

PacketCable™ 2.0

Presence Specification

PKT-SP-PRS-I01-070628

ISSUED

Notice

This PacketCable specification is a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. (CableLabs®) for the benefit of the cable industry. Neither CableLabs, nor any other entity participating in the creation of this document, is responsible for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document by any party. This document is furnished on an AS-IS basis and neither CableLabs, nor other participating entity, provides any representation or warranty, express or implied, regarding its accuracy, completeness, or fitness for a particular purpose.

© Copyright 2007 Cable Television Laboratories, Inc.

All rights reserved.

Document Status Sheet

Document Control Number:	PKT-SP-PRS-I01-070628			
Document Title:	Presence Specification			
Revision History:	I01 - Released 07/06/28			
Date:	June 28, 2007			
Status:	Work in Progress	Draft	Issued	Closed
Distribution Restrictions:	Author Only	GL/Member	GL/Member/Vendor	Public

Key to Document Status Codes

Work in Progress	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks

CableLabs®, DOCSIS®, EuroDOCSIS™, eDOCSIS™, M-CMTS™, PacketCable™, EuroPacketCable™, PCMM™, CableHome®, CableOffice™, OpenCable™, OCAP™, CableCARD™, M-Card™, and DCAS™ are trademarks of Cable Television Laboratories, Inc.

Table of Contents

PRESENCE SPECIFICATION	I
1 INTRODUCTION	1
1.1 PACKETCABLE OVERVIEW	1
1.2 GENERAL DESCRIPTION OF PRESENCE.....	1
1.3 REQUIREMENTS	2
2 REFERENCES	3
2.1 NORMATIVE REFERENCES	3
2.2 INFORMATIVE REFERENCES.....	3
2.3 REFERENCE ACQUISITION	5
3 TERMS AND DEFINITIONS	6
4 ABBREVIATIONS AND ACRONYMS	7
5 PACKETCABLE PRESENCE BACKGROUND	9
5.1 RELATIONSHIP WITH EXISTING PRESENCE SOLUTIONS.....	9
5.2 PACKETCABLE PRESENCE DOCUMENT ORGANIZATION	10
6 PACKETCABLE PRESENCE ARCHITECTURE	11
6.1 PRESENCE FUNCTIONAL COMPONENTS	13
6.1.1 <i>Presence Server</i>	13
6.1.2 <i>Presence Source</i>	14
6.1.3 <i>Watcher</i>	14
6.1.4 <i>Resource List Server (RLS)</i>	15
6.1.5 <i>XML Document Management Server (XDMS)</i>	15
6.1.6 <i>XML Document Management Client (XDMS)</i>	15
6.1.7 <i>Aggregation Proxy</i>	16
6.1.8 <i>Content Server</i>	16
6.2 PRESENCE REFERENCE POINTS.....	16
6.2.1 <i>PRS-1: Presence Source – SIP Core Network</i>	16
6.2.2 <i>PRS-2: Watcher – SIP Core Network</i>	16
6.2.3 <i>PRS-3: Presence Server – SIP Core Network</i>	16
6.2.4 <i>PRS-4: Resource List Server – SIP Core Network</i>	17
6.2.5 <i>PRS-5: Presence Server – Shared XDMS</i>	17
6.2.6 <i>PRS-6: Presence XDMS – SIP Core Network</i>	17
6.2.7 <i>PRS-7: Presence XDMS – Aggregation Proxy</i>	17
6.2.8 <i>PRS-8: Presence Server – Presence XDMS</i>	17
6.2.9 <i>PRS-9: Resource List Server – Shared XDMS</i>	17
6.2.10 <i>PRS-10: Resource List Server – RLS XDMS</i>	18
6.2.11 <i>PRS-11: RLS XDMS – SIP Core Network</i>	18
6.2.12 <i>PRS-12: RLS XDMS – Aggregation Proxy</i>	18
6.2.13 <i>PRS-13: Presence Source – Content Server</i>	18
6.2.14 <i>PRS-14: Watcher – Content Server</i>	18
6.2.15 <i>PRS-15: Presence Server – Content Server</i>	18
6.2.16 <i>XDM-1: XDM Client – SIP Core Network</i>	19
6.2.17 <i>XDM-2: Shared XDMS – SIP Core Network</i>	19
6.2.18 <i>XDM-3: XDM Client – Aggregation Proxy</i>	19
6.2.19 <i>XDM-4: Shared XDMS – Aggregation Proxy</i>	19
6.2.20 <i>IP-1: SIP/IP Core network – Remote Presence Network (based on a SIP/IP Core)</i>	19

6.3 PRESENCE PROTOCOLS AND DATA FORMATS19

 6.3.1 *Event Notification*19

 6.3.2 *Publishing Presence Information*20

 6.3.3 *Presence Information Data Format (PIDF)*.....21

 6.3.4 *Watcher Information*.....21

 6.3.5 *Presence Authorization Rules*.....22

 6.3.6 *Performance Enhancements*23

6.4 XDM PROTOCOLS23

6.5 CLIENT PROVISIONING AND MANAGEMENT24

6.6 ACCOUNTING24

6.7 SECURITY24

6.8 CODEC.....25

7 PACKETCABLE PRESENCE REQUIREMENTS.....26

 7.1 OMA PRESENCE ENABLER.....26

 7.1.1 *OMA Presence SIMPLE Specification*.....26

 7.1.2 *OMA Presence SIMPLE XDM Specification*.....27

 7.1.3 *OMA Presence SIMPLE RLS XDM Specification*27

 7.2 OMA XDM ENABLER.....27

 7.2.1 *OMA XDM Core Specification*27

 7.2.2 *OMA XDM Shared Specification*.....27

 7.3 PROVISIONING AND MANAGEMENT27

APPENDIX I INFORMATIVE CALL FLOW.....28

APPENDIX II ACKNOWLEDGEMENTS30

Figures

FIGURE 1 - OMA PRESENCE ARCHITECTURE11

FIGURE 2 - PACKETCABLE PRESENCE ARCHITECTURE: FUNCTIONAL COMPONENTS12

FIGURE 3 - PACKETCABLE PRESENCE ARCHITECTURE: REFERENCE POINTS13

FIGURE 4 - INFORMATIVE CALL FLOW ILLUSTRATING A PRESENCE SUBSCRIPTION29

Tables

TABLE 1 – OMA PRESENCE-RELATED DOCUMENTS9

TABLE 2 – OMA TO PACKETCABLE PRESENCE PACM CONFIGURATION ELEMENT MAPPING27

1 INTRODUCTION

1.1 PacketCable Overview

The ever-increasing capability and potential of telecommunications networks and the emerging converged fixed-mobile technologies are providing innovative services beyond today's basic high-speed Internet to a vast number of cable broadband customers. Real-time IP-based communication services such as Voice over IP are being offered to consumers through a wide-range of client devices and media types. Cutting edge technologies such as Video over IP communications and the ability to display voice and video mail message notifications on TV-sets are changing the way communication and entertainment services are offered. These are all new opportunities for cable operators to offer new value added services to consumers in a cost-effective manner. The PacketCable project is a CableLabs specification initiative designed to do just that by extending cable network's real-time IP communication service spectrum and accelerating the introduction of converged IP multimedia and mobility technologies into the cable network. It defines a platform consisting of architecture and a set of open interfaces, which leverages emerging communications technologies, such as the IETF Session Initiation Protocol (SIP) and the 3GPP IP Multimedia Subsystem (IMS), to support the rapid introduction of new IP-based services onto the cable network.

PacketCable leverages open standards and specifications wherever possible and provides enhancements to ensure it addresses requirements that may be specific to the Cable Industry that are not already addressed by other standards bodies. Based on a modular approach to the design and development of the specification, the PacketCable specifications allow cable operators to flexibly deploy those network capabilities that are required for their specific service offerings in order to meet a large number of applications while maintaining interoperability across a variety of devices from multiple suppliers. Intentionally being designed as "non-service-specific," the PacketCable platform currently includes the basic capabilities necessary for cable operators to deploy services in areas such as Enhanced Residential VoIP and IP Video Communications, mobility services through integration with cellular and other wireless networks, QoS-enabled audio and video streaming, and Residential SIP Telephony Extensions. A more detailed and complete listing of the current PacketCable capabilities can be found in the PacketCable Architecture Framework Technical Report document [ARCH-FRM-TR].

This specification adds the Presence functionality to the PacketCable architecture. This enables the exchange of dynamic information about the state of a logical entity (e.g., person, device) and its availability and willingness to communicate or to share. This information can be exploited, for example, by application servers to optimize or customize services. This specification describes the PacketCable Presence architecture that is aligned with the OMA Presence service. This specification also adopts the normative requirements from the relevant OMA specifications and identifies the exceptions where applicable (Section 7).

1.2 General Description of Presence

Presence can be thought of as dynamic information about a presentity that is made available to others. Many people have become familiar with presence through popular applications such as AOL Instant Messenger, which has a "buddy list" that allows a watcher to see who is available for messaging. Presence enables more effective communication by allowing a user to advertise his availability and willingness to communicate, and by what means.

Although presence has its roots in instant messaging, the concept of presence has broadened to encompass any information that is relevant for watchers to know and that the presentity is willing to share. This includes "context sharing," such as mood, location, and activity. Other presence states may convey information about bandwidth, device capabilities, or roaming status, which can be exploited by application servers to optimize or customize services such as gaming or Push-to-Talk over Cellular (PoC). As illustrated in the last example, watchers or presentities need not be human, but could be devices, applications, or any number of things.

