc2616.txt

[RFC2818] IETF RFC 2818 “HTTP Over TLS”, E. Rescorla, May 2000,
URL: http://www.ietf.org/rfc/rfc2818.txt

[RFC3261] IETF RFC 3261 "Session Initiaton Protocol (SIP)", J. Rosenberg et al., Jun 2002,
URL: http://www.ietf.org/rfc/rfc3261.txt

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

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

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

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

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

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

[RFC3515] IETF RFC 3515 "The Session Initiation Protocol (SIP) REFER Method", R. Sparks, Apr 2003, URL: http://www.ietf.org/rfc/rfc3515.txt

[RFC3840] IETF RFC 3840 "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", Rosenberg, J., et al., Aug 2004, RFC 3840, URL: http://www.ietf.org/rfc/rfc3840.txt

[RFC3841] IETF RFC 3841 "Caller Preferences for the Session Initiation Protocol (SIP)", J. Rosenberg et al., Aug 2004, URL: http://www.ietf.org/rfc/rfc3841.txt

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

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

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

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

[RFC3863] IETF RFC 3863 "Presence Information Data Format (PIDF)", H. Sugano et al., Aug 2004, URL: http://www.ietf.org/rfc/rfc3863.txt

[RFC3903] IETF RFC 3903 "An Event State Publication Extension to the Session Initiation Protocol (SIP)", A. Niemi, Oct 2004, URL: http://www.ietf.org/rfc/rfc3903.txt

[RFC3966] IETF RFC 3966 "The tel URI for Telephone Numbers", H. Schulzrinne, Dec 2004, URL: http://www.ietf.org/rfc/rfc3966.txt

[RFC4077] IETF RFC 4077 "A Negative Acknowledgement Mechanism for Signaling Compression", A.B. Roach, May 2005, URL: http://www.ietf.org/rfc/rfc4077.txt

[RFC4474] IETF RFC 4474 "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", J. Peterson et al., Aug 2006, URL: http://www.ietf.org/rfc/rfc4474.txt

[RFC4483] IETF RFC 4483 "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages", E. Burger, Ed., May 2006, URL: http://www.ietf.org/rfc/rfc4483.txt

[RFC4488] IETF RFC "Suppression of Session Initiation Protocol (SIP) REFER Method Implicit Subscription", O. Levin, May 2006, URL: http://www.ietf.org/rfc/rfc4488.txt

[RFC4660] IETF RFC 4660 "Functional Description of Event Notification Filtering", H.Khartabil et al., Sep 2006, URL: http://www.ietf.org/rfc/rfc4660.txt

[RFC4661] IETF RFC 4661 "An Extensible Markup Language (XML) Based Format for Event Notification Filtering", H. Khartabil et al., Sep 2006, URL: http://www.ietf.org/rfc/rfc4661.txt

[RFC4662] IETF RFC 4662 "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists", A. B. Roach et al., Aug 2006, URL: http://www.ietf.org/rfc/rfc4662.txt

[RFC4896] IETF RFC 4896 "Signaling Compression (SigComp) Corrections and Clarifications", A. Surtees et al., Jun 2007, URL: http://www.ietf.org/rfc/rfc4896.txt

[RFC5025] IETF RFC 5025 "Presence Authorization Rules", J. Rosenberg, Dec 2007, URL: http://www.ietf.org/rfc/rfc5025.txt

[RFC5049] IETF RFC 5049 "Applying Signaling Compression (SigComp) to the Session Initiation Protocol (SIP)", C. Bormann et al., Dec 2007,
URL: http://www.ietf.org/rfc/rfc5049.txt

[RFC5112] IETF RFC 5112 "The Presence-Specific Static Dictionary for Signaling Compression (Sigcomp)", M. Garcia-Martin, Jan 2008,
URL: http://www.ietf.org/rfc/rfc5112.txt

[RFC5262] IETF RFC 5262 "Presence Information Data format (PIDF) Extension for Partial Presence", M. Lonnfors et al., Sep 2008,
URL: http://www.ietf.org/rfc/rfc5262.txt

[RFC5263] IETF RFC 5263 "Session Initiation Protocol (SIP) extension for Partial Notification of Presence Information", M.Lonnfors et al., Sep 2008,
URL: http://www.ietf.org/rfc/rfc5263.txt

[RFC5264] IETF RFC 5264 "Publication of Partial Presence Information", M. Lonnfors et al., Sep 2008,
URL: http://www.ietf.org/rfc/rfc5264.txt

[RFC5367] IETF RFC 5367 "Subscriptions to Request-Contained Resource Lists in the Session Initiation Protocol (SIP)", G. Camarillo et al., Oct 2008,
URL: http://www.ietf.org/rfc/rfc5367.txt

3GPP/3GPP2

[3GPP-TS_23.228] 3GPP TS 23.228 "IP Multimedia Subsystem (IMS); Stage 2",
URL: http://www.3gpp.org/ftp/Specs/archive/23_series/23.228/

[3GPP-TS_23.141] 3GPP TS 23.141 "Presence Service; Architecture and functional description",
URL: http://www.3gpp.org/ftp/Specs/archive/23_series/23.141/

[3GPP-TS_24.109] 3GPP TS 24.109 "Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details ; Stage 3",
URL: http://www.3gpp.org/ftp/Specs/archive/24_series/24.109/

[3GPP-TS_24.141] 3GPP TS 24.141 "Presence service using the IP Multimedia (IM) Core Network (CN) subsystem; Stage-3",
URL: http://www.3gpp.org/ftp/Specs/archive/24_series/24.141/

[3GPP-TS_24.229] 3GPP TS 24.229 "Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3",
URL: http://www.3gpp.org/ftp/Specs/archive/24_series/24.229/

[3GPP-TS_26.141] 3GPP TS 26.141 "IP Multimedia System (IMS) Messaging and Presence; Media formats and codecs",
URL: http://www.3gpp.org/ftp/Specs/archive/26_series/26.141/

[3GPP-TS_32.240] 3GPP TS 32.240 "Charging management; Charging architecture and principles",
URL: http://www.3gpp.org/ftp/Specs/archive/32_series/32.240/

[3GPP-TS_32.260] 3GPP TS 32.260 "Charging Management; IP Multimedia Subsystem (IMS) Charging",
URL: http://www.3gpp.org/ftp/Specs/archive/32_series/32.260/

[3GPP-TS_33.203] 3GPP TS 33.203 "Access Security for IP-based services",
URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.203/

[3GPP2-C.P0071] 3GPP2 C.P0071 "IP Multimedia Domain(MMD) Codecs and Transport Protocols",
URL: http://www.3gpp2.org/Public_html/specs/index.cfm

[3GPP2-S.R0086] 3GPP2 S.R0086 "IMS Security Framework",
URL: http://www.3gpp2.org/Public_html/specs/index.cfm

[3GPP2-X.S0013-002] 3GPP2 X.S0013-002 "All-IP Core Network Multimedia Domain: IP Multimedia Subsystem - Stage 2",
URL: http://www.3gpp2.org/Public_html/specs/index.cfm

[3GPP2-X.S0013-004] 3GPP2 X.S0013-004 "All-IP Core Network Multimedia Domain: IP Multimedia Call Control Protocol Based on SIP and SDP Stage 3",
URL: http://www.3gpp2.org/Public_html/specs/index.cfm

[3GPP2-X.S0013-007] 3GPP2 X.S0013-007 "All-IP Core Network Multimedia Domain: IP Multimedia Subsystem – Charging Architecture",
URL: http://www.3gpp2.org/Public_html/specs/index.cfm

[3GPP2-X.S0013-008] 3GPP2 X.S0013-008 "All-IP Core Network Multimedia Domain: IP Multimedia Subsystem – Offline Accounting, Information Flows and Protocol",
URL: http://www.3gpp2.org/Public_html/specs/index.cfm

[3GPP2-X.S0027-001] 3GPP2 X.S0027-001 "Presence Service; Architecture and functional description",
URL: http://www.3gpp2.org/Public_html/specs/index.cfm

2.2 Informative References

Void.

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

Application Usage	Use definition from [XDM_Core].
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	Use definition from [PRS_AD].
Event Package	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]
Fetcher	Use definition from [PRS_RD].
Global Tree	Use definition from [XDM_Core].
Permanent Presence State	Use definition from [PRS_AD].
Presence External Agent (PEA)	Use definition from [PRS_RD].
Presence Information	Use definition from [PRS_RD].
Presence Information Element	Use definition from [PRS_RD].
Presence List	Use definition from [PRS_AD].
Presence Network Agent (PNA)	Use definition from [PRS_RD].
Presence Publication Rules	Use definition from [PRS_AD].
Presence Service	Use definition from [PRS_RD].
Presence Source	Use definition from [PRS_RD].
Presence Subscription Rules	Use definition from [PRS_AD].
Presence User Agent (PUA)	Use definition from [PRS_RD].
Presentity	Use definition from [PRS_RD].
Publication Authorization Rules	Use definition from [PRS_AD].
Publication Content Rules	Use definition from [PRS_AD].
Request-contained Presence	Use definition from [PRS_AD].

List

Request-contained Watcher Information List	Use definition from [PRS_AD].
Resource List Server (RLS)	Use definition from [PRS_AD].
Subscribed-watcher	Use definition from [PRS_RD].
Subscription Authorization Rules	Use definition from [PRS_AD].
Subscription Content Rules	Use definition from [PRS_AD].
Users Tree	Use definition from [XDM_Core].
XCAP Server	Use definition from [XDM_Core].
Watcher	Use definition from [PRS_RD].
Watcher Information	Use definition from [PRS_RD].
Watcher Information Subscriber	Use definition from [PRS_RD].

3.3 Abbreviations

3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AC	Application Characteristics
AD	Architecture Document
AS	Application Server
CID	Content ID
DM	Device Management
EPA	Event Publication Agent
ESC	Event State Compositor
GRUU	Globally Routable UA URI
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extensions
MMD	Multimedia Domain
MO	Management Object
OMA	Open Mobile Alliance
OMNA	Open Mobile Naming Authority
OTA	Over the Air
PEA	Presence External Agent
PIDF	Presence Information Data Format
PNA	Presence Network Agent
PoC	Push-to-talk over Cellular
PRS	Presence SIMPLE
PS	Presence Server
PUA	Presence User Agent
RD	Requirement Document
RFC	Request For Comments
RLMI	Resource List Meta-Information
RLS	Resource List Server
SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
TLS	Transport Layer Security
UA	User Agent
UE	User Equipment
URI	Uniform Resource Identifier
XCAP	XML Configuration Access Protocol

XDM	XML Document Management
XDMC	XML Document Management Client
XDMS	XML Document Management Server
XML	eXtensible Markup Language
XUI	XCAP User Identifier

4. Introduction

This document defines an application level specification for the OMA PRS enabler.

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 implemented in the user's terminal or within a network entity.

When the SIP/IP Core 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] respectively.

5.1.1 General

The Presence Source:

- SHALL support the presence data model defined in [PDE_DDS] "*Presence Data Model*";
- SHALL use the elements defined in [PDE_DDS] "*Presence Information Element Definitions*" when providing Presence Information with semantics identical to those elements; and
- MAY support other PIDF extensions to provide elements whose semantics do not match with those defined in [PDE_DDS], as long as a Watcher that does not understand those extensions can ignore them without changing the meaning of the Presence Information Elements that are understood.

NOTE: For a given Presentity, the Presence Information provided by each Presence Source is composed into a single raw presence document as described in section 5.5.3.2.

The Presence Source SHALL be free to provide any value of the "id" (instance identifier) attributes for <tuple>, <person> and <device> (see [PDE_DDS]) 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 the Watcher.

The Presence Source SHALL support one or both of the following mechanisms for providing Presence Information about a given Presentity to the PS:

- Publication of Presence Information using SIP, as described in section 5.1.2; and/or
- Manipulation of Permanent Presence State using XCAP, as described in section 5.1.3.

5.1.2 Publication of Presence Information using SIP

If the Presence Source supports publication of Presence Information using SIP, then the following procedures apply.

The Presence Source:

- SHALL implement the Event Publication Agent (EPA) function and support the PUBLISH method according to the procedures described in [RFC3903]; and
- SHALL support the 'application/pidf+xml' content type, according to [RFC3863].

The Presentity SHALL be identified by a SIP URI (as defined in [RFC3261]), and may additionally be identified by a tel URI (as defined in [RFC3966]) or a pres URI (as defined in [RFC3859]). The tel URI SHALL take the international public telecommunication number format with a leading "+" sign. If the Presence Source is aware of the SIP URI of the Presentity, the Presence Source SHOULD insert the SIP URI in the Request-URI of the PUBLISH request rather than a pres URI or a tel URI.

The Presence Source SHALL insert the same URI in both the "entity" attribute of the <presence> element of the presence document and the Request-URI of the PUBLISH request.

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

- The Presence Source SHALL set the "entity" attribute of the <presence> element of the presence document, defined in [RFC3863], to the Presentity's public user identity, defined in [3GPP-TS_24.229] and [3GPP2-X.S0013-004].
- If the publication is being performed by the same Presentity, the Presence Source:
  - SHALL, if implemented in a UE, set the value of the P-Preferred-Identity header field (if included), described in [3GPP-TS_24.229] and [3GPP2-X.S0013-004], of the PUBLISH request to the same value as the "entity" attribute of the <presence> element in the presence document; and
  - SHALL, if implemented in an AS within the same trust domain as the PS, set the value of the P-Asserted-Identity header field, described in [3GPP-TS_24.229] and [3GPP2-X.S0013-004], of the PUBLISH request to the same value as the "entity" attribute of the <presence> element in the presence document.
- If the publication is being performed on behalf of another Presentity, the Presence Source:
  - SHALL, if implemented in a UE, set the value of the P-Preferred-Identity header field (if included), described in [3GPP-TS_24.229] and [3GPP2-X.S0013-004], of the PUBLISH request to the authenticated originator identity that it intends to use for publication authorization; and
  - SHALL, if implemented in an AS within the same trust domain as the PS, set the value of the P-Asserted-Identity header field, described in [3GPP-TS_24.229] and [3GPP2-X.S0013-004], of the PUBLISH request to the authenticated originator identity that it intends to use for publication authorization.
  - If the Presence Source receives a 488 (Not Acceptable Here) response containing a Policy-Contact header field as defined in [IETF-SessionPol], the Presence Source:
    - MAY fetch the Publication Content Rules Presence Source View document from the Presence XDMS via an XDMC using the URI from the Policy-Contact header field; and
    - MAY evaluate the Publication Content Rules Presence Source View and re-publish a presence document satisfying the Publication Content Rules Presence Source View.

5.1.2.1 Partial Publication

Partial publication is a mechanism that enables a Presence Source to 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 partial publication procedure, according to [RFC5264]; and
- SHALL support partial presence extension to PIDF, according to [RFC5262].

5.1.2.2 Handling of Large MIME Objects

5.1.2.2.1 Publishing MIME Objects using Content Indirection

A Presence Source MAY support the content indirection mechanism [RFC4483]. If the following conditions are true:

- the Presence Source supports the content indirection mechanism;
- the value of the Presence Information Element is a MIME object; and
- the Presence Source decides to use the content indirection mechanism for publishing an initial or modified value of the Presence Information Element,

then the Presence Source:

1) SHALL store the MIME object in the Content Server.

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

   The Presence Source MAY be provisioned with the HTTP URI, 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 C.

2) SHALL construct an HTTP URI, or optionally an HTTPS URI, referencing the stored MIME object.

3) SHALL 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) SHALL set a cid URI as described in [RFC2392] referencing to the 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) SHALL 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; and

   c) SHALL 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.

The MIME object format SHALL conform to [3GPP-TS_26.141] and [3GPP2-C.P0071].

5.1.2.2.2 Publishing MIME Objects using Direct Content

A Presence Source MAY support the 'multipart/related' content type as described in [RFC2387]. If the following conditions are true:

- the Presence Source supports the 'multipart/related' content type;
- the value of the Presence Information Element is a MIME object; and
- the Presence Source decides to publish the MIME object as direct content inside the presence document,

then the Presence Source:

1) SHALL utilize the 'multipart/related' content type as described in [RFC2387] in the PUBLISH request;

2) SHALL set a cid URI as described in [RFC2392] referencing to the multipart body which contains the MIME object; and

3) SHALL 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 objects sent as 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 the CLIENT-OBJ-DATA-LIMIT parameter defined in [PRS_AC] and [PRS_MO].

If the Presence Source does not support OTA Provisioning, the size limit for MIME objects sent as 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.

The MIME object format SHALL conform to [3GPP-TS_26.141] and [3GPP2-C.P0071].

5.1.2.2.3 Publishing MIME Objects using Presence Content XDMS

A Presence Source MAY support storing MIME objects in the Presence Content XDMS and publishing the URI of the stored MIME object as the value of the Presence Information Element. If the following conditions are true:

- the Presence Source is co-located with an XDMC;

- the value of the Presence Information Element is a MIME object; and
- the Presence Source decides to publish an initial or modified value of the Presence Information Element using the Presence Content XDMS,

then the Presence Source:

1) SHALL use XDMC procedures as described in [XDM_Core] "*Procedures at the XDM Client*" to store the MIME object in the Presence Content XDMS; and

2) SHALL include the 'etag' attribute [PDE_DDS] in the appropriate element in the presence document.

The MIME object format SHALL conform to [PRS_ContXDM].

5.1.2.3 Limiting the Rate of Publications

The service provider MAY configure a Presence Source with the shortest allowed time period between two PUBLISH requests. This can be done with OTA Provisioning or local configuration. In case of OTA Provisioning, it SHALL use the value of SOURCE-THROTTLE-PUBLISH (defined in Appendix C).

If such configuration is present for the Presence Source, the Presence Source SHALL NOT generate PUBLISH requests more often than instructed by the configured value.

5.1.2.4 Compression of a PUBLISH Request

In order to reduce the amount of access network bandwidth needed to transmit the PUBLISH request, the Presence Source implemented in a UE SHOULD support Signaling Compression (SigComp) according to [RFC3320] and updated by [RFC4896], procedures to apply SigComp to SIP according to [RFC5049] and mechanisms for discovering SigComp support at the SIP layer according to [RFC3486].

If the Presence Source implemented in a UE supports all these functionalities, the Presence Source:

- SHALL support the SIP dictionary specified in [RFC3485] and updated by [RFC4896];
- SHALL support the Presence-specific static dictionary specified in [RFC5112];
- SHALL use both dictionaries to compress the first message;
- SHALL send compressed SIP messages in accordance with [RFC3486] and [RFC5049]; and
- MAY support the negative acknowledgement mechanism specified in [RFC4077].

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the signaling compression procedures as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] SHALL be used.

5.1.2.5 Optimizing Publication of Presence Information

The Presence Source MAY support optimizing publication of Presence Information as described in this section. If supported, the Presence Source:

- SHALL be co-located with a Watcher Information Subscriber, which subscribes to the Watcher Information of each Presentity that the Presence Source is publishing on behalf of, and provides the Watcher Information notifications to the Presence Source; and
- MAY, if the Presence Source is implemented in a network element, support the procedure for the handling of requests to trigger Subscription to Watcher Information as described in section 5.1.2.5.1.

The Presence Source SHALL publish Presence Information only upon receiving an indication in the Watcher Information notification that there is at least one authorized Subscribed-watcher or Fetcher for the Presentity who is subscribed for particular Presence Information this Presence Source is responsible for publishing.

The Watcher Information notification can further include other Watcher-specific attributes that refine the publication content or the publication frequency of the Presence Source. Section 5.5.4.4 defines the full list of these attributes as extensions to

the Watcher Information Event Package. If the Watcher Information notification includes an 'application/simple-filter+xml' document, the Presence Source SHOULD publish only the Presence Information according to the received filter.

5.1.2.5.1 Handling of Requests to Trigger Subscription to Watcher Information

The Presence Source that supports optimizing publication of Presence Information MAY support triggering subscription to Watcher Information. If supported, the Presence Source SHALL support the REFER method according to [RFC3515] together with the extension defined in [RFC4488].

Before accepting a REFER request, the Presence Source SHALL perform authorization of the REFER request, per local policy. The default local policy SHOULD be to allow to trigger subscription to Watcher Information only by the PS associated with the Presentity's domain on behalf of the Presentity. This is equivalent to a REFER request where both the originator identity of the request and the Refer-To header field have the value of the Presentity URI.

In case of successful authorization, the Presence Source SHALL check the "method" parameter of the Refer-To header field. For any values other than "method=SUBSCRIBE?Event=presence.winfo", the Presence Source SHALL reject the REFER request with a 403 (Forbidden) response.

If the "method" parameter of the Refer-To header field has the value "SUBSCRIBE?Event=presence.winfo", the Presence Source:

1) SHALL accept the REFER request and send a 200 (OK) response;
2) SHALL, if the REFER request included a Refer-Sub header field set to "false", include a Refer-Sub header field set to "false" in the 200 (OK) response according to the procedures described in [RFC4488]; and
3) SHALL subscribe to the Watcher Information Event Package through the co-located Watcher Information Subscriber according to the procedures described in section 5.3.1.

5.1.2.6 PS-controlled Presence Information Re-publication

The PS can request the Presence Source to re-publish Presence Information for a previously established publication.

If PS-controlled Presence Information re-publication is supported, the Presence Source:

- SHALL include a Contact header field in PUBLISH requests when publishing Presence Information as described in section 5.1.2. The Contact header field SHALL include a SIP URI that can be used by the PS to contact the Presence Source for subsequent REFER requests. A Presence Source implemented in a UE MAY support the GRUU mechanism as specified in [IETF-GRUU]. If the Presence Source and the PS are in different domains and the Presence Source supports the GRUU mechanism, the Presence Source SHOULD populate the Contact header field of the PUBLISH request as described in [IETF-GRUU] section 4.4. When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the procedures to populate the Contact header with a GRUU are described in [3GPP-TS_24.229] section 5.1.2A;
- SHALL include an Allow header field with the value of "REFER" in PUBLISH requests when publishing Presence Information as described in section 5.1.2;
- SHALL support the REFER method according to [RFC3515] together with the extension defined in [RFC4488].

Upon receiving a REFER request, the Presence Source:

1) SHALL perform authorization of the Presence Information re-publication per local policy, before accepting the REFER request. The default local policy SHOULD be to allow Presence Information re-publication only if it is requested by the PS associated with the Presentity's domain on behalf of the Presentity. This is equivalent to a REFER request where both the originator identity of the request and the Refer-To header field have the same value of the Presentity URI. If unauthorized, the Presence Source SHALL reject the REFER request with a 403 (Forbidden) response;
2) SHALL, in case of successful authorization, check the "method" parameter of the Refer-To header field. For any other values than "method=PUBLISH?event=presence", the Presence Source SHALL reject the REFER request with a 403 (Forbidden) response;

3) SHALL check if a valid publication exists for the Presentity. In case of no publication, the Presence Source SHALL reject the REFER request with a 403 (Forbidden) response;

4) SHALL check if the REFER request includes a SIP-If-Match header field. If included, the Presence Source SHALL check if the value of the SIP-If-Match header field matches any locally stored entity-tag of an established publication. In case of no match, the Presence Source SHALL reject the REFER request with a 403 (Forbidden) response;

5) SHALL accept the REFER request and send a 200 (OK) response;

6) SHALL, if the REFER request included a Refer-Sub header field set to "false", include a Refer-Sub header field set to "false" in the 200 (OK) response according to the procedures described in [RFC4488]; and

7) SHALL perform a one-time re-publication of Presence Information for the previously established publication according to the procedures described in section 5.1.2. The re-publication can be a refresh, modify or remove operation as described in [RFC3903]. If the Presence Source maintains multiple publications for the same Presentity and the REFER request did not include a SIP-If-Match header field, the Presence Source SHALL re-publish all of the non-expired publications. If the REFER request included a SIP-If-Match header field matching a locally stored entity-tag, the Presence Source SHALL construct a PUBLISH request that includes a SIP-If-Match header field with the same entity-tag as the REFER request.

5.1.3 Manipulation of Permanent Presence State using XCAP

If the Presence Source supports the manipulation of Permanent Presence State, then the following procedures apply.

The Presence Source SHALL manipulate the Permanent Presence State via an XDMC using the Permanent Presence State Application Usage described in [PRS_PresXDM] "*Permanent Presence State*".

When manipulating Permanent Presence State, the Presence Source SHALL insert the same URI in both the "entity" attribute of the <presence> element of the presence document, and the XUI part of the Request-URI of the XCAP request.

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

- The Presence Source SHALL set the "entity" attribute of the <presence> element of the presence document, defined in [RFC3863], to the Presentity's public user identity, defined in [3GPP-TS_24.229] and [3GPP2-X.S0013-004].
- If the publication is being performed by the same Presentity, the Presence Source:
  - SHALL, if implemented in a UE, set the value of the X-3GPP-Intended-Identity header field (if included), described in [XDM_Core], of the XCAP request to the same value as the "entity" attribute of the <presence> element in the presence document.
- If the publication is being performed on behalf of another Presentity, the Presence Source:
  - SHALL, if implemented in a UE, set the value of the X-3GPP-Intended-Identity header field (if included), described in [XDM_Core], of the XCAP request to the authenticated originator identity that it intends to use for publication authorization.

5.2 Watcher

The Watcher is an entity that subscribes to Presence Information about a Presentity or list of Presentities (e.g. Presence List).

When the SIP/IP Core 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] respectively.

5.2.1 Subscription to Presence Information

5.2.1.1 General Procedures

A Watcher SHALL support subscription and notification of Presence Information, according to the subscriber procedures described in [RFC3265] and [RFC3856] with the following clarifications:

- If the Watcher is aware of the SIP URI of the Presentity, the Watcher SHOULD insert the SIP URI in the Request-URI of the SUBSCRIBE request rather than a pres URI or a tel URI; and
- If the Watcher only knows the tel URI or pres URI of the Presentity, the tel URI or pres URI may get translated to a SIP URI by the SIP/IP Core. 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.

A Watcher implemented in a UE MAY support the GRUU mechanism as specified in [IETF-GRUU]. If the Watcher and the PS are in different domains and the Watcher supports the GRUU mechanism, the Watcher SHOULD populate the Contact header field of the SUBSCRIBE request as described in [IETF-GRUU] section 4.4. When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the procedures to populate the Contact header with a GRUU are described in [3GPP-TS_24.229] section 5.1.2A.

A Watcher MAY include multiple content (e.g. 'application/resource-lists+xml' and 'application/simple-filter+xml') in the body of the SUBSCRIBE request. In this case, the Watcher SHALL implement the 'multipart/mixed' content type as described in [RFC2046], in order to aggregate the multiple content in the body of the SUBSCRIBE request.

5.2.1.2 Subscription to a Presence List and Request-contained Presence List

5.2.1.2.1 Subscription to a Presence List

Subscription to a Presence List enables 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].

NOTE: As described in section 5.6.2, the RLS can enforce a limit on the number of back-end subscriptions allowed for a single Presence List subscription, in which case the Watcher will not receive <instance> elements for those <resource> elements corresponding to Presentities that could not be subscribed by the RLS. The Watcher may be configured with the MAX-NUMBER-OF-SUBSCRIPTIONS-IN-PRESENCE-LIST parameter (defined in Appendix C) to indicate the limit being enforced by the RLS. How the Watcher makes use of this parameter is out of scope of this specification.

5.2.1.2.2 Subscription to a Request-contained Presence List

Subscription to a Request-contained Presence List enables a Watcher to subscribe to multiple Presentities using a single subscription.

A Watcher MAY support subscription to a Request-contained Presence List. If supported, the Watcher SHALL follow User Agent Client procedures as described in [RFC5367] sections "*User Agent Client Procedures*" and "*URI-List Document Format*" with the following clarifications:

- The Watcher SHALL NOT use hierarchical lists, <entry-ref> elements, and <external> elements when listing the Presentities in the SUBSCRIBE request.

  NOTE 1: [RFC5367] section "*URI-List Document Format*" states that a User Agent Client SHOULD NOT use hierarchical lists, <entry-ref> elements and <external> elements.

NOTE 2: [RFC5367] section *"Providing a URI to Manipulate a Presence List"* is outside the scope of the present specification.

The Watcher MAY be provisioned with the SIP URI of the RLS. Provisioning can be done with OTA Provisioning or local configuration. In case of OTA Provisioning, the Watcher SHALL use the value of RLS-URI (defined in Appendix C) as the value of the SUBSCRIBE Request-URI when subscribing to multiple Presentities using a Request-contained Presence List.

NOTE 3: When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, provisioning of the RLS-URI may not be necessary. The S-CSCF can route the SUBSCRIBE request to the RLS based on matching of an appropriate initial filter criteria. The value of the SUBSCRIBE Request-URI can be set to the originator's identity.

NOTE 4: As described in section 5.6.2, the RLS can enforce a limit on the number of back-end subscriptions allowed for a single Request-contained Presence List subscription, in which case the Watcher will not receive <instance> elements for those <resource> elements corresponding to Presentities that could not be subscribed by the RLS. The Watcher may be

configured with the MAX-NUMBER-OF-SUBSCRIPTIONS-IN-PRESENCE-LIST parameter (defined in Appendix C) to indicate the limit being enforced by the RLS. How the Watcher makes use of this parameter is out of scope of this specification.

5.2.2 Presence Information Processing

The Watcher SHALL support the presence data model defined in [PDE_DDS] "*Presence Data Model*", and interpret the received Presence Information according to the watcher processing rules defined in [PDE_DDS] "*Presence Information Element Definitions*".

