



Converged IP Messaging Architecture

Approved Version 1.0 – 12 Jun 2012

Open Mobile Alliance
OMA-AD-CPM-V1_0-20120612-A

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2012 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

1. SCOPE (INFORMATIVE)	6
2. REFERENCES	7
2.1 NORMATIVE REFERENCES	7
2.2 INFORMATIVE REFERENCES	7
3. TERMINOLOGY AND CONVENTIONS	9
3.1 CONVENTIONS	9
3.2 DEFINITIONS	9
3.3 ABBREVIATIONS	11
4. INTRODUCTION (INFORMATIVE)	13
4.1 VERSION 1.0	13
4.2 FUTURE VERSIONS	14
4.3 SECURITY CONSIDERATIONS	15
4.3.1 SIP Signalling Security	15
4.3.2 Media Plane Security	15
4.3.3 CPM Interworking Function Security	15
4.3.4 Message Storage Security	16
5. ARCHITECTURAL MODEL	17
5.1 DEPENDENCIES	17
5.2 ARCHITECTURAL DIAGRAM	18
5.3 FUNCTIONAL COMPONENTS AND INTERFACES DEFINITION	20
5.3.1 CPM Functional Components	20
5.3.1.1 CPM Client	20
5.3.1.2 Message Storage Client	22
5.3.1.3 CPM Participating Function	23
5.3.1.4 CPM Controlling Function	25
5.3.1.5 Interworking Selection Function	26
5.3.1.6 Interworking Function	27
5.3.1.7 Message Storage Server	28
5.3.2 External Functional Components	30
5.3.2.1 SIP/IP Core	30
5.3.2.2 Presence Enabler	30
5.3.2.3 XML Document Management Enabler	31
5.3.2.4 Device Management Enabler	31
5.3.2.5 Push Enabler	31
5.3.2.6 Non-CPM Communication Services	31
5.3.2.7 Remote CPM Environment	31
5.3.3 CPM Interfaces	32
5.3.3.1 CPM-PF1	32
5.3.3.2 CPM-PF2	32
5.3.3.3 CPM-CF	33
5.3.3.4 CPM-IW1	33
5.3.3.5 CPM-IW2	34
5.3.3.6 CPM-MSG	34
5.4 FLOWS	35
APPENDIX A. CHANGE HISTORY (INFORMATIVE)	36
A.1 APPROVED VERSION 1.0 HISTORY	36
APPENDIX B. DEPLOYMENT SCENARIOS (INFORMATIVE)	37
B.1 DEPLOYMENT CONFIGURATIONS	37
B.1.1 Deployment without Network Storage	37
B.1.2 Deployment for Basic Communications Service	38
B.2 CO-LOCATION DEPLOYMENT SCENARIOS	38
B.2.1 Interworking Selection Function	38
B.2.1.1 ISF Co-located with the CPM Participating Function	39

B.2.1.2 ISF Co-located with the Interworking Function 39

B.2.1.3 ISF as a Separate Physical Component 40

B.2.2 CPM Controlling Function and CPM Participating Function 40

B.2.2.1 CPM Controlling Function and CPM Participating Function deployed separately 40

B.2.2.2 CPM Controlling Function and CPM Participating Function deployed co-located 41

APPENDIX C. ARCHITECTURE FOR FUTURE CPM FUNCTIONALITY (INFORMATIVE)..... 42

C.1 ADDITIONAL SECURITY CONSIDERATIONS 42

C.1.1 Content Storage Security 42

C.2 ADDITIONAL DEPENDENCIES..... 42

C.3 EVOLVED ARCHITECTURE DIAGRAM 43

C.4 ADDITIONS TO EXISTING CPM ENABLER FUNCTIONAL COMPONENTS 44

C.4.1 CPM Client 44

C.4.2 CPM Participating Function..... 45

C.4.3 CPM Controlling Function..... 46

C.4.4 Message Storage Server 46

C.5 ADDITIONAL CPM ENABLER FUNCTIONAL COMPONENTS..... 46

C.5.1.1 Content Storage Client..... 46

C.5.1.2 Content Storage Server 47

C.5.1.3 Application Control Function 49

C.6 ADDITIONS TO EXISTING EXTERNAL FUNCTIONAL COMPONENTS..... 49

C.6.1 Presence Enabler 49

C.7 ADDITIONAL EXTERNAL FUNCTIONAL COMPONENTS 49

C.7.1 Charging Enabler 49

C.7.2 Converged Address Book Enabler 50

C.7.3 VAS Applications 50

C.8 ADDITIONS TO EXISTING INTERFACES..... 50

C.8.1 CPM-MSG 50

C.9 ADDITIONAL INTERFACES 50

C.9.1 CPM-CNT..... 50

C.9.2 CPM-VAS..... 51

C.10 ADDITIONAL DEPLOYMENT SCENARIOS 51

C.10.1 Additional Deployment Configuration 51

C.10.1.1 Deployment without Content Storage 51

C.10.2 Additional Co-Location Deployment Scenarios 52

C.10.2.1 Application Control Function..... 52

C.7.2.1.1 Application Control Function as a Separate Physical Component..... 52

C.7.2.1.2 Application Control Function Co-located with the CPM Participating Function 53

Figures

Figure 1: Architecture Diagram Conventions 9

Figure 2: Schematic to illustrate the modularity and extensibility of the CPM Enabler. 13

Figure 3: CPM architecture diagram. 18

Figure 4: Interactions between the CPM Enabler functional components and the supporting Enablers. 20

Figure 5: Deployment configuration without network storage and VAS Applications 37