Presence information can be provided by a variety of sources to form an aggregated view of the presentity's presence state. A calendar application on a PC might automatically publish that the presentity is in a meeting, a mobile device

might publish the presentity's desire not to be disturbed based on real-time user input, and the network might publish a presentity's availability based on successful registration of a device. When combined together, the presentity is viewed as being "online," but in a meeting and not willing to be disturbed.

A watcher can fetch information from the presence service on a per-request basis or subscribe to receive immediate notification of future changes in the information. This means, for example, that if the user turns off his mobile device and the device deregisters with the network, then the network may publish this event and subscribed watchers would immediately receive notification that the presentity is now "offline."

Due to the personal nature of presence information, it is distributed according to policies configured by the presentity and/or service administrator, which dictate what presence information can be made available to whom. Such policies enable a presentity to provide different views of his presence state based on whether he is at work or at home, or whether a watcher is a co-worker or a family member. For example, a user may wish to appear available to family members at all times, but to co-workers only during working hours.

Presence provides the framework for exchanging information that enables more convenient communication and customization of services.

1.3 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST"	This word means that the item is an absolute requirement of this specification.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[OMA TS-PRS]	OMA-TS-Presence_SIMPLE-V1_0_1-20061128-A, Presence SIMPLE Specification, November 2006, Open Mobile Alliance.
[OMA TS-PRSXDM]	OMA-TS-Presence_SIMPLE_XDM-V1_0_1-20061128-A, Presence XDM Specification, November 2006, Open Mobile Alliance.
[OMA TS-RLSXDM]	OMA-TS-Presence_SIMPLE_RLS_XDM-V1_0_1-20061128-A, Resource List Server (RLS) XDM Specification, November 2006, Open Mobile Alliance.
[OMA TS-SHDXDM]	OMA-TS-XDM_Shared-V1_0_1-20061128-A, Shared XDM Specification, November 2006, Open Mobile Alliance.
[OMA TS-XDM]	OMA-TS-XDM_Core-V1_0_1-20061128-A, XML Document Management (XDM) Specification, November 2006, Open Mobile Alliance.

2.2 Informative References

The following informative references are used in this specification.

[ACCT]	PacketCable 2.0 Accounting Specification. PKT-SP-ACCT-I02-061013, October 13, 2006, Cable Television Laboratories, Inc.
[ARCH-FRM-TR]	PacketCable Architecture Framework Technical Report, PKT-TR-ARCH- PKT-TR-ARCH-FRM-V02-061013, October 13, 2006, Cable Television Laboratories, Inc.
[CODEC-MEDIA]	PacketCable 2.0 Codec and Media Specification, PKT-SP-CODEC-MEDIA-I02-061013, October 13, 2006, Cable Television Laboratories, Inc.
[ID PARTNOT]	IETF draft, draft-ietf-simple-partial-notify-08, Session Initiation Protocol (SIP) extension for Partial Notification of Presence Information, July 2006.
[ID PRRULES]	IETF draft, draft-ietf-simple-presence-rules-08, Presence Authorization Rules, October 2006.
[ID XCAP]	IETF draft, draft-ietf-simple-xcap-12, The Extensible Markup Language (XML) Configuration Access Protocol (XCAP), October 2006.
[OMA AD-PRS]	OMA-AD-Presence_SIMPLE-V1_0-20061128-A, Presence SIMPLE Architecture Document, November 2006, Open Mobile Alliance.
[OMA AD-XDM]	OMA-AD-XDM-V1_0-20060612-A, XML Document Management Architecture, June 2006, Open Mobile Alliance.
[OMA RD-PRS]	OMA-RD-Presence_SIMPLE-V1_0-20060725-A, Presence SIMPLE Requirements, July 2006, Open Mobile Alliance.
[OMA RD-XDM]	OMA-RD-XDM-V1_0-20060612-A, XML Document Management Requirements, June 2006, Open Mobile Alliance.
[OMA PRES-MO]	OMA-TS-Presence_SIMPLE_MO-V1_0_1-20061128-A, OMA Management Object for SIMPLE Presence, November 2006, Open Mobile Alliance.
[OMA XDM-MO]	OMA-TS-XDM_MO-V1_0_1-20061128-A, OMA Management Object for XML Document Management, November 2006, Open Mobile Alliance.

- [PACM] PacketCable 2.0 Provisioning, Activation, Configuration and Management Specification, PKT-SP-PACM-I02-061013, October 13, 2006, Cable Television Laboratories, Inc.
- [PKT 33.203] PacketCable 2.0, IMS Delta Specifications, 3G security; Access security for IP-based services Specification 3GPP TS 33.203, PKT-SP-33.203-I02-061013, October 13, 2006, Cable Television, Inc.
- [PKT 33.210] PacketCable 2.0 IMS Delta Specifications, 3G Security; Network Domain Security; IP network layer security Specification 3GPP TS 33.210, PKT-SP-33.210-I02-061013, October 13, 2006, Cable Television, Inc.
- [PKT 33.220] PacketCable 2.0, IMS Delta Specifications, Generic Authentication Architecture (GAA); Generic bootstrapping architecture Specification, 3GPP TS 33.220, PKT-SP-33.220-I02-061013, October 13, 2006, Cable Television Laboratories, Inc.
- [PRS PACM] PacketCable 2.0 Presence Provisioning, Activation, Configuration and Management Specification, PKT-SP-PRS-PACM-I01-070628, June 28, 2007, Cable Television Laboratories, Inc.
- [RFC 2617] IETF RFC 2617, HTTP Authentication: Basic and Digest Access Authentication, June 1999.
- [RFC 3265] IETF RFC 3265, Session Initiation Protocol (SIP)-Specific Event Notification, June 2002.
- [RFC 3856] IETF RFC 3856, A Presence Event Package for the Session Initiation Protocol (SIP), August 2004.
- [RFC 3857] IETF RFC 3857, A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP), August 2004.
- [RFC 3863] IETF RFC 3863, Presence Information Data Format (PIDF), August 2004.
- [RFC 3903] IETF RFC 3903, Session Initiation Protocol (SIP) Extension for Event State Publication, October 2004.
- [RFC 4119] IETF RFC 4119, A Presence-based GEOPRIV Location Object Format, December 2005.
- [RFC 4480] IETF RFC 4480, RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF), July 2006.
- [RFC 4483] IETF RFC 4483, A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages, May 2006.
- [RFC 4660] IETF RFC 4660, Functional Description of Event Notification Filtering, September 2006.
- [RFC 4662] IETF RFC 4662, A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists, August 2006.
- [TS 22.141] 3GPP TS 22.141 V6.5.0, Presence Service; Stage 1 (Release 6), December 2005.
- [TS 23.141] 3GPP TS 23.141 V6.9.0, Presence Service; architecture and functional description (Release 6), December 2005.
- [TS 24.141] 3GPP TS 24.141 V6.7.0, Presence service using the IP Multimedia (IM) Core Network (CN) subsystem; Stage 3 (Release 6), September 2006.
- [TS 26.141] 3GPP TS 26.141 V6.3.0, IP Multimedia System (IMS) Messaging and Presence; Media formats and codecs (Release 6), March 2006.
- [TS 33.141] 3GPP TS 33.141 V6.2.0, Presence service; Security (Release 6), September 2005.
- [TS 33.222] 3GPP TS 33.222 V6.6.0, Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS), March 2006.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; Internet: <http://www.cablelabs.com/>.
- Internet Engineering Task Force (IETF), Internet: <http://www.ietf.org/>.

Note: Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time.

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

- 3rd Generation Partnership Project (3GPP), ETSI Mobile Competence Centre, 650 route des Lucioles, 06921 Sophia-Antipolis Cedex, France, Internet: <http://www.3gpp.org/>.
- Open Mobile Alliance (OMA), OMA Office, 4275 Executive Square, Suite 240, La Jolla, CA 92037, Fax +1-858-623-0743, Internet: <http://www.openmobilealliance.com/>.

3 TERMS AND DEFINITIONS

This specification uses the following terms and definitions:

Application Usage	Detailed information on the interaction of an application with the XCAP server [ID XCAP].
Enabler	A term used by OMA to mean "a technology intended for use in the development, deployment or operation of a service." Examples of OMA Enablers are Device Management, Push-to-Talk over Cellular (PoC), Presence SIMPLE, and XML Document Management (XDM).
Fetcher	A form of Watcher that has asked the Presence Service for the Presence Information of one or more Presentities, but is not requesting a Notification from the Presence Service of (future) changes in a Presentity's Presence Information [OMA RD-PRS].
Notification	A message sent from the Presence Service to a Subscriber when there is a change in the Presence Information of some Presentity of interest, as recorded in one or more Subscriptions [OMA RD-PRS].
Presence Information	Dynamic set of information pertaining to a Presentity that may include Presence Information Elements such as the status, reachability, willingness, and capabilities of that Presentity [OMA RD-PRS].
Presence Information Element	A basic unit of Presence Information [OMA RD-PRS].
Presence List	A list of presentities that can have their individual states subscribed to with a single subscription request (e.g., a list subscription).
Presence Server	A logical entity that receives Presence Information from a multitude of Presence Sources pertaining to the Presentities it serves and makes this information available to Watchers according to the rules associated with those Presentities [OMA RD-PRS].
Presence Service	The capability to support management of Presence Information between Watchers and Presentities, in order to enable applications and services to make use of Presence Information [OMA RD-PRS].
Presence Source	A logical entity that provides Presence Information pertaining to exactly one or more Presentities to the Presence Server. 3GPP Presence User Agents, Presence Network Agents, and Presence External Agents are examples of Presence Sources [OMA RD-PRS].
Presentity	A logical entity that has Presence Information associated with it. This Presence Information may be composed from a multitude of Presence Sources. A Presentity is most commonly a reference for a person, although it may represent a role such as "help desk" or a resource such as "conference room #27." Presentities are generally referenced by distinguished names, such as "john.smith@example.com" or by phone numbers like "+19724735455." In SIMPLE, presentities are generally referenced using a sip:, pres: or tel: URL [OMA RD-PRS].
Subscriber	A form of watcher that has asked the Presence service to notify it immediately of changes in the Presence Information of one or more presentities [OMA RD-PRS].
Subscription	The information kept by the presence service about a subscriber's request to be notified of changes in the Presence Information of one or more presentities [OMA RD-PRS].
Watcher	Any uniquely identifiable entity that requests Presence Information about a presentity, or watcher information about a watcher, from the Presence service. Special types of watchers are fetchers and subscribers [OMA RD-PRS].
Watcher Information	Information about watchers that have received or may receive Presence Information about a particular presentity within a particular recent span of time [OMA RD-PRS].