5.2.3 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 SIP extension for partial notifications, according to the Watcher procedures described in [RFC5263]; and
- SHALL support Partial presence extension to PIDF, according to [RFC5262].

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

5.2.5 Handling of Large MIME Objects

5.2.5.1 Direct Content

A Watcher MAY implement the 'multipart/related' content type as described in [RFC2387], in order to extract different MIME objects from the body of the SIP NOTIFY request. In this case, the Watcher SHALL indicate its support for the 'multipart/related' content type by using the Accept header field in the SUBSCRIBE request.

5.2.5.2 Fetching Indirect Content

A Watcher MAY support the content indirection mechanism [RFC4483]. If supported, the Watcher:

- SHALL support the 'multipart/related' content type as described in [RFC2387]; and
- SHALL indicate its support for the 'multipart/related' and 'message/external-body' content types 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 described in [RFC4483].

If the URI of an indirect content received in the NOTIFY request is an HTTPS URI, the Watcher SHALL perform the procedures described in [RFC2818].

5.2.5.3 Fetching Presence Content from the Presence Content XDMS

The MIME object stored in the Presence Content XDMS is indicated to a Watcher using the "etag" attribute (defined in [PDE_DDS]) included in the Presence Information Element containing the URI.

A Watcher MAY support fetching the MIME object from the Presence Content XDMS. If supported, the Watcher:

- SHALL use XDMC procedures as described in [XDM_Core] "*Procedures at the XDM Client*" to retrieve the MIME object from the Presence Content XDMS; and
- SHALL, if the "etag" attribute value is different from the locally stored value, fetch the latest version of the MIME object from the Presence Content XDMS.

5.2.6 Conditional Event Notification

Conditional event notification is a mechanism that allows the Watcher to condition the subscription request to whether the state has changed since the previous notification was received. When such a condition is met, either the body of the presence event notification or the entire notification message is suppressed.

A Watcher MAY issue a conditional SUBSCRIBE request according to the subscriber procedures defined in [IETF-SubNotEtag]. If supported, the SUBSCRIBE request SHALL include a Suppress-If-Match header field to indicate the conditional subscription.

5.2.7 Event Notification Throttling

Event notification throttling is a mechanism for limiting the rate of SIP event notifications.

A Watcher subscribing to Presence Information MAY request event notification throttling. A Watcher requesting event notification throttling SHALL support the subscriber procedures described in [IETF-EventThrottle].

5.2.8 Event Notification Suppression

5.2.8.1 Direct Event Notification Suppression

Direct event notification suppression is a mechanism that enables Watchers to request the PS or RLS to suppress event notifications while keeping the corresponding event subscription state active.

The Watcher MAY request direct event notification suppression. If so, the Watcher SHALL generate the event notification suppression request according to one of the following options:

- If the Watcher supports conditional event notification procedures as described in section 5.2.6, the Watcher MAY issue a SUBSCRIBE request to refresh the subscription and include a wildcarded Suppress-If-Match header field using the special "*" entity-tag value as described in [IETF-SubNotEtag] "*Generating SUBSCRIBE Requests*"; or
- If the Watcher supports event notification throttling procedures as described in section 5.2.7, the Watcher MAY issue a SUBSCRIBE request to refresh the subscription and include a throttle parameter set to the remaining subscription expiration value as described in [IETF-EventThrottle] "*Selecting the Throttle Interval*".
- If the Watcher supports both of the above options, the Watcher SHALL indicate the presence notification suppression request using the conditional event notification procedure.

5.2.8.2 Conditional Event Notification Suppression

Conditional event notification suppression is a mechanism that enables Watchers to request the Watcher Agent to conditionally suppress event notifications based on the Watcher's own presence state. Such conditions are specified in the presence-based event notification suppression filters as defined in Appendix D.

The Watcher MAY request conditional event notification suppression. If so, the Watcher:

1) SHALL specify the conditions of its own presence state when the Watcher does not wish to receive event notifications using the presence-based event notification suppression filters; and

2) SHALL include the filters as an 'application/vnd.oma.suppnot+xml' content type in the body of the SUBSCRIBE request.

The Watcher MAY change/cancel the previously set presence-based event notification suppression filter by sending the re-SUBSCRIBE request with the updated/empty filter.

5.2.9 Compression of Subscription Signaling

5.2.9.1 Compression of the SIP Signaling

In order to reduce the amount of access network bandwidth needed to transmit the SUBSCRIBE and NOTIFY requests, the Watcher implemented in a UE SHOULD support Signaling Compression (SigComp) according to [RFC3320] and updated by [RFC4896], procedures to apply SigComp to SIP according to [RFC5049] and mechanisms for discovering SigComp support at the SIP layer according to [RFC3486].

If the Watcher implemented in a UE supports all these functionalities, the Watcher:

- SHALL support the SIP dictionary specified in [RFC3485] and updated by [RFC4896];
- SHALL support the Presence-specific static dictionary specified in [RFC5112];
- SHALL use both dictionaries to compress the first message;
- SHALL send compressed SIP messages in accordance with [RFC3486] and [RFC5049]; and
- MAY support the negative acknowledgement mechanism specified in [RFC4077].

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the signaling compression procedures as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] SHALL be used

5.2.9.2 Compression of the Body of a NOTIFY Request

A Watcher implemented in a UE subscribing for Presence Information MAY, if it does not support SIP signaling compression according to [RFC3320], [RFC4896], [RFC3485], [RFC5112], [RFC3486] and [RFC5049, or it detects that [RFC5112] is not supported by the SIP/IP Core, indicate that it supports to compress the body of a NOTIFY request by the GZIP algorithm [RFC1952] by including an Accept-Encoding header field with the value 'gzip' in the SUBSCRIBE request.

A Watcher indicating support for GZIP compression SHALL, when receiving a NOTIFY request with the Content-Encoding header field with the value 'gzip', decompress the received body as defined by [RFC1952] before performing Presence Information processing (defined in section 5.2.2).

5.3 Watcher Information Subscriber

The Watcher Information Subscriber is an entity that subscribes to the dynamically changing set of Watchers and the state of their subscriptions.

When the SIP/IP Core 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

5.3.1.1 General Procedures

A Watcher Information Subscriber

- SHALL support subscription and notification of Watcher Information, according to the subscriber procedures described in [RFC3265] and [RFC3857]; and
- SHALL support the 'application/watcherinfo+xml' content type, according to [RFC3858].

A Presentity is expected to have a Watcher Information Subscriber and maintain an active Watcher Information subscription Package to support reactive authorization; a Presentity can perform reactive authorization by being notified of the Watcher status in Watcher Information and updating the Presence Subscription Rules in Presence XDMS if it elects to allow the Watcher to access its Presence Information.

A Watcher Information Subscriber implemented in a UE MAY support the GRUU mechanism as specified in [IETF-GRUU]. If the Watcher Information Subscriber and the PS are in different domains and the Watcher Information Subscriber supports the GRUU mechanism, the Watcher Information Subscriber SHOULD populate the Contact header field of the SUBSCRIBE request as described in [IETF-GRUU] section 4.4. When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the procedures to populate the Contact header with a GRUU are described in [3GPP-TS_24.229] section 5.1.2A.

5.3.1.2 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 Event notification filtering, according to the subscriber procedures described in [RFC4660]; and
- SHALL support the ‘application/simple-filter+xml’ content type, according to [RFC4661].

5.3.1.3 Procedures when co-located with Presence Source

If the Watcher Information Subscriber is co-located with a Presence Source that supports the procedures of section 5.1.2.5, then the following applies:

The Watcher Information Subscriber SHALL support the ‘multipart/mixed’ content type according to [RFC2046] and the ‘application/simple-filter+xml’ content type according to [RFC4661], and advertise the support for these content types using the Accept header field.

The Watcher Information Subscriber SHALL subscribe to Watcher Information of each Presentity that the co-located Presence Source is publishing on behalf of, as follows:

- If the co-located Presence Source is implemented in a UE, the Watcher Information Subscriber SHALL maintain an active subscription for the Watcher Information.
- If the co-located Presence Source is implemented in a network element, the Watcher Information Subscriber SHALL, based on local policy, do one of the following:
    - Trigger a subscription for the Watcher Information on receipt of a REFER request as described in section 5.1.2.5.1. In this case, the Watcher Information Subscriber SHALL maintain the subscription for the Watcher Information as long as there is at least one active Watcher for the Presentity.
    - Maintain an active subscription for the Watcher Information.

The Watcher Information Subscriber SHALL provide the received Watcher Information notifications to the co-located Presence Source.

NOTE: The interface between the Watcher Information Subscriber and co-located Presence Source is out of scope of this specification.

5.3.1.3.1 Subscription to a Request-contained Watcher Information List

Subscription to a Request-contained Watcher Information List enables a Watcher Information Subscriber to subscribe to multiple Presentities using a single subscription.

A Watcher Information Subscriber co-located with a Presence Source MAY support subscription to a Request-contained Watcher Information List. If supported, the Watcher Information Subscriber SHALL follow User Agent Client procedures as described in [RFC5367] sections “*User Agent Client Procedures*” and “*URI-List Document Format*” with the following clarifications:

- The Watcher Information Subscriber SHALL NOT use hierarchical lists, <entry-ref> elements, and <external> elements when listing the Presentities in the SUBSCRIBE request.

  NOTE 1: [RFC5367] section "*URI-List Document Format*" states that a User Agent Client SHOULD NOT use hierarchical lists, <entry-ref> elements and <external> elements.

NOTE 2: [RFC5367] section "*Providing a URI to Manipulate a Presence List*" is outside the scope of the present specification.

The Watcher Information Subscriber MAY be configured with the SIP URI of the RLS. If configured, the Watcher Information Subscriber SHALL insert the configured value to the Request-URI of the SUBSCRIBE request when subscribing to multiple Presentities using a Request-contained Presence List.

NOTE 3: When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, configuring of the SIP URI of the RLS may not be necessary. The S-CSCF can route the SUBSCRIBE request to the RLS based on matching of an appropriate initial filter criteria. The value of the SUBSCRIBE Request-URI can be set to the originator's identity.

5.3.2 Conditional Event Notification

Conditional event notification is a mechanism that allows the Watcher Information Subscriber to condition the subscription request to whether the state has changed since the previous notification was received. When such a condition is met, either the body of the presence event notification or the entire notification message is suppressed.

A Watcher Information Subscriber MAY issue a conditional SUBSCRIBE request according to the subscriber procedures defined in [IETF-SubNotEtag]. If supported, the SUBSCRIBE request SHALL include a Suppress-If-Match header field to indicate the conditional subscription.

5.3.3 Compression of Watcher Information Signaling

5.3.3.1 Compression of SIP Signaling

In order to reduce the amount of access network bandwidth needed to transmit the SUBSCRIBE and NOTIFY requests, the Watcher Information Subscriber implemented in a UE SHOULD support Signaling Compression (SigComp) according to [RFC3320] and updated by [RFC4896], procedures to apply SigComp to SIP according to [RFC5049] and mechanisms for discovering SigComp support at the SIP layer according to [RFC3486].

If the Watcher Information Subscriber implemented in a UE supports all these functionalities, the Watcher Information Subscriber:

- SHALL support the SIP dictionary specified in [RFC3485] and updated by [RFC4896];
- SHALL support the Presence-specific static dictionary specified in [RFC5112];
- SHALL use both dictionaries to compress the first message;
- SHALL send compressed SIP messages in accordance with [RFC3486] and [RFC5049]; and
- MAY support the negative acknowledgement mechanism specified in [RFC4077].

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the signaling compression procedures as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] SHALL be used

5.3.3.2 Compression of the Body of a NOTIFY Request

A Watcher Information Subscriber implemented in a UE subscribing for Watcher Information MAY, if it does not support SIP signaling compression according to [RFC3320], [RFC3485], [RFC5112] and [RFC3486] or detects that [RFC5112] is not supported by the SIP/IP Core, indicate that it supports that the body of a NOTIFY request is compressed by the GZIP algorithm [RFC1952] by including an Accept-Encoding header field with the value 'gzip' in the SUBSCRIBE request.

A Watcher Information Subscriber indicating support for GZIP compression SHALL, when receiving a NOTIFY request with the Content-Encoding header field with the value 'gzip', decompress the received body as defined by [RFC1952] before performing Presence Information processing (defined in section 5.2.2).

5.4 Watcher Agent

The Watcher Agent is an entity that controls the Watcher's access to the Presence Service and optimizes the notification traffic based on the Watcher's preferences or local policy.

5.4.1 Watcher Service Authorization

Upon receiving the SUBSCRIBE request from a Watcher, the Watcher Agent:

1) SHALL, if a local policy for Watcher service authorization exists, check whether the Watcher is authorized to use the Presence Service per the local policy and generate a 403 (Forbidden) response to the Watcher if authorization fails;
2) SHALL, if limiting the number of subscriptions is supported, perform the procedures of section 5.4.2;
3) SHALL, if event notification suppression is supported, perform the procedures of section 5.4.3; and
4) SHALL, if event notification suppression is not supported, forward the received SUBSCRIBE request to the SIP/IP Core.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the above functionalities of the Watcher Agent MAY be implemented in the P-CSCF and/or S-CSCF as defined in [3GPP-TS_23.141] and [3GPP2-X.S0027-001] respectively.

NOTE: The method how the SUBSCRIBE request is routed to the Watcher Agent depends on the underlying SIP/IP Core and is out of scope of this specification.

5.4.2 Limiting the Number of Subscriptions

The Watcher Agent MAY have a local policy to limit the maximum number of simultaneous subscriptions for a Watcher. If the Watcher Agent determines to reject an initial subscription due to the current number of active subscriptions initiated by the Watcher being equal to or greater than the maximum, the Watcher Agent SHALL send a 480 (Maximum number of subscriptions exceeded) response. The response MAY include the Retry-After header field (e.g. based on the expiry of active subscriptions initiated by the Watcher) in order to suggest to the Watcher not to retry the subscription prior to the Retry-After time.

5.4.3 Handling of Event Notification Suppression

The Watcher Agent MAY support event notification suppression. If supported, the Watcher Agent:

- SHALL support the procedures described in section 5.2.8.1 to suppress notifications at the PS or RLS; and
- SHALL support the handling of event notification suppression conditions. These conditions MAY be based on a local policy, or supplied by the presence-based event notification suppression filters set by the Watcher as described in Appendix D.1, or the combination of local policy and the presence-based event notification suppression filters.

Upon successful authorization of the SUBSCRIBE request from a Watcher, the Watcher Agent:

1) SHALL check whether the body contains a valid 'application/vnd.oma.suppnot+xml' content as described in Appendix D.1 or whether there is any other event notification suppression conditions set by the local policy. In case of invalid content and no local policy the Watcher Agent SHALL forward the SUBSCRIBE request targeted to the PS or RLS;
2) SHALL, in case of a valid 'application/vnd.oma.suppnot+xml' content or event notification suppression conditions by local policy, terminate the SUBSCRIBE request, install the subscription and send a 202 (Accepted) response to the Watcher as described in [RFC3265]. The Watcher Agent SHALL also extract the presence-based event notification suppression filters from the 'application/vnd.oma.suppnot+xml' content;
3) SHALL generate a back-end presence subscription request targeted to the PS or RLS according to the procedures described in section 5.2.1. If there are other contents than the 'application/vnd.oma.suppnot+xml' content in the

body of the received SUBSCRIBE request the back-end presence subscription request SHALL include those contents in the body of the request;

4) SHALL generate a presence subscription request to the Watcher's Presence Information according to the procedures described in section 5.2.1; and

5) SHALL, upon receiving a response for the back-end presence subscription from the PS or RLS, send a NOTIFY request to the Watcher containing a Subscription-State header with the value of 'active'.

During the Watcher's subscription lifetime, the Watcher Agent:

1) SHALL evaluate the presence-based event notification suppression filters against the Watcher's Presence Information; and

2) SHALL, if a match is found, request the PS or RLS to suppress the presence notifications according to the procedures described in 5.2.8.1.

5.5 Presence Server

The Presence Server (PS) is an entity that accepts, stores and distributes Presence Information.

When the SIP/IP Core 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] respectively.

5.5.1 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].

5.5.1.1 Applying Presence Publication

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

Before accepting a PUBLISH request, the PS:

- SHALL perform identity verification of the Presence Source; and
- SHALL perform publication authorization as described in section 5.5.3.1.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, 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] section 5.7.1.4.

If the Presentity is identified by a SIP URI and also a pres URI or a tel URI, the PS SHALL consider these URIs to be 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].

5.5.1.2 Handling of Partial Publications

The PS SHALL support partial publication.

If the Presence Source generates a partial publication request as described in 5.1.2.1 using the 'application/pidf-diff+xml' content type defined in [RFC5262] the PS SHALL process the PUBLISH request in accordance with [RFC3903] and [RFC5264].

5.5.1.3 Handling of Large MIME Objects

The PS MAY support direct MIME objects in a presence publication. If supported, the PS SHALL support the 'multipart/related' content type in accordance with [RFC2387].

The PS MAY support indirect MIME objects in a presence publication. If supported, the PS:

- SHALL support the 'multipart/related' content type in accordance with [RFC2387]; and
- SHALL support the 'message/external-body' content type and content indirection in accordance with [RFC4483].

The PS SHALL process a presence document represented as 'multipart/related' content type as follows:

- If the 'multipart/related' content type is supported and it contains a direct MIME object, the PS:
  - SHALL stop processing and return the 413 (Request Entity Too Large) response if the size of the direct MIME object exceeds the limit defined by PS local policy for the Presence Source; or
  - SHALL either store the MIME object in case of an initial publication or replace an existing MIME object in case of a modify operation if the size of the direct MIME object is within the PS's limit.
- If the 'multipart/related' content type is supported and it contains an indirect MIME object included in a 'message/external-body' content type, the PS:
  - SHALL associate the value of the relevant Presence Information Element with the external content if the content indirection [RFC4483] mechanism is supported by the PS; or
  - SHALL send a 415 (Unsupported Media Type) response and indicate the supported content types in the Accept header field if the content indirection [RFC4483] mechanism is not supported by the PS.
- If the 'multipart/related' content type is not supported, the PS SHALL send a 415 (Unsupported Media Type) response and indicate the supported content types in the Accept header field.

5.5.1.4 Permanent Presence State

The PS MAY support Permanent Presence State. If supported, the PS SHALL use the Permanent Presence State as input for Presence Information processing. The PS SHALL ensure it has the latest available Permanent Presence State when applying the composition policy. It MAY do so by subscribing to or fetching the Permanent Presence State document from the Presence XDMS.

When fetching the Permanent Presence State document, the PS SHALL use the procedures defined in [XDM_Core] "*Document Management*". When constructing the HTTP GET request, the PS:

1) SHALL set the XCAP Root URI as described in [XDM_Core];
2) SHALL set the AUID to "pidf-manipulation" as defined in [PRS_PresXDM];
3) SHALL set the XUI to the SIP URI or tel URI of the Presentity;
4) SHALL set the document name to "perm-presence" as defined in [PRS_PresXDM]; and
5) SHALL set the X-3GPP-Asserted-Identity header field as defined in [3GPP-TS_24.109] or the X-XCAP-Asserted-Identity header field as defined in [XDM_Core] to the SIP URI or tel URI of the Presentity.

If a <timestamp> element exists in a <tuple> element, <person> element or <device> element part of the Permanent Presence State, the PS SHALL ignore its value and remove the <timestamp> element respectively before using the Permanent Presence State as input for Presence Information processing.

5.5.1.5 PS-controlled Presence Information Re-publication

The PS MAY have a local policy containing conditions for Presence Information re-publication.

NOTE: An example local policy condition can include the following: an initial or refresh Watcher subscription occurs and the PS determines that an established publication is older than a predefined value and the expiration of the established publication is also later than another predefined value.

When the local policy conditions are present and evaluate to true, the PS SHALL check if the previous PUBLISH request from the Presence Source included an Allow header field with the value of "REFER". If included, the PS MAY, depending on local policy, issue a REFER request to trigger to re-publish Presence Information for the previously established publication according to the procedures in [RFC3515] and [RFC4488].

For the REFER request the PS:

1) SHALL set the Request-URI to the SIP URI from the Contact header field of the previous PUBLISH request;
2) SHALL set the Refer-To header field to the Presentity URI whose Presence Information the PS is requesting to re-publish;
3) SHALL set the "method" parameter of the Refer-To header field to the value "PUBLISH?Event=presence";
4) SHALL include a Refer-Sub header field set to "false" according to the procedures described in [RFC4488];
5) SHALL, if the Presence Source maintains multiple publications for the same Presentity and not all of the publications are requested to be re-published, include a SIP-If-Match header field containing the entity-tag of the publication to be re-published; and
6) SHALL set the originator identity of the REFER request to the Presentity URI as if the request was sent by the Presentity.

When the SIP/IP Core corresponds to 3GPP IMS or 3GPP2 MMD networks, the PS SHALL follow the procedures described in section 5.7.3 of [3GPP-TS_24.229] and [3GPP2-X.S0013-004] and insert the Presentity URI in the P-Asserted-Identity header field used in the REFER request.

5.5.2 Presence Event Package

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

- In case of an initial subscription for the presence event package which includes the Expires header set to "0", if:
  - the Watcher Information Subscriber co-located with the Presence Source is subscribed for the Watcher Information Event Package and indicated its support for "application/simple-filter+xml" content type as described in section 5.3.1.3;
  - the PS triggers the Presence Source to subscribe for Watcher Information in order to support optimizing publication of Presence Information as described in section 5.5.4.2; or
  - the PS triggers the Presence Source to re-publish the Presence Information as described in section 5.1.2.6,

  the PS SHALL, opposed to [RFC3265], delay sending the NOTIFY request for a local policy defined time in order to allow the optimized publication of Presence Information or PS-controlled Presence Information re-publication.

The PS MAY have a local policy to limit the maximum number of simultaneous subscriptions for a Presentity. If the PS determines to reject an intial subscription due to the current number of active subscriptions to the Presentity being equal to or greater than the maximum, the PS SHALL send a 503 (Maximum number of subscriptions exceeded) response. The response SHOULD include the Retry-After header field (e.g. based on the expiry of active subscriptions), in order to suggest to the Watcher not to retry the subscription prior to the Retry-After time.

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 5.5.3.3. If the PS accepts the SUBSCRIBE request, the PS SHALL process the SUBSCRIBE request in accordance with [RFC3265] and [RFC3856] with the following clarification:

- the PS SHALL NOT terminate a subscription because the Presence Information of the Presentity being monitored does not exist. This allows a Watcher to remain subscribed to the Presentity and receive its Presence Information whenever it is available.

If the Presentity is identified by a SIP URI and also a pres URI or a tel URI, the PS SHALL consider these URIs equivalent for the purposes of presence event package subscriptions.

5.5.2.1 Handling of Large MIME Objects

If the Presence Information formatted as 'application/pidf+xml' or 'application/pidf-diff+xml' includes references to other MIME objects included as direct content, the PS:

- SHALL generate notifications using the 'multipart/related' content type in accordance with [RFC2387], if the Watcher indicated support for the 'multipart/related' content type using the Accept header field in the SUBSCRIBE request; or
- SHALL exclude the MIME object from the notification if the Watcher did not indicate support for the 'multipart/related' content type using the Accept header field in the SUBSCRIBE request.

If the Presence Information formatted as 'application/pidf+xml' or 'application/pidf-diff+xml' includes references to other MIME objects included as indirect content, the PS:

- SHALL generate notifications using content indirection in accordance with [RFC4483], if the Watcher indicated support for the 'multipart/related' and 'message/external-body' content types using the Accept header field in the SUBSCRIBE request;
- SHALL fetch the content using the HTTP GET method defined in [RFC2616] and include as direct content in the notification, if the Watcher indicated support for the 'multipart/related' content type using the Accept header field in the SUBSCRIBE request; or
- SHALL exclude the MIME object from the notification if the Watcher did not indicate support for the 'multipart/related' content type using the Accept header field in the SUBSCRIBE request.

When sending the MIME object as direct content, the PS SHALL modify the value of the relevant Presence Information Element in the presence document to refer to the MIME object included in the 'multipart/related' content type.

If content indirection is used in a notification, access to the 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.

If the size of the MIME object in the NOTIFY request exceeds a maximum limit defined by the local policy, then the PS

- SHALL handle the MIME object data as indirect content, i.e. store the MIME object in the Content Server and include an HTTP URI, or optionally HTTPS URI, in the notification pointing to the stored MIME object, if the Watcher indicated support for the 'multipart/related' and 'message/external-body' content types using the Accept header field in the SUBSCRIBE request; or
- SHALL exclude the MIME object from the notification, if the Watcher did not indicate support for the 'message/external-body' content type.

5.5.3 Presence Information Processing

The PS processes the Presence Information published by the Presence Sources before delivering it to the Watchers by applying the following steps in this order (see Figure 1):

1) Presence Publication Rules (see section 5.5.3.1)
2) Composition Policy (see section 5.5.3.2)
3) Presence Subscription Rules (see section 5.5.3.3)
4) Event notification suppression (see section 5.5.3.4)
5) Event notification filtering (see section 5.5.3.5)
6) Event notification throttling (see section 5.5.3.6)

7) Partial notification processing (see section 5.5.3.7)

8) Entity-tag generation (see section 5.5.3.8)

9) Notification generation (see section 5.5.3.9)

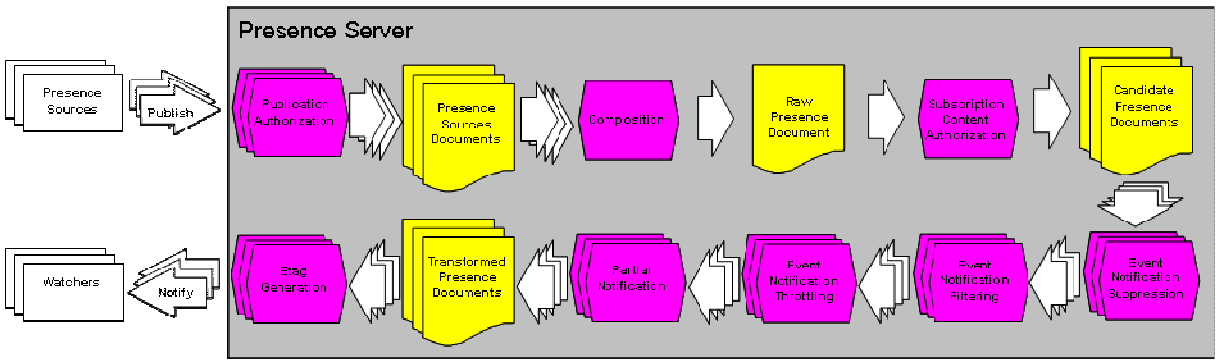

Figure 1: Presence Information Processing Steps

5.5.3.1 Applying Presence Publication Rules

The PS MAY support Presence Publication Rules. If supported, the PS SHALL verify if the Presence Publication Rules document exists in the Presence XDMS.

In case the PS does not support Presence Publication Rules or the document does not exist, the PS SHALL apply the default publication authorization policy. The default publication authorization policy SHALL authorize the publication if the authenticated originator identity is the Presentity, and SHOULD reject the publication if the authenticated originator identity is any user other than the Presentity.

In case the Presence Publication Rules document exists in the Presence XDMS, the PS SHALL apply the Presence Publication Rules to all authenticated PUBLISH requests for the presence event package.

NOTE: Applying Presence Publication Rules when publishing Permanent Presence State is described in [PRS_PresXDM] "*Authorization Policies*".

When the Presentity changes the Presence Publication Rules, the PS SHALL ensure it applies the rules with the most recent changes (see section 5.5.5).

When a PUBLISH request is received for the presence event package, the PS SHALL fetch the Presentity's Presence Publication Rules document stored in the Presence XDMS according to the procedures defined in [XDM_Core] "*Document Management*". When constructing the HTTP GET request, the PS:

- SHALL set the XCAP Root URI as defined in [XDM_Core];
- SHALL set the AUID to "org.openmobilealliance.pub-rules" as defined in [PRS_PresXDM];
- SHALL set the XUI to the SIP URI or tel URI of the Presentity;
- SHALL set the document name to "pub-rules" as defined in [PRS_PresXDM]; and
- SHALL set the X-3GPP-Asserted-Identity header field as defined in [3GPP-TS_24.109] or the X-XCAP-Asserted-Identity header field as defined in [XDM_Core] to the SIP URI or tel URI of the Presentity.

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

The PS SHALL determine which rules in the Presence Publication Rules document are applicable and evaluate the combined permissions according to the procedures described in [XDM_Core] "*Combining Permissions*" with the following clarifications:

- When realized in 3GPP IMS or 3GPP2 MMD networks, the PS SHALL use the received P-Asserted-Identity header field (as defined in [3GPP-TS_24.229] and [3GPP2-X.S0013-004]) in the PUBLISH request to determine the URI value(s) used for matching against a conditions element.
- The PS SHALL reject presence publications that are identified as anonymous (see section 6.1.4).
- If an attempt to resolve an <external-list> condition element fails, the PS SHALL regard the Publication Authorization 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 publication according to the default publication authorization policy.

The PS MAY determine that the Presence Publication Rules have been updated by subscribing to changes made to XML documents stored in the Presence XDMS and Shared List XDMS.

5.5.3.1.1 Applying Publication Authorization Rules

As defined in [PRS_PresXDM] the Publication Authorization Rules defined by the Presentity determine which other users are allowed to publish the Presentity's Presence Information.