Figure 6: Deployment configuration for a basic communications service 38

Figure 7: ISF co-located with the CPM Participating Function. 39

Figure 8: ISF co-located with the Interworking Function..... 39

Figure 9: ISF as a separate physical component. 40

Figure 10: CPM Participating Function and CPM Controlling Function deployed separately. 41

Figure 11: CPM Participating Function Co-located with the CPM Controlling Function.	41
Figure 12: A possible future CPM architecture diagram.....	43
Figure 13: Interactions between the CPM Enabler functional components and the supporting Enablers.	44
Figure 14: Deployment configuration without Content Storage.....	52
Figure 15: Application Control Function deployed as a Separate Physical Component.....	53
Figure 16: Application Control Function Co-located with the CPM Participating Function.....	53

1. Scope

(Informative)

This document describes the architecture designed to fulfil all the requirements outlined in the Converged IP Messaging Requirements document [OMA-CPM-RD] for CPM V1.0 and for future CPM versions; the major differences between CPM V1.0 and the future CPM versions are described in section 4.1, section 4.2 and Appendix C. The description of the architecture comprises the definition of functional components and the interfaces used or exposed by these functional components.

Additionally, this document identifies the relationship of the CPM Enabler to external functional components that either are utilized by the CPM Enabler or take advantage of the CPM Enabler. The description of these relationships comprises the functional components and interfaces that reside at the boundary between the CPM Enabler and the external technology domain.

2. References

2.1 Normative References

- [3GPP TS23.228] “IP Multimedia Subsystem (IMS); Stage 2”, 3GPP, TS23.228, URL:<http://www.3gpp.org/>
- [3GPP2 X.S0013-002] “All-IP Core Network Multimedia Domain - IP Multimedia Subsystem (IMS) - Stage-2”, 3GPP2, X.S0013-002-A, URL:<http://www.3gpp2.org/>
- [OMA-CPM-RD] “Converged IP Messaging Requirements”, Open Mobile Alliance™, OMA-RD-CPM-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-CPM-SD] “Converged IP Messaging System Description”, Open Mobile Alliance™, OMA-TS-CPM_System_Description-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-DM-PRO] “Device Management Protocol”, Open Mobile Alliance™, OMA-TS-DM_Protocol-V1_2, URL:<http://www.openmobilealliance.org/>
- [OMA-IMS-AD] “Utilization of IMS capabilities Architecture”, Open Mobile Alliance™, OMA-AD-IMS-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-PRS-AD] “Presence SIMPLE Architecture”, Open Mobile Alliance™, OMA-AD-Presence_SIMPLE-V2_0, URL:<http://www.openmobilealliance.org/>
- [OMA-PRS-RD] “Presence SIMPLE Requirements”, Open Mobile Alliance™, OMA-RD-Presence_SIMPLE-V2_0, URL:<http://www.openmobilealliance.org/>
- [OMA-PUSH-AD] “Push Architecture”, Open Mobile Alliance™, OMA-AD-Push-V2_2, URL:<http://www.openmobilealliance.org/>
- [OMA-XDM-AD] “XML Document Management Architecture”, Open Mobile Alliance™, OMA-AD-XDM-V2_1, URL:<http://www.openmobilealliance.org/>
- [OSE] “OMA Service Environment”, Open Mobile Alliance™, OMA-AD-Service_Environment-V1_0, URL:<http://www.openmobilealliance.org/>
- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL:<http://www.ietf.org/rfc/rfc2119.txt>
- [RFC3261] “SIP: Session Initiation Protocol”, J. Rosenberg et al, June 2002, URL:<http://www.ietf.org/rfc/rfc3261.txt>
- [RFC3325] “Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks”, C. Jennings et al, November 2002, URL:<http://www.ietf.org/rfc/rfc3325.txt>
- [RFC3428] “Session Initiation Protocol (SIP) Extension for Instant Messaging”, B. Campbell et al, December 2002, URL:<http://www.ietf.org/rfc/rfc3428.txt>
- [RFC3501] “INTERNET MESSAGE ACCESS PROTOCOL (IMAP4) - VERSION 4rev1”, M. Crispin, March 2003. URL:<http://www.ietf.org/rfc/rfc3501.txt>
- [RFC3550] “RTP: A Transport Protocol for Real-Time Applications”, H. Schulzrinne, July 2003, URL:<http://www.ietf.org/rfc/rfc3550.txt>
- [RFC4566] “SDP: Session Description Protocol”, M. Handley et al, July 2006, URL:<http://www.ietf.org/rfc/rfc4566.txt>
- [RFC4975] “The Message Session Relay Protocol”, B. Campbell et al, September 2007, URL:<http://www.ietf.org/rfc/rfc4975.txt>

2.2 Informative References

- [3GPP TS33.203] “3G security; Access security for IP-based services”, 3GPP, TS33.203, URL:<http://www.3gpp.org/>
- [3GPP TS33.210] “3G security; Network Domain Security (NDS); IP network layer security”, 3GPP, TS33.210, URL:<http://www.3gpp.org/>
- [3GPP2 S.R0086-0] “IMS Security Framework”, 3GPP2, S.R0086-0, URL:<http://www.3gpp2.org/>
- [OMA-CAB-AD] “Converged Address Book Architecture”, Open Mobile Alliance™, OMA-AD-CAB-V1_0, URL:<http://www.openmobilealliance.org/>