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations and acronyms:

3GPP	3 rd Generation Partnership Project
3GPP2	3 rd Generation Partnership Project 2
AP	Aggregation Proxy
AS	Application Server
AUID	Application Unique ID
AVP	Attribute-Value Pair
BSF	Bootstrapping Server Function
CDF	Charging Data Function
CGF	Charging Gateway Function
CSCF	Call Session Control Function
DM	Device Management
DMC	Device Management Client
DMS	Device Management Server
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
HTTP	Hyper Text Transfer Protocol
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extensions
MMD	Multimedia Domain
MO	Management Object
OMA	Open Mobile Alliance
PACM	Provisioning, Activation, Configuration, and Management
P-CSCF	Proxy Call Session Control Function
PIDF	Presence Information Data Format
PoC	Push-to-Talk over Cellular
QoS	Quality of Service
RLS	Resource List Server
SIMPLE	SIP for Instant Message and Presence Leveraging Extensions
SIP	Session Initiation Protocol
TLS	Transport Layer Security
UE	User Equipment
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VoIP	Voice over Internet Protocol
XCAP	XML Configuration Access Protocol

XDM	XML Document Management
XDMC	XML Document Management Client
XDMS	XML Document Management Server
XML	Extensible Markup Language

5 PACKETCABLE PRESENCE BACKGROUND

5.1 Relationship with Existing Presence Solutions

As one of its design goals, PacketCable seeks to leverage existing industry standards and open protocols whenever possible. Specifically, PacketCable is aligned with the IP Multimedia Subsystem (IMS) as defined by the 3rd Generation Partnership Project (3GPP).

3GPP includes a set of specifications ([TS 22.141], [TS 23.141], [TS 24.141], [TS 26.141], and [TS 33.141]) which define a SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE)-based presence service framework as an IMS service. However, early in 2004, 3GPP agreed to transition its Presence work item to the Open Mobile Alliance (OMA).

OMA was formed in 2002 as an industry forum aiming to consolidate, under one umbrella, all mobile service enabler specification development. OMA had begun working on its own set of specifications for a SIMPLE-based presence service. Aware that several standards committees (e.g., 3GPP, 3GPP2, and IETF) were already engaged in parallel presence-related activities, the OMA work proceeded with the following goals:

- to identify consolidated presence service requirements, which define a generalized set of presence features and capabilities that satisfy the needs of a variety of presence-enabled applications (e.g., PoC, IM, Gaming);
- to define a common framework based on SIP/SIMPLE that would be agnostic to the underlying network, in order to assure interoperability amongst presence implementations; and
- to describe the relationship of the common presence framework with existing SIP/IP core networks (e.g., 3GPP IMS and 3GPP2 MMD).

There are two OMA enablers which, together, specify a complete presence service using a 3GPP IMS core network. The OMA Presence SIMPLE enabler specifies the presence protocol and data formats based on SIP/SIMPLE, and the OMA XML Document Management (XDM) enabler specifies the means for accessing and managing user information (e.g., presence lists, presence authorization policy, etc) that is stored in the network. Table 1 shows a list of OMA specifications that make up the two enablers.

Table 1 – OMA Presence-related Documents

Document Title	Document Description	Reference
Presence SIMPLE:		
Presence SIMPLE Requirements	Contains use cases and requirements for the presence service.	[OMA RD-PRS]
Presence SIMPLE Architecture Document	Describes a network-agnostic model for mobile presence based on the IETF SIP, SIMPLE, and XCAP framework.	[OMA AD-PRS]
Presence SIMPLE Specification	Describes how the set of IETF specifications and OMA-specified extensions are used to create a uniform presence service, using a SIP/IP Core based on 3GPP IMS capabilities.	[OMA TS-PRS]
Presence XDM Specification	Describes the XCAP application usage for presence authorization rules.	[OMA TS-PRSXDM]
Resource List Server (RLS) XDM Specification	Describes the XCAP application usage for presence lists.	[OMA TS-RLSXDM]

Document Title	Document Description	Reference
XML Document Management (XDM):		
XML Document Management Requirements	Contains use cases and requirements for the management of user information stored in the network.	[OMA RD-XDM]
XML Document Management Architecture	Introduces the functionality and architecture of the XDM service enabler.	[OMA AD-XDM]
XML Document Management Specification	Describes the common protocols and application usages needed to provide XDM services to other enablers, using a SIP/IP Core based on 3GPP IMS capabilities.	[OMA TS-XDM]
Shared XDM Specification	Describes the XCAP application usages for URI lists and group usage lists, which may be shared by multiple service enablers.	[OMA TS-SHDXDM]

PacketCable adopts the approach of basing its Presence architecture on OMA, for reasons that include:

- improved interoperability – OMA specifies presence composition policy, and detailed semantics of the supported presence elements;
- more flexible architecture – OMA separates application data from the Presence Server by introducing the XDMS functional element; and
- future evolution – 3GPP has transferred its work item on Presence enhancements to OMA.

5.2 PacketCable Presence Document Organization

The organization of the PacketCable Presence-related documents is based upon the need to closely align with OMA. There are two specifications that specify the PacketCable Presence architecture: the Presence Specification (this document) and the Presence PACM Specification.

This document (Presence Specification) describes the PacketCable Presence architecture and its relationship with the OMA Presence model. An overview of the functionalities supported by the OMA Presence SIMPLE enabler and the OMA XML Document Management enabler is also provided. This document also specifies the normative requirements on the functional components for support of presence. This is done by adopting the relevant specifications from OMA and listing the differences and exceptions for the PacketCable environment.

The provisioning parameters used by Presence are described in the PacketCable Presence PACM Specification [PRS PACM]. [PRS PACM] defines the data models that are required by the PacketCable Presence architecture. Additionally, [PRS PACM] describes the architecture and functional elements required to enable configuration of a UE that supports Presence as defined in this specification. It also specifies the normative requirements on the elements of that architecture that are in addition to those provided by the underlying PacketCable PACM specifications [PACM].

6 PACKETCABLE PRESENCE ARCHITECTURE

The PacketCable presence architecture is closely aligned with the OMA presence architecture, which in turn is based on the IETF SIMPLE framework and provides a network-agnostic model for presence services. The OMA presence architecture, as defined in [OMA AD-PRS], is shown in Figure 1.