After evaluating the combined permissions the PS SHALL handle the publication from this Presence Source based on the value of the <pub-handling> action as follows:

- if the value is "block" or there is no value, then the PS SHALL reject the publication by responding to the PUBLISH request with a 403 (Forbidden) response according to procedures described in [RFC3903] section 6; and
- if the value is "allow", then the PS SHALL apply the Publication Content Rules according to procedures described in section 5.5.3.1.2.

The PS SHALL also perform authorization of the publication by verifying that the identity of the Request-URI of the PUBLISH request matches against the value of the "entity" attribute of the <presence> element in the presence document as described in [RFC3863]. In case of no match, the PS SHALL reject the publication by responding to the PUBLISH request with a 403 (Forbidden) response according to procedures described in [RFC3903] section 6.

5.5.3.1.2 Applying Publication Content Rules

As defined in [PRS_PresXDM], the Publication Content Rules determine the subset of the Presentity's Presence Information the Presence Source is allowed to publish. The PS SHALL apply the Publication Content Rules after applying the Publication Authorization Rules by checking the <transformations> element of the combined permissions as specified in [PRS_PresXDM].

The PS SHALL evaluate the published Presence Information against the Publication Content Rules:

- If the published Presence Information conforms to the Publication Content Rules, the PS SHALL accept the publication by responding to the PUBLISH request with a 200 (OK) response according to procedures described in [RFC3903] section 6.
- If at least part of the published Presence Information does not conform to the Publication Content Rules, the PS SHALL reject the PUBLISH request with a 488 (Not Acceptable Here) response according to [IETF-SessionPol] and include a Policy-Contact header field as defined in [IETF-SessionPol] to convey the URI of the Publication Content Rules Presence Source View document stored in the Presence XDMS.

5.5.3.2 Applying Composition Policy

The function of the composition is to combine different input publications from various Presence Sources into a single raw presence document for a particular Presentity.

The PS SHALL use the following input to composition:

- publications from PUBLISH requests, as described in section 5.5.1.1, if available; and
- Permanent Presence State, as described in section 5.5.1.4, if available.

The PS SHALL support the presence data model defined in [PDE_DDS] "*Presence Data Model*".

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 a <timestamp> element value on publication refreshes.

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

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.

5.5.3.2.1 Composition Policy

The PS SHALL compose the Presence Information from the different Presence Sources according to the following rules, based on the "service", "device", and "person" components of the presence data model (see [PDE_DDS] "*Presence Data Model*"):

- Service component:

  If the following conditions all apply:

  - If one <tuple> element includes a <contact> element, other <tuple> elements include an identical <contact> element;
  - If one <tuple> element includes a <service-description> element, other <tuple> elements include an identical <service-description> element. Two <service-description> elements are considered identical if they contain identical <service-id> and <version> elements;
  - If one <tuple> element includes a <servcaps> element with an <audio> element valued "true", other <tuple> elements include an identical <servcaps> element;
  - If one <tuple> element includes a <servcaps> element with a <video> element valued "true", other <tuple> elements include an identical <servcaps> element;
  - If one <tuple> element includes a <class> element, other <tuple> elements include an identical <class> element; and
  - If 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:

  1) SHALL 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;
  2) SHALL 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;
  3) SHALL set the <timestamp> of the aggregated <tuple> to the most recent one among the ones that contribute to the aggregation (a <tuple> element without a <timestamp> element corresponds with a <tuple> element with the oldest <timestamp> element); and
  4) SHALL 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 component:

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

  1) SHALL aggregate the non-conflicting elements within one <device> element;

  2) SHALL set the <timestamp> of the aggregated <device> element to the most recent one among the ones that contribute to the aggregation (a <device> element without a <timestamp> element corresponds with a <device> element with the oldest <timestamp>); and

  3) SHALL use the element from the most recent publication for conflicting elements.

- Person component:

  If the following conditions all apply:

  - If one <person> element includes a <class> element, other <person> elements include an identical <class> element; and
  - If there are no conflicting elements (same elements with different values or attributes) under the <person> elements. Identical elements with the same value SHALL not be duplicated. Different <timestamp> values are not considered as a conflict;

  then the PS:

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

  2) SHALL set the <timestamp> of the aggregated <person> element to the most recent one among the ones that contribute to the aggregation (a <person> element without a <timestamp> element corresponds with a <person> element with the oldest <timestamp> element during comparison).

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

The PS SHALL ignore the values of the "id" (instance identifier) attributes of <tuple>, <person> and <device> elements when applying composition policy.

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

5.5.3.3 Applying Presence Subscription Rules

The authorization decision in the PS SHALL be determined based on authorization policies defined by the service provider (local policy) and the Presence Subscription Rules document stored in the Presence XDMS.

Presence Information is considered very sensitive personal information; therefore, an authorization mechanism SHALL be supported.

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

When the Presentity changes the Presence Subscription Rules, the PS SHALL ensure it applies the Presence Subscription Rules with those most recent changes (see section 5.5.5).

As defined in [PRS_PresXDM] the Presence Subscription Rules has two parts defined by the Presentity:

- Subscription Authorization Rules, which determine if a Watcher is allowed to subscribe to the Presentity's Presence Information; and
- Subscription Content Rules, which determine the subset of the Presentity's Presence Information the Watcher is allowed to receive.

When a SUBSCRIBE request is received for the presence event package, the PS SHALL fetch the Presentity's Presence Subscription Rules document stored in the Presence XDMS according to the procedures defined in [XDM_Core] "*Document Management*". When constructing the HTTP GET request, the PS:

- SHALL set the XCAP Root URI as defined in [XDM_Core];
- SHALL set the AUID to "org.openmobilealliance.pres-rules" as defined in [PRS_PresXDM];
- SHALL set the XUI to the SIP URI or tel URI of the Presentity;
- SHALL set the document name to "pres-rules" as defined in [PRS_PresXDM]; and
- SHALL set the X-3GPP-Asserted-Identity header field as defined in [3GPP-TS_24.109] or the X-XCAP-Asserted-Identity header field as defined in [XDM_Core] to the SIP URI or tel URI of the Presentity.

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

The PS SHALL determine which rules in the Presence Subscription Rules document are applicable and evaluate the combined permissions according to the procedures described in [XDM_Core] "*Combining Permissions*", with the following clarifications:

- When realized in 3GPP IMS or 3GPP2 MMD networks, the PS SHALL use the received P-Asserted-Identity header field (as defined in [3GPP-TS_24.229] and [3GPP2-X.S0013-004]) in the SUBSCRIBE request to determine the URI value(s) used for matching against a conditions element.
- If a presence subscription is identified as anonymous (see section 6.1.4), the PS SHALL evaluate the rule with the <anonymous-request> condition element (if present) as defined in [XDM_Core].
- If an attempt to resolve an <external-list> condition element fails, the PS SHALL regard the Presence Subscription 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 according to rules and procedures of [RFC5025], section 3.2;
- if the value is "polite-block", then the PS SHALL politely block the subscription following the procedures defined in section 5.5.3.3.1;
- if the value is "confirm", then the PS SHALL place the subscription in the "pending" state according to rules and procedures of [RFC5025], section 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 authorization" from the Presentity; and
- if the value is "allow", then the PS SHALL place the subscription in the "active" state according to rules and procedures of [RFC5025], section 3.2 and apply the Subscription Content Rules defined under the "transformations" element of the matched rules as specified in [PRS_PresXDM].

While a Watcher's subscription is 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 an active subscription to the Presentity, the PS terminates the subscription and blocks any future subscription requests from this Watcher.

Furthermore, while a Watcher's subscription is active, a Presentity may update its Subscription Content Rules. The PS SHALL re-determine the subset of the Presentity's Presence Information the Watcher is allowed to receive. For example, a

Presentity may decide to stop disseminating specific Presence Information Elements to its Watchers. In such a case the PS will generate presence notifications that will omit those specific Presence Information Elements.

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

5.5.3.3.1 Polite Blocking

Polite blocking is a mechanism to deny providing Presence Information updates , while indicating to the Watcher that the subscription is active.

If the result of applying Subscription Authorization Rules is to perform polite blocking (see section 5.5.3.3), the PS:

1) SHALL respond to the SUBSCRIBE request according to rules and procedures of [RFC5025], section 3.2; and

2) SHALL then send only one NOTIFY request with the following content:

   a) 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 [PDE_DDS] "*Presence Information Element Definitions*") for 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;

   b) not provide the <device> and <person> elements if existing in the Presentity's "raw presence document"; and

   c) perform all the subsequent steps in the Presence Information processing framework, as they are listed in section 5.5.3 and detailed in relevant sub-sections (e.g. apply filtering, partial notifications, throttling, etc).

5.5.3.4 Applying Event Notification Suppression

The PS SHALL support event notification suppression according to the procedures described in this section.

If the PS receives a SUBSCRIBE request including:

- a wildcarded Suppress-If-Match header field using the special "*" entity-tag value as described in [IETF-SubNotEtag] "*Generating SUBSCRIBE Requests*"; or
- a throttle parameter set to the remaining subscription expiration value as described in [IETF-EventThrottle] "*Selecting the Throttle Interval*",

the PS SHALL suppress the generation of event notifications until a Watcher cancels the suppression with a re-SUBSCRIBE request or the subscription state changes.

5.5.3.5 Applying Event Notification Filtering

The PS SHOULD 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 body 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 5.5.3.3, followed by the event notification filtering procedures described in this section.
- does not understand the particular filter included in the body of the SUBSCRIBE request as requested by the Watcher, the PS SHALL indicate it to the Watcher as specified in [RFC4660].

5.5.3.6 Applying Event Notification Throttling

The PS MAY have a local throttling configuration setting that limit the rate at which notifications are generated (i.e. the shortest time period between two NOTIFY requests for a given Watcher). In this case, the PS SHALL NOT generate NOTIFY requests more often than the throttling configuration dictates, except when generating the notification either upon receipt of a SUBSCRIBE request or upon subscription state changes.

The PS SHALL support Watcher requested event notification throttling. The PS SHALL follow the notifier procedures described in [IETF-EventThrottle].

If the PS has a local throttling configuration setting and it is lower than the Watcher proposed throttle value, the PS SHALL accept the Watcher proposed throttle value.

If the local throttling configuration setting is higher than the Watcher proposed throttle value, the PS SHALL adjust the Watcher proposed throttle value to the local throttling configuration setting and send it back to the Watcher as described in [IETF-EventThrottle].

5.5.3.7 Applying Partial Notification

The PS SHALL support partial notifications. If the Watcher indicates support for partial notifications in the SUBSCRIBE request for the presence event package, the PS SHALL generate partial notifications in accordance with [RFC5263] and [RFC5262].

5.5.3.8 Generating Entity Tags

The PS SHALL support the notifier procedures defined in [IETF-SubNotEtag]. The PS:

- SHALL generate entity tags for presence documents. The entity tag SHALL be unique to the presence document over time, i.e the PS SHALL generate the same entity tag for the same presence document in different time samples. The algorithm to generate such entity tags is out of scope of this specification.

  NOTE: The presence document here refers to the document the PS generates after "Event Notification Filtering" as shown in Figure 1. Several Watchers can receive the same presence document if they share common Presence Subscription Rules and apply the same event notification filtering.

- SHALL include the entity tag in all NOTIFY requests as described in [IETF-SubNotEtag].

5.5.3.9 Generation of Notifications

At the last step of Presence Information processing, the PS SHALL generate new NOTIFY requests 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 to the Watcher.

If a Watcher requested a condition for suppressing a NOTIFY request or a NOTIFY request body using the Suppress-If-Match header field and the condition evaluates to true, the PS SHALL suppress the NOTIFY request or the NOTIFY request body appropriately as described in [IETF-SubNotEtag].

If the PS supports view sharing and determines to use it for this subscription, it SHALL follow the procedures in [IETF-ViewShare] section 4.

The PS 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.

5.5.4 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 originator's identity is equal to the Presentity URI or a configured URI of a Presence Source that is publishing on behalf of the Presentity, and to reject all other subscriptions. The PS SHALL reject Watcher Information subscriptions that are identified as anonymous (see section 6.1.4).

If the PS accepts the SUBSCRIBE request, the PS SHALL process the SUBSCRIBE request in accordance with [RFC3265], [RFC3857], and [RFC3858] with the exceptions and clarifications described below.

Contrary to [RFC3857], the PS SHALL generate Watcher Information notifications when the Watcher Information state machine defined in [RFC3857] moves from "init" to "active" state, even if the change is transient. This behaviour enables the PS to also include authorized Fetchers in Watcher Information notifications. When indicating the existence of a Fetcher, the PS SHALL include the "expiration" attribute set to "0" in the appropriate <watcher> element. This information can be utilized by Presence Sources co-located with a Watcher Information Subscriber when optimizing publication of Presence Information according to section 5.1.2.5.

The PS SHALL indicate the existence of a presence subscription requesting privacy as defined by section 6.1.4 by including a <watcher> element as defined in [RFC3858] containing a SIP URI with the the value "sip:anonymous@anonymous.invalid" in the NOTIFY request sent to the Watcher Information Subscriber.

5.5.4.1 Applying Event Notification Filtering

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

- Event notification filtering, according to the server 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 body of the SUBSCRIBE request, the PS SHALL apply the requested filter.
- does not understand the particular filter included in the body of the SUBSCRIBE request, the PS SHALL indicate it to the Watcher Information Subscriber as specified in [RFC4660] and [RFC4661].

5.5.4.2 Generating Entity Tags

The PS SHALL support the notifier procedures defined in [IETF-SubNotEtag]. The PS:

- SHALL generate entity tags for Watcher Information documents. The entity tag SHALL be unique to the Watcher Information document over time, i.e the PS SHALL generate the same entity tag for the same Watcher Information document in different time samples. The algorithm to generate such entity tags is out of scope of this specification.
- SHALL include the entity tag in all NOTIFY requests as described in [IETF-SubNotEtag].

5.5.4.3 Triggering Subscription to Watcher Information

The PS MAY be configured with a (list of) URI(s) of Presence Sources which implement the optimized publication of Presence Information according to section 5.1.2.5. If configured and the Watcher Information state changes from having no authorized Subscribed-watchers or Fetchers to having at least one authorized Subscribed-watcher or Fetcher, the PS SHALL send a REFER request towards those URIs which do not maintain an active Watcher Information subscription for the Presentity.

The REFER request SHALL be formulated according to the procedures in [RFC3515] and [RFC4488]. For each REFER request the PS:

1) SHALL set the Request-URI to the configured URI;

   NOTE: the local configuration may include multiple target URIs. In that case, multiple REFER requests will be issued.

2) SHALL set the Refer-To header field to the Presentity URI;
3) SHALL set the "method" parameter of the Refer-To header field to the value "SUBSCRIBE?Event=presence.winfo";
4) SHALL include a Refer-Sub header field set to "false" according to the procedures described in [RFC4488]; and

5) SHALL set the originator identity of the request to the Presentity URI as if the request was sent on behalf of the Presentity.

When the SIP/IP Core corresponds to 3GPP IMS or 3GPP2 MMD networks, the PS SHALL follow the procedures described in section 5.7.3 of [3GPP-TS_24.229] and [3GPP2-X.S0013-004] and insert the Presentity URI in the P-Asserted-Identity header field used in the REFER request.

5.5.4.4 Watcher Information Content

In order to support the optimized publication of Presence Information as described in section 5.1.2.5, the PS SHOULD support additional content in Watcher Information notifications in addition to the basic 'application/watcherinfo+xml' format defined in [RFC3858]. If supported and the Watcher Information Subscriber advertised support for the additional content using the Accept header field with values "multipart/mixed" and "application/simple-filter+xml", the PS:

- SHALL combine presence event notification filtering information from all Watchers, including the namespace binding synchronization, in one 'application/simple-filter+xml' [RFC4661] document with the following restrictions:
    - The <filter-set> element SHALL include only one <filter> child element;
    - The <filter> element SHALL include only the <what> child element;
    - The <filter> element SHALL include the "uri" attribute with the URI identifying the Presentity; and
    - The <filter> element SHALL NOT include the "enabled" and "domain" attributes.

  The PS MAY also consider the Presence Information content the Watchers are authorized to see and local policy restrictions for generating the combined filtering document. The resulting combined filtering document is the one which filters the largest set of Presence Information from the presence document; and
- SHALL include the resulting combined filtering document in the notification.

If the additional content is included in Watcher Information notifications, such content SHALL be included in 'multipart/mixed' content according to [RFC2046].

5.5.5 XDM Functions

Certain PS functionality depends on particular XML documents stored in the Presence XDMS and Shared List XDMS. In order to provide this functionality the PS:

- SHALL support retrieval of XML documents stored in the Presence XDMS and Shared List XDMS, according to [XDM_Core] "*Document Management*" (via the PRS-8 and PRS-5 reference points, respectively);
- SHALL support the Presence Subscription Rules Application Usage as specified in [PRS_PresXDM] "*Presence Subscription Rules*", and the URI List Application Usage as specified in [XDM_List] "*URI List*";

If the PS supports the Permanent Presence State functionality as specified in section 5.5.1.4, the PS SHALL support the Permanent Presence State Application Usage as specified in [PRS_PresXDM] "*Permanent Presence State*".

If the PS supports the Presence Publication Rules functionality as specified in section 5.5.3.1, the PS SHALL support the Presence Publication Rules Application Usage as specified in [PRS_PresXDM] "*Presence Publication Rules*".

The PS MAY subscribe to changes made to XML documents stored in the Presence XDMS and Shared List XDMS. If so, the PS SHALL follow the procedure defined in [XDM_Core] "*Subscribing to Changes in the XML Documents*" (via the PRS-3 reference point).

5.5.6 Compression of Presence Traffic

5.5.6.1 Compression of the Body of a NOTIFY Request

If a received SUBSCRIBE request contains an Accept-Encoding header field with the value 'gzip', the PS SHALL, dependent on local policy, compress the NOTIFY request body using the GZIP algorithm [RFC1952] and add a Content-Encoding header field with the value 'gzip' to the NOTIFY request before sending the NOTIFY request to the SIP/IP Core.

5.6 Resource List Server

The Resource List Server (RLS) accepts and manages subscriptions to:

- Presence Lists and Request-contained Presence Lists, which enable a Watcher to subscribe to the Presence Information of multiple Presentities using a single subscription; and
- Request-contained Watcher Information Lists, which enable a Watcher Information Subscriber to subscribe to the Watcher Information of multiple Presentities using a single subscription.

When the SIP/IP Core 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] respectively.

5.6.1 General

The RLS:

- SHALL support subscriptions to Presence Lists, according to the RLS procedures described in [RFC4662];
- MAY support subscriptions to Request-contained Presence Lists according to the RLS procedures described in [RFC5367] sections "*URI-List Document Format*" and "*Resource List Server Behavior*"; and
- MAY support subscriptions to Request-contained Watcher Information Lists according to the RLS procedures described in [RFC5367] sections "*URI-List Document Format*" and "*Resource List Server Behavior*".

If subscriptions to Request-contained Presence Lists or Request-contained Watcher Information Lists are supported , the RLS SHALL support "multipart/mixed" content type in SUBSCRIBE requests as described in [RFC2046].

The RLS SHALL, before accepting a subscription to a Presence List, perform authorization of the usage of the Presence List by the Watcher, per local policy. If the Presence List subscription is authorized, the RLS SHALL resolve the Presence List into individual Presentities according to section 5.6.6.

When sending a list notification, the RLS SHALL set the "uri" attribute of each <resource> element included in the RLMI document to the URI of the Presentity in the Presence List, Request-contained Presence List or Request-contained Watcher Information List.

NOTE: If a Presentity is identified by a pres URI or a tel URI in the Presence List, Request-contained Presence List or Request-contained Watcher Information List, the pres URI or the tel URI is included in the RLMI document even if the RLS has knowledge of an equivalent SIP URI.

5.6.2 Back-end Subscriptions

If the RLS supports the view sharing procedures described in [IETF-ViewShare] section 3, it SHOULD indicate support for this extension and optimize the number of back-end subscriptions.

For back-end subscriptions using SIP, the RLS:

- SHALL support the 'application/pidf+xml' content type, according to [RFC3863];

- if the Request-contained Watcher Information List is supported, SHALL support the 'application/watcherinfo+xml' content type according to [RFC3858], the 'multipart/mixed' content type according to [RFC2046] and the 'application/simple-filter+xml' content type according to [RFC4661];
- SHALL support subscription and notification of Presence Information, according to the subscriber procedures described in [RFC3265] and [RFC3856];
- if the Request-contained Watcher Information List is supported, SHALL support subscription and notification of Watcher Information, according to the subscriber procedures described in [RFC3265] and [RFC3857];
- SHALL support SIP extension for partial notifications, according to the Watcher procedures described in [RFC5263] and partial presence extension to PIDF, according to [RFC5262];
- SHOULD support event notification filtering, according to the procedures described in section 5.6.3;
- SHALL support the 'multipart/related' content type as described in [RFC2387] and advertise its support for the 'multipart/related' content type by using the Accept header field in the SUBSCRIBE request for the back-end subscription;
- SHALL support the content indirection mechanism described in [RFC4483]. If the Watcher advertised the support for the 'message/external-body' content type by using the Accept header field in the SUBSCRIBE request, the RLS SHALL advertise the support for the 'message/external-body' content type by using the Accept header field in the SUBSCRIBE request for the back-end subscription;
- SHALL support conditional subscriptions according to the subscriber procedures described in [IETF-SubNotEtag];
- SHALL support event notification supression according to the procedures in described in section 5.6.4; and
- MAY indicate that it supports that the body of a NOTIFY request is compressed by the GZIP algorithm [RFC1952] by including an Accept-Encoding header field with the value 'gzip' in the SUBSCRIBE request. An RLS indicating support for GZIP compression SHALL, when receiving a NOTIFY request with the Content-Encoding header field with the value 'gzip', decompress the received body as defined by [RFC1952] before processing the body of the SIP NOTIFY request.

When the SIP/IP Core 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.S0013-004] and insert a URI value from the P-Asserted-Identity header field of the incoming SUBSCRIBE request (as defined in [3GPP-TS_24.229] and [3GPP2-X.S0013-004]) to the SUBSCRIBE request of the back-end subscription, as opposed to acting as an authentication service (as defined in [RFC4474]) as required by [RFC4662].

If the list subscription is identified as anonymous (see section 6.1.4), the RLS SHALL generate back-end subscriptions as anonymous using the Watcher Privacy procedures as defined in section 6.1.1.

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
- SHALL 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 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 list.

When the Watcher removes Presentities from the 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 list.

The Presence List can be changed either directly, when the Presence List document stored in RLS XDMS is updated, or indirectly, when the URI List stored in the Shared List XDMS and referenced in the Presence List document is updated.

When the Watcher refreshes the subscription, the RLS SHOULD refresh the back-end subscriptions accordingly. The RLS SHOULD try to re-generate the back-end subscriptions for those Presentities whose corresponding <resource> element in the last list notification:

- did not include an <instance> element, if the omission was not caused by a limit to the maximum number of back-end subscriptions; or
- included an <instance> element whose "state" attribute was set to "terminated".

5.6.3 Event Notification Filtering

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

- Event notification filtering, according to the RLS and notifier procedures described in [RFC4660] with the clarifications described in this section; 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, if the filter element contains a "uri" attribute and its value matches with the URI of a Presentity in the Presence List, Request-contained Presence List or Request-contained Watcher Information List, supply the filter document in the back-end subscription to the matching Presentity;
    - SHALL, if the filter element contains a "domain" attribute and its value matches with the domain of a set of Presentities in the Presence List, Request-contained Presence List or Request-contained Watcher Information List, supply the filter document in the back-end subscriptions to the Presentities matching the "domain" attribute, but not matching the "uri" attribute in other filters in the filter-set; and
    - SHALL, if the filter element does not contain a "uri" or "domain" attribute, supply the filter document in the back-end subscriptions to all Presentities in the Presence List, Request-contained Presence List or Request-contained Watcher Information List not matching a "uri" or a "domain" attribute in other filters in the filter-set.
- does not understand the particular filter included in the payload of the SUBSCRIBE request as requested by the Watcher or Watcher Information Subscriber, the RLS SHALL indicate it to the Watcher or Watcher Information Subscriber as specified in [RFC4660].

For every filter propagated in a back-end subscription targeted to a Presentity, the RLS SHALL remove the "uri" or "domain" attribute if included in the RLS filter obtained from the Watcher or Watcher Information Subscriber.

5.6.4 Conditional Event Notifications

The RLS SHALL support the notifier procedures defined in [IETF-SubNotEtag].

5.6.4.1 Generating Entity Tags

The RLS:

- SHALL generate entity tags for the full resource list state including the RLMI and the list of presence documents or Watcher Information documents as described in [IETF-SubNotEtag] "*List Subscriptions*"; and
- SHALL include the entity tag in all NOTIFY requests as described in [IETF-SubNotEtag].

5.6.4.2 Generation of Notifications

If the Watcher or Watcher Information Subscriber requested condition for suppressing a NOTIFY request or a NOTIFY request body evaluates to true, the RLS SHALL suppress the NOTIFY request or the NOTIFY request body appropriately as described in [IETF-SubNotEtag].

5.6.5 Handling of Event Notification Suppression

The RLS SHALL support event notification suppression according to the procedures described in this section.

If the RLS receives a SUBSCRIBE request including:

- a wildcarded Suppress-If-Match header field using the special "*" entity-tag value as described in [IETF-SubNotEtag] "*Generating SUBSCRIBE Requests*"; or
- a throttle parameter set to the remaining subscription expiration value as described in [IETF-EventThrottle] "*Selecting the Throttle Interval*",

the RLS SHALL suppress the generation of event notifications until a Watcher or Watcher Information Subscriber cancels the suppression with a re-SUBSCRIBE request or the subscription state changes.

5.6.6 XDM Functions

In order to resolve Presence Lists into individual Presentities, the RLS:

- SHALL support retrieval of XML documents stored in the RLS XDMS and Shared List XDMS, according to [XDM_Core] "*Document Management*" (via the PRS-10 and PRS-9 reference points, respectively);
- SHALL support the Presence List Application Usage as specified in [PRS_RLSXDM] "*Presence List*"; and
- SHALL support the URI List Application Usage as specified in [XDM_List] "*URI List*".

On receiving a SUBSCRIBE request directed at a Presence List identified by a Request-URI, the RLS either:

- SHALL access the "index" document in the Global Tree using the XCAP path [XCAP Root URI]/rls-services/global/index; or
- SHALL access the "index" document in the Users Tree using the XCAP path [XCAP Root URI]/rls-services/users/[XUI]/index, if the RLS has knowledge about the XUI of the Primary Principal of the Presence List.

  NOTE: The latter procedure may be preferred when the RLS has a need to handle multiple Presence Lists owned by a single Primary Principal (i.e. contained in the same "index" document) and the XUI is known (e.g. included as part of the Presence List URI in the Request-URI of the SUBSCRIBE request as defined by the URI template in [XDM_Core] "*Provisioned XDMC Parameters*").

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 List XDMS. If the RLS is unable to retrieve a URI List from the Shared List 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 List XDMS. If so, the RLS SHALL follow the procedures defined in [XDM_Core] "*Subscribing to Changes in the XML Documents*" (via the PRS-4 reference point).

When realized in 3GPP IMS or 3GPP2 MMD networks, the RLS SHALL insert a URI from the received P-Asserted-Identity header field (as defined in [3GPP-TS_24.229] and [3GPP2-X.S0013-004]) from the SUBSCRIBE request in the X-3GPP-Asserted-Identity header field, as defined in [3GPP-TS_24.109] or the X-XCAP-Asserted-Identity header field as defined in [XDM_Core], of the HTTP GET request.

5.6.7 Applying Event Notification Throttling

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

The RLS MAY have local throttling configuration settings that limit the rate at which notification are generated (i.e. the shortest time period between two NOTIFY requests). In this case, the RLS SHALL NOT generate NOTIFY requests more often than the throttling configuration dictates, except when generating the notification either upon receipt of a SUBSCRIBE request or upon subscription state changes.

The RLS SHALL support Watcher requested event notification throttling. The RLS SHALL follow the notifier procedures described in [IETF-EventThrottle].

If the RLS has a local throttling configuration setting and it is lower than the Watcher proposed throttle value, the RLS SHALL accept the Watcher proposed throttle value.

If the local throttling configuration setting is higher than the Watcher proposed throttle value, the RLS SHALL adjust the Watcher proposed throttle value to the local throttling configuration setting and send it back to the Watcher as described in [IETF-EventThrottle].

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 request) and transmit them when those restrictions expire. The mechanism by which multiple notifications are aggregated is described in [RFC4662].

5.6.8 Compression of Presence Subscription Traffic

5.6.8.1 Compression of the Body in a NOTIFY Request

If the RLS receives a SUBSCRIBE request containing an Accept-Encoding header field with the value ‘gzip’, the RLS SHALL, dependent on local policy, compress the NOTIFY request body using the GZIP algorithm [RFC1952] and add a Content-Encoding header field with the value ‘gzip’ to the NOTIFY request before sending the NOTIFY request to the SIP/IP Core.

5.7 XDM Client

The XDMC SHALL support the following:

- XDMC procedures described in [XDM_Core] “*Procedures at the XDM Client*”;
- Presence Subscription Rules Application Usage as specified in [PRS_PresXDM] “*Presence Subscription Rules*”;
- Presence List Application Usage as specified in [PRS_RLSXDM] “*Presence List*”; and
- URI List Application Usage as specified in [XDM_List] “*URI List*”.