[OMA-CHRG-AD]	“Charging Architecture”, Open Mobile Alliance™, OMA-AD-Charging-V1_0, URL: http://www.openmobilealliance.org/
[OMA-DICT]	“Dictionary for OMA Specifications”, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_7, URL: http://www.openmobilealliance.org/
[OMA-DS]	“DS 1.2 Architecture”, Open Mobile Alliance™, OMA-ERP-DS-V1_2, URL: http://www.openmobilealliance.org/
[OMA-OWSER]	“OMA Web Services”, Open Mobile Alliance™, OMA-ERELED-OWSER-V1_1, URL: http://www.openmobilealliance.org/
[OMA-POC]	“Push to talk Over Cellular”, Open Mobile Alliance™, OMA-ERELED-POC-V2_0, URL: http://www.openmobilealliance.org/
[OMA-SEC_CF]	“Application Layer Security Common Functions”, Open Mobile Alliance™, OMA-ERELED-SEC_CF-V1_0, URL: http://www.openmobilealliance.org/
[OMA-SIMPLE-IM]	“Instant Messaging”, Open Mobile Alliance™, OMA-ERELED-SIMPLE_IM-V1_0, URL: http://www.openmobilealliance.org/
[RFC2617]	“HTTP Authentication: Basic and Digest Access Authentication”, J. Franks et al, June 1999, URL: http://www.ietf.org/rfc/rfc2617.txt
[RFC4918]	“HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)”, L. Dusseault, Ed., June 2007, URL: http://www.ietf.org/rfc/rfc4918.txt

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.

All architecture diagrams in this document adhere to the conventions depicted in Figure 1.

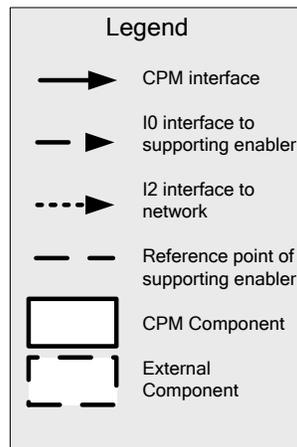


Figure 1: Architecture Diagram Conventions.

3.2 Definitions

Authorized Principal	See [OMA-DICT].
Charging Event	See [OMA-DICT].
Communication Capabilities	See [OMA-CPM-RD].
Content Screening	See [OMA-CPM-RD].
Content Storage Client	A functional component of the CPM architecture, as defined in this document.
Content Storage Server	A functional component of the CPM architecture, as defined in this document.
CPM-based Service	A Service that uses the functionalities of the framework provided by the CPM Enabler to fulfil its communication needs.
CPM Ad-hoc Group	See [OMA-CPM-RD].
CPM Address	See [OMA-CPM-RD].
CPM Chat Message	See [OMA-CPM-RD].
CPM Client	A functional component of the CPM architecture, as defined in this document.
CPM Controlling Function	A functional component of the CPM architecture, as defined in this document.
CPM Conversation	See [OMA-CPM-RD].
CPM Conversation History	See [OMA-CPM-RD].

CPM File Transfer	See [OMA-CPM-RD].
CPM File Transfer History	See [OMA-CPM-RD].
CPM Group	See [OMA-CPM-RD].
CPM Group Session	See [OMA-CPM-RD].
CPM Message	See [OMA-CPM-RD].
CPM Participating Function	A functional component of the CPM architecture, as defined in this document.
CPM Pre-defined Group	See [OMA-CPM-RD].
CPM Session	See [OMA-CPM-RD].
CPM Session History	See [OMA-CPM-RD].
CPM Session Invitation	See [OMA-CPM-RD].
CPM Standalone Message	See [OMA-CPM-RD].
CPM User	User of a CPM-based Service.
CPM User Preferences	A functional component of the CPM architecture, as defined in this document.
Deferred Delivery Messaging	See [OMA-DICT].
Device	See [OMA-DICT].
Enabler	See [OMA-DICT].
Internet Message Access Protocol 4	See [RFC3501].
Immediate Messaging	See [OMA-DICT].
Interworking Function	A functional component of the CPM architecture, as defined in this document.
Interworking Selection Function	A functional component of the CPM architecture, as defined in this document.
Large Message Mode	A method to send a single large CPM Message using MSRP.
Malware	See [OMA-CPM-RD].
Media	See [OMA-CPM-RD].
Media Type	See [OMA-DICT].
Media Object	A piece of discrete Media that can be referenced by a URL, or included directly in a CPM Message.
Media Plane	The Media and Media control signalling (e.g. floor control) between CPM functional components. Media Plane is also known as “User Plane” in other specifications.
Media Stream	See [RFC3264].
Media Stream Type	The Media Stream Type corresponds to the <media> parameter of a Media description, referred to as “media types” in [RFC4566].
Message Storage Client	A functional component of the CPM architecture, as defined in this document.
Message Storage Server	A functional component of the CPM architecture, as defined in this document.
Message Session Relay Protocol	See [RFC4975].
Non-CPM Communication Service	See [OMA-CPM-RD].
Offline Charging	See [OMA-DICT].
OMA Data	See [OMA-DS].

Synchronisation

Online Charging	See [OMA-DICT].
Pager Mode	A method to send a single CPM Message of limited size without requiring the establishment of an MSRP session.
Participant	See [OMA-DICT].
Presence Information	See [OMA-PRS-RD].
Principal	See [OMA-DICT].
Pseudonym	See [OMA-CPM-RD].
Pull	See [OMA-DICT].
Push	See [OMA-DICT].
Real-Time Transport Protocol	See [RFC3550].
RTP Control Protocol	See [RFC3550].
Service	See [OMA-DICT].
Session Description Protocol	See [RFC4566].
Session Initiation Protocol	See [RFC3261].
User	See [OMA-DICT].
User Preferences Profile	See [OMA-CPM-RD].
Value Added Service	See [OMA-DICT].
VAS Application	An external entity offering a VAS which uses the functionalities of the framework provided by the CPM Enabler as part of its service logic.
Web Distributed Authoring and Versioning	See [RFC4918].