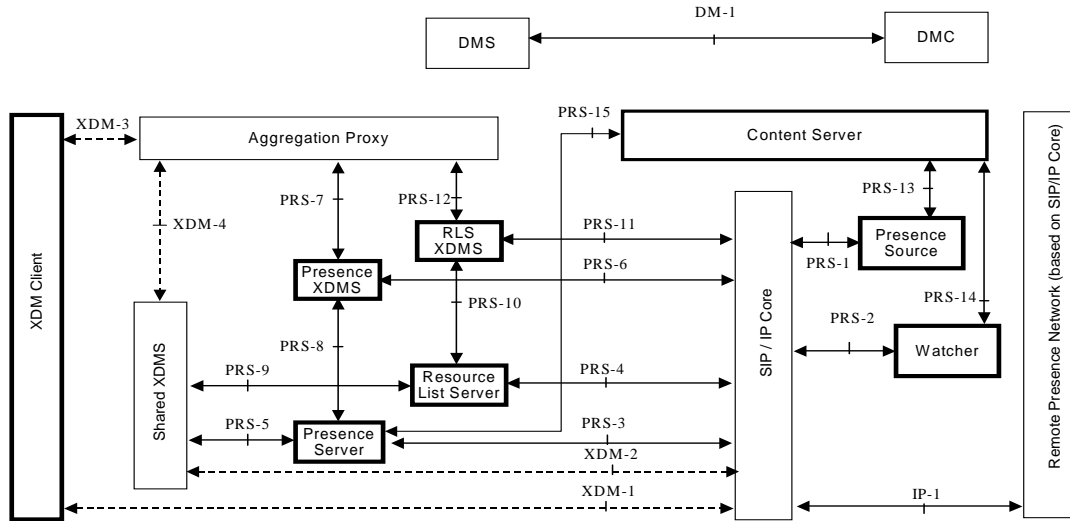
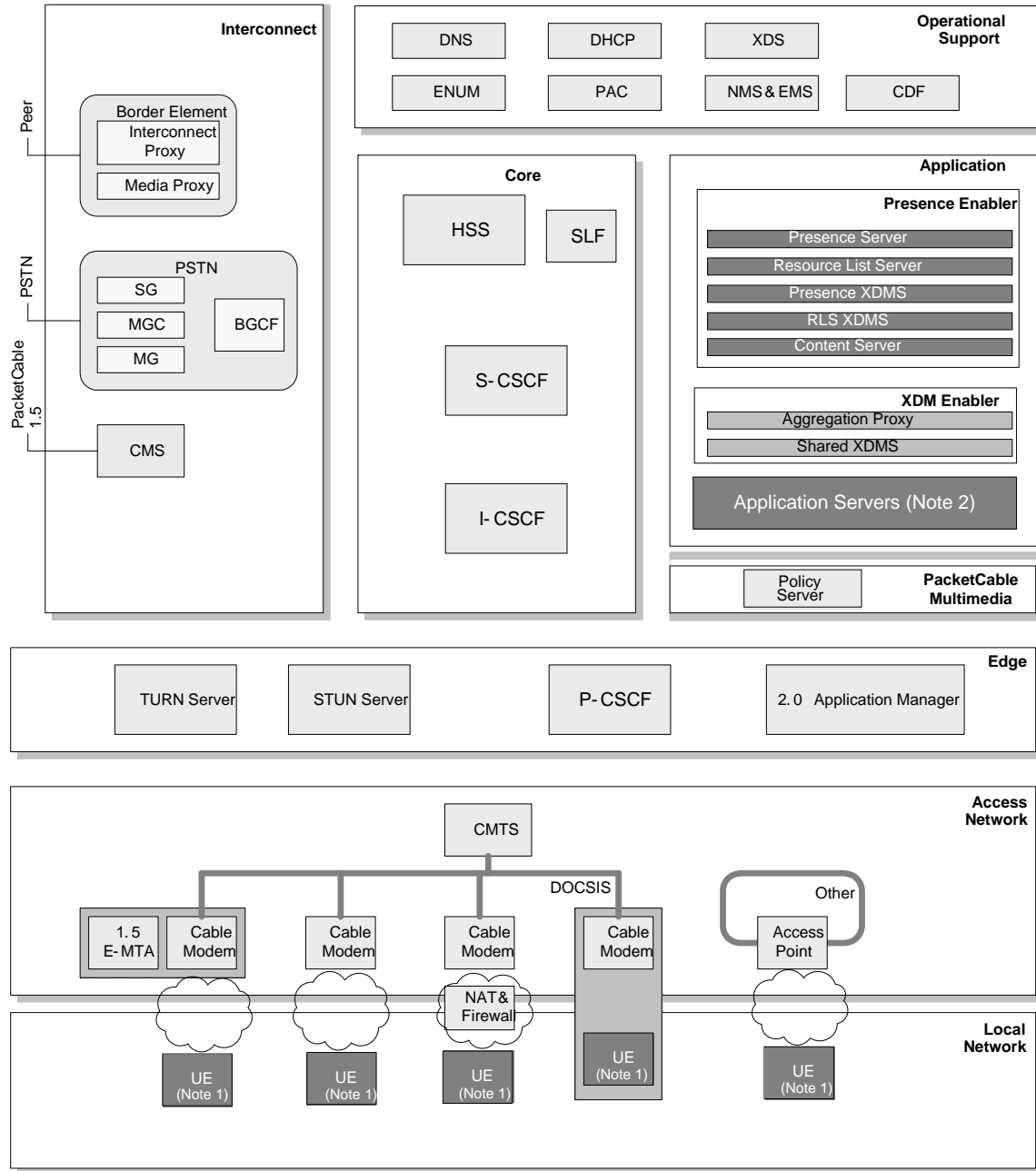


Figure 1 - OMA Presence Architecture

Figure 2 maps the OMA Presence architecture to the PacketCable architecture and highlights the Presence-related functional components within the PacketCable architecture. Collectively, these components represent the extensions to the base PacketCable architecture [ARCH-FRM-TR] in support of the Presence functionality. As illustrated, the Presence Server, the Resource List Server, the Presence XDMS, the RLS XDMS, the Shared XDMS and the Aggregation Proxy in the OMA Presence model are all mapped to application components. The Shared XDMS and the Aggregation Proxy are associated with the OMA XDM Enabler [OMA AD-XDM], and the rest are associated with the OMA Presence SIMPLE Enabler [OMA AD-PRS]. Furthermore, the XDMC, the Presence Source, and the Watcher in the OMA Presence model are mapped to Application Servers and UEs, typically as embedded elements.



Note 1: The UE may support any combined functionality of Presence Source, Watcher and XDMC.
Note 2: The AS may support any combined functionality of Presence Source, Watcher and XDMC.

Figure 2 - PacketCable Presence Architecture: Functional Components

Providing another view of the PacketCable Presence architecture, Figure 3 shows the reference points related to the Presence functional components in Figure 2, with each reference point being marked according to their dual-roles within the OMA Presence application architecture and the PacketCable network architecture. A detailed description of these reference points is in Section 6.2.

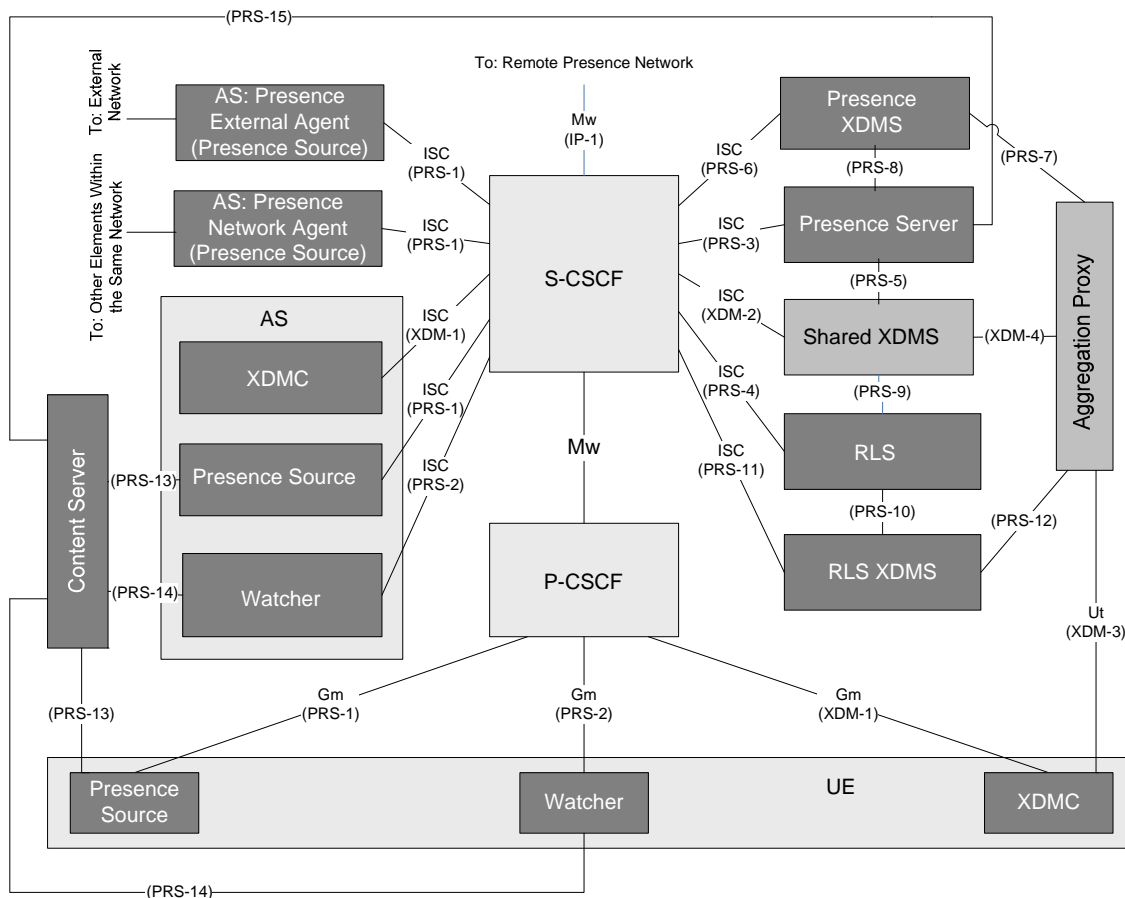


Figure 3 - PacketCable Presence Architecture: Reference Points

As shown in Figure 3, there is no one-to-one mapping between the OMA reference points and the PacketCable reference points. For instance, the ISC reference point in PacketCable is mapped from multiple OMA reference points (XDM-1, XDM-2, PRS-1, PRS-2, PRS-3, PRS-4, PRS-6 and PRS-11). On the other hand, each of the OMA reference points XDM-1, PRS-1 and PRS-2 is mapped to two different reference points in PacketCable, depending on whether the associated OMA functionality resides in a UE or an AS. Furthermore, some OMA reference points (XDM-4, PRS-5, PRS-7, PRS-8, PRS-9, PRS-10, PRS-12, PRS-13, PRS-14 and PRS-15) do not correspond to any reference points in the base PacketCable architecture [ARCH-FRM-TR]; they are adopted by the PacketCable Presence architecture.

6.1 Presence Functional Components

6.1.1 Presence Server

The Presence Server is responsible for managing presence information on behalf of a presentity. The Presence Server performs the following functions:

- receives and manages presence information that is published by the Presence Source;
- aggregates presence information received for a particular presentity from multiple Presence Sources, and composes the information into a single presence document;
- allows watchers to fetch or subscribe to the presence information of a presentity (i.e., acts as presence agent);
- provides a configurable filtering function that is used to limit the information that is sent to a watcher;

- generates partial notifications to watchers that have indicated the capability to process them;
- allows watchers to fetch or subscribe to watcher information;
- enforces presence authorization rules;
- fetches XML documents (i.e., presence authorization rules and URI lists) stored in the Presence XDMS and Shared XDMS; and
- subscribes to the SIP event package for modifications to XML documents stored in the Presence XDMS and Shared XDMS.