The XDMC MAY support the following:

- Permanent Presence State Application Usage as specified in [PRS_PresXDM] “*Permanent Presence State*”;
- Presence Content Application Usage as specified in [PRS_ContXDM] “*Presence Content*”; and
- Presence Publication Rules as specified in [PRS_PresXDM] “*Presence Publication Rules*”.

5.8 Presence XDMS

The Presence XDMS SHALL support the XDMS procedures described in [XDM_Core] “*Procedures at the XDM Server*”, and the Application Usages described in [PRS_PresXDM].

5.9 RLS XDMS

The RLS XDMS SHALL support the XDMS procedures described in [XDM_Core] "*Procedures at the XDM Server*", and the Application Usages described in [PRS_RLSXDM].

5.10 Content Server

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

When processing an HTTP PUT request, the Content Server SHALL store a MIME object when received in the HTTP PUT request behind the HTTP URI therein.

When processing an HTTP GET request, the Content Server:

- SHALL return a MIME object in a 200 (OK) response; and
- 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 5.1.2.2, Watchers as described in 5.2.5 and the PS as described in sections 5.5.1.3 and 5.5.2.1.

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

5.11 Presence Content XDMS

The Presence Content XDMS SHALL support the XDMS procedures described in [XDM_Core] "*Procedures at the XDM Server*", and the Application Usage described in [PRS_ContXDM].

An XDMC performing a retrieve operation of the Presence Content document from the "oma_status-icon" subfolder in the Users Tree of the Presence Content Application Usage SHALL be authorized using Presence Subscription Rules as specified in [PRS_ContXDM] "*Authorization Policies*".

6. Security

The security mechanism provides protection to the presence service environment.

6.1 Privacy

6.1.1 Watcher Privacy

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

The Watcher MAY indicate further privacy preferences in accordance with [RFC3323].

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

6.1.2 Watcher Information Subscriber Privacy

Watcher Information Subscriber privacy is not supported.

6.1.3 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 5.5.3.3.

6.1.4 Anonymous SIP Request

The PS and RLS SHALL consider a SIP request as anonymous if it contains a From header indicating an anonymous value as defined in [RFC3323].

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the PS SHALL follow the procedures described in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] section 5.7.1.4 to determine if the SIP request is anonymous.

6.2 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 to accomplish user identity verification.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks the authentication mechanism SHALL be as specified in [3GPP-TS_33.203] / [3GPP2-S.R0086], and the PS or RLS:

- SHALL authenticate the SIP request originator as specified in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] section 5.7.1.4; and
- SHALL, when acting on behalf of the Presence Source or the Watcher, populate security related SIP header fields according to the procedures given in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] 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] section 5.7.3.

6.3 Integrity and Confidentiality Protection

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

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the integrity and confidentiality protection mechanism is specified in [3GPP-TS_33.203] / [3GPP2-S.R0086].

7. Charging

7.1 Charging Architecture

Since both online and offline charging SHOULD be supported according to [PRS_RD], there are two different charging architectures, as described in the following sub-sections.

7.1.1 Offline Charging Architecture

When the SIP/IP Core 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] [3GPP2-X.S0013-008] for 3GPP2.

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

7.1.2 Online Charging Architecture

When the SIP/IP Core 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] [3GPP2-X.S0013-008] for 3GPP2.

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

8. Registration

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the Presence Source, the Watcher and the Watcher Information Subscriber implemented in a UE SHALL use the 3GPP IMS or 3GPP2 MMD networks registration mechanisms as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] with the following clarifications:

- If a Presence Source implemented in a UE supports the PS-controlled retrieval of Presence Information re-publication as described in section 5.1.2.6, the Presence Source MAY support the GRUU mechanism as specified [IETF-GRUU]. If the Presence Source and the PS are in different domains and the Presence Source supports the GRUU mechanism, the Presence Source SHOULD obtain a GRUU as described in section 5.1.1.2.1 of [3GPP-TS_24.229] for the purpose of populating the Contact header field of the PUBLISH request as described in section 5.1.2.6.
- A Watcher MAY support the GRUU mechanism as specified [IETF-GRUU]. If the Watcher and the PS are in different domains and the Watcher supports the GRUU mechanism, the Watcher SHOULD obtain a GRUU as described in section 5.1.1.2.1 of [3GPP-TS_24.229] for the purpose of populating the Contact header field of the SUBSCRIBE request as described in sections 5.2.1, 5.2.1.1.1 and 5.2.1.1.2.
- A Watcher Information Subscriber MAY support the GRUU mechanism as specified [IETF-GRUU]. If the Watcher Information Subscriber and the PS are in different domains and the Watcher Information Subscriber supports the GRUU mechanism, the Watcher Information Subscriber SHOULD obtain a GRUU as described in section 5.1.1.2.1 of [3GPP-TS_24.229] for the purpose of populating the Contact header field of the SUBSCRIBE request as described in section 5.3.1.1.

In a non-3GPP/3GPP2 network, this document has the following requirements regarding the SIP registration procedures:

- If a Presence Source implemented in a UE supports the PS-controlled retrieval of Presence Information re-publication as described in section 5.1.2.6, the Presence Source MAY support the GRUU mechanism as specified [IETF-GRUU]. If the Presence Source and the PS are in different domains and the Presence Source supports the GRUU mechanism, the Presence Source SHOULD obtain a GRUU as described in section 4.1 of [IETF-GRUU] for the purpose of populating the Contact header field of the PUBLISH request as described in section 5.1.2.6.
- A Watcher .MAY support the GRUU mechanism as specified [IETF-GRUU]. If the Watcher and the PS are in different domains and the Watcher supports the GRUU mechanism, the Watcher SHOULD obtain a GRUU as described in section 4.1 of [IETF-GRUU] for the purpose of populating the Contact header field of the SUBSCRIBE request as described in sections 5.2.1, 5.2.1.1.1 and 5.2.1.1.2.

- A Watcher Information Subscriber MAY support the GRUU mechanism as specified [IETF-GRUU]. If the Watcher Information Subscriber and the PS are in different domains and the Watcher Information Subscriber supports the GRUU mechanism, the Watcher Information Subscriber SHOULD obtain a GRUU as described in section 4.1 of [IETF-GRUU] for the purpose of populating the Contact header field of the SUBSCRIBE request as described in section 5.3.1.1.

9. Content of the Presence Document

The presence data model and the content of the presence document is described in [PDE_DDS].

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

A.2 Draft/Candidate Version 2.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-TS-Presence_SIMPLE-V2_0	14 Feb 2006	All	Created based on OMA-TS-Presence_SIMPLE-V1_0-20060110, agreed through OMA-PAG-2006-0054.
	12 Apr 2006	10.4.8.3 10.5.1.2 10.5.1.3 10.5.1.4 10.4.15.4 10.4.16.3 10.4.16.5 B1 5.4.3.1.1 5.3.1	Incorporated CR OMA-PAG-2006-0085 Incorporated CR OMA-PAG-2006-0103 Incorporated CR OMA-PAG-2006-0104 Incorporated CR OMA-PAG-2006-0105 Incorporated CR OMA-PAG-2006-0157 Incorporated CR OMA-PAG-2006-0196
	21 Jun 2006	2.1	Incorporated CRs: OMA-PAG-2006-0228 OMA-PAG-2006-0229 OMA-PAG-2006-0230 OMA-PAG-2006-0231 OMA-PAG-2006-0339 OMA-PAG-2006-0305R01 OMA-PAG-2006-0307R03 OMA-PAG-2006-0363 OMA-PAG-2006-0367 OMA-PAG-2006-0374R01 OMA-PAG-2006-0376R01 OMA-PAG-2006-0377OMA-PAG-2006-0381
	13 Jul 2006	2.2, App. E	Incorporated CRs: OMA-PAG-2006-0325R02
	02 Aug 2006	2.1, 4, 5.1.1.2, 5.2.6.1, 5.4.1.3, 5.9,	Incorporated CR: OMA-PAG-2006-0393R02

Document Identifier	Date	Sections	Description
	01 Sep 2006	2.1, 4, 5.1.1, 5.2.3, 10.1, 10.1, 10.1.1, 10.1.2, 10.1.3, 10.3,, 10.4.13.3, 10.4.13.4, 10.4.15.3, 10.4.16.3, 10.5.1.4 10.4, 10.4.6.3 10.4.7.3 10.4.9.3 10.4.9.5 10.4.10.3 10.4.11.3 10.4.14.3 A.6 2.1 B.1 2.1 Appendix B Appendix C 10.6, 10.3, 2.1, 5.1.1 5.2.1 5.4.1.2 5.4.3.6 2.1 10.4 10.5.1 10.2 5.4, 5.4.1.1 5.4.3.1, 10.1	Incorporated CRs: OMA-PAG-2006-0417 OMA-PAG-2006-0415 OMA-PAG-2006-0453R03 OMA-PAG-2006-0470R03 OMA-PAG-2006-0480 OMA-PAG-2006-0457R02 OMA-PAG-2006-0459R01 OMA-PAG-2006-0455R01 OMA-PAG-2006-0454R01 OMA-PAG-2006-0452 OMA-PAG-2006-0451 OMA-PAG-2006-0430
	28 Sep 2006	5.2.3 10.4.17, 10.5.2, 2.1, 4, 5.2.2.2, 5.5.1, 5.5.2, 5.5.5	Incorporated CRs: OMA-PAG-2006-0429R05 OMA-PAG-2006-0515
	10 Nov 2006	2.1, 2.2 4 5.2.5 5.3.1.1 5.4.3.3 5.5.3, 10.1.2.1 10.3 10.4.17 C.1.2.8	Incorporated CRs: OMA-PAG-2006-0188R06 OMA-PAG-2006-0543 OMA-PAG-2006-0544

Document Identifier	Date	Sections	Description
	23 Nov 2006	2.1, 4, 5.4.1.5, 5.4.5, 5.5.1, 5.6, 10.3, 10.4, 10.6	Incorporated CRs: OMA-PAG-2006-0638R03 OMA-PAG-2006-0663 OMA-PAG-2006-0725 OMA-PAG-2006-0730 OMA-PAG-2006-0778
	17 Jan 2007	2.1, 2.2, 3.2, 5.1.1, 5.1.1.2 5.1.1.2, 5.2.6, 5.4.1.2 5.4.2, 5.4.2.1, 5.4.3.1.1, 5.4.3.2, 5.4.3.2.1, 5.4.3.6, 5.5.1, 5.5.2, 5.5.4, 5.5.5, 10.1, 10.5.1.2, 10.6, C.1.4, C.1.6, D.5.2, D.6.2,	Incorporated CRs: OMA-PAG-2006-0780 OMA-PAG-2006-0798 OMA-PAG-2006-0802 OMA-PAG-2006-0838 OMA-PAG-2006-0862R01 Editorial corrections to references.
	19 Feb 2007	5.4.2, 6.1.1, 6.1.2	Incorporated CRs: OMA-PAG-2007-0011R01 OMA-PAG-2007-0060
	21 May 2007	2.2, 9.1, 9.2, 5.4.1.5,	Incorporated CRs: OMA-PAG-2007-0170R02 OMA-PAG-2007-0212R03
	01 Jul 2007	2.1, 2.2, 5.1.1.2, 5.1.1.2.1, 5.1.1.2.2, 5.3.7, 5.4, 5.4.2, 5.4.2.2, 5.4.3, 5.4.3.4, 5.4.3.6, 5.4.3,7, 5.5.1, 5.5.2, 10.4, 10.5.1	Incorporated CRs: OMA-PAG-2007-0297 OMA-PAG-2007-0346R01 OMA-PAG-2007-0370 OMA-PAG-2007-0442R01 OMA-PAG-2007-0477
	04 Sep 2007	5.2.8, 5.4, 5.4.3, 5.4.3.5, 5.5.5,	Incorporated CRs: OMA-PAG-2007-0368R03 OMA-PAG-2007-0492 OMA-PAG-2007-0505R01 OMA-PAG-2007-0506R01 OMA-PAG-2007-0580

Document Identifier	Date	Sections	Description
	11 Oct 2007	2.1, 5.4.3.2, 5.4.3.2.1	Incorporated CR: OMA-PAG-2006-0394
	28 Nov 2007	2, 4, 5, App.D	Incorporated CRs: OMA-PAG-2007-0646R02 OMA-PAG-2007-0737R03 OMA-PAG-2007-0762 OMA-PAG-2007-0763 OMA-PAG-2007-0764 OMA-PAG-2007-0766 OMA-PAG-2007-0767 OMA-PAG-2007-0768 OMA-PAG-2007-0769 OMA-PAG-2007-0770 OMA-PAG-2007-0771 OMA-PAG-2007-0778
	17 Jan 2008	2,3,4,5,6,10, App. A, D	OMA-PAG-2007-0367R02 OMA-PAG-2007-0532R06 OMA-PAG-2007-0812R03 OMA-PAG-2007-0820 OMA-PAG-2007-0821R01 OMA-PAG-2007-0862 OMA-PAG-2007-0863R01 OMA-PAG-2008-0005 OMA-PAG-2008-0010R01
	23 Jan 2008	all	Editorial corrections to references and editor's notes CR correction: OMA-PAG-2007-0821R01
	1 Feb 2008	2.1, 5.5.3.2, 5.5, 5.5.2, App. B	Editorial corrections to cross-references and formatting Incorporated CRs: OMA-PAG-2007-0816R01 OMA-PAG-0833R05 OMA-PAG-2008-0021R02 OMA-PAG-2008-0041
	5 Mar 2008	all	Incorporated CRs: OMA-PAG-2007-0738R06 OMA-PAG-2008-0024R06 OMA-PAG-2008-0028R02 OMA-PAG-2008-0035R03 OMA-PAG-2008-0096R03 OMA-PAG-2008-0109R02 OMA-PAG-2008-0117R03 OMA-PAG-2008-0135R01 OMA-PAG-2008-0142R01 OMA-PAG-2008-0144 OMA-PAG-2008-0148R01
	12 Mar 2008	All	Editorial cleanup based on OMA-PAG-2008-0154R01
	21 Apr 2008	All	Incorporated CR: OMA-PAG-2008-0235R01
	11 Jun 2008	All	Incorporated CRs: OMA-PAG-2008-0351R01 OMA-PAG-2008-0362 OMA-PAG-2008-0374 OMA-PAG-2008-0375R01 OMA-PAG-2008-0376 OMA-PAG-2008-0380 OMA-PAG-2008-0394 OMA-PAG-2008-0395

Document Identifier	Date	Sections	Description
	02 Jul 2008	All	Incorporated CRs: OMA-PAG-2008-0361R03 OMA-PAG-2008-0373R03 OMA-PAG-2008-0391R01 OMA-PAG-2008-0402R01 OMA-PAG-2008-0403 OMA-PAG-2008-0404 OMA-PAG-2008-0406 OMA-PAG-2008-0407R01 OMA-PAG-2008-0408R01 OMA-PAG-2008-0409 OMA-PAG-2008-0410 OMA-PAG-2008-0411R01 OMA-PAG-2008-0416 OMA-PAG-2008-0438R01 OMA-PAG-2008-0440R01 OMA-PAG-2008-0442R01 OMA-PAG-2008-0448 OMA-PAG-2008-0449 OMA-PAG-2008-0450R01 OMA-PAG-2008-0455R01 OMA-PAG-2008-0469R02 OMA-PAG-2008-0485R02
	06 Aug 2008	All	Incorporated CRs: OMA-PAG-2008-0494 OMA-PAG-2008-0453R02 OMA-PAG-2008-0454R01
	26 Aug 2008	All	Incorporated CRs: OMA-PAG-2008-0405R03 OMA-PAG-2008-0413R03 OMA-PAG-2008-0452R03 OMA-PAG-2008-0497R02 OMA-PAG-2008-0508R01 OMA-PAG-2008-0509R02 OMA-PAG-2008-0514R01 OMA-PAG-2008-0515 OMA-PAG-2008-0526 OMA-PAG-2008-0535R01 OMA-PAG-2008-0552R01 OMA-PAG-2008-0570R01

Document Identifier	Date	Sections	Description
	01 Oct 2008	All	Incorporated CRs: OMA-PAG-2008-0439R05 OMA-PAG-2008-0445R04 OMA-PAG-2008-0447R03 OMA-PAG-2008-0492R02 OMA-PAG-2008-0513R03 OMA-PAG-2008-0519R03 OMA-PAG-2008-0530R03 OMA-PAG-2008-0571R02 OMA-PAG-2008-0578R02 OMA-PAG-2008-0611 OMA-PAG-2008-0612R02 OMA-PAG-2008-0613R01 OMA-PAG-2008-0614R03 OMA-PAG-2008-0615R01 OMA-PAG-2008-0623R01 OMA-PAG-2008-0625 OMA-PAG-2008-0626 OMA-PAG-2008-0627R01 OMA-PAG-2008-0628R01 OMA-PAG-2008-0629R01 OMA-PAG-2008-0631R01 OMA-PAG-2008-0633R01 OMA-PAG-2008-0641R02 OMA-PAG-2008-0653 OMA-PAG-2008-0654
	14 Oct 2008	All	Incorporated CRs: OMA-PAG-2008-0441R01 OMA-PAG-2008-0525R02 OMA-PAG-2008-0616R03 OMA-PAG-2008-0630R02 OMA-PAG-2008-0632R02 OMA-PAG-2008-0634R02 OMA-PAG-2008-0637R03 OMA-PAG-2008-0639R03 OMA-PAG-2008-0656 OMA-PAG-2008-0663
	22 Oct 2008	All	Incorporated CRs: OMA-PAG-2008-0518R03 OMA-PAG-2008-0520R02 OMA-PAG-2008-0683 OMA-PAG-2008-0684 OMA-PAG-2008-0685 OMA-PAG-2008-0720R01 OMA-PAG-2008-0721 OMA-PAG-2008-0722

Document Identifier	Date	Sections	Description
	27 Oct 2008	All	Incorporated CRs: OMA-PAG-2008-0531R02 OMA-PAG-2008-0532R01 OMA-PAG-2008-0682R01 OMA-PAG-2008-0688R02 OMA-PAG-2008-0689R02 OMA-PAG-2008-0727 OMA-PAG-2008-0729R01 OMA-PAG-2008-0731 OMA-PAG-2008-0739 OMA-PAG-2008-0742 OMA-PAG-2008-0743R01 OMA-PAG-2008-0745R02 OMA-PAG-2008-0764
	05 Nov 2008	All	Incorporated CRs: OMA-PAG-2008-0715R06 OMA-PAG-2008-0754R02 OMA-PAG-2008-0770R03 OMA-PAG-2008-0773 OMA-PAG-2008-0778 OMA-PAG-2008-0783
	13 Nov 2008	All	Incorporated CR: OMA-PAG-2008-0788R01
Candidate Version OMA-TS-Presence_SIMPLE-V2_0	23 Dec 2008	N/A	Status changed to Candidate by TP TP ref # OMA-TP-2008-0490-INP_Presence_SIMPLE_V2_0_ERP_for_Candidate_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

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

- Presence Source;
- Presence Server;
- Watcher Information Subscriber;
- Resource List Server;
- Watcher;
- Watcher Agent;
- XDM Client;
- Presence XDMS; and
- RLS XDMS.

The following tags are used in the Function column to identify the release of the Presence SIMPLE enabler that the requirement was introduced:

- PRSv1.1 – Requirement was introduced in Presence SIMPLE 1.1.
- PRSv2.0 – Requirement was introduced in Presence SIMPLE 2.0.

B.1 Presence Source

Item	Function	Reference	Requirement
PRS-SRC-C-001-M	Providing Presence Information to the Presence Server (PRSv1.1)	5.1.1	PRS-SRC-C-003-O OR PRS-SRC-C-016-O
PRS-SRC-C-002-M	Presence Data Model (PRSv1.1)	5.1.1	
PRS-SRC-C-003-O	Event Publication Agent and sending SIP PUBLISH (PRSv1.1)	5.1.2	PRS-SRC-C-006-O AND (PRS-SRC-C-004-O OR PRS-SRC-C-005-O)
PRS-SRC-C-004-O	Publication performed by the same Presentity, using SIP PUBLISH (PRSv1.1)	5.1.2	
PRS-SRC-C-005-O	Publication performed on behalf of another Presentity, using SIP PUBLISH (PRSv2.0)	5.1.2	
PRS-SRC-C-006-O	application/pidf+xml content type, according to [RFC3863] (PRSv1.1)	5.1.2	
PRS-SRC-C-007-O	Partial Publication (PRSv1.1)	5.1.2.1	
PRS-SRC-C-008-O	Publishing large objects using content indirection (PRSv1.1)	5.1.2.2.1	
PRS-SRC-C-009-O	Publishing large objects using direct content (PRSv1.1)	5.1.2.2.2	
PRS-SRC-C-010-O	Publishing large objects via Presence Content XDMS (PRSv2.0)	5.1.2.2.3	
PRS-SRC-C-011-O	Limiting the rate of publications (PRSv1.1)	5.1.2.3	
PRS-SRC-C-012-O	Compression of a PUBLISH request (PRSv1.1)	5.1.2.4	
PRS-SRC-C-013-O	Optimizing publication of Presence Information (PRSv2.0)	5.1.2.5	
PRS-SRC-C-014-O	Handling of requests to Trigger Subscription to Watcher Information (PRSv2.0)	5.1.2.5.1	
PRS-SRC-C-015-O	PS-controlled Presence Information Re-publication (PRSv2.0)	5.1.2.6	
PRS-SRC-C-016-O	Manipulation of Permanent Presence State (PRSv2.0)	5.1.3	PRS-SRC-C-017-O OR PRS-SRC-C-018-O
PRS-SRC-C-017-O	Publication performed by the same Presentity, using XCAP (PRSv2.0)	5.1.3	

Item	Function	Reference	Requirement
PRS-SRC-C-018-O	Publication performed on behalf of another Presentity, using XCAP (PRSv2.0)	5.1.3	

B.2 Presence Server

Item	Function	Reference	Requirement
PRS-PS-S-001-M	Event State Compositor and handling SIP PUBLISH, according to [RFC3903] (PRSv1.1)	5.5.1 5.5.1.1	
PRS-PS-S-002-M	application/pidf+xml content type, according to [RFC3863] (PRSv1.1)	5.5.1	
PRS-PS-S-003-M	Handling presence publication (PRSv1.1)	5.5.1	
PRS-PS-S-004-O	Handling partial publications (PRSv1.1)	5.5.1.2	
PRS-PS-S-005-O	Handling publication of large objects using content indirection (PRSv1.1)	5.5.1.3	
PRS-PS-S-006-O	Handling publication of large objects using direct content (PRSv1.1)	5.5.1.3	
PRS-PS-S-007-O	Permanent Presence State (PRSv2.0)	5.5.1.4	PRS-PS-S-008-O OR PRS-PS-S-009-O
PRS-PS-S-008-O	Fetch Permanent Presence State Document (PRSv2.0)	5.5.1.4	
PRS-PS-S-009-O	Subscription to Permanent Presence State document changes (PRSv2.0)	5.5.1.4	
PRS-PS-S-010-O	PS-controlled Presence Information Re-publication (PRSv2.0)	5.5.1.5	
PRS-PS-S-011-M	Presence Information Subscriptions (PRSv1.1)	5.5.2	
PRS-PS-S-012-O	Local policy to limit the maximum number of simultaneous subscriptions to a Presentity (PRSv2.0)	5.5.2	
PRS-PS-S-013-M	Presence Information Notifications (PRSv1.1)	5.5.2 5.5.3.9	
PRS-PS-S-014-O	Handling notification of large objects using content indirection (PRSv1.1)	5.5.2.1	
PRS-PS-S-015-O	Handling notification of large objects using direct content (PRSv1.1)	5.5.2.1	
PRS-PS-S-016-O	Apply Presence Publication Rules (PRSv2.0)	5.5.3.1	PRS-PS-S-017-O OR PRS-PS-S-018-O
PRS-PS-S-017-O	Fetch Presence Publication Rules from Presence XDMS (PRSv2.0)	5.5.3.1 5.5.5	

Item	Function	Reference	Requirement
PRS-PS-S-018-O	Subscription to Presence Publication Rules changes (PRSv2.0)	5.5.3.1 5.5.5	
PRS-PS-S-019-O	Apply Publication Authorization Rules (PRSv2.0)	5.5.3.1.1	
PRS-PS-S-020-O	Apply Publication Content Rules (PRSv2.0)	5.5.3.1.2	
PRS-PS-S-021-M	Apply Composition Policy (PRSv1.1)	5.5.3.2	
PRS-PS-S-022-M	Presence Data Model (PRSv1.1)	5.5.3.2	
PRS-PS-S-023-M	Applying Presence Subscription Rules	5.5.3.3	
PRS-PS-S-024-M	Fetch Presence Subscription Rules from Presence XDMS (PRSv1.1)	5.5.3.3 5.5.5	
PRS-PS-S-025-O	Subscription to Presence Subscription Rules changes (PRSv2.0)	5.5.3.3 5.5.5	
PRS-PS-S-026-M	Fetch URI List(s) from Shared List XDMS (PRSv1.1)	5.5.3.3 5.5.5	
PRS-PS-S-027-O	Subscribe to URI List(s) changes (PRSv2.0)	5.5.3.3 5.5.5	
PRS-PS-S-028-M	Polite Blocking (PRSv1.1)	5.5.3.3.1	
PRS-PS-S-029-M	Apply event notification suppression (PRSv2.0)	5.5.3.4	
PRS-PS-S-030-O	Apply event notification filtering (PRSv1.1)	5.5.3.5	
PRS-PS-S-031-M	Apply event notification throttling according to [IETF-EventThrottle] (PRSv2.0)	5.5.3.6	
PRS-PS-S-032-M	Apply partial notification (PRSv1.1)	5.5.3.7	
PRS-PS-S-033-M	Generate entity tag according to [IETF-SubNotEtag] and suppressing notifications (PRSv2.0)	5.5.3.8	
PRS-PS-S-034-M	Watcher Information subscriptions (PRSv1.1)	5.5.4	
PRS-PS-S-035-M	Watcher Information notifications (PRSv1.1)	5.5.4	
PRS-PS-S-036-O	Triggering subscription to Watcher Information (PRSv2.0)	5.5.4.3	
PRS-PS-S-037-O	Extensions to Watcher Information content (PRSv2.0)	5.5.4.4	

Item	Function	Reference	Requirement
PRS-PS-S-038-M	Compression of NOTIFY body using gzip (PRSv2.0)	5.5.6.1	

B.3 Watcher Information Subscriber

Item	Function	Reference	Requirement
PRS-WIS-C-001-M	Watcher Information subscriptions (PRSv1.1)	5.3.1.1	
PRS-WIS-C-002-M	Watcher Information notifications (PRSv1.1)	5.3.1.1	
PRS-WIS-C-003-O	Event notification filtering (PRSv1.1)	5.3.1.2	
PRS-WIS-C-004-O	Procedures when co-located with Presence Source (PRSv2.0)	5.3.1.3	
PRS-WIS-C-005-O	Request-contained Watcher Information List Subscription (PRSv2.0)	5.3.1.3.1	
PRS-WIS-C-006-O	Conditional event notification according to [IETF-SubNotEtag] (PRSv2.0)	5.3.2	
PRS-WIS-C-007-O	Compression of a NOTIFY request (PRSv1.1)	5.3.3.1	
PRS-WIS-C-008-O	Compression of Watcher Information in a NOTIFY body using gzip (PRSv2.0)	5.3.3.2	

B.4 RLS Server

Item	Function	Reference	Requirement
PRS-RLS-S-001-M	Subscriptions to Presence List (PRSv1.1)	5.6.1	
PRS-RLS-S-002-O	Subscriptions to Request-contained Presence List according to [RFC5367] (PRSv2.0)	5.6.1	
PRS-RLS-S-003-O	Subscriptions to Request-contained Watcher Information List according to [RFC5367] (PRSv2.0)	5.6.1	
PRS-RLS-S-004-M	List notifications (PRSv1.1)	5.6.1	
PRS-RLS-S-005-M	Back-end subscriptions (PRSv1.1)	5.6.2	
PRS-RLS-S-006-O	Limiting the number of back-end subscriptions (PRSv1.1)	5.6.2	
PRS-RLS-S-007-O	Conditional back-end subscriptions (PRSv2.0)	5.6.2	
PRS-RLS-S-008-O	Event notification filtering for the back-end subscriptions (PRSv1.1)	5.6.2	

Item	Function	Reference	Requirement
PRS-RLS-S-009-M	Handling of event notification suppression for the back-end subscriptions (PRSv2.0)	5.6.2	
PRS-RLS-S-010-O	Apply event notification throttling according to [IETF-Throttle] for the back-end subscriptions (PRSv2.0)	5.6.2	
PRS-RLS-S-011-O	Event notification filtering for the list notifications (PRSv1.1)	5.6.3	
PRS-RLS-S-012-M	Conditional event notifications for the list notifications (PRSv2.0)	5.6.4	
PRS-RLS-S-013-M	Handling of event notification suppression for the list notifications (PRSv2.0)	5.6.5	
PRS-RLS-S-014-M	Fetch Presence List from Presence XDMS (PRSv1.1)	5.6.6	
PRS-RLS-S-015-O	Subscription to Presence List changes (PRSv2.0)	5.6.6	
PRS-RLS-S-016-M	Fetch URI List(s) from Shared List XDMS (PRSv1.1)	5.6.6	
PRS-RLS-S-017-O	Subscribe to URI List(s) changes (PRSv2.0)	5.6.6	
PRS-RLS-S-018-M	Apply event notification throttling according to [IETF-Throttle] for the list notifications (PRSv2.0)	5.6.7	
PRS-RLS-S-019-O	Compression of NOTIFY body using gzip (PRSv2.0)	5.6.8.1	