3.3 Abbreviations

3GPP	3 rd Generation Partnership Project
3GPP2	Third Generation Partnership Project 2
CAB	Converged Address Book
CPM	Converged IP Messaging
DM	Device Management
HTTP	Hyper Text Transfer Protocol
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMAP	Internet Message Access Protocol
IMPS	Instant Messaging and Presence Service
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IP-SM-GW	IP-Short-Message-Gateway
ISF	Interworking Selection Function
ISIM	IP Multimedia Services Identity Module

MGCF	Media Gateway Control Function
MMD	3GPP2 Multimedia Domain
MMS	Multimedia Messaging Service
MSRP	Message Session Relay Protocol
OMA	Open Mobile Alliance
OSE	OMA Service Environment
PoC	Push to talk over Cellular
RFC	Request For Comments
RTCP	RTP Control Protocol
RTP	Real-Time Transport Protocol
R-UIM	Removable – User Identity Module
SDP	Session Description Protocol
SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SMS	Short Message Service
UIM	User Identity Module
URL	Uniform Resource Locator
USIM	Universal Subscriber Identity Module
VAS	Value Added Service
WebDAV	Web Distributed Authoring and Versioning
XCAP	XML Configuration Access Protocol
XDM	XML Document Management
XML	Extensible Markup Language

4. Introduction

(Informative)

The Converged IP Messaging (CPM) Enabler provides the convergence of multi-media communication services while leveraging standardized service functionalities from existing communication enablers like Instant Messaging [OMA-SIMPLE-IM] or Push to talk over Cellular [OMA-POC].

The CPM Enabler provides a framework by defining a horizontal Enabler built on top of a SIP/IP core infrastructure. This framework comprises a set of functional components and interfaces that have been designed to facilitate easy deployment of existing and future communication services. The components of the CPM Enabler are independently reusable. The set of functions interact with one another via the framework provided by the CPM Enabler. The CPM Enabler offers multimedia communication functionalities and can be used to build Services, called CPM-based Services. An instantiation of a CPM communication can have one or more media types. The CPM Enabler has also been designed to be extensible with VAS Applications using specifically designed interfaces.

All CPM-based Services will use the functional components and interfaces provided by the framework. Non-CPM Communication Services require Interworking Functions that provide adaptation/mapping between the CPM Enabler and the different technologies to communicate with the framework.

Both CPM-based Services and Interworking Functions for the Non-CPM Communication Services are a set of functions that are deployed by service providers. Figure 2 below shows how the CPM Enabler can be extended with CPM-based Services and Non-CPM Communication Services.

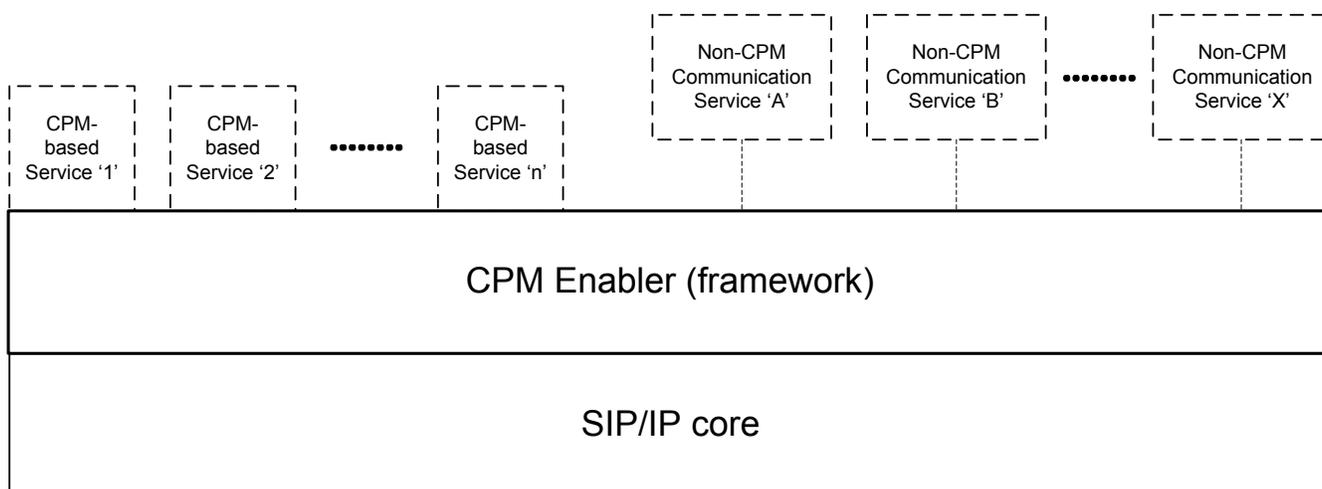


Figure 2: Schematic to illustrate the modularity and extensibility of the CPM Enabler.

4.1 Version 1.0

CPM Enabler version 1.0 offers:

- support for the following CPM conversation requirements:
 - Immediate Messaging and Deferred Delivery Messaging (with temporary server storage and subsequent delivery).
 - Exchanging files with multimedia contents after explicit recipient authorization (file transfer feature).
 - Add or remove Media Streams at any time during a CPM Session.
 - Add or remove users at any time during a CPM Session.
- support for discrete (e.g. text, image, video clip, audio clip, voice clip, binary file) Media Types and continuous (e.g. bidirectional voice, streaming video) Media Stream Types.