In the PacketCable Presence architecture, the user's presence authorization rules are managed by the Presence XDMS (Section 6.1.5).

6.1.2 Presence Source

A Presence Source provides Presence information to the Presence Server. A Presence Source may be located in the UE or in the network. Presence information for a single presentity may come from multiple Presence Sources.

The Presence Source performs the following functions:

- collects presence information associated with a presentity;
- assembles the presence information into a suitable format;
- publishes it to the Presence Server; and
- identifies itself uniquely among other Presence Sources of the same presentity when publishing presence information.

It should be noted that the 3GPP and IETF specifications use different terms for this entity, as detailed below. To avoid confusion, OMA has adopted the term Presence Source [OMA RD-PRS] to cover any entity that collects and sends presence information on behalf of the presentity. PacketCable adopts the OMA terminology for the Presence Source.

3GPP distinguishes three types of Presence Sources [TS 23.141]:

- The Presence User Agent (PUA) collects and sends user-related presence information to the Presence Server on behalf of the presentity. The PUA can be located in either the terminal or network. A 3GPP PUA also manages subscription authorization policy.
- The Presence Network Agent (PNA) collects and sends network-related presence information to the Presence Server on behalf of the presentity. The PNA is located in the network.
- The Presence External Agent (PEA) supplies presence information from external networks, and is responsible for handling the interworking and security issues involved with interfacing to external networks. Examples of presence information that the PEA may provide include third party services (e.g., calendar applications) and non SIMPLE-based presence services.

The IETF defines only one type of Presence Source, which it also calls a Presence User Agent [RFC 3856], but defines it differently than 3GPP.

6.1.3 Watcher

A Watcher requests information from the Presence Server. The Watcher performs the following functions:

- fetches or subscribes to presence information about a presentity or list of presentities; and
- fetches or subscribes to watcher information.

6.1.4 Resource List Server (RLS)

The RLS accepts and manages subscriptions to presence lists, which enable a Watcher to request the presence information of multiple presentities using a single transaction. The RLS performs the following functions:

- accepts subscriptions to presence lists;
- authorizes the watcher's usage of the presence list;
- creates and manages back-end subscriptions to all presentities in the presence list, on behalf of the watcher;
- sends notifications to the watcher, based on information received from the back-end subscriptions;
- applies aggregation and rate control mechanisms to the notifications, as appropriate;
- fetches XML documents (i.e., presence lists and URI lists) stored in the RLS XDMS and Shared XDMS; and
- subscribes to the SIP event package for modifications to XML documents stored in the RLS XDMS and Shared XDMS.

The RLS has many similar functions as the Presence Server, and, in a physical implementation, may in fact be collocated. However, one important architectural distinction is that a subscription request to a presence list is routed to the RLS of the watcher, while a subscription request to a presentity is sent to the Presence Server of the presentity.

6.1.5 XML Document Management Server (XDMS)

The XDMS performs the following functions:

- manages documents reused by multiple services (i.e., Shared XDMS), or documents that are service-specific (i.e., Presence XDMS and RLS XDMS);
- provides storage of the XML documents managed by it; and
- accepts subscriptions for modifications to XML documents managed by it, and acts as notifier when modifications are made.

In the PacketCable Presence architecture, the user's Presence Lists are managed by the RLS XDMS. Documents representing user's Presence Authorization Rules are stored in the Presence XDMS.

The Shared XDMS manages URI lists, which is a convenient way to group URIs that can then be referenced by documents stored in other, service-specific XDMS (e.g., Presence XDM, RLS XDMS, etc.). This enables multiple services to reuse a common list.

As an example, the presence user can create and store a list in the Shared XDMS, where the example list is given the identity sip:friends@example.com. The user might then reuse the list as follows:

- to create a presence list that references sip:friends@example.com as the list of presentities; and
- to create a presence authorization policy that references sip:friends@example.com as the accept list (i.e., those who are automatically authorized to subscribe to the users full presence information).

The XDMS includes XCAP Server and SIP Notifier [RFC 3265] functionality.

6.1.6 XML Document Management Client (XDMC)

The XML Document Management Client (XDMC) performs the following functions:

- manages XML documents (e.g., presence authorization rules, presence lists, and URI lists) stored in various XDMS; and

- subscribes to the SIP event package for modifications to XML documents stored in the various XDMS.

The XDMS can be implemented in both UE and application servers (e.g., Presence Server and RLS). It includes XCAP Client and SIP User Agent functionality.

6.1.7 Aggregation Proxy

The Aggregation Proxy is essentially an HTTP proxy that hides the XDMS configuration from the XDMS. The XDMS targets all XDM requests to the Aggregation Proxy, which then routes the request to the appropriate XDMS based on the Application Unique ID (AUID) contained in the request.

6.1.8 Content Server

The Content Server manages presence-related MIME objects. It allows the Presence Source and the Presence Server to store MIME objects, and the Watcher and the Presence Server to retrieve these stored objects as required for content indirection [RFC 4483].

6.2 Presence Reference Points

Each reference point in Figure 3 plays a specific role in the OMA Presence model [OMA AD-PRS], and at the same time can be mapped to its corresponding reference point in the PacketCable architecture [ARCH-FRM-TR], as described below.

6.2.1 PRS-1: Presence Source – SIP Core Network

The PRS-1 reference point supports the communications between the Presence Source and the SIP Core Network. In particular, it enables the Presence Source to push presence status information to the Presence Server via the SIP Core Network, using the SIP PUBLISH mechanism (Section 6.3.2.1).

If the Presence Source resides in a UE, the PRS-1 reference point is mapped to the Gm reference point in the PacketCable architecture; otherwise, if the Presence Source resides in an AS, the PRS-1 reference point is mapped to the ISC reference point in the PacketCable architecture.

6.2.2 PRS-2: Watcher – SIP Core Network

The PRS-2 reference point supports the communications between the Watcher and the SIP Core Network. It enables the Watcher to subscribe to the presence events using the SIP SUBSCRIBE message (Section 6.3.1.1) and receive the corresponding SIP NOTIFY messages (Section 6.3.1.2) from the Presence Server or RLS, via the SIP Core Network.

If the Watcher resides in a UE, the PRS-2 reference point is mapped to the Gm reference point in the PacketCable architecture; otherwise, if the Watcher resides in an AS, the PRS-2 reference point is mapped to the ISC reference point in the PacketCable architecture.

6.2.3 PRS-3: Presence Server – SIP Core Network

The PRS-3 reference point supports the communications between the Presence Server and the SIP Core Network. It enables the SUBSCRIBE and NOTIFY messages between the Presence Server and the Watcher via the SIP Core Network. It also enables the PUBLISH messages from the Presence Source to the Presence Server via the SIP Core Network. In addition, this reference point allows the Presence Server to subscribe to the changes of the XML documents stored on any XDMS.

The PRS-3 reference point is mapped to the ISC reference point in the PacketCable architecture.

6.2.4 PRS-4: Resource List Server – SIP Core Network

The PRS-4 reference point supports the communications between the RLS and the SIP Core Network. It enables the SUBSCRIBE and NOTIFY messages between the RLS and the SIP Core Network in support of presence list subscriptions, and between the RLS and the SIP Core Network in support of the RLS's gathering of presence information concerning any presentity in a presence list.

The PRS-4 reference point is mapped to the ISC reference point in the PacketCable architecture.

6.2.5 PRS-5: Presence Server – Shared XDMS

The PRS-5 reference point supports the communications between the Presence Server and the Shared XDMS. It allows the Presence Server to retrieve URI Lists from the Shared XDMS.

The PRS-5 reference point as defined in the OMA Presence architecture is adopted by the PacketCable Presence architecture.

6.2.6 PRS-6: Presence XDMS – SIP Core Network

The PRS-6 reference point supports the communications between the Presence XDMS and the SIP Core Network. It enables the SUBSCRIBE and NOTIFY messages between the Presence XDMS and the SIP Core Network in support of the subscription to and notification of any modification to the presence-specific XML documents stored in the Presence XDMS.

The PRS-6 reference point is mapped to the ISC reference point in the PacketCable architecture.

6.2.7 PRS-7: Presence XDMS – Aggregation Proxy

The PRS-7 reference point supports the communications between the Presence XDMS and the Aggregation Proxy. It supports the management (e.g., creation, modification, retrieval, deletion) of the XML documents stored in the Presence XDMS.

The PRS-7 reference point as defined in the OMA Presence architecture is adopted by the PacketCable Presence architecture.

6.2.8 PRS-8: Presence Server – Presence XDMS

The PRS-8 reference point supports the communications between the Presence Server and the Presence XDMS. It allows the Presence Server to manage (e.g., create, modify, retrieve, delete) the XML documents stored in the Presence XDMS.

The PRS-8 reference point as defined in the OMA Presence architecture is adopted by the PacketCable Presence architecture.

6.2.9 PRS-9: Resource List Server – Shared XDMS

The PRS-9 reference point supports the communications between the RLS and the Shared XDMS. It allows the RLS to retrieve URI Lists from the Shared XDMS.

The PRS-9 reference point as defined in the OMA Presence architecture is adopted by the PacketCable Presence architecture.

6.2.10 PRS-10: Resource List Server – RLS XDMS

The PRS-10 reference point supports the communications between the RLS and the RLS XDMS. It allows the RLS to manage (e.g., create, modify, retrieve, delete) the XML documents stored in the RLS XDMS.