B.5 Watcher

Item	Function	Reference	Requirement
PRS-WTR-C-001-M	Presence subscriptions (PRSv1.1)	5.2.1	
PRS-WTR-C-002-O	Presence List subscriptions (PRSv1.1)	5.2.1.2.1	
PRS-WTR-C-003-O	Request-contained Presence List Subscription (PRSv2.0)	5.2.1.2.2	
PRS-WTR-C-004-M	Presence notifications (PRSv1.1)	5.2.1	
PRS-WTR-C-005-O	Presence List notifications (PRSv1.1)	5.2.1.2.1	
PRS-WTR-C-006-M	Presence Information Processing based on Presence Data Model (PRSv1.1)	5.2.2	
PRS-WTR-C-007-O	Partial notification (PRSv1.1)	5.2.3	
PRS-WTR-C-008-O	Event notification filtering (PRSv1.1)	5.2.4	

Item	Function	Reference	Requirement
PRS-WTR-C-009-O	Handling notification of large objects using direct content (PRSv1.1)	5.2.5.1	
PRS-WTR-C-010-O	Handling notification of large objects using content indirection (PRSv1.1)	5.2.5.2	
PRS-WTR-C-011-O	Handling notification of large objects via Presence Content XDMS (PRSv2.0)	5.2.5.3	
PRS-WTR-C-012-O	Conditional event notification according to [IETF-SubNotEtag] (PRSv2.0)	5.2.6	
PRS-WTR-C-013-O	Event notification throttling according to [IETF-EventThrottling] (PRSv2.0)	5.2.7	
PRS-WTR-C-014-O	Direct event notification suppression (PRSv2.0)	5.2.8.1	
PRS-WTR-C-015-O	Conditional event notification suppression (PRSv2.0)	5.2.8.2	
PRS-WTR-C-016-O	Presence-based event notification suppression filter (PRSv2.0)	D.1	
PRS-WTR-C-017-O	Compression of a NOTIFY request (PRSv1.1)	5.2.9.1	
PRS-WTR-C-018-O	Compression of Presence Information in a NOTIFY body using gzip (PRSv2.0)	5.2.9.2	

B.6 Watcher Agent

Item	Function	Reference	Requirement
PRS-WA-S-001-O	Watcher service authorization (PRSv2.0)	5.4.1	
PRS-WA-S-002-O	Limiting number of subscriptions (PRSv2.0)	5.4.2	
PRS-WA-S-003-O	Handling of event notification suppression (PRSv2.0)	5.4.3	
PRS-WA-S-004-O	Presence-based event notification suppression filter (PRSv2.0)	5.4.3	PRS-WA-S-003-O
PRS-WA-S-005-O	Direct event notification suppression (PRSv2.0)	5.4.3	PRS-WTR-C-014-O

B.7 XDM Client

Item	Function	Reference	Requirement
PRS-XDM-C-001-M	Mandatory XDMC functions (PRSv1.1)	5.7	
PRS-XDM-C-002-O	Optional XDMC functions (PRSv1.1)	5.7	

Item	Function	Reference	Requirement
PRS-XDM-C-003-M	Presence Subscription Rules Application Usage (PRSv1.1)	5.7	
PRS-XDM-C-004-M	Presence List Application Usage (PRSv1.1)	5.7	
PRS-XDM-C-005-M	URI List Application Usage (PRSv1.1)	5.7	
PRS-XDM-C-006-O	Permanent Presence State Application Usage (PRSv2.0)	5.7	
PRS-XDM-C-007-O	Presence Content Application Usage (PRSv2.0)	5.7	
PRS-XDM-C-008-O	Presence Publication Rules Application Usage (PRSv2.0)	5.7	

B.8 Presence XDMS

Item	Function	Reference	Requirement
PRS-PRSXDM-S-001-M	Mandatory XDMS functions (PRSv1.1)	5.8	
PRS-PRSXDM-S-002-O	Optional XDMS functions (PRSv1.1)	5.8	
PRS-PRSXDM-S-003-M	Presence Subscription Rules Application Usage (PRSv1.1)	5.8	
PRS-PRSXDM-S-004-O	Permanent Presence State Application Usage (PRSv2.0)	5.8	
PRS-PRSXDM-S-005-O	Presence Publication Rules Application Usage (PRSv2.0)	5.8	
PRS-PRSXDM-S-006-M	Subscription to XML Document Changes (PRSv2.0)	5.8	

B.9 RLS XDMS

Item	Function	Reference	Requirement
PRS-RLSXDM-S-001-M	Mandatory XDMS functions (PRSv1.1)	5.9	
PRS-RLSXDM-S-002-O	Optional XDMS functions (PRSv1.1)	5.9	
PRS-RLSXDM-S-003-M	Presence List Application Usage (PRSv1.1)	5.9	
PRS-RLSXDM-S-004-M	Subscription to XML Document Changes (PRSv2.0)	5.9	

B.10 Presence Content XDMS

Item	Function	Reference	Requirement
PRS-CNTXDM-S-001-M	Mandatory XDMS functions (PRSv2.0)	5.11	
PRS-CNTXDM-S-002-O	Optional XDMS functions (PRSv2.0)	5.11	
PRS-CNTXDM-S-003-M	Presence Content Application Usage (PRSv2.0)	5.11	
PRS-CNTXDM-S-004-O	Subscription to XML Document Changes (PRSv2.0)	5.11	

Appendix C. Presence Client Provisioning (Normative)

This appendix specifies the parameters that are needed for initiation of Presence Service by the presence client, as well as continuous provisioning by the Service Provider. These parameters are specified in Client Provisioning Application Characteristics document (AC file) [CP_ProvCont] and Device Management Management Objects (DM MOs) [DM_StdObj]. Existing parameters in [CP_ProvCont] and [DM_StdObj] are re-used; those without corresponding parameters are defined and to be registered in OMNA through the OMA official registration process.

The AC file or DM MOs MAY be used for initial provisioning of parameters as specified in [DM_ERELD], and the DM MOs SHOULD be used for continuous provisioning of parameters according to [DM_ERELD], if required by the Service Provider to update service configurations.

C.1 Presence Client Provisioning Parameters

The parameters listed in the table below are needed for Presence Client provisioning:

ID	Name	Description	Mandatory (M) / Optional (O)
1	Application identity	Uniquely identifies the application	M
2	Application name	User displayable name for the Presence service	M
3	Provider–ID	Identity of the Presence Service Provider	O
4	XDM reference to SIP/IP Core	Reference to the SIP/IP Core for accessing the Presence Service using the referenced SIP/IP Core.	M
5	Network Access Definitions	Reference to the network access point used for the application.	M
6	Client Object Data Limit	Size limit of the MIME object when PUBLISH requests are used in the Presence Source.	M
7	Content Server URI	HTTP URI of the Content Server to be used for content indirection	O
8	Source Throttle Publish	Minimum time interval between two consecutive publications from a Presence Source	O
9	Max number of subscription in Presence List	Maximum number of back-end subscriptions allowed for a Presence List	O
10	Service URI Template	Syntax of the Service URI Template as specified in [XDM_Core] “Provisioned XDMC Parameters”	O
11	RLS URI	SIP URI of the Resource List Server to be used for Request-contained Presence List subscription	O

C.2 Application Characteristics

The Application Characteristics file for PRS 2.0 service MAY be used for initial provisioning of the Presence Client.

This chapter describes the provisioning document structure as described in [CP_ProvCont].

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

Parameter Name	Req / Opt	Instances	Default
Standard Application Characteristic fields as defined in [CP_ProvCont]			
APPID	Required	1	“ap0009”
NAME	Required	1	None
PROVIDER-ID	Optional	0 or 1	None
TO-APPREF	Required	1	None
TO-NAPID	Required	1 or more	None
Application Characteristic fields specifically required for the Presence Enabler			
CLIENT-OBJ-DATA-LIMIT	Required	1	None
CONTENT-SERVER-URI	Optional	0 or 1	None
SOURCE-THROTTLE-PUBLISH	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
RLS-URI	Optional	0 or 1	None

The Application Characteristics file for PRS 2.0 service is defined in [PRS_AC].

C.3 Management Objects

The Management Objects for PRS 2.0 service MAY be used for initial provisioning of the Presence Client and SHOULD be used for continuous provisioning by the Service Provider.

The Management Objects for PRS 2.0 service is defined in [PRS_MO].

Appendix D. Common Content Types (Normative)

The common content types for this specification are described in this Appendix.

D.1 Presence-based Event Notification Suppression Filter

The presence-based event notification suppression filter specifies the conditions when the Watcher wishes not to receive event notifications. A condition is evaluated by comparing the values of the condition with the Watcher's Presence Information. If they match, the condition evaluates to true.

D.1.1 MIME Type

The MIME type for the presence-based event notification suppression filter SHALL be "application/vnd.oma.suppnot+xml".

Editor's Note: The MIME type shall be OMNA registered.

D.1.2 XML Schema

The presence-based event notification suppression filter SHALL conform to the XML schema described in [XSD_suppNot].

D.1.3 Structure and Data Semantics

The presence-based event notification suppression filter SHALL conform to the structure and semantics as described in this subclause.

The root element <suppnot-filter-set>:

a) MAY include any other attributes for the purposes of extensibility;

b) MAY include a <ns-bindings> element that contains the namespace bindings according to [RFC4661] "*The <ns-bindings> Element*";

c) SHALL include zero or more <suppnot-filter> elements that contain the conditions for event notification suppression.

The <ns-bindings> element:

a) SHALL include one or more <ns-binding> elements, each of which SHALL contain the binding between the prefix and the namespace in a "prefix" attribute and a "namespace" attribute, respectively. This is used to express the XPATH formed Presence Information Elements or Presence Information Element attributes under <presattrib> elements.

The <suppnot-filter> element:

a) SHALL include a "id" attribute that contains the unique identification for the filter;

b) MAY include any other attribute for the purposes of extensibility;

c) MAY include one or more <presattrib> elements that contain the Presence Information Elements or Presence Information Element attributes that decide the suppression of the notifications;

d) MAY include any other elements from other namespaces for the purposes of extensibility.

The <presattrib> element:

a) MAY include any other attribute for the purposes of extensibility;

b) MAY include one or more <suppress-if-match> elements, each of which contains the XPATH expression according to [RFC4661] "*Syntax for Referencing XML Items and Making Logical Expressions*", that identifies the Presence Information Elements or Presence Information Element attributes to be matched;

c) MAY include any other elements from other namespaces for the purposes of extensibility.

The <suppress-if-match> element:

a) MAY include a "type" attribute that contains the expression type of the Presence Information Elements or Presence Information Element attributes in a <suppress-if-match> element. The default value is "xpath" in case of the absence of this attribute;

b) MAY include any other attribute for the purposes of extensibility.

D.1.4 Evaluation

The evaluation of the presence-based event notification suppression filter is achieved as following:

- The empty <suppnot-filter-set> element SHALL remove any existing filters set in the Watcher Agent;
- The evaluation of each <suppnot-filter> element under the root element <suppnot-filter-set> SHALL be logically ORed;
- The evaluation of each <presattrib> child element under a <suppnot-filter> element SHALL be logically ANDed;
- The evaluation of each <suppress-if-match> child element under a <presattrib> element SHALL be logically ANDed;
- The evaluation of a <suppress-if-match> element SHALL be TRUE if the corresponding expression in the content results in identification of one or more elements in the Watcher's Presence Information;
- The evaluation of an empty <suppress-if-match> element SHALL be FALSE.

D.1.5 Examples (Informative)

The following is an example of the presence-based event notification suppression filter.

```
<?xml version="1.0" encoding="UTF-8"?>
<suppnot-filter-set xmlns="urn:oma:xml:prs:pidf:oma-suppnotfilter"
   xmlns:sf="urn:ietf:params:xml:ns:simple-filter">

  <ns-bindings>
   <sf:ns-binding prefix="pdm" urn="urn:ietf:params:xml:ns:pidf:data-model"/>
   <sf:ns-binding prefix="rpid" urn="urn:ietf:params:xml:ns:pidf:rpid"/>
   <sf:ns-binding prefix="op" urn="urn:oma:xml:prs:pidf:oma-pres"/>
   <sf:ns-binding prefix="pde" urn="urn:oma:xml:pde:pidf:ext"/>
  </ns-bindings>

  <!-- Condition1: Notification will be suppressed if the Watcher's device is participating in PoC
   session -->
  <suppnot-filter id="45i0s">
   <presattrib>
      <suppress-if-match>//pdm:tuple[*/op:service-id="org.openmobilealliance:PoC-session" and
   pdm:deviceID="urn:uuid:d27459b7-8213-4395-aa77-ed859a3e5b3a"]/op:session-
   participation[op:basic="open"]</suppress-if-match>
   </presattrib>
  </suppnot-filter>

  <!--Condition2: Notification will be suppressed if the Watcher's presence is 'away' -->
  <suppnot-filter id="fe23de">
   <presattrib>
      <suppress-if-match>//pdm:person/rpid:activities/rpid:away</suppress-if-match>
   </presattrib>
  </suppnot-filter>

  <!-- Condition3: Notification will be suppressed if the Watcher's device is under roaming -->
  <suppnot-filter id="we34is">
```

```
    <presattrib>
       <suppress-if-match>//pdm:device[pdm:deviceID="urn:uuid:d27459b7-8213-4395-aa77-ed859a3e5b3a"
    and op:network-availability/op:network/pde:visited]</suppress-if-match>
    </presattrib>
   </suppnot-filter>

</suppnot-filter-set>
```

NOTE: It is out of scope of this specification how a Presence Source residing in the network (e.g. application server) learns about the device ID of a device and publishes the corresponding <deviceID> element for the device.

Appendix E. Example Realizations of a Presence Source (Informative)

E.1 Presence User Agent

A Presence Source can be implemented as a Presence User Agent (PUA) as defined by 3GPP/3GPP2 in [3GPP-TS_23.141] and [3GPP2-X.S0027-001] 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.

E.2 Presence Network Agent

A Presence Source can be implemented as a Presence Network Agent (PNA) as defined by 3GPP/3GPP2 in [3GPP-TS_23.141] and [3GPP2-X.S0027-001] respectively. The PNA collects the network related Presence Information from the various network elements and sends it to the PS.

The interfaces between the PNA and the various elements are defined in 3GPP/3GPP2 (see Figure 2 and Figure 3) and are out of scope of the current specification.

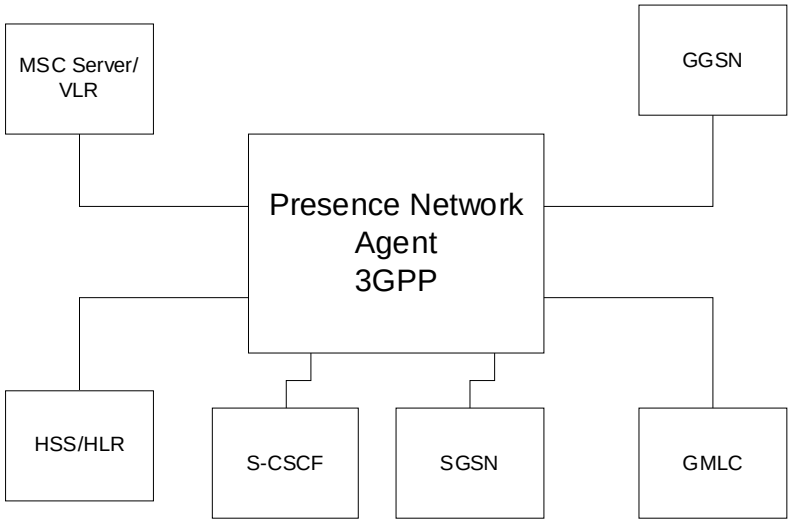

Figure 2: PNA in 3GPP

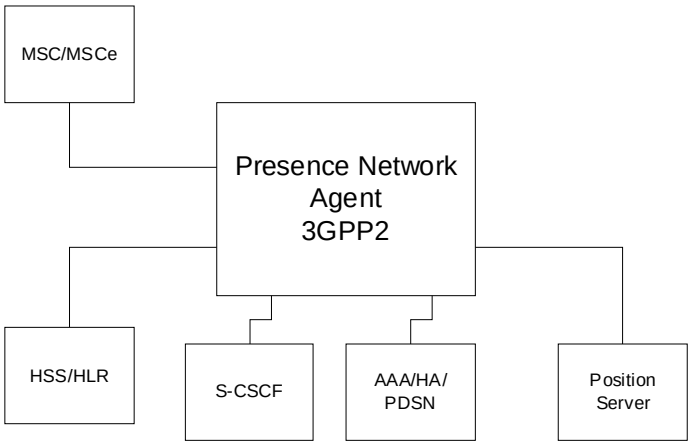

Figure 3: PNA in 3GPP2

The options of using a PNA in a non-3GPP/3GPP2 environment are shown in Figure 4:

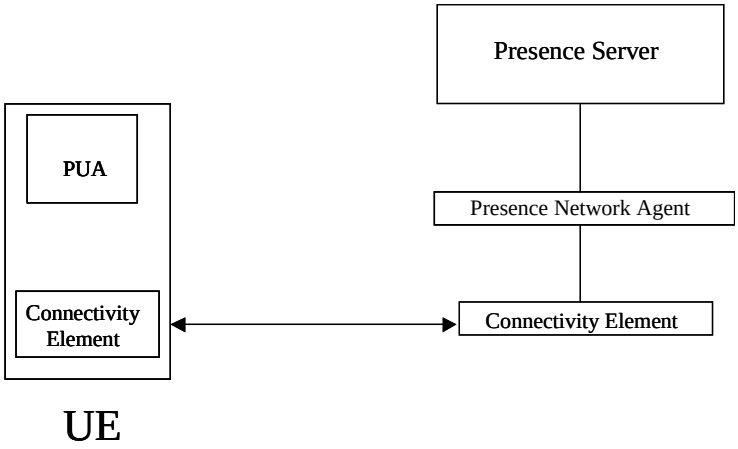

Figure 4: PNA in a non-3GPP/3GPP2 architecture.

Presence Information can be aggregated either directly in the PS or via a PNA.

E.3 Presence External Agent

A Presence Source can be implemented as a Presence External Agent (PEA) as defined by 3GPP/3GPP2 in [3GPP-TS_23.141] and [3GPP2-X.S0027-001] 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; and
- 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; and
- Non SIMPLE-based presence services.

Appendix F. SIP Methods (Informative)

F.1 SUBSCRIBE Method

When the SIP/IP Core is realized with 3GPP IMS or 3GPP2 MMD networks, the full list of supported headers and parameters of the SUBSCRIBE method and its responses is available in [3GPP-TS_24.229] and [3GPP2-X.S0013-004] respectively.

In the context of other realizations of the SIP/IP Core, the full list of supported headers and parameters of the SUBSCRIBE method and its responses is available in [RFC3265], [RFC3857], [RFC3856], [IETF-EventThrottle] and [IETF-SubNotEtag].

F.2 PUBLISH Method

When the SIP/IP Core is realized with 3GPP IMS or 3GPP2 MMD networks, the full list of supported headers and parameters of the PUBLISH method and its responses is available in [3GPP-TS_24.229] and [3GPP2-X.S0013-004] respectively.

In the context of other realizations of the SIP/IP Core, the full list of supported headers and parameters of the PUBLISH method and its responses is available in [RFC3903] and [IETF-SessionPol].

F.3 NOTIFY Method

When the SIP/IP Core is realized with 3GPP IMS or 3GPP2 MMD networks, the full list of supported headers and parameters of the NOTIFY method and its responses is available in [3GPP-TS_24.229] and [3GPP2-X.S0013-004] respectively.

In the context of other realizations of the SIP/IP Core, the full list of supported headers and parameters of the NOTIFY method and its responses is available in [RFC3265], [RFC3857], [RFC3856], [IETF-EventThrottle] and [IETF-SubNotEtag].

F.4 REFER Method

When the SIP/IP Core is realized with 3GPP IMS or 3GPP2 MMD networks, the full list of supported headers and parameters of the REFER method and its responses is available in [3GPP-TS_24.229] and [3GPP2-X.S0013-004] respectively.

In the context of other realizations of the SIP/IP Core, the full list of supported headers and parameters of the REFER method and its responses is available in [RFC3515] and [RFC4488].

Appendix G. Presence Signaling Flows (Informative)

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

G.1 Subsystem Collaboration

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

G.1.1 Signaling Flows for Publishing Presence Information

G.1.1.1 Publishing Presence Information

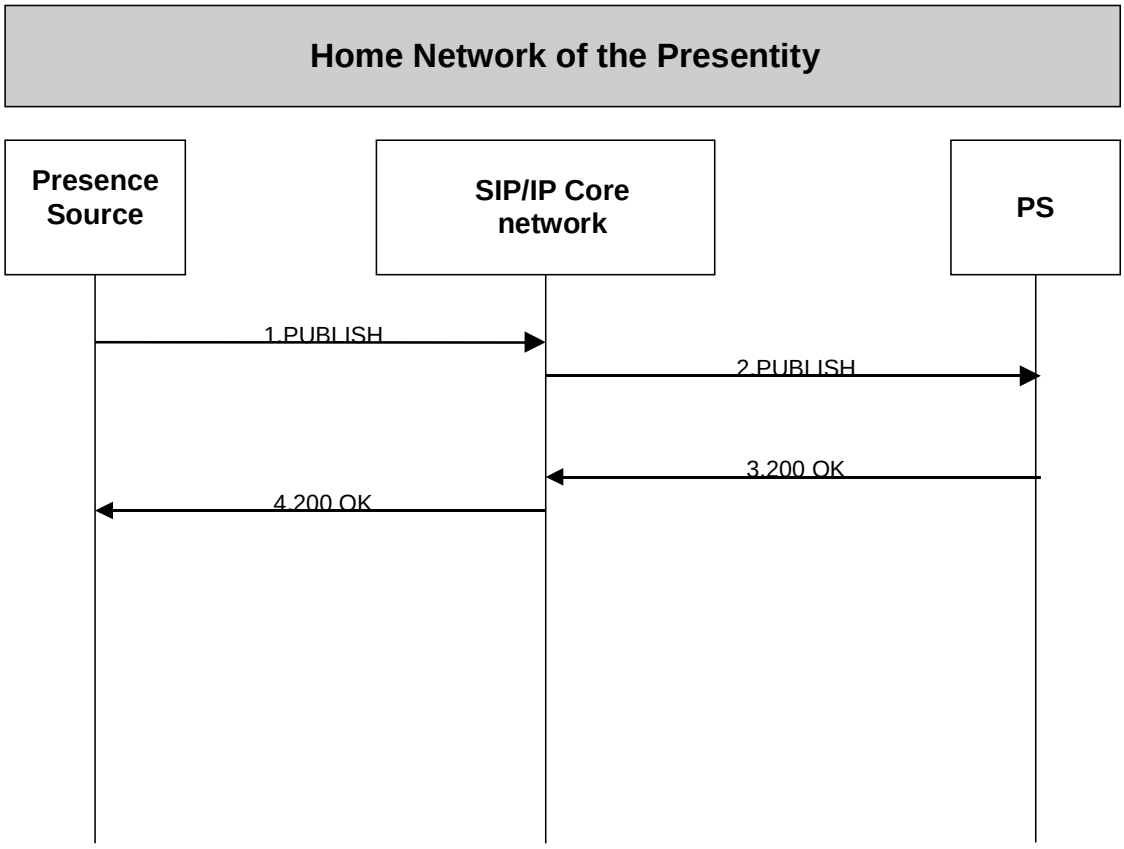

Figure 5: Publishing Presence Information

1. The Presence Source generates a SIP PUBLISH request, which contains a presence document.
2. The SIP/IP Core routes the request to the correct PS.
3. The PS authorizes 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 the Presence Source.
4. The SIP/IP Core forwards the response back to the Presence Source.

G.1.1.2 Publishing Presence Information on behalf of Another Presentity

G.1.1.2.1 Successful Attempt

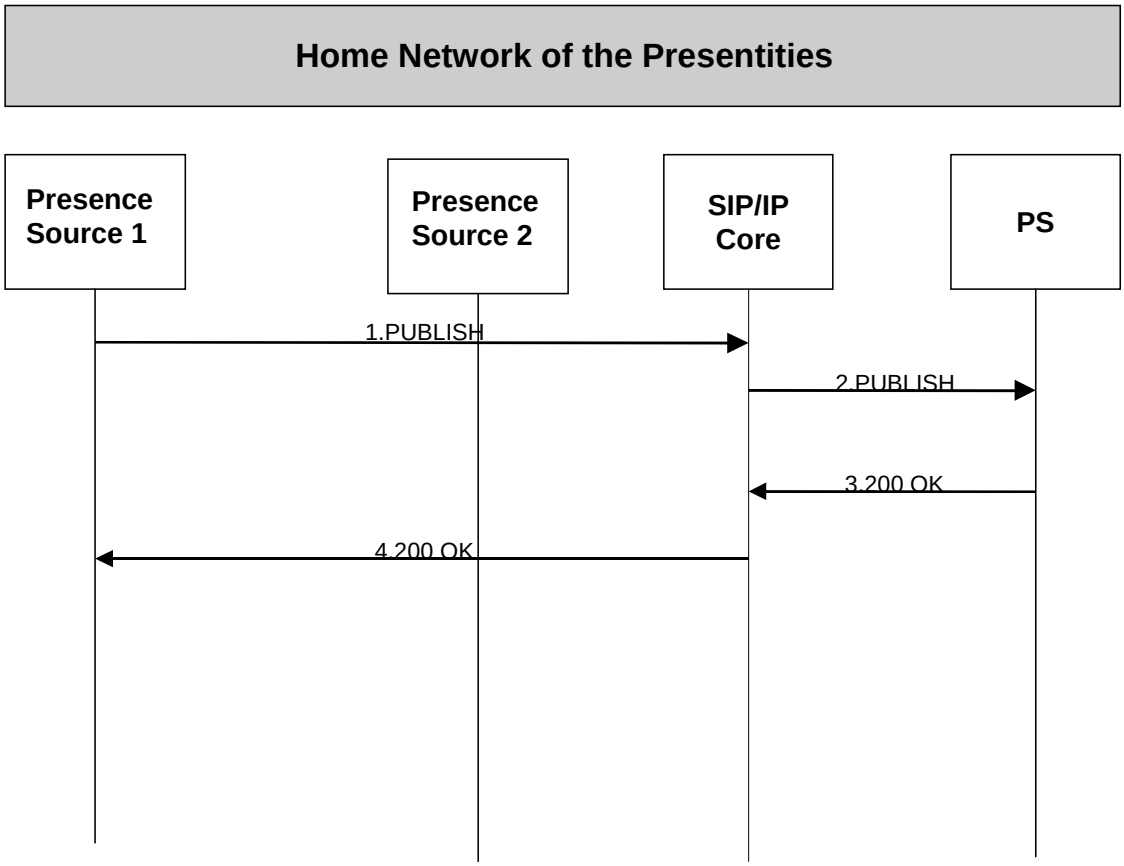

Figure 6: Aggregating published Presence Information from multiple Presence Sources

1. Presence Source1 generates a SIP PUBLISH request, which contains Presence Information relating to Presence Source2's Presentity.
2. The SIP/IP Core forwards the SIP PUBLISH request to the appropriate PS.
3. The PS authorizes 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.
4. The SIP/IP Core forwards the SIP 200 OK response back to the Presence Source1.

G.1.1.2.2 Unsuccessful Attempt: PUBLISH Request Not Authorized

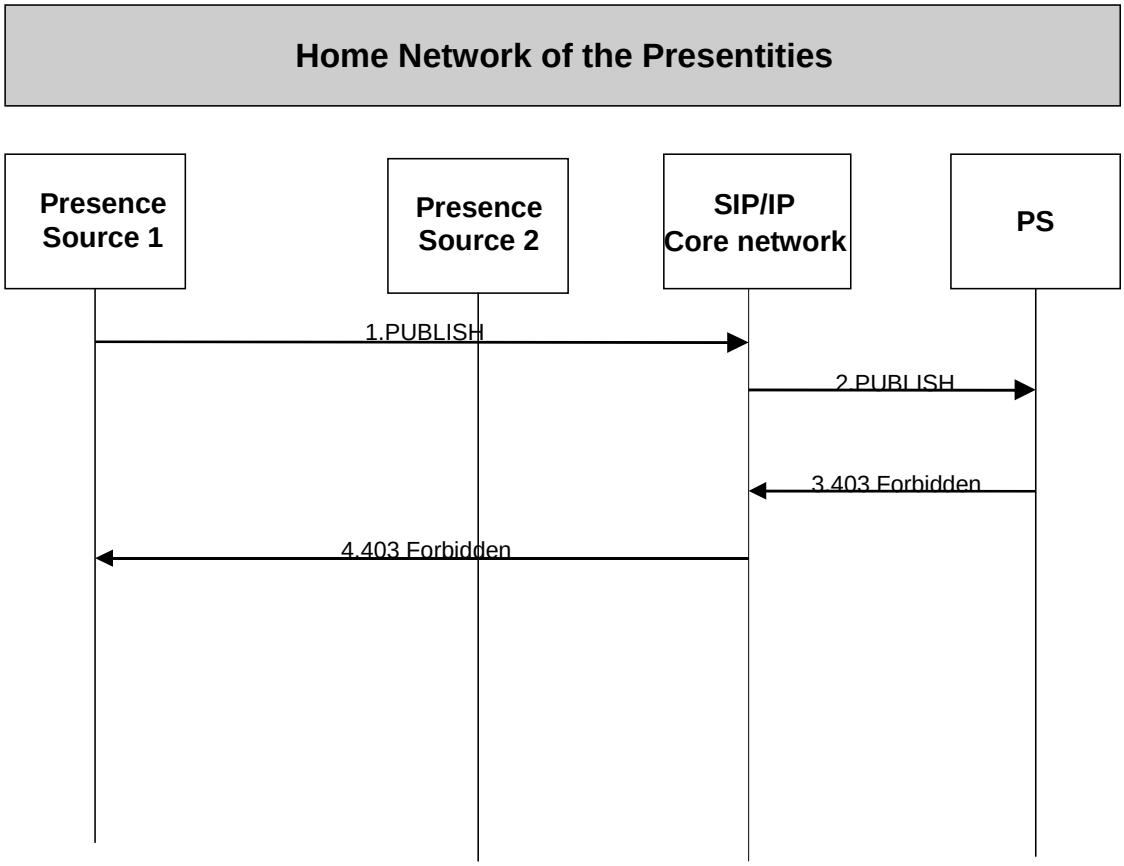

Figure 7: Unsuccessful attempt: PUBLISH request not authorized

1. Presence Source1 generates a SIP PUBLISH request, which contains Presence Information relating to Presence Source2's Presentity.
2. The SIP/IP Core forwards the SIP PUBLISH request to the appropriate PS.
3. The PS does not authorize the request and sends a SIP 403 (Forbidden) response back to the SIP/IP Core.
4. The SIP/IP Core forwards the SIP 403 (Forbidden) response back to the Presence Source1.