- support for the initiation of CPM Conversations for CPM Pre-defined Groups and CPM Ad-hoc Groups.
- support for CPM Users to set up several User Preferences Profiles like Office, Home, Meeting, etc. to which the preferences of the CPM User are associated.
- an environment supporting multiple CPM Addresses and multiple Devices aiming for best user experience in today's heterogeneous world for services, networks and Devices.
- interaction with the Presence Enabler.
- support for interworking with Non-CPM Communication Services.
- inclusion of a network-based storage for CPM Messages, CPM File Transfer Histories, CPM Session Histories, and MIME format wrapped Standalone Media Objects. All these data can be synchronized to all the Devices of the CPM User.
- support for interoperability between multiple networks, i.e. CPM Conversations between Principals from different CPM service providers.

4.2 Future Versions

In addition to the functionalities offered by CPM Enabler version 1.0, functionality for future versions of the CPM Enabler has been discussed as follows:

- support for the following additional conversation requirements:
 - Messaging on behalf of a user
 - Reminders for Deferred Messaging
 - Selective Participant Information options (e.g. Participant is not part of the conversation) in CPM Sessions
 - Selective Participation modes (e.g. hidden, re-negotiation of pseudonyms during sessions) in CPM Sessions
 - Searching for public chat rooms, or ongoing Group Sessions based on criteria
 - Sharing of CPM Session History during conversations to joining Participants
 - Selective protection mechanism (e.g. content screening)
 - One-to-one and one-to-many CPM Conversations among CPM Users and between CPM Users and VAS Applications with the selection of any kind of Media (single or multiple).
- support for filtering of continuous Media exchanged simultaneously in the same session.
- support for creation of CPM Pre-defined Groups on behalf of another user including transfer of ownership and respective notifications
- inclusion of a network-based storage for standalone Media Objects. The Media Objects can be synchronized to all the Devices of the CPM User and can be shared among CPM Users.
- support for various activation schemes for an active User Preferences Profile for any address and Device combination.
- support for various additional mechanisms to manage the multiple Devices of a CPM User:
 - Allow pending invitations
 - Session transfer to another Device
 - Selection of media per Device per session
 - Provide listings of session(s) and capabilities of Devices

- support for CPM-specific presence parameters on behalf of the CPM Users that derive from different Communication Capabilities
- enhanced application support:
 - Media modification functions (e.g. conferencing, adding/removing Media)
 - Control of enabled CPM features on a per-VASP basis.
- support for a generalized interface to communicate with VAS Applications.
- support for interaction with the CAB Enabler for user experience enhancements.

More details on the architecture related to these functionalities are given in Appendix C.

4.3 Security Considerations

The following security aspects are considered in this document:

- SIP signalling security,
- Media Plane security,
- Interworking Function security,
- Message storage security,

The CPM Enabler shall provide at least the same level of confidentiality and integrity as existing Enablers in the area of messaging and communication services.

4.3.1 SIP Signalling Security

The CPM Client (representing the CPM User) shall be authenticated prior to accessing a CPM-based Service via the SIP-based interfaces. The CPM Participating Function should rely on signalling access security and network domain security provided by the underlying SIP/IP core for securing the service environments, e.g. authentication of the Service usage. The SIP signalling access security is defined in [RFC3261] and network domain security is defined in [3GPP TS33.210].

The SIP/IP core needs to ensure during registration that the registered CPM Address is allocated and authorised to be used by that CPM User in order to prevent spoofing attacks. When the SIP/IP core corresponds with 3GPP IMS/3GPP2 MMD and the Device contains USIM/ISIM or UIM/R-UIM/ISIM, mutual authentication is applied as specified in [3GPP TS 33.203]/[3GPP2 S.R0086-0].

For SIP signalling, an integrity protection mechanism shall be used as specified in [3GPP TS 33.203]/[3GPP2 S.R0086-0].

[RFC3261] mandates the support for HTTP digest authentication [RFC2617], if authentication is performed.

4.3.2 Media Plane Security

To protect Media Plane communication against eavesdropping, modification and spoofing, the CPM Media Plane communication should be protected to support its integrity and confidentiality. This protection should be subject to the CPM User's request and service provider policies. A suitable mechanism for integrity and confidentiality protection is specified in [OMA-SEC_CF].

4.3.3 CPM Interworking Function Security

Assuming the existence of full security features compatibility on the part of non-CPM networks, the CPM network shall apply the same security measures to the signalling received from non-CPM networks as it would to the signalling received from CPM networks. Otherwise, when the CPM Enabler cannot apply equivalent security measures, the integrity of the security protection might be compromised and interworking is not recommended. Also, the CPM network should apply the same security measures to the Media Plane communication received from non-CPM networks as it would to the Media Plane

communication received from CPM networks. To these two cases, the security measures defined in subsections 4.3.1 "SIP Signalling Security" and 4.3.2 "Media Plane Security" will apply.

4.3.4 Message Storage Security

The Message Storage Server shall authenticate the Message Storage Client by checking the Principal's address/certificate or by using pre-shared key authentication mechanisms prior to granting access to the CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, Media Objects attached to them and folders stored in the Message Storage Server, when a Message Storage Client directly accesses a Message Storage Server. The Message Storage Client shall authenticate the Message Storage Server.

NOTE: Any other untrusted functional component could be authenticated using the same authentication principles.

The Message Storage Server shall check whether the Principal is authorized for accessing the requested resources stored in the Message Storage Server before granting the access, i.e. validate that the Principal is either the owner of the resource being accessed, or has been given access to it by the owner of the resource being accessed.