The PRS-10 reference point as defined in the OMA Presence architecture is adopted by the PacketCable Presence architecture.

6.2.11 PRS-11: RLS XDMS – SIP Core Network

The PRS-11 reference point supports the communications between the RLS XDMS and the SIP Core Network. It enables the SUBSCRIBE and NOTIFY messages between the RLS XDMS and the SIP Core Network in support of the subscription to and notification of any modification to the resource-list-specific XML documents stored in the RLS XDMS.

The PRS-11 reference point is mapped to the ISC reference point in the PacketCable architecture.

6.2.12 PRS-12: RLS XDMS – Aggregation Proxy

The PRS-12 reference point supports the communications between the RLS XDMS and the Aggregation Proxy. It supports the management (e.g., creation, modification, retrieval, deletion) of the XML documents stored in the RLS XDMS.

The PRS-12 reference point as defined in the OMA Presence architecture is adopted by the PacketCable Presence architecture.

6.2.13 PRS-13: Presence Source – Content Server

The PRS-13 reference point supports the communications between the Presence Source and the Content Server. It allows the Presence Source to store MIME objects related to presence publication in the Content Server. It is the Presence Source's responsibility to correlate the presence publication with the MIME objects it has previously stored in the Content Server.

The PRS-13 reference point as defined in the OMA Presence architecture is adopted by the PacketCable Presence architecture.

6.2.14 PRS-14: Watcher – Content Server

The PRS-14 reference point supports the communications between the Watcher and the Content Server. It allows the Watcher to retrieve MIME objects related to presence notification from the Content Server.

The PRS-14 reference point as defined in the OMA Presence architecture is adopted by the PacketCable Presence architecture.

6.2.15 PRS-15: Presence Server – Content Server

The PRS-15 reference point supports the communications between the Presence Server and the Content Server. It allows the Presence Server to store and retrieve MIME objects related to presence publication and notification.

The PRS-15 reference point as defined in the OMA Presence architecture is adopted by the PacketCable Presence architecture.

6.2.16 XDM-1: XDM Client – SIP Core Network

The XDM-1 reference point supports the communications between the XDMC and the SIP Core Network. It enables the SUBSCRIBE and NOTIFY messages between the XDMC and the SIP Core Network in support of the subscription to and notification of any modification to the XML documents stored in the XDMS.

If the XDMC resides in a UE, the XDM-1 reference point is mapped to the Gm reference point in the PacketCable architecture; otherwise, if the XDMC resides in an AS, the XDM-1 reference point is mapped to the ISC reference point in the PacketCable architecture.

6.2.17 XDM-2: Shared XDMS – SIP Core Network

The XDM-2 reference point supports the communications between the Shared XDMS and the SIP Core Network. It enables the SUBSCRIBE and NOTIFY messages between the Shared XDMS and the SIP Core Network in support of the subscription to and notification of any modification to the XML documents stored in the Shared XDMS.

The XDM-2 reference point is mapped to the ISC reference point in the PacketCable architecture.

6.2.18 XDM-3: XDM Client – Aggregation Proxy

The XDM-3 reference point supports the communications between the XDMC and the Aggregation Proxy. It supports the mutual authentication between the XDMC and the Aggregation Proxy, and allows the XDMC to manage (e.g., create, modify, retrieve, delete) the XML documents stored in the XDMS.

With the XDMC residing in a UE, the XDM-3 reference point is mapped to the Ut reference point in the PacketCable architecture.

6.2.19 XDM-4: Shared XDMS – Aggregation Proxy

The XDM-4 reference point supports the communications between the Shared XDMS and the Aggregation Proxy. It supports the management (e.g., create, modify, retrieve, delete) of the XML documents stored in the Shared XDMS.

The XDM-4 reference point as defined in the OMA Presence architecture is adopted by the PacketCable Presence architecture.

6.2.20 IP-1: SIP/IP Core network – Remote Presence Network (based on a SIP/IP Core)

The IP-1 reference point supports the communication between the SIP/IP Core network and a Remote Presence Network based on a SIP/IP Core network.

The IP-1 reference point is mapped to the Mw reference point in the PacketCable architecture.

6.3 Presence Protocols and Data Formats

This section provides an overview of the SIP/SIMPLE presence protocol, and the various data formats used.

6.3.1 Event Notification

A generic framework for SIP-based event notification, using the SUBSCRIBE and NOTIFY methods, is described in [RFC 3265]. The framework needs to be extended by an event package that describes the semantics and behavior associated with a particular event type, such as presence. The event package for presence is described in [RFC 3856], which essentially defines the usage of SIP as a presence protocol.

The general concept is that a watcher subscribes to the presence state of a presentity, and then receives notifications about the current presence state as well as updates about state changes.

6.3.1.1 SIP SUBSCRIBE method

A watcher requests the presence state of a presentity by sending a SUBSCRIBE message and specifying the 'presence' token in the Event header. The Request-URI of the message is used for routing to the appropriate Presence Server, as well as for identifying the presentity whose presence state is desired. The watcher can specify the desired duration of the subscription using the Expires header.

The Presence Server must authorize the subscription request (Section 6.3.5.1). If the subscription request is successful, the watcher receives a 200-class response that includes the actual duration of the subscription in the Expires header.

A 200-class response is always followed by an immediate notification (Section 6.3.1.2). Therefore, a SUBSCRIBE with an Expires value of 0 constitutes a "fetch" of current presence state, while an Expires value greater than 0 constitutes both a request for current presence state and future updates.

Because subscriptions expire, they must be refreshed by new SUBSCRIBE requests from the watcher, which can be sent at any time before the subscription expires. To unsubscribe or terminate the subscription, the watcher sends a SUBSCRIBE refresh with an Expires value of 0.

The body of the SUBSCRIBE message can either be empty, or can include a filter to request that notifications be triggered only by certain presence events and/or contain only certain data (Section 6.3.6.1).

6.3.1.2 SIP NOTIFY method

The Presence Server communicates the current presence information and subscription state to the watcher by sending a NOTIFY message. The NOTIFY conveys the following information:

- Subscription state: The subscription state is indicated by the Subscription-State header, and has one of three possible values:
 - Active: indicates that the subscription has been accepted and authorized.
 - Pending: indicates that the subscription has been accepted, but may or may not have been authorized.
 - Terminated: indicates that the subscription is not active.
- Presence state: The current state of presence information is conveyed by a presence document (Section 6.3.3) carried in the NOTIFY body. The presence document contains whatever presence information the watcher is authorized to receive according to the presentity's content policy (Section 6.3.5.2).

For Active or Pending subscriptions, a NOTIFY can be sent at any time, but is typically sent whenever subscribed information changes or the subscription state changes. However, the exact timing of NOTIFY messages may be impacted by local policy (e.g., throttling).

6.3.2 Publishing Presence Information

A general framework for SIP-based publication of event state, using the PUBLISH method, is described in [RFC 3903]. The RFC also describes the specific usage of the framework for publication of presence state.

The concept is that a presence source publishes presence state to the network for storage and distribution. The presence source is then responsible for modifying the published presence state (i.e., when there are changes), and for periodically refreshing the presence state.

6.3.2.1 SIP PUBLISH method

A Presence Source publishes some or all of the presence state of a presentity by sending a PUBLISH message that carries a presence document in the message body (Section 6.3.3) and specifies the 'presence' token in the Event

header. The addressing of a PUBLISH is identical to that of a SUBSCRIBE, i.e., the Request-URI of the message is used for routing to the appropriate Presence Server, as well as for identifying the presentity whose presence state is being published. The Presence Source can specify a suggested lifetime of the published presence state using the Expires header.

For each successful PUBLISH, the Presence Source receives a 200-class response, which must include the actual lifetime of the publication in the Expires header. Because publications expire, they must be refreshed by new PUBLISH requests from the Presence Source.

A Presence Source can send PUBLISH requests at any time (e.g., whenever presence state changes), subject to throttling constraints that may be locally configured or provisioned by the service provider.

6.3.3 Presence Information Data Format (PIDF)

For the presence event package, the base data format of a presence information document is defined in [RFC 3863], and is called the Presence Information Data Format (PIDF). The PIDF defines the minimal set of information in a presence document and provides an extensibility framework for defining additional information.

The presence document carries presence tuples, which are the basic units of presence information. PIDF defines a basic set of presence information elements and specifies an extensibility framework based on XML namespaces. This allows applications to define their own presence elements in order to provide richer and more meaningful information about a presentity, while avoiding element naming conflicts by associating the extensions with a globally unique namespace URI. There are several IETF specifications that define PIDF extensions (e.g., [RFC 4480] and [RFC 4119]).

Below is a sampling of presence information elements that are added via PIDF extensions:

- <activity> describes the current activity of the presentity, such as "on-the-phone," "holiday," "steering," or "sleeping.'
- <contact-type> describes the type of the tuple, such as "device," "service," or "presentity.'
- <placetype> describes the type of place the presentity is located, such as "home," "office," or "mall.'
- <privacy> indicates whether third parties are likely to see or hear communications.
- <icon> provides a URI pointing to an image that can be used to represent the tuple or presentity on a graphical user interface.
- <location-info> and <method> provide location information and the method used to obtain it (i.e., "GPS").

6.3.4 Watcher Information

In the context of a presence service, watcher information refers to the dynamically changing set of watchers that are subscribed to a particular presentity, and the state of the subscriptions.

A watcher information event template-package is defined in [RFC 3857]. An event template-package is a special type of event package that is always associated with some other event package, such as presence.