G.1.1.2.3 Unsuccessful First Attempt: PUBLISH Request with Partially Authorized Presence Information

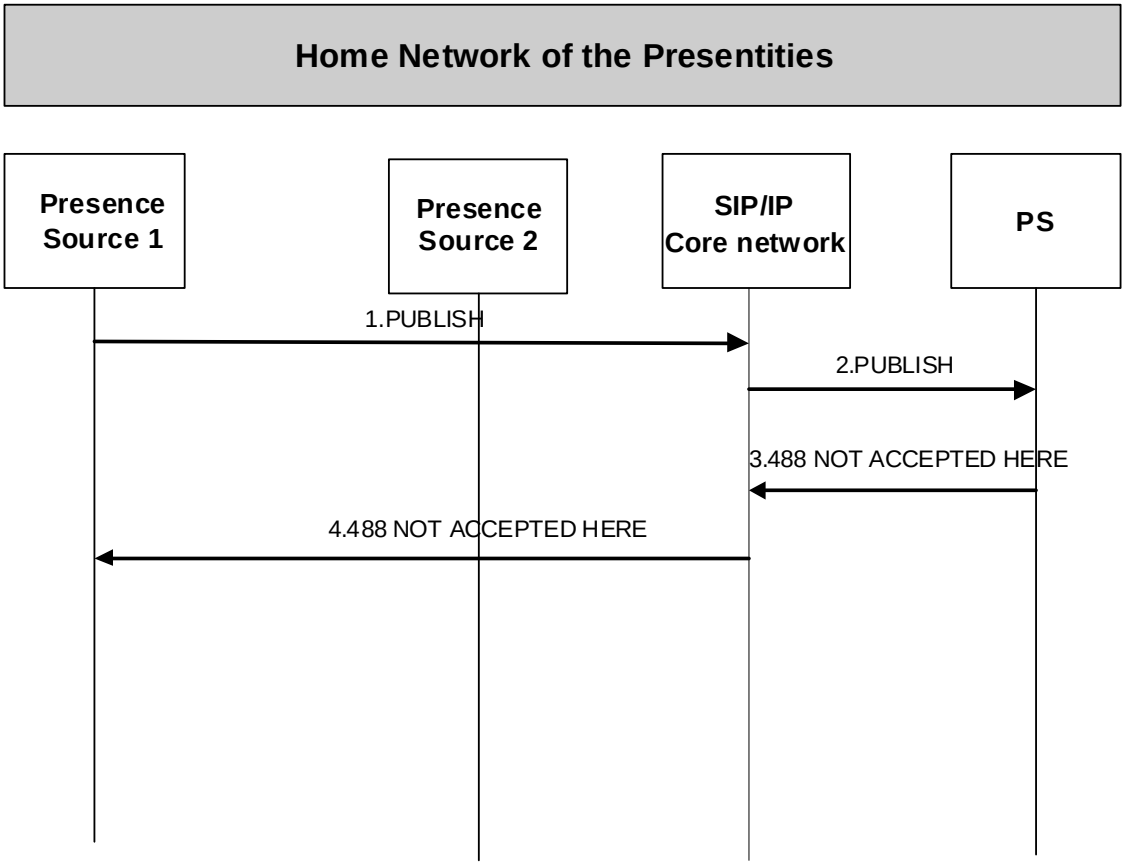

Figure 8: Unsuccessful first attempt: PUBLISH request with partially authorized Presence Information

1. Presence Source1 generates a SIP PUBLISH request, which contains Presence Information relating to Presence Source2's Presentity.

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

3. The PS performs authorization of the request but finds that the Publication Content Rules do not authorize the Presence Information contained in the request and sends a SIP 488 (Not Accepted Here) response back to the SIP/IP Core with a Policy-Contact header with a URI containing the XCAP URI of the Presentity's Publication Content Rules Presence Source View document.

4. The SIP/IP Core forwards the SIP 488 (Not Accepted Here)) response back to Presence Source1.
   Presence Source1 fetches the indicated document using the received URI as described in the example in [PRS_PresXDM] "*Obtaining A Publication Content Rules Presence Source View Document*".
   The Presence Source1 checks the received document and generates a new SIP Publish request with only authorized Presence Information as shown in Figure 6.

G.1.1.2.4 Aggregating Published Presence Information from Multiple Presence Sources

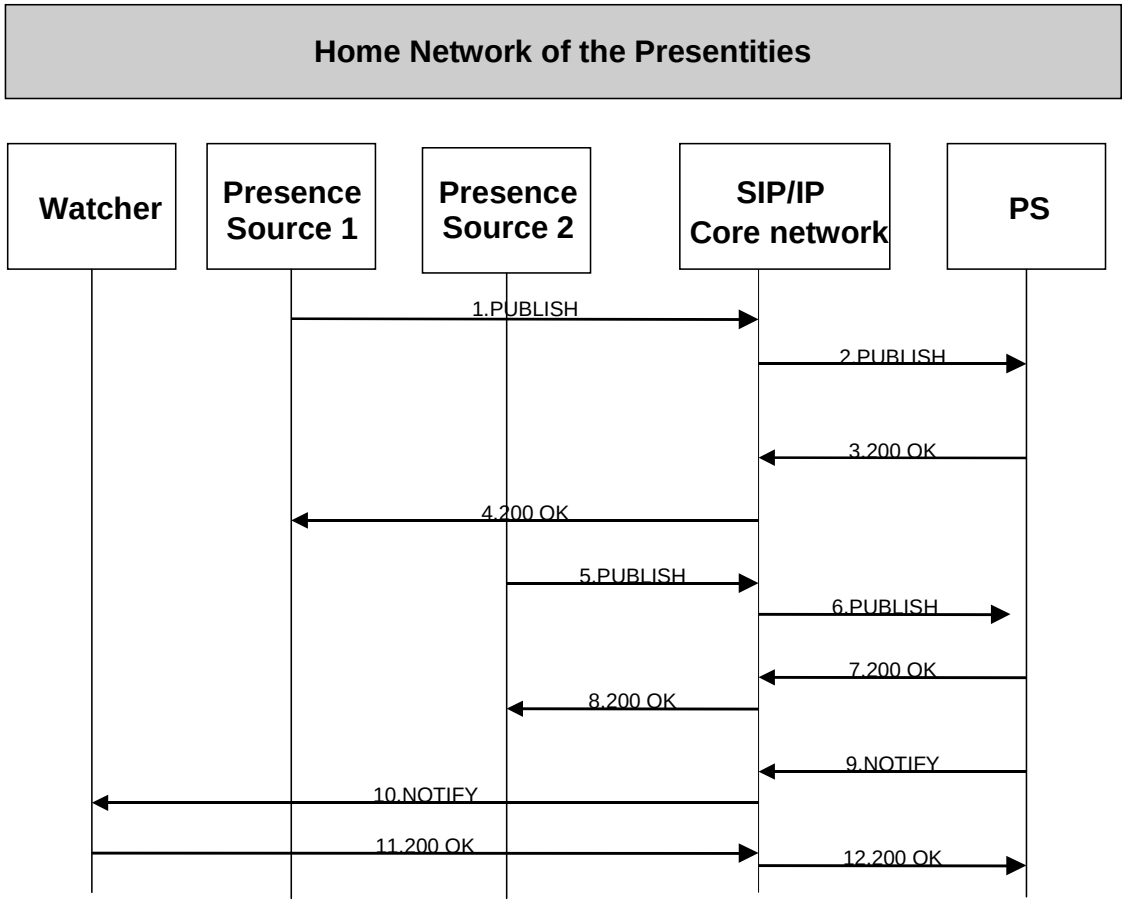

Figure 9: Aggregating published Presence Information from multiple Presence Sources

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 forwards the SIP PUBLISH request to the appropriate PS.
3. The PS authorizes 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 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 forwards the SIP PUBLISH request to the appropriate PS.
7. The PS authorizes 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 Presence Information Presence Source1 has published. The PS sends a SIP 200 (OK) response back to the SIP/IP Core.
8. The SIP/IP Core forwards the SIP 200 (OK) response back to the Presence Source2.

9. The PS determines which authorized 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 of the Watcher.
10. The SIP/IP Core 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.
12. The SIP/IP Core of the Watcher forwards the SIP 200 (OK) response to the PS.

G.1.2 Signaling Flows for Watchers Subscribing to Presence Event Notification

G.1.2.1 Subscribing to Presence Information State Changes - Proactive Authorization

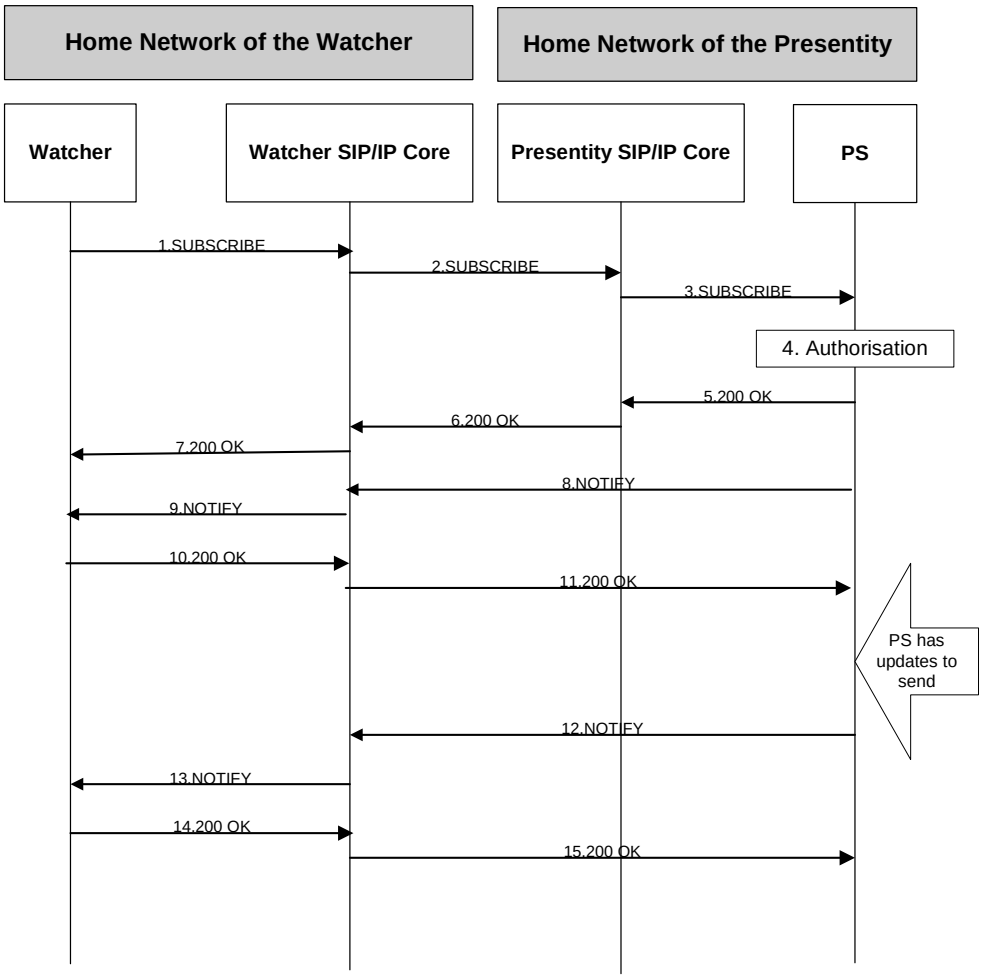

Figure 10: Subscribing to Presence Information state changes (Watcher and Presentity are in different networks) – Proactive Authorization

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 of the Watcher resolves the address of the Presentity and forwards the request to the SIP/IP Core of the Presentity

NOTE 1: In the case Watcher service authorization is applied, the SIP SUBSCRIBE is routed to the Watcher Agent prior to step 2 (see G.1.2.9).

3. The SIP/IP Core of the Presentity routes the SIP SUBSCRIBE request to the correct PS.

4. The PS performs the necessary authorization checks on the originator to ensure it is allowed to watch the Presentity.

NOTE 2: In the case where the privacy/authorization 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 of the Presentity.

6. The SIP/IP Core of the Presentity forwards the response to the SIP/IP Core of the Watcher.

7. The SIP/IP Core 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 authorized to receive. The SIP NOTIFY request is sent to the SIP/IP Core of the Watcher. 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 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.

11. The SIP/IP Core of the Watcher forwards the SIP 200 (OK) response to the PS.

12. When the Presence Information for the Presentity changes, 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 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 request with a SIP 200 (OK) response to its SIP/IP Core.

15. The SIP/IP Core of the Watcher forwards the SIP 200 (OK) response to the PS.

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

G.1.2.2 Fetching Presence Information State – Proactive Authorization

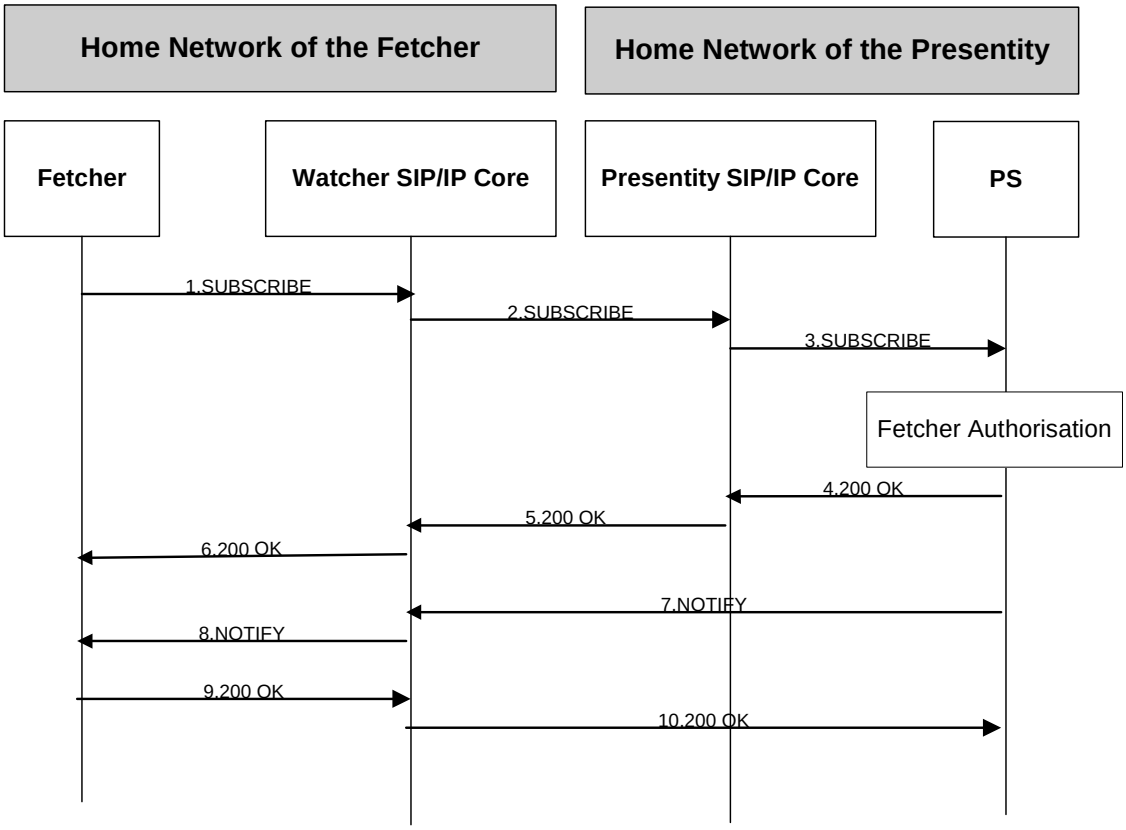

Figure 11: Fetching Presence Information state (fetcher and Presentity are in different networks)

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 resolves the address of the SIP/IP Core of the Presentity and forwards the request.

NOTE 1: In the case Watcher service authorization is applied, the SIP SUBSCRIBE is routed to the Watcher Agent prior to step 2 (see G.1.2.9).

3. The SIP/IP Core 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 sends a SIP 200 (OK) response to the SIP/IP Core of the Presentity.

5. The SIP/IP Core of the Presentity forwards the SIP 200 (OK) response to the SIP/IP Core of the Watcher.

6. The SIP/IP Core 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 receive. The SIP NOTIFY request is sent along the path of the SUBSCRIBE dialog to the SIP/IP Core of the Watcher.

8. The SIP/IP Core 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 to the SIP/IP Core of the Watcher.

10. The Watcher's SIP/IP Core forwards the SIP 200 (OK) response to the PS.

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

G.1.2.3 Subscribing to Presence Information State Changes - Reactive Authorization

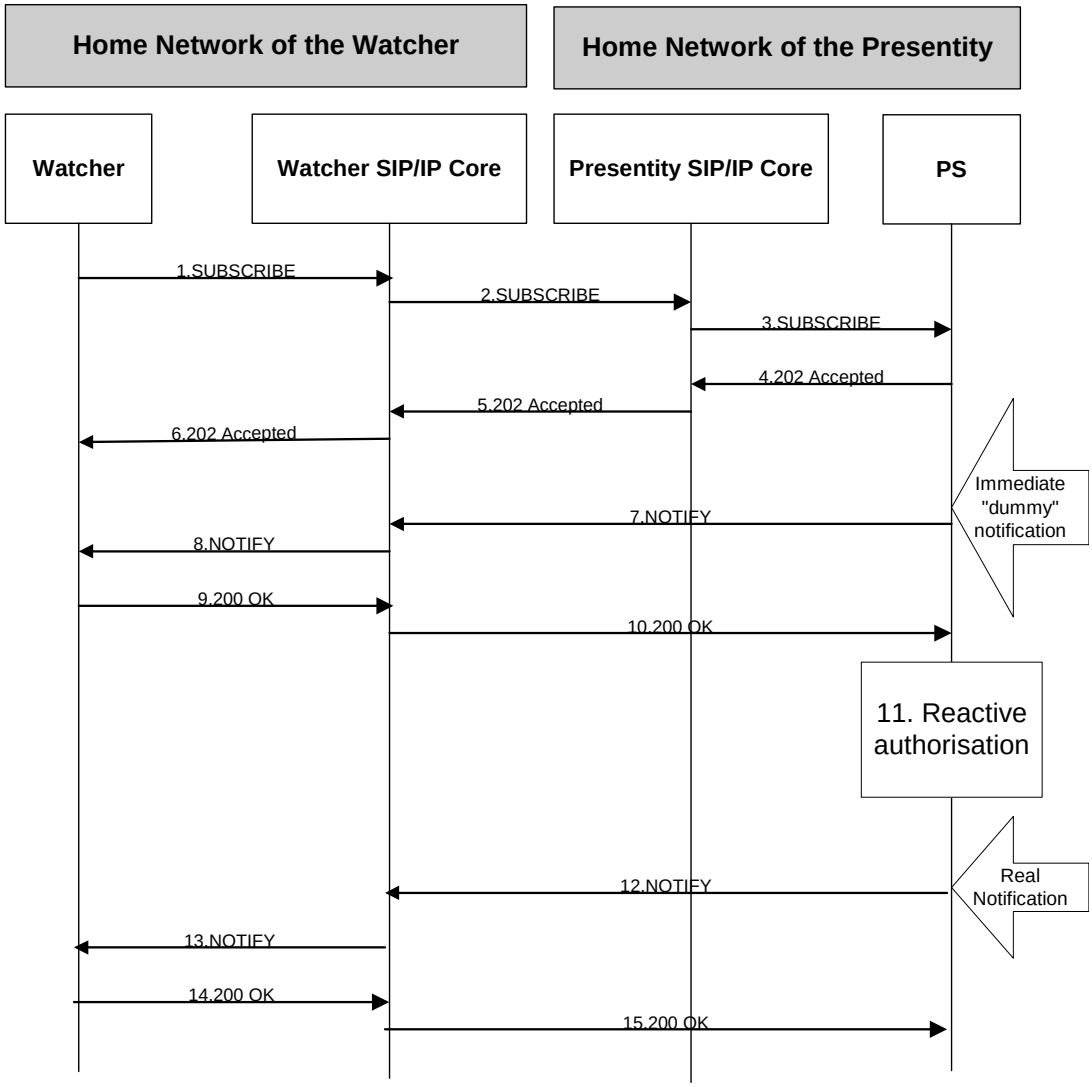

Figure 12: Subscribing to Presence Information state changes (Watcher and Presentity are in different networks) - Reactive Authorization

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 of the Watcher resolves the address of the Presentity and forwards the request to the SIP/IP Core of the Presentity.

NOTE 1: In the case Watcher service authorization is applied, the SIP SUBSCRIBE is routed to the Watcher Agent prior to step 2 (see G.1.2.9).

3. The SIP/IP Core of the Presentity routes the SIP SUBSCRIBE request to the correct PS.
4. The PS acknowledges the request with a SIP 202 (Accepted) response sent to the SIP/IP Core of the Presentity.
5. The SIP/IP Core of the Presentity forwards the SIP 202 (Accepted) response to the SIP/IP Core of the Watcher.
6. The SIP/IP Core of the Watcher forwards the SIP 202 (Accepted) 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 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.
10. The SIP/IP Core of the Watcher forwards the SIP 200 (OK) response to the PS.
11. The PS authorizes the Watcher, after the Presentity modifies the Presence Subscription Rules (see section 5.5.3.3).
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.
15. The SIP/IP Core of the Watcher forwards the SIP 200 (OK) response to the PS.

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

NOTE 3: 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.

G.1.2.4 Receiving a Presence Notification for an Existing Subscription

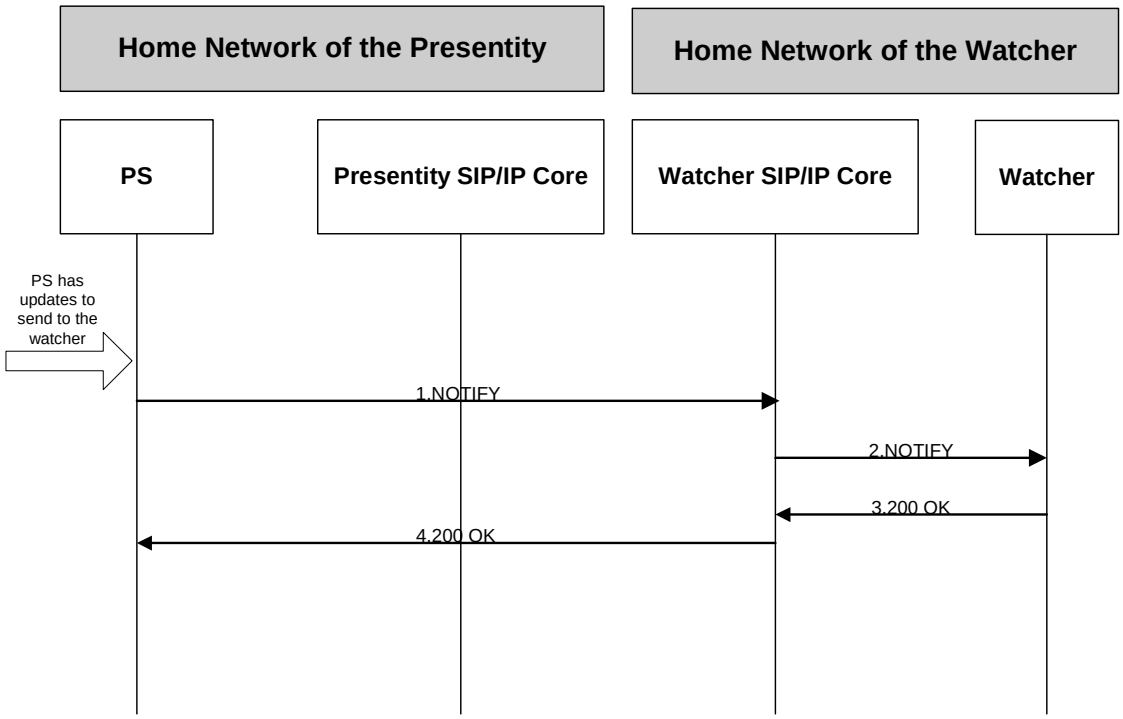

Figure 13: Receiving a presence notification

1. The PS determines which authorized 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 of the Watcher.
2. The SIP/IP Core 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.
4. The SIP/IP Core of the Watcher forwards the SIP 200 (OK) response to the PS.

G.1.2.5 Partial Notifications

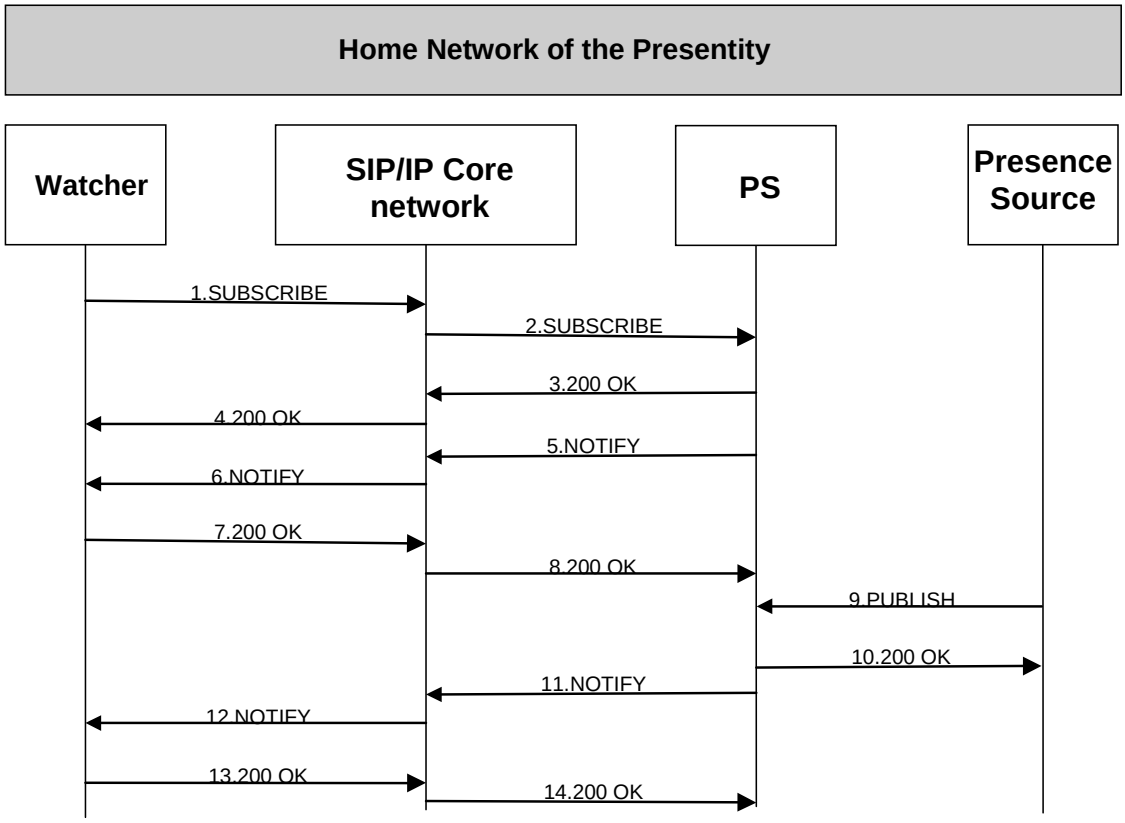

Figure 14: Partial Notifications Information Flow

1. A Watcher sends a SIP SUBSCRIBE request to the PS indicating the support for the default Presence Information Data Format defined in [RFC3863] and the partial PIDF defined in [RFC5262]. The Watcher also indicates the support for the partial notification mechanism according to [RFC5263].
2. The SIP/IP Core 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.
4. The SIP/IP Core forwards the SIP 200 (OK) response to the Watcher.
5. The PS, based on the Watcher's indication that it supports the partial notification mechanism, generates a SIP NOTIFY request, which includes a full state presence document formulated according to [IETF-ParNot]. The SIP NOTIFY request is forwarded to the SIP/IP Core.
6. The SIP/IP Core forwards the SIP NOTIFY request to the Watcher.
7. The Watcher sends a SIP 200 (OK) response to the SIP/IP Core to acknowledge the SIP NOTIFY request.
8. The SIP/IP Core 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 [RFC5262] showing only the changed Presence Information.
12. The SIP/IP Core 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 forwards the SIP 200 (OK) response to the PS.