In order to protect the interaction between a Message Storage Client and a Message Storage Server against eavesdropping, modification and spoofing, the interaction should be protected to support its integrity and confidentiality, if requested by the Principal and subject to service provider policies.

5. Architectural Model

The CPM Enabler is realized as a framework for communication services, and is realized using a client - server architecture concept. It interacts with other network elements and re-uses concepts and technologies specified by other OMA Enablers (e.g. SIMPLE IM [OMA-SIMPLE-IM], PoC [OMA-POC]) and in non-OMA specifications / documents. The architecture is based on the requirements defined in [OMA-CPM-RD].

5.1 Dependencies

The CPM Enabler utilizes several existing technologies specified in OMA Enablers and other non-OMA specifications (IETF, 3GPP, 3GPP2). Where the CPM Enabler interacts directly with external OMA Enablers, it does so under the Enabler exposure control environment called for in [OSE]. Depending upon the specific policies applicable between the CPM service provider and the provider of the external Enablers, a policy enforcement step may be required.

The CPM Enabler depends on at least the following technologies:

- Session Initiation Protocol (SIP) as described in [RFC3261]
- Session Initiation Protocol (SIP) Extension for Instant Messaging as described in [RFC3428]
- Session Description Protocol (SDP) as described in [RFC4566]
- Message Session Relay Protocol (MSRP) as described in [RFC4975]
- Real-Time Transport Protocol (RTP) and RTP Control Protocol (RTCP) as described in [RFC3550]
- Internet Message Access Protocol version 4 (IMAP4) protocol as described in [RFC3501]

The CPM Enabler depends on technologies provided by external OMA Enablers, including the following:

- Presence technology: Presence Enabler as described in [OMA-PRS-AD]
- XML document management technology: XDM Enabler as described in [OMA-XDM-AD]
- Device provisioning technology : Device Management Enabler as described in [OMA-DM-PRO]
- Notification delivery technology: Push Enabler as described in [OMA-PUSH-AD]

Furthermore, the CPM Enabler is dependent on an underlying SIP/IP core infrastructure to transport SIP messages between the CPM functional components. A particular instantiation of the SIP/IP core infrastructure is the IP Multimedia Subsystem as specified in either [3GPP TS23.228] or [3GPP2 X.S0013-002].

5.2 Architectural Diagram

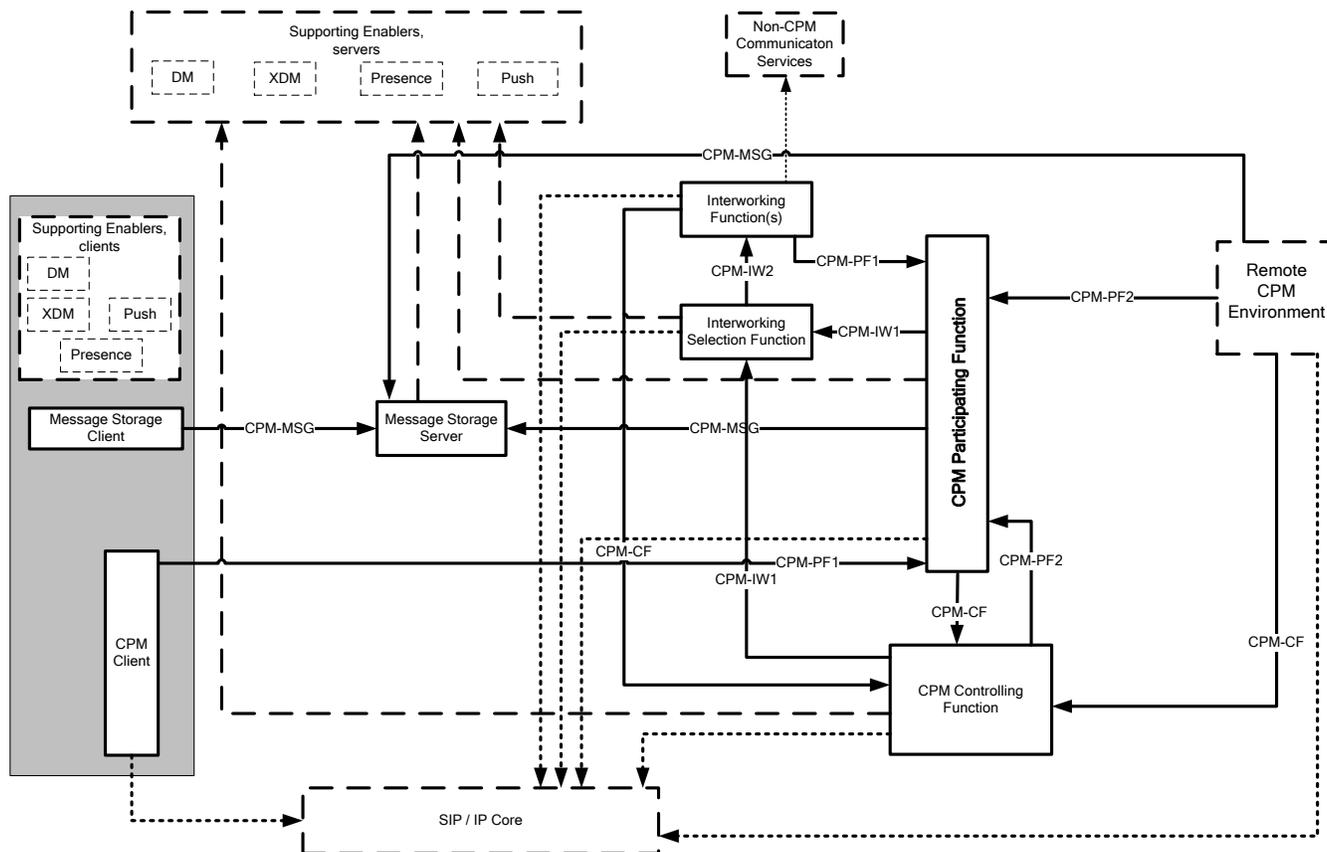


Figure 3: CPM architecture diagram.