A user requests the watcher information of a particular presentity (usually himself) by sending a SUBSCRIBE message, and specifying the 'presence.winfo' token in the Event header. The NOTIFY body then contains a watcher information document.

The primary motivation for subscribing to watcher information is to support Reactive Authorization (Section 6.3.5.1). For example, a user can subscribe to his own watcher information so that he is alerted when an authorization decision is required.

6.3.5 Presence Authorization Rules

Presence authorization rules control how the Presence Server is to disseminate a presentity's presence information. There are at least two parts to the presence authorization rules:

- Subscription authorization rules
- Presence content rules

The presence authorization rules can be defined by the presentity, and are represented by an XML document stored in the Presence XDMS. The structure of the XML document is specified by [ID PRRULES] and extended by [OMA TS-PRSXDM]. Each rule in a presence authorization document expresses a set of conditions which, if met, result in a set of permissions being granted.

It should be noted that a presentity's policies can be overridden by local administrator, service provider, and/or implementation policies. Also, any watcher preferences (e.g., notification filtering) are applied after all other relevant policies.

6.3.5.1 Subscription Authorization Rules

Subscription authorization rules are applied by the Presence Server to a subscription request once the subscriber has been authenticated. It determines which watchers are authorized to receive the presentity's presence information and which are not.

There are two types of subscription authorization:

- Proactive: describes a policy where the presentity has preconfigured which watchers are authorized to receive its presence information so that the authorization decision can be performed automatically by the Presence Server.
- Reactive: describes a policy where the Presence Server interactively queries the presentity for authorization of a subscription request.

Both types of authorization can be supported by allowing the presentity to conceptually configure the following rules:

- Allow: Identifies the watchers for whom authorization is automatically successful. This results in a 200 OK response.
- Block: Identifies the watchers for whom authorization automatically fails, and is typically the default. This results in a 403 Forbidden response.
- Confirm: Identifies the watchers for whom authorization is pending user input. This results in a 202 Accepted response. In this case, the user has presumably subscribed to watcher information (Section 6.3.4) and will be informed about the subscription request.

6.3.5.2 Presence Content Rules

Presence content rules determine what presence information an authorized watcher is allowed to receive, or in other words, the content of the NOTIFY body sent to the watcher. The policy defines a set of rules for how presence information elements should be transformed for particular watchers.

An example of content policy is polite blocking, which is a method of rejecting a subscription request without revealing to the watcher that its request was rejected. This is accomplished by generating a 200 OK response authorizing the subscription request, but transforming the contents of the presence document sent to the watcher, such that it does not reveal the presentities "true" presence information (for example, it always indicates that the presentity is unavailable).

6.3.6 Performance Enhancements

As stated above, the Presence Server may send a NOTIFY at any time, and the NOTIFY includes the complete state that the watcher is authorized to receive. In order to conserve bandwidth and minimize the impact to mobile battery life, it is desirable to limit the size and frequency of NOTIFY messages sent to a watcher.

6.3.6.1 Presence Lists

A presence list is a list of presentities that can have their individual states subscribed to with a single SUBSCRIBE request. An event notification extension for subscriptions to presence lists is defined in [RFC 4662]. The watcher subscribes to a presence list in the same way it subscribes to an individual presentity (Section 6.3.1.1). The RLS accepts subscriptions to presence lists and acts as notifier for the list.

The resource list mechanism reduces the presence messaging traffic for subscribing to multiple presentities, since:

- Only one initial SUBSCRIBE message is needed for the entire list.
- Only one SUBSCRIBE message is needed to refresh the subscription.
- Notifications can be aggregated by the RLS, reducing the number of NOTIFY messages.

6.3.6.2 Event Notification Filtering

Event notification filtering, described in [RFC 4660], is a mechanism for the watcher to control the content and frequency of NOTIFY messages sent to it.

The watcher can describe its preferences for NOTIFY content and triggering by creating a filter and including it in the body of the SUBSCRIBE request.

It should be noted that the filter expresses the watcher's preference only. It is up to the Presence Server to determine whether to honor the filter.

6.3.6.3 Partial Notifications

Partial notifications, described in [ID PARTNOT], is a mechanism for sending only those parts of the presence information that have changed since the last notification sent to the watcher rather than the full presence state.

When the Presence Server receives the subscription request and determines to use partial notifications, then the first NOTIFY message contains a full presence document, but subsequent NOTIFY messages contain a presence document that includes only the subset of presence tuples that have changed. When the watcher receives a partial notification, it updates its locally cached copy of the full presence document accordingly.

6.3.6.4 Content Indirection

Content indirection, described in [RFC 4483], is useful for handling of large MIME objects. Rather than including the MIME object directly in the body of a SIP request (e.g., PUBLISH or NOTIFY), it can be referred to indirectly using a URI.

In the OMA Presence architecture, the MIME object can be stored in the Content Server.

6.4 XDM Protocols

XDM requires a common protocol for accessing and manipulating the XML documents stored in the various XDMSs. The protocol defined for this purpose is the XML Configuration Access Protocol (XCAP), which is described in [ID XCAP].

XCAP maps XML document sub-trees and element attributes to HTTP URIs, so that these components can be directly accessed by HTTP. This mapping allows an XCAP client to use HTTP commands to directly manipulate elements and attributes within an XML document.

Enablers (such as Presence) and applications that make use of XCAP define an application usage, which defines what the enabler or application needs to do in order to be used with XCAP. This includes defining: an application unique ID (AUID), an XML schema and semantics for the data, naming conventions, data interdependencies (for server computed data), and authorization policies.

The use of XCAP within the PacketCable Presence architecture is specified in [OMA TS-PRSXDM] and is based on the specification of the generic XDM framework provided in [OMA TS-XDM]. The requirements from both these OMA specifications are adopted by the PacketCable Presence architecture with a few minor exceptions noted elsewhere in this document.

The XDMC and XDMS in the PacketCable Presence architecture contain all the functionality of an XCAP Client and XCAP Server, respectively.

6.5 Client Provisioning and Management

OMA defines the required parameters that are subject to real-time provisioning that are needed by the OMA Presence service. These parameters are represented by OMA using an OMA DM Management Object (MO). The OMA Presence architecture notes that OMA DM can be used to configure Presence entities. However, there is no interface defined between an OMA DM client and a Presence entity; the two can be viewed as independent. Reference the use of DMS (Device Management Server) and DMC (Device Management Client) shown in Figure 1.

PacketCable Presence makes use of PACM [PRS PACM] to provision and provide ongoing management of Presence entities residing within a UE. As already noted, as the use of OMA Presence is independent architecturally from OMA DM then the use of PACM to provide configuration and provisioning details for PacketCable Presence should be feasible, provided that the requirements for provisioned elements defined within [OMA RD-PRS] are met by the PACM data structures. This requires that the OMA defined requirements and OMA DM representation of these requirements are analyzed and reduced to a data model that allows subsequent modeling using XSD format.

PACM requires the use of two additional interfaces at the UE in addition to those illustrated in Figure 3: pkt-pacm-3 and pkt-pacm-4. These are as defined within [PRS PACM] wherein pkt-pacm-3 is realized by the SIP SUBSCRIBE-NOTIFY model and pkt-pacm-4 by XCAP.

Details with regard to the representation of the OMA identified provisioned items and how they are maintained are within the PacketCable Presence PACM Specification [PRS PACM].

6.6 Accounting

Accounting of Presence services uses the Event Based charging model described in [ACCT], which is based on the 3GPP IMS charging specifications. Charging Events related to presence can be generated by PacketCable Core Network elements (CSCFs) and/or by the Presence Application Server. Presence based Charging Events are identified by the SIP-Method and Event DIAMETER AVPs, which identify the SIP Method associated with the Charging Event (e.g., PUBLISH) and the event package associated with the SIP Method (e.g., Message Waiting Indication) respectively. These Charging Events are forwarded to the Offline Charging entity (CDF/CGF). The events which may trigger Presence charging are described in [TS 23.141].

6.7 Security

Procedures for XDMC authentication over the XDM-3 reference point are defined in [OMA TS-XDM]. The Aggregation Proxy (AP) authenticates XCAP requests received from an XDMC, and performs identity assertion prior to forwarding to the appropriate XDMS. The Aggregation Proxy behaves as an Authentication Proxy as

described in [TS 33.141]. [TS 33.141] allows for two methods of authentication. The default method is HTTP Digest, as defined in [RFC 2617]. [TS 33.141] also allows the use of the Generic Authentication Architecture (GAA) [TS 33.222]. GAA makes use of bootstrapped credentials using the Generic Bootstrapping Architecture (GBA) [PKT 33.220]. Bootstrapped credentials are obtained from a GBA run between the UE and the Bootstrapping Server Function (BSF). Reference points and components of GAA can be found in [TS 33.222]. The Presence Server, RLS, and various XDMS authenticate SIP requests.

TLS provides confidentiality and integrity for the XDM-3 reference point between the XDMC and the AP. Communications between the AP and the Application Servers on the Zb reference point can be protected using TLS or IPsec, as described in [PKT 33.210]. Procedures for securing the Gm reference point between the UE and the P-CSCF are described in [PKT 33.203].

6.8 Codec

The PacketCable Codec Specification [CODEC-MEDIA] describes audio and video codecs that are optional for use by PacketCable applications. The PacketCable Presence specification does not call out the support of codecs. However, the Presence-enabled applications should refer to [CODEC-MEDIA] to define their individual audio and video codec requirements.

7 PACKETCABLE PRESENCE REQUIREMENTS

The following exceptions apply to all the OMA specifications that are adopted by PacketCable from the Presence Enabler and the XDM Enabler in the following sub-sections.