NOTE: If the Watcher and the Presentity reside in different domains, the SIP/IP Core of the Watcher will perform address resolution on the address of the Presentity to forward the SIP SUBSCRIBE request to the SIP/IP Core of the Presentity. Then the SIP/IP Core of the Presentity will route the SIP SUBSCRIBE request to the PS (see step 2 and 3 as well as 5 and 6 in appendix D.1.2.1).

G.1.2.6 Expiry of Published Presence Information

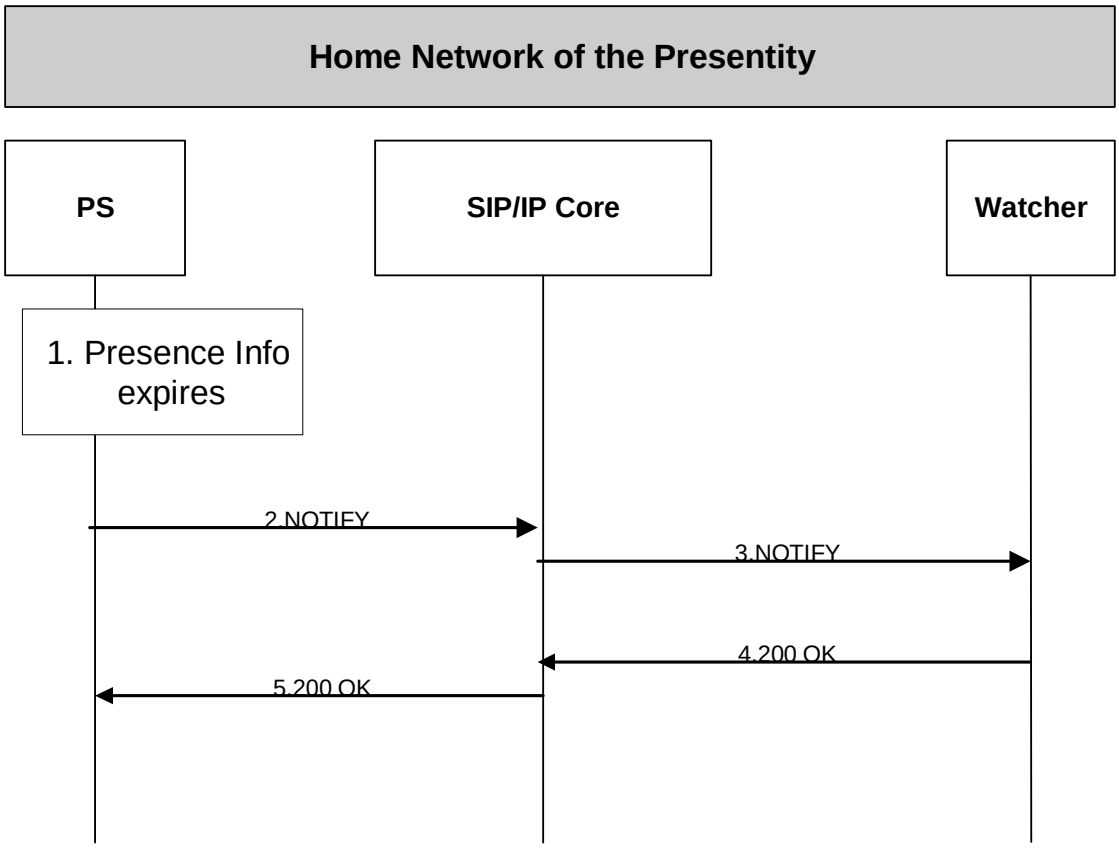

Figure 15: Expiry of published Presence Information

1. The lifetime of some Presence Information expires 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 forwards the SIP NOTIFY request to the Watcher.
4. The Watcher sends a 200 (OK) response to the SIP/IP Core to acknowledge the SIP NOTIFY request.
5. The SIP/IP Core forwards the 200 (OK) response to the PS.

G.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”).

G.1.2.7.1 Blocking

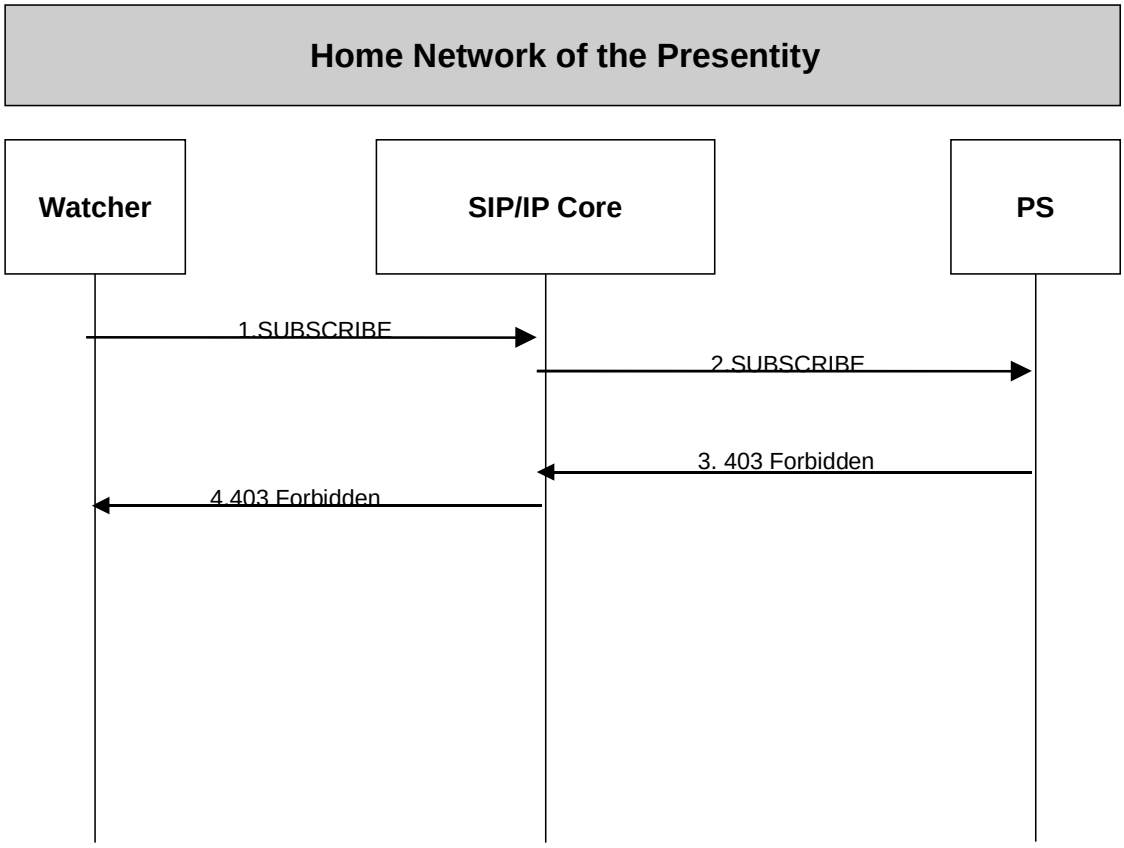

Figure 16: Blocking

1. A Watcher wishing to subscribe to Presence Information about a Presentity, sends a SIP SUBSCRIBE request to the SIP/IP Core.
2. The SIP/IP Core 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 Rules of the Presentity, the PS determines to reject the subscription request. The PS sends a SIP 403 (Forbidden) response to the SIP/IP Core.
4. The SIP/IP Core forwards the SIP 403 (Forbidden) response to the Watcher.

NOTE: If the Watcher and the Presentity reside in different domains, the SIP/IP Core of the Watcher will perform address resolution on the address of the Presentity to forward the SIP SUBSCRIBE request to the SIP/IP Core of the Presentity. Then the SIP/IP Core of the Presentity will route the SIP SUBSCRIBE request to the PS (see step 2 and 3 as well as 5 and 6 in appendix G.1.2.1).

G.1.2.7.2 Polite Blocking

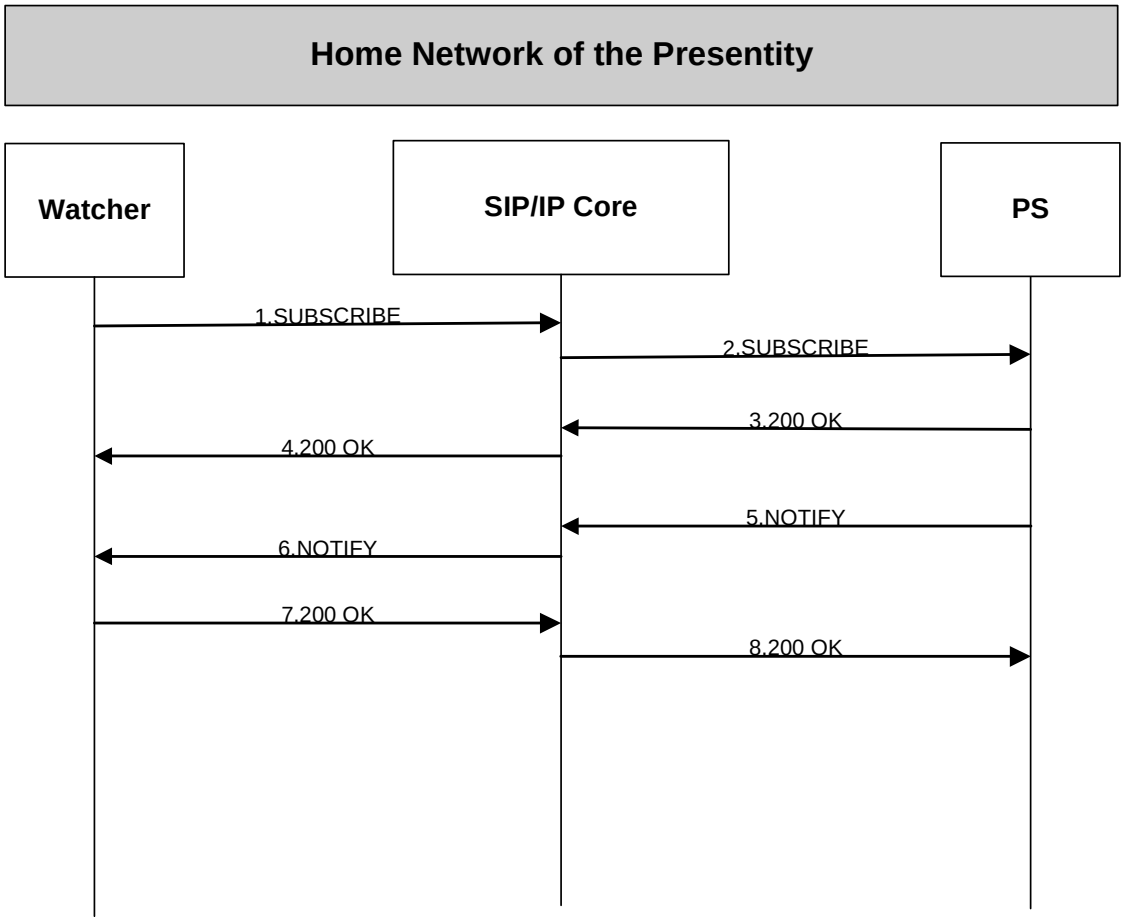

Figure 17: Polite Blocking

1. A Watcher wishing to subscribe to Presence Information about a Presentity, sends a SIP SUBSCRIBE request to the SIP/IP Core.
2. The SIP/IP Core 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 Rules 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 5.5.3.3.1). The PS sends a 200 (OK) to the SIP/IP Core.
4. The SIP/IP Core 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 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 forwards the SIP 200 (OK) response to the appropriate PS.

NOTE: If the Watcher and the Presentity reside in different domains, the SIP/IP Core of the Watcher will perform address resolution on the address of the Presentity to forward the SIP SUBSCRIBE request to the SIP/IP Core of the Presentity. Then

the SIP/IP Core of the Presentity will route the SIP SUBSCRIBE request to the PS (see step 2 and 3 as well as 5 and 6 in appendix D.1.2.3).

G.1.2.8 Subscription Filters

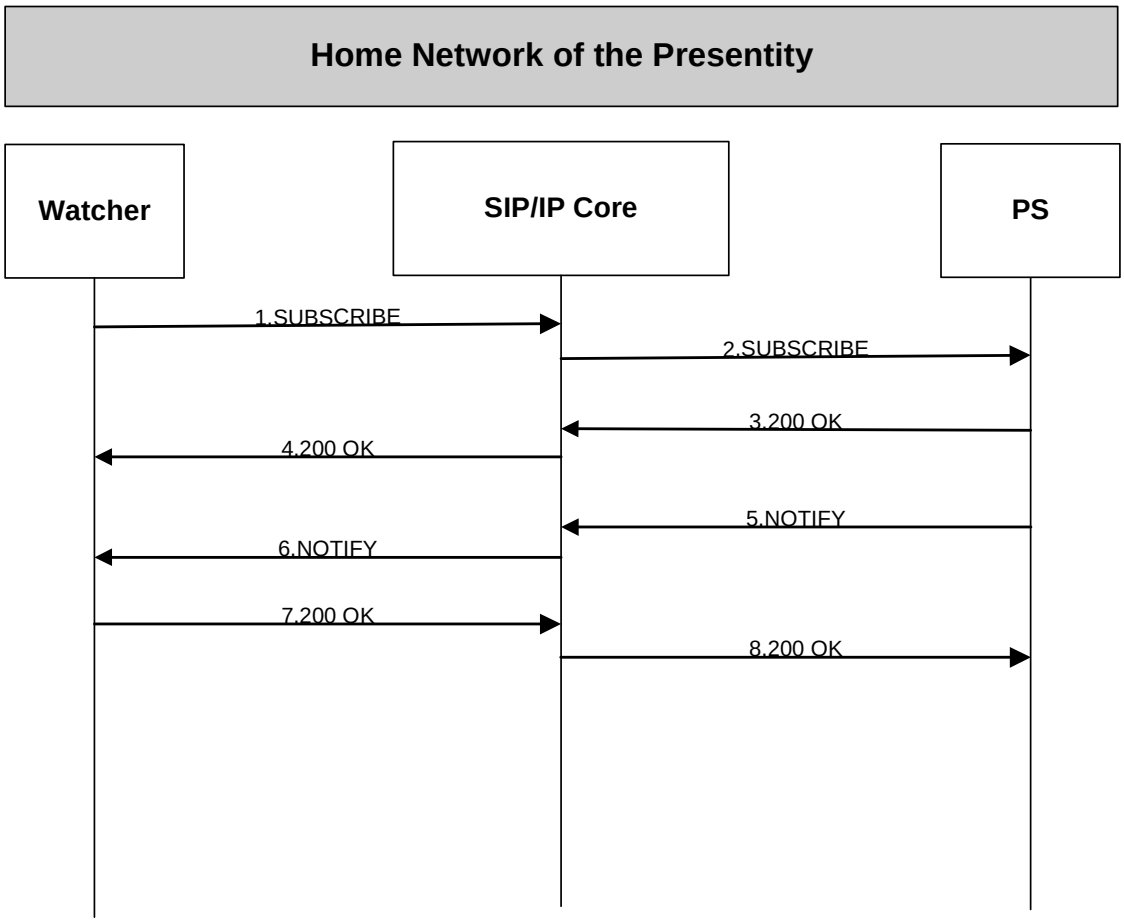

Figure 18: Subscription Filters

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 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 indicating that the subscription has been accepted and the subscription filter understood.
4. The SIP/IP Core forwards the SIP 200 (OK) response to the Watcher.
5. The PS sends a SIP NOTIFY request to the the SIP/IP Core including only the Instant Messaging related tuple that was requested by the Watcher's subscription filter.
6. The SIP/IP Core 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 forwards the SIP 200 (OK) response to the PS.

NOTE: If the Watcher and the Presentity reside in different domains, the SIP/IP Core of the Watcher will perform address resolution on the address of the Presentity to forward the SIP SUBSCRIBE request to the SIP/IP Core of the Presentity. Then the SIP/IP Core of the Presentity will route the SIP SUBSCRIBE request to the PS (see step 2 and 3 as well as 5 and 6 in appendix D.1.2.3).

G.1.2.9 Subscribing to Presence Information State Changes with Watcher Service Authorization

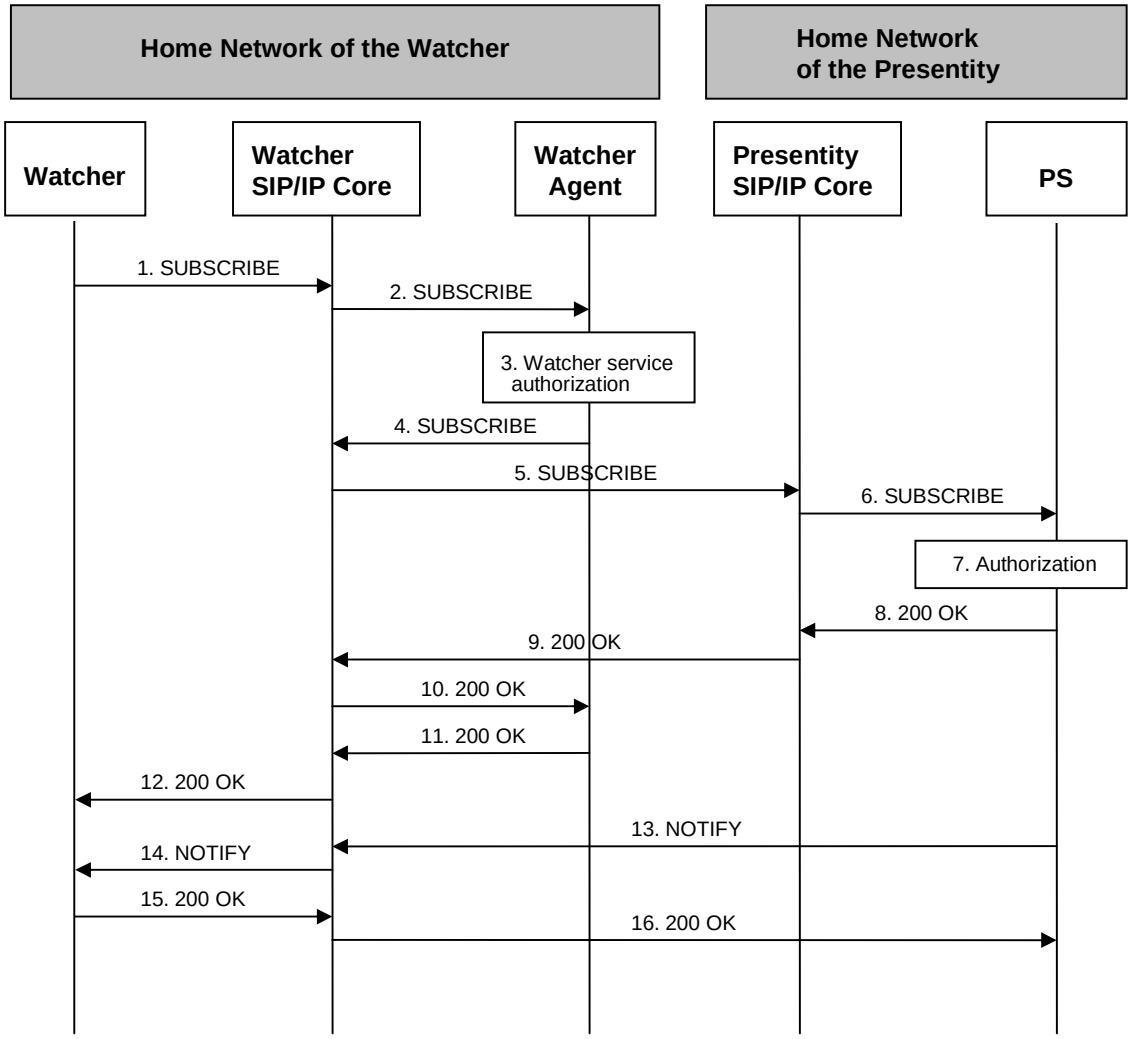

Figure 19: Subscribing to Presence Information state changes (Watcher and Presentity are in different networks) – Watcher service authorization

1. A Watcher wishing to watch a Presentity's Presence Information sends a SIP SUBSCRIBE request for the Presence Event Package including an indication of the duration this subscription should last.

2. The SIP/IP Core of the Watcher forwards the SIP SUBSCRIBE request to the Watcher Agent.

3. The Watcher Agent performs the necessary authorization checks on the originator to ensure the originator is allowed to use the Presence Service and issue the SIP SUBSCRIBE request.

NOTE 1: In the case where the authorization checks fail, then a negative acknowledgement is sent to the Watcher.

4. Upon successful authorization, the Watcher Agent forwards the request to the SIP/IP Core of the Watcher.
5. The SIP/IP Core of the Watcher resolves the address of the Presentity and forwards the request to the SIP/IP Core of the Presentity.
6. The SIP/IP Core of the Presentity routes the SIP SUBSCRIBE request to the correct PS.
7. The PS performs the necessary authorization checks on the originator to ensure it is allowed to watch the Presentity.

NOTE 2: In the case where the privacy/authorization checks fail, then a negative acknowledgement is sent to the Watcher.

8. Once all privacy conditions are met, the PS issues a SIP 200 (OK) response to the SIP/IP Core of the Presentity.
9. The SIP/IP Core of the Presentity forwards the response to the SIP/IP Core of the Watcher.
10. The SIP/IP Core of the Watcher forwards the response to the Watcher Agent.
11. The Watcher Agent forwards the response to the SIP/IP Core of the Watcher.
12. The SIP/IP Core of the Watcher forwards the response to the Watcher.
13. 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 authorized to receive. The SIP NOTIFY request is sent to the SIP/IP Core of the Watcher.
14. The SIP/IP Core of the Watcher forwards the SIP NOTIFY request to the Watcher.
15. The Watcher acknowledges the receipt of the SIP NOTIFY request with a SIP 200 (OK) response sent to the SIP/IP Core of the Watcher.
16. The SIP/IP Core of the Presentity forwards the SIP 200 (OK) response to the PS.
    When the Presence Information for the Presentity changes, the PS will send additional SIP NOTIFY requests with the udpated Presence Information towards the Watcher.

NOTE 3: Steps 5 and 6 as well as steps 8 and 9 are combined if the Watcher resides in the same domain as the Presentity.

G.1.2.10 Subscribing to Presence Information State Changes with Direct Event Notification Suppression

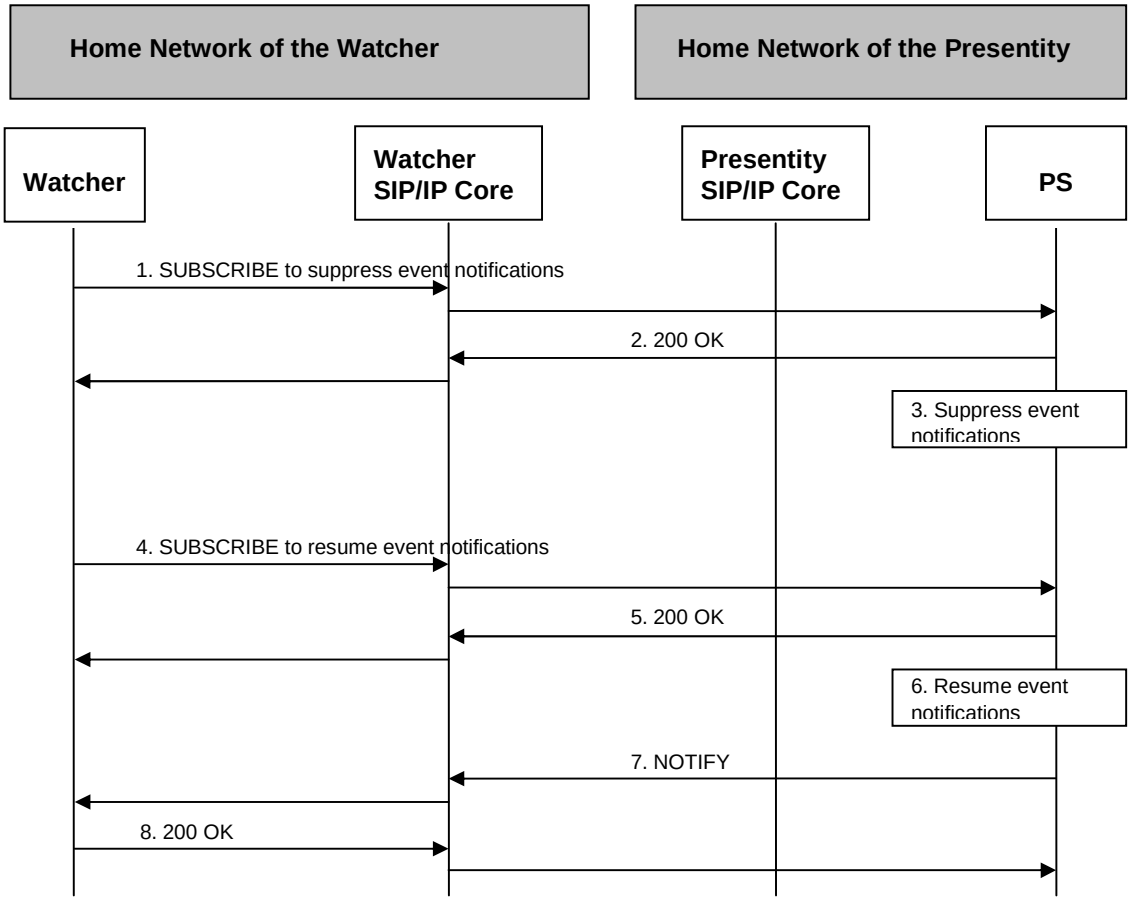

Figure 20: Subscribing to Presence Information state changes (Watcher and Presentity are in different networks) – Direct event notification suppression

1. When the Watcher wishes to suppress the event notifications, the Watcher sends a SIP re-SUBSCRIBE request for the Presence Event Package including a Suppress-If-Match header set to "*" according to [IETF-SubNotEtag] or including a throttling parameter set to the remaining subscription expiration value according to [IETF-EventThrottle]. The SIP SUBSCRIBE request is sent through the SIP/IP Core of the Watcher to the PS.

2. The PS checks whether the event notification suppression request is acceptable, and if acceptable, the PS sends a 200 (OK) response to the Watcher through the SIP/IP Core of the Watcher.

NOTE 1: In the case where the event notification suppression request checks fail, then a negative acknowledgement is sent to the Watcher.

3. The PS starts to suppress event notifications towards the Watcher.

4. When the Watcher wishes to resume the event notifications, the Watcher sends another SIP re-SUBSCRIBE request for the Presence Event Package including a Suppress-If-Match header set to the previous Entity-tag available to the Watcher according to [IETF-SubNotEtag] or including a throttling parameter set to "0" according to [IETF-EventThrottle]. The SIP SUBSCRIBE request is sent through the SIP/IP Core of the Watcher to the PS.

5. The PS checks whether the event notification resumption request is acceptable, and if acceptable, the PS sends a 200 (OK) response to the Watcher through the SIP/IP Core of the Watcher.

6. The PS starts to resume event notifications towards the Watcher.

7. When the Presence Information for the Presentity changes, the PS will send the SIP NOTIFY request with the udpated Presence Information towards the Watcher through the SIP/IP Core of the Watcher.

8. The Watcher acknowledges the receipt of the SIP NOTIFY request with a SIP 200 (OK) response sent through the SIP/IP Core of the Watcher to the PS.

NOTE 2: The SIP/IP Core of the Watcher and that of the Presentity are combined if the Watcher resides in the same domain as the Presentity.