Figure 3 gives a representation of the CPM architecture model, depicting the CPM Enabler functional components and their interactions with each other and with external functional components. In this figure the supporting Enablers are split into a clients-part (which reside on the Device) and a servers-part (which reside in the network). The exact interactions between the CPM functional components and the supporting Enablers are depicted in Figure 4 below.

NOTE 1: Due to the framework nature of the CPM Enabler, functional components may or may not be deployed, depending on the required functionality for a deployment. When functional components are not deployed, then that deployment will not be able to fulfil the functionalities offered by the not deployed functional components in the CPM specified manner. Appendix B.1 shows a number of deployment configurations of the CPM Enabler.

The CPM Enabler consists of the following CPM functional components:

- The CPM Client, which allows the CPM User to use CPM-based Services by interacting with other CPM functional components such as the CPM Participating Function. See section 5.3.1.1 for details of this functional component.
- The CPM Participating Function, which acts as the network-side surrogate for the CPM User, including the coordination of the interaction with the multiple CPM Clients the CPM User may have. See section 5.3.1.3 for details of this functional component.
- The CPM Controlling Function, which is responsible for handling CPM Standalone Messages and CPM Sessions that need group functionalities (e.g. Pre-defined Groups, control of Media Streams, ...). See section 5.3.1.4 for details of this functional component.

- The Interworking Selection Function, which selects an Interworking Function that should perform the actual interworking for a CPM Message, CPM File Transfer or CPM Session. See section 5.3.1.5 for details of this functional component.
- The Interworking Function(s), which provides interworking capabilities between CPM-based Services and external Non-CPM Communication Services. Multiple instances of the Interworking Function may exist in a deployment (e.g. specialized Interworking Functions for the various Non-CPM Communication Services). See section 5.3.1.6 for details of this functional component.
- The Message Storage Client, which manages CPM Messages, CPM File Transfer Histories, CPM Conversation Histories, CPM Session Histories and Media Objects attached to them stored at Message Storage Server in the network as well as at the Device's local storage. See section 5.3.1.2 for details of this functional component.
- The Message Storage Server, which provides management and storage functions for CPM Messages, CPM File Transfer Histories, CPM Conversation Histories, CPM Session Histories and Media Objects attached to them in the network. See section 5.3.1.7 for details of this functional component.

The CPM functional components expose their functionalities to the following external functional components:

- A remote CPM environment, which is a CPM environment residing in another (remote) network. See section 5.3.2.7 for details of this external functional component.

The CPM functional components interact with the following external functional components to be able to provide the functionalities of the CPM Enabler:

- A SIP/IP core, which is an underlying infrastructure that provides SIP-based and IP-based functionalities that are needed to support the CPM Enabler. See section 5.3.2.1 for details of this external functional component.
- Supporting Enablers, which are other OMA Enablers being used to support the CPM Enabler. The following supporting Enablers are used by the CPM Enabler:
 - The Presence Enabler for retrieving Presence Information from and publishing Presence Information to.
 - The XDM Enabler for storage and manipulation of CPM User preferences, User Preferences Profiles, CPM Group definitions and user list definitions.
 - The Push Enabler for the delivery of notifications to the Device(s) of a CPM User.
 - The Device Management Enabler for the management of the CPM Client's and the Message Storage Client's settings.

See sections **Error! Reference source not found.** to 5.3.2.5 for details on why and how the CPM Enabler uses these supporting Enablers.

- Non-CPM Communication Services, which are other communication Services to which the CPM Enabler connects via the Interworking Function(s). See section 5.3.2.6 for details of this external functional component.

NOTE 2: Appendix B.2 contains co-location deployment scenarios for a number of CPM functional components, such as the CPM Interworking Selection Function and the CPM Controlling and Participating Functions.

Figure 4 shows the details of the interactions between the functional components of the CPM Enabler and the supporting Enablers.