- 3GPP2 specifications and references are not applicable.
- The following 3GPP Presence specifications and references are out of scope for the current version of the PacketCable Presence specification: [TS 22.141], [TS 23.141], and [TS 24.141].
- PacketCable IMS delta specifications supersede the corresponding 3GPP specifications where applicable. This includes nested references.
- Where there is mention of a SIP/IP core network corresponding with 3GPP IMS, 3GPP IMS is replaced by PacketCable.
- OMA Device Management enabler specifications and references are superseded by [PRS PACM].
- Appendices listing Static Conformance Requirements do not create any new normative requirements beyond what is present in the specification text, so they are considered informative for PacketCable.
- The exceptions noted in this section and the sub-sections also apply to nested references.

7.1 OMA Presence Enabler

The sub-sections adopt the individual specifications of the OMA Presence Enabler, indicating the applicable exceptions.

7.1.1 OMA Presence SIMPLE Specification

The normative requirements specified in [OMA TS-PRS] MUST be met, with the exceptions noted here and in Section 7:

- Normative requirements that map OMA Presence reference points to 3GPP Presence reference points (Pep, Pex, Pen, Pw and Pwp) are out of scope for the current version of the PacketCable Presence specification. These reference points are used by 3GPP Presence for the sole purpose of differentiating the context of the presence source and/or the watcher.
- Appendix B is superseded by [PRS PACM].
- Where use is made of 'OTA Provisioning' or 'OTA Provisioning or local configuration,' it is replaced by PacketCable Presence PACM [PRS PACM]. Subsequently defined use of OMA DM based configuration elements is replaced according to Table 2.

Table 2 – OMA to PacketCable Presence PACM Configuration Element Mapping

OMA Defined Configuration Element	PacketCable Presence PACM Defined Configuration Element
CONTENT-SERVER-URI	pktcPRSContentServerURI
CLIENT-OBJ-DATA-LIMIT	pktcPRSClientObjDataLimit
SOURCE-THROTTLE-PUBLISH	pktcPRSSourceThrottlePublish
MAX-NUMBER-OF-PRESENCE-SUBSCRIPTIONS	pktcPRSMaxNumberOfPresenceSubscriptions
MAX-NUMBER-OF-SUBSCRIPTIONS-IN-PRESENCE-LIST	pktcPRSMaxNumberOfSubscriptionsInPresenceList

7.1.2 OMA Presence SIMPLE XDM Specification

The normative requirements specified in [OMA TS-PRSXDM] MUST be met, with the exceptions noted in Section 7.

7.1.3 OMA Presence SIMPLE RLS XDM Specification

The normative requirements specified in [OMA TS-RLSXDM] MUST be met, with the exceptions noted here and in Section 7:

- Reference to [OMA PRES-MO] is superseded by [PRS PACM].

7.2 OMA XDM Enabler

The sub-sections adopt the individual specifications of the OMA XDM Enabler, indicating the applicable exceptions.

7.2.1 OMA XDM Core Specification

The normative requirements specified in [OMA TS-XDM] MUST be met, with the exceptions noted here and in Section 7:

- Appendix C is superseded by [PRS PACM].
- Reference to [OMA XDM-MO] is superseded by [PRS PACM].

7.2.2 OMA XDM Shared Specification

The normative requirements specified in [OMA TS-SHDXDM] MUST be met, with the exceptions noted in Section 7.

7.3 Provisioning and Management

The OMA specifications [OMA PRES-MO] and [OMA XDM-MO] define the OMA DM management objects for the Presence and XDM enablers. For PacketCable, these are superseded by [PRS PACM].

Appendix I Informative Call Flow

This appendix contains an informative call flow illustrating a presence subscription request by a watcher and subsequent publishing of presence information by a presentity.

Part 1: The presentity (UE2) subscribes to watcher information, to be notified of presence subscription requests for the presentity. The Presence Server authorizes the subscription (based on policy that a presentity is allowed to subscribe to its own watcher information) and sends a notification with current subscription information.

Part 2: At some later point the watcher (UE1) subscribes to presence for a presentity (UE2)

- The Presence Server obtains the presence authorization rules for the presentity, from the Presence XDMS. In this example, the rules indicate that the watcher's subscription request is to be confirmed with the presentity, and local policy for confirmation is "reactive authorization". The Presence Server sends a notification to UE2 indicating that a presence subscription has been attempted by the watcher. The subscription is placed into a "pending" state.
- UE2 receives the notification that there is a pending subscription and authorizes the request, and in this example the subscription is allowed. UE2 updates the authorization rules on the Presence XDMS (via the Aggregation Proxy) to indicate that the subscription is allowed from the watcher. (Details of authentication of UE2 for this operation are not shown in this example).
- The Presence Server receives a notification of the change in the presence authorization rules. (In this example, the Presence Server is assumed to have previously subscribed to changes in the presence authorization rules). The "pending" subscription is placed into an "accepted" state.

Part 3: At some later point the presentity (UE2) publishes presence information.

- The Presence Server sends a notification to the watcher (UE1).

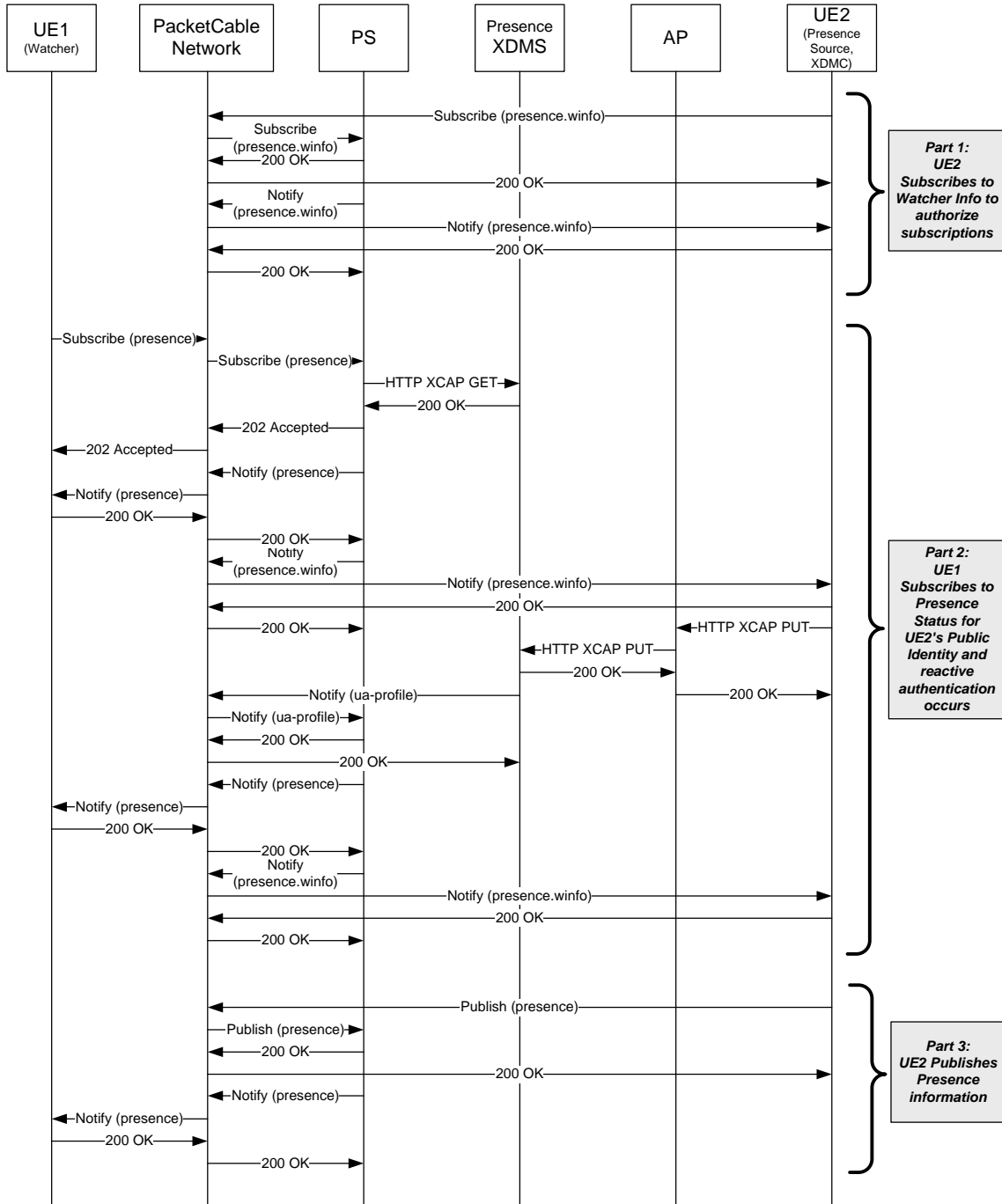


Figure 4 - Informative call flow illustrating a presence subscription

Appendix II Acknowledgements

CableLabs wishes to thank the following PacketCable focus team participants for authoring various sections that led to the development of this specification (alphabetical by company name):

Moh Torabi (Alcatel-Lucent)

Gordon Li (Broadcom)

Sean Schneyer (Ericsson)

Sean Kelley (Motorola)

Brian Lindsay (Nortel)

Mark Trayer (Samsung)

Charles (Chuck) Chaney (Siemens)

Special thanks are extended to Sean Kelley (Motorola) for being the primary author of this specification and other reviewers who provided valuable feedback.

Sandeep Sharma and the PacketCable Architecture Team, CableLabs.