G.1.2.11 Conditional Event Notification Suppression: setting up presence-based event notification filter

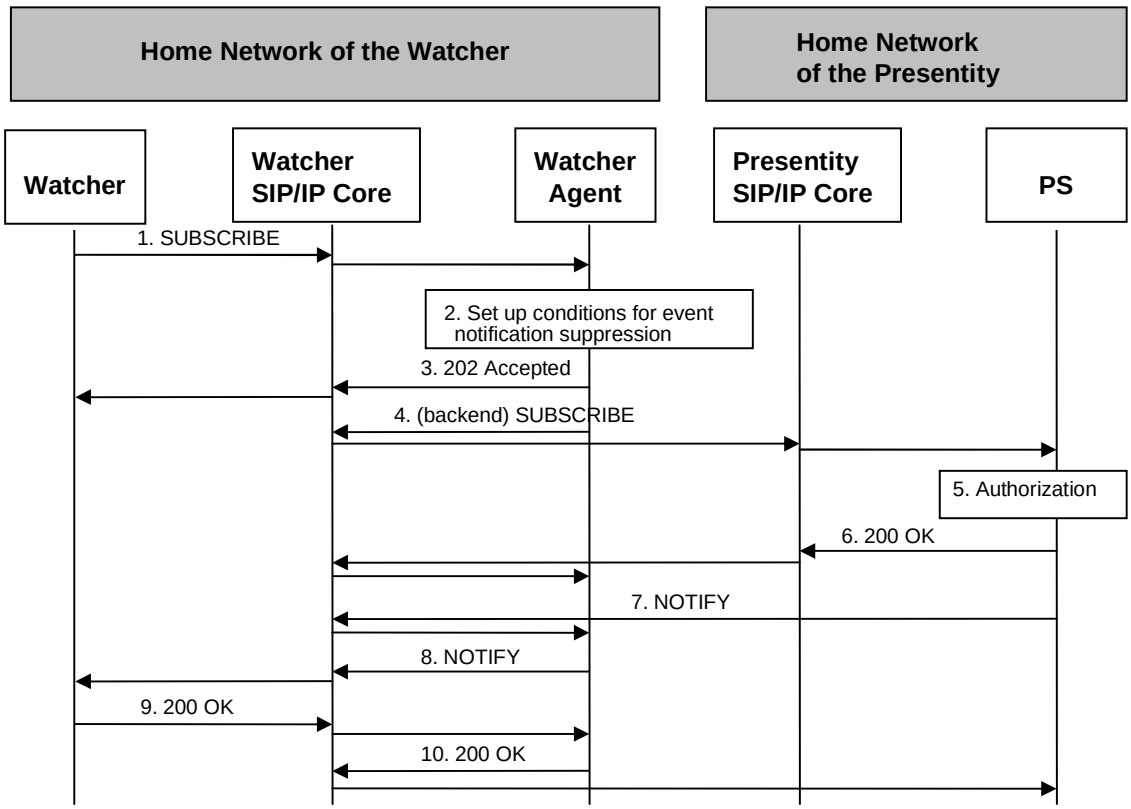

Figure 21: Conditional event notification suppression: setting up presence-based event notification filter

1. A Watcher wishing to watch a Presentity's Presence Information with conditional event notification suppression sends a SIP SUBSCRIBE request for the Presence Event Package including a presence-based event notification suppression filter in the body that describes the condition of the Watcher's own presence state when the Watcher does not wish to receive event notifications.
   The SIP/IP Core of the Watcher forwards the SIP SUBSCRIBE request to the Watcher Agent.

2. The Watcher Agent checks whether the presence-based event notification suppression filter is acceptable. If it accepts, the Watcher Agent extracts and store the the presence-based event notification suppression filter.

NOTE 1: In the case Watcher service authorization is applied, the SIP SUBSCRIBE is routed to the Watcher Agent prior to step 2 (see G.1.2.9).

3. The Watcher Agent generates a 202 (Accepted) response towards the Watcher through the SIP/IP Core of the Watcher.

NOTE 2: In the case where the presence-based event notification suppression filter is not acceptable, then a negative acknowledgement is sent to the Watcher.

4. The Watcher Agent generates the backend SIP SUBSCRIBE request, and the SIP/IP Core of the Watcher forwards the SIP SUBSCRIBE request to the correct PS of the Presentity through the SIP/IP Core of the Presentity.

5. The PS performs the necessary authorization checks on the Watcher to ensure it is allowed to watch the Presentity.

NOTE 3: In the case where the privacy/authorization checks fail, then a negative acknowledgement is sent towards the Watcher Agent, which is then forwarded to the Watcher.

6. Once all privacy conditions are met, the PS issues a SIP 200 (OK) response to the SIP/IP Core of the Presentity, then the SIP/IP Core of the Presentity forwards the SIP 200 (OK) response to the Watcher Agent through the SIP/IP Core of the Watcher.

7. 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 authorized to receive. The SIP NOTIFY request is sent to the SIP/IP Core of the Watcher, then which forwards the SIP NOTIFY request to the Watcher Agent.

8. The Watcher Agent generates a SIP NOTIFY request to the Watcher through the SIP/IP Core of the Watcher, including the Presence Information as received from the PS.

9. The Watcher acknowledges the receipt of the SIP NOTIFY request with a SIP 200 (OK) response sent through the SIP/IP Core of the Watcher to the Watcher Agent.

10. The Watcher Agent generates a SIP 200 (OK) response to the PS through the SIP/IP Core of the Watcher. When the Presence Information for the Presentity changes, the PS will send additional SIP NOTIFY requests with the updated Presence Information towards the Watcher through the Watcher Agent.

G.1.2.12 Conditional Event Notification Suppression: suppressing and resuming event notifications

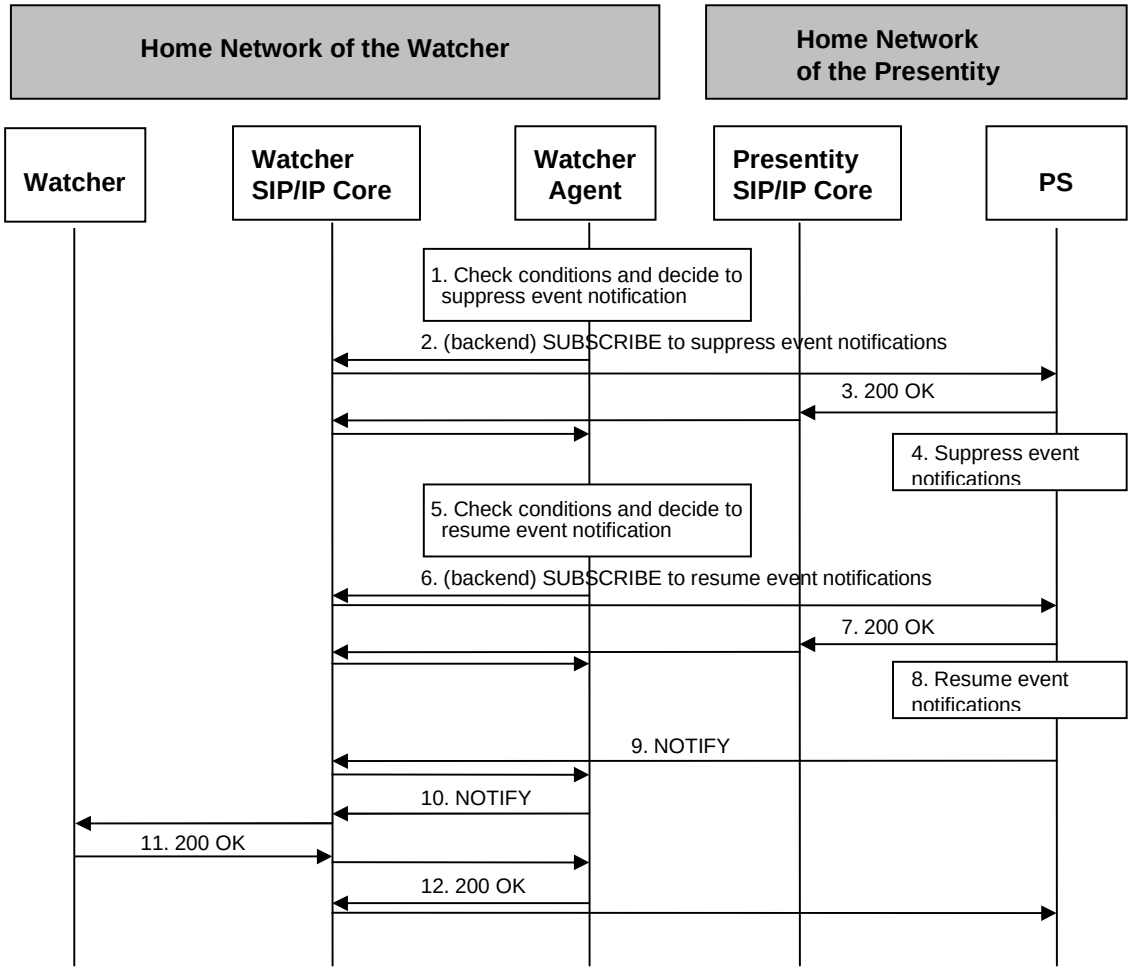

Figure 22: Conditional event notification suppression: suppressing and resuming event notifications

1. The Watcher Agent evaluates the presence-based event notification suppression filter against the Watcher's Presence Information and, if a match is found, will request the PS to suppress the event notifications.

2. When the Watcher Agent wishes to suppress the event notifications, the Watcher Agent sends a SIP re-SUBSCRIBE request for the Presence Event Package including a Suppress-If-Match header set to "*" according to [IETF-SubNotEtag] or including a throttling parameter set to the remaining subscription expiration value according to [IETF-EventThrottle]. The SIP SUBSCRIBE request is sent through the SIP/IP Core of the Watcher to the PS.

3. The PS checks whether the event notification suppression request is acceptable, and if acceptable, the PS sends a 200 (OK) response to the Watcher Agent through the SIP/IP Core of the Watcher.

NOTE 1: In the case where the event notification suppression request checks fail, then a negative acknowledgement is sent to the Watcher Agent.

4. The PS starts to suppress event notifications towards the Watcher through the Watcher Agent.

5. The Watcher Agent evaluates the presence-based event notification suppression filter against the Watcher's Presence Information and, if there is no match, will request the PS to resume the event notifications.

6. When the Watcher Agent wishes to resume the event notifications, the Watcher Agent sends another SIP re-SUBSCRIBE request for the Presence Event Package including a Suppress-If-Match header set to the previous Entity-tag available to the Watcher Agent according to [IETF-SubNotEtag] or including a throttling parameter set to "0" according to [IETF-EventThrottle]. The SIP re-SUBSCRIBE request is sent through the SIP/IP Core of the Watcher to the PS.

7. The PS checks whether the event notification resumption request is acceptable, and if acceptable, the PS sends a 200 (OK) response to the Watcher Agent through the SIP/IP Core of the Watcher.

8. The PS starts to resume event notifications towards the Watcher through the Watcher Agent.

9. When the Presence Information for the Presentity changes, the PS will send the SIP NOTIFY request with the udpated Presence Information towards the Watcher Agent through the SIP/IP Core of the Watcher.

10. The Watcher Agent generates a SIP NOTIFY request to the Watcher through the SIP/IP Core of the Watcher, including the updated Presence Information as received from the PS.

11. The Watcher acknowledges the receipt of the SIP NOTIFY request with a SIP 200 (OK) response sent through the SIP/IP Core of the Watcher to the Watcher Agent.

12. The Watcher Agent generates a SIP 200 (OK) response to the PS through the SIP/IP Core of the Watcher.
When the Presence Information for the Presentity changes, the PS will send additional SIP NOTIFY requests with the udpated Presence Information towards the Watcher through the Watcher Agent.

NOTE 2: The SIP/IP Core of the Watcher and that of the Presentity are combined if the Watcher resides in the same domain as the Presentity.

G.1.3 Signaling Flows for Watchers Terminating a Subscription

G.1.3.1 Watcher-initiated Subscription Termination

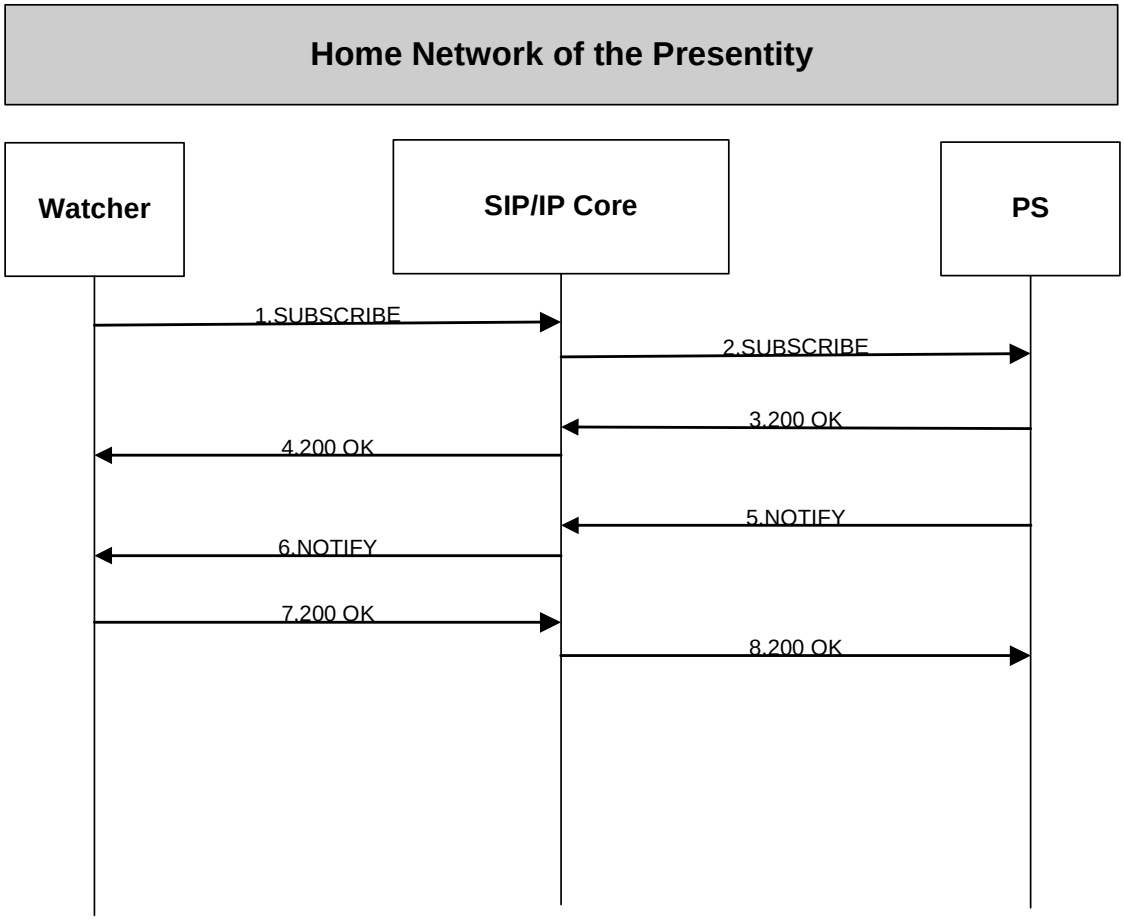

Figure 23: Watcher-initiated Subscription Termination

1. A Watcher sends a SIP SUBSCRIBE request to the SIP/IP Core with the Expires header field set to 0 indicating the terminating of the subscription, according to [RFC3265].
2. The SIP/IP Core forwards the SIP SUBSCRIBE request to the PS.

   NOTE: Even when the Watcher and the Presentity reside in different domains, the SIP/IP Core of the Watcher will foward the SIP 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 SIP SUBSCRIBE request with the Expires header set to 0 indicating the terminating a subscription operation, and sends a 200 (OK) response to the SIP/IP Core.
4. The SIP/IP Core forwards the 200 (OK) response to the Watcher.
5. The PS sends a SIP NOTIFY request to the SIP/IP Core with a Subscription-State header field set to “terminated” indicating that the subscription has been terminated, according to [RFC3265].
6. The SIP/IP Core forwards the SIP NOTIFY request to the Watcher.
7. The Watcher sends a SIP 200 (OK) response to the SIP/IP Core to acknowledge the SIP NOTIFY request.
8. The SIP/IP Core forwards the SIP 200 (OK) to the PS.

G.1.3.2 PS-initiated Subscription Termination

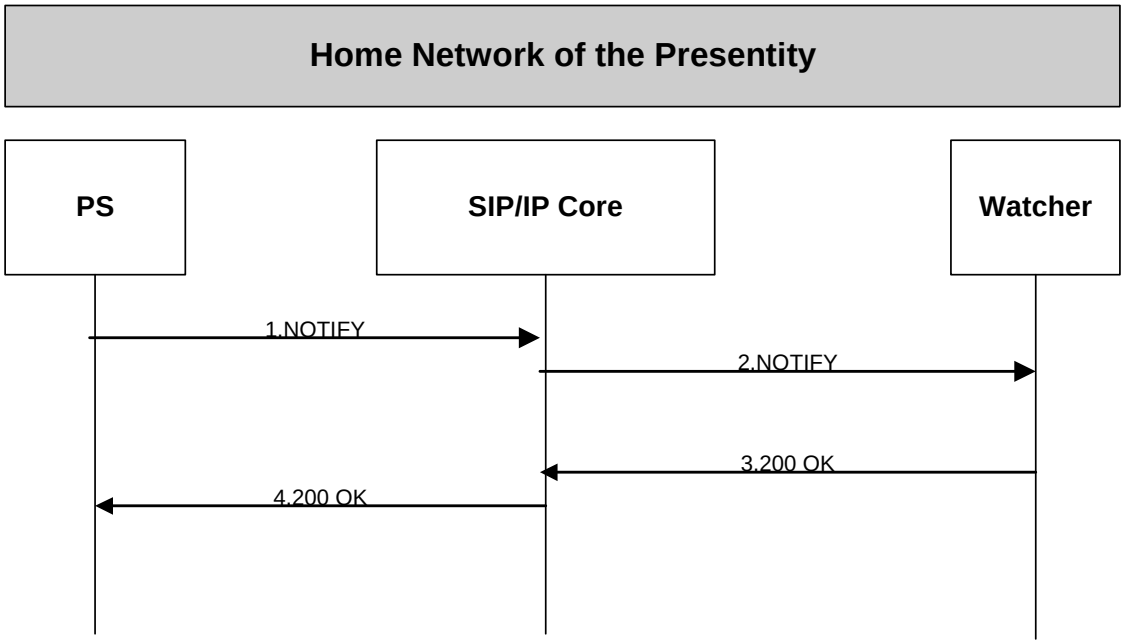

Figure 24: PS-initiated Subscription Termination

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 [RFC3265].
2. The SIP/IP Core forwards the SIP NOTIFY request to the Watcher.

   NOTE: Even when the Watcher and the Presentity reside in 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 to acknowledge the SIP NOTIFY request.
4. The SIP/IP Core forwards the SIP 200 (OK) to the PS.

G.1.4 PS Subscribing to Changes Made to Presence Subscription Rules

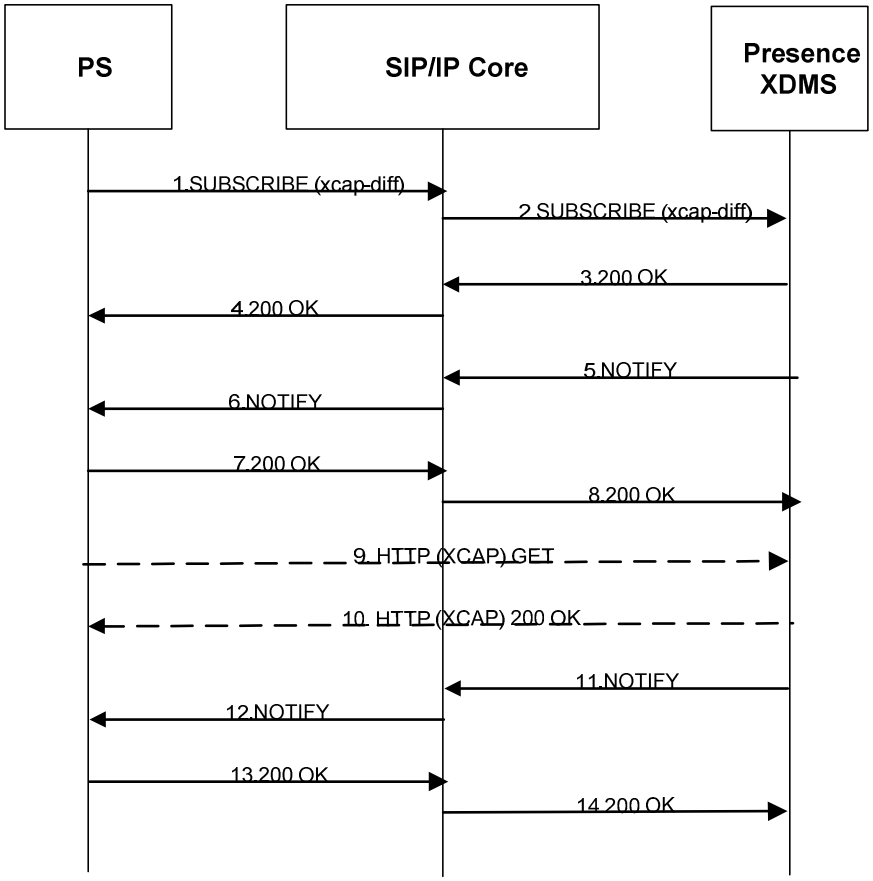

Figure 25: PS subscribing to changes made to a Presentity's Presence Subscription Rules

1. A PS that wishes to subscribe to changes made to a Presentity's Presence Subscription Rules document, sends a SIP SUBSCRIBE request with the Event header field set to "xcap-diff" as described in [XDM_Core] "*XDMC residing in an Application Server*".
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, which is used in order to synchronize the Presence XDMS and PS on a common "baseline" document as described in [IETF-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 [IETF-XCAP_diff] and [XDM_Core].
10. The version of the document requested is provided by the Presence XDMS.
11. When changes occur in the Presence Subscription 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 request with a 200 (OK) response.

14. The SIP/IP Core forwards the 200 (OK) response to the Presence XDMS.

G.1.5 Subscribing to Watcher Information State Changes

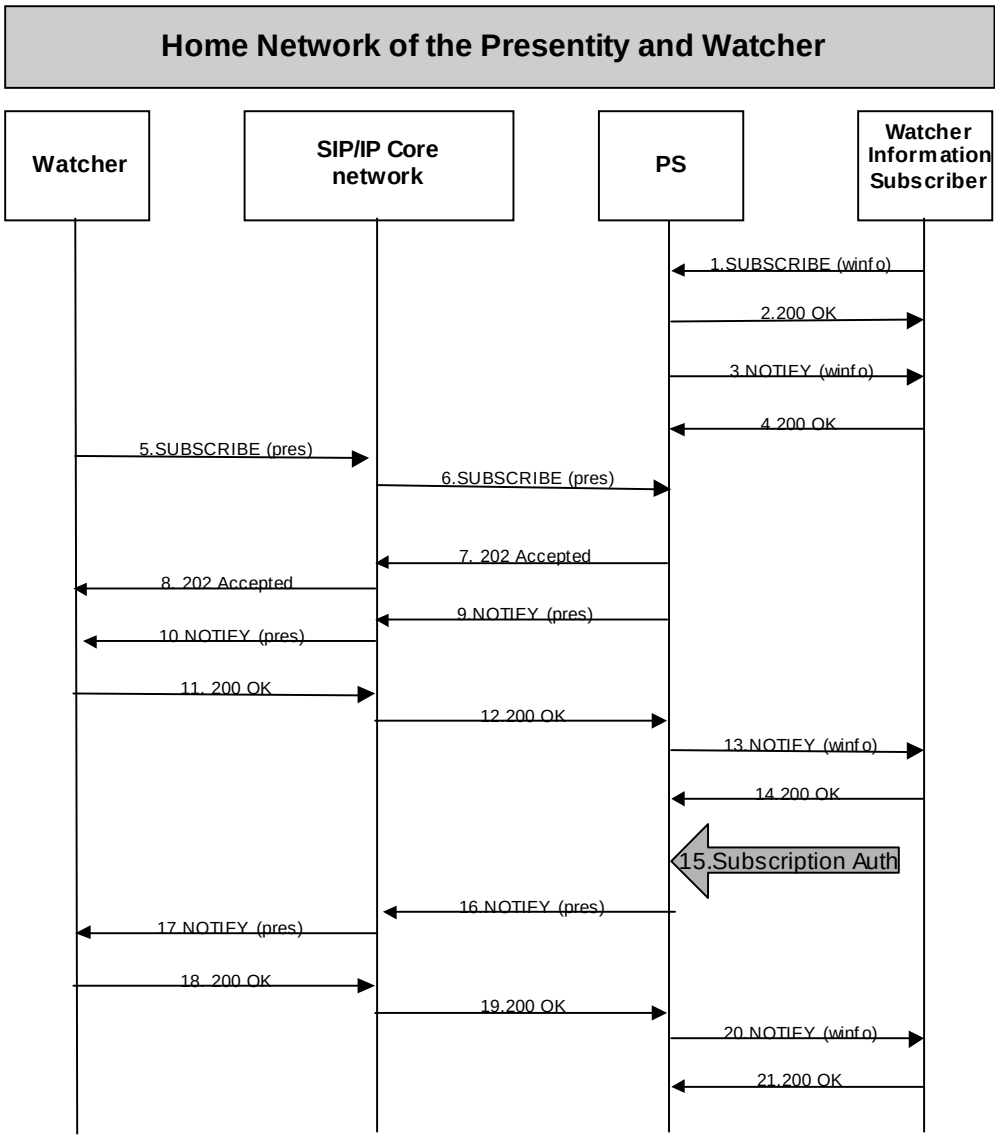

Figure 26: Watcher Information (Subscriptions/Notifications)

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 applying the Subscription Authorization Rules to the Watcher results in placing the subscription into the "pending" state.

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 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 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 Rules is insufficient to accept or deny the subscription at this time.
10. The SIP/IP Core 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 forwards the SIP 200 (OK) response to the PS.
13. As the Watcher Information state for the Presentity changes (e.g. a Watcher has requested to subscribe to the Presence Information), the PS sends a SIP NOTIFY request to indicate the change (e.g. 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 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 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.

G.1.6 Sending Different Presence Information to Different Watchers

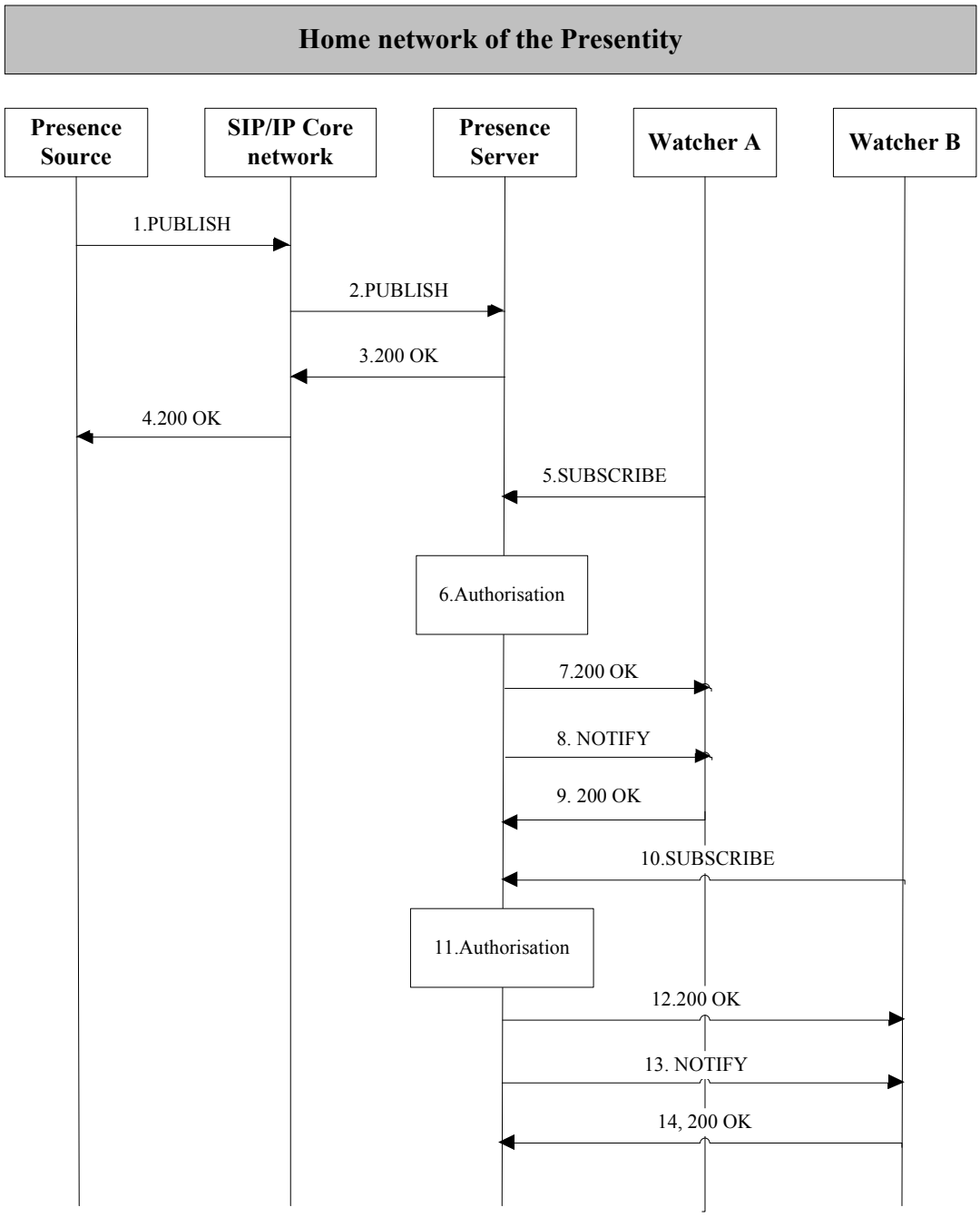

Figure 27 : Sending different Presence Information to different Watchers

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 values. The association of tuples to different Watchers and Watcher groups is based on the Presence Subscription Rules.

2. The SIP/IP Core routes the request to the corresponding PS.
3. The PS authorizes 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 the Presence Source.
4. The SIP/IP Core 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 authorization checks on Watcher A to ensure it is allowed to watch the Presentity and to watch 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 the 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 authorization checks on Watcher B to ensure it is allowed to watch the Presentity and to watch 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 the PS.