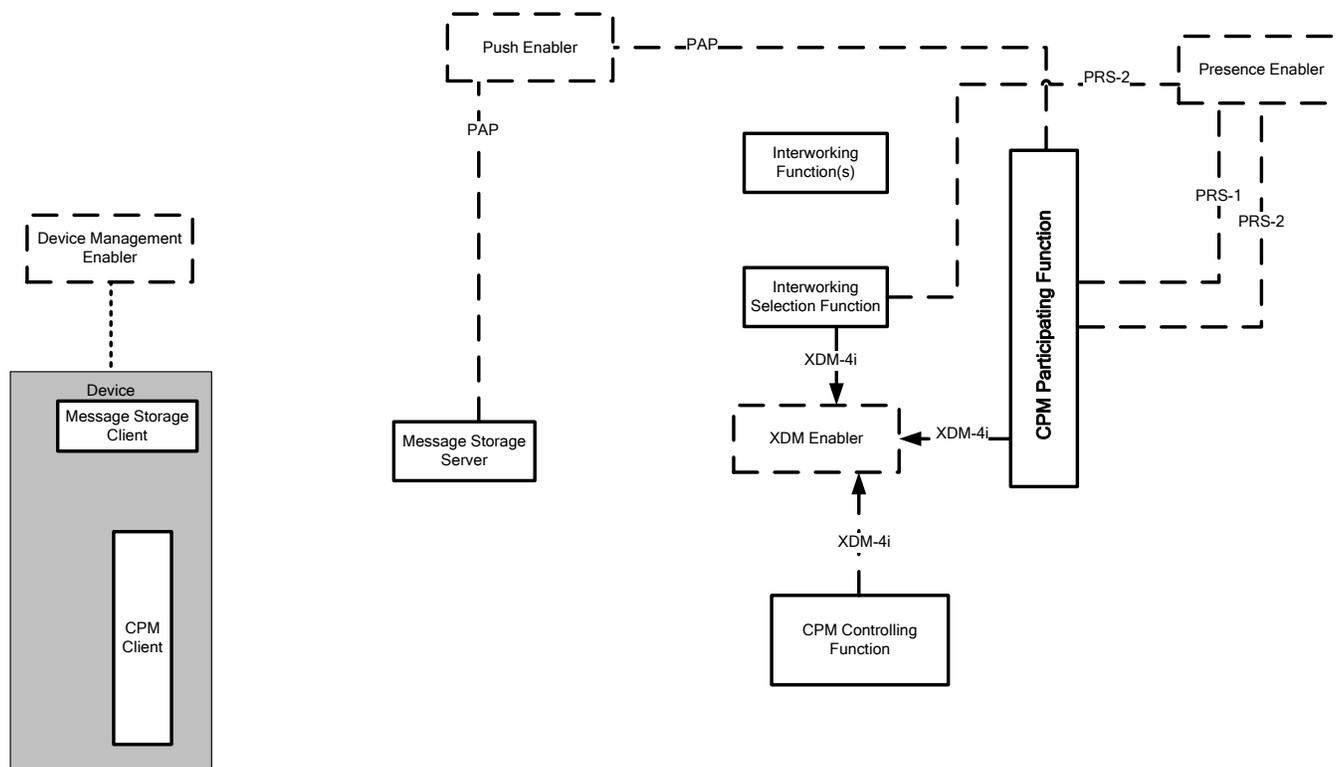


Figure 4: Interactions between the CPM Enabler functional components and the supporting Enablers.

5.3 Functional Components and Interfaces Definition

5.3.1 CPM Functional Components

5.3.1.1 CPM Client

The CPM Client resides in a Device. It is used to access network-based CPM functional components. The other functional components in the Device may communicate internally with the CPM Client. More detail on these functional components can be found in section 5.3.1.2.

NOTE: Specifying interfaces or protocols for internal communication within a Device is out of scope of the CPM Enabler.

The CPM Client is involved in the following high level functionalities:

- SIP/IP core related communication
- Media Plane communication
- Client-side CPM service logic execution
- Communication internal to the Device with the supporting Enablers clients and the Message Storage Client.

The CPM Client is responsible for generating and receiving SIP requests and responses. The CPM Client SHALL be able to use related functionalities provided by the SIP/IP core.

The CPM Client SHALL support the following registration-related functionalities:

- Registration of the CPM Client, the CPM User's Address(es) and the CPM device name.

The CPM Client SHALL support the following client-side CPM service logic related functionalities:

- Supporting the CPM User to use his/her associated CPM Addresses in parallel.
- Generating, sending and receiving CPM Standalone Messages and CPM Chat Messages
 - Sending a CPM Message
 - Indicating the sensitive nature of a CPM Message
 - Indicating the priority level of a CPM Message
 - Indicating one CPM Address to be used for reply for CPM Standalone Messages
 - Receiving a CPM Message
 - Requesting to forward a CPM Message without downloading the CPM Message
 - Requesting and receiving as well as generating and sending delivery notifications and/or read reports
 - Attaching Media Objects, CPM Messages, CPM File Transfer Histories, CPM Session Histories or CPM Conversation Histories to a CPM Message:
 - from the local storage of the Device
 - or by providing references to their locations on either the Message Storage Server or the Content Storage Server (without requiring their download to the Device)
 - Activating/de-activating storage of CPM Standalone Messages during a CPM Conversation
- Initiating and receiving files with Media Objects via CPM File Transfers within and outside of CPM Sessions.
- Performing CPM Session handling related functions
 - CPM Session initiation, joining and termination
 - Negotiating the CPM Session characteristics (e.g. Media Stream Types and codecs of used Media Streams)
 - Indicating the preferred Media Stream
 - CPM Session modification
 - Dynamically adding/modifying/removing Media Streams during a CPM Session
 - Activating/de-activating storage (in the network) of a CPM Session for a CPM User
 - CPM Session information request and reception
 - Requesting information such as the Media Stream Types of the Media Streams currently being used in an ongoing CPM Session of the CPM User and receiving this information
- Performing CPM Group Session handling related functions
 - Using CPM Pre-defined Group definition to initiate a CPM Group Session
 - Initiating ad-hoc type of CPM Group Sessions
 - Receiving invitations to CPM Group Sessions
 - Requesting CPM Group Session information (e.g. Participant information), and receiving notifications of changes in the CPM Group Session information

- Setting and modifying its own SIP session parameters
- Terminating its own participation
- Setting and modifying CPM Group Session parameters
- Terminating a CPM Group Session
- Adding and removing Participants
- Providing the CPM User with the capability to negotiate a unique Pseudonym when requesting to join anonymously in a CPM Group Session
- Performing CPM Conversation handling related functions
 - Starting a CPM Conversation by sending a CPM Standalone Message, initiating a CPM File Transfer or establishing a CPM Session.

The CPM Client SHALL support the following client-side CPM service logic related functionalities related to the usage of multiple Devices:

- Allowing the CPM User to create/modify a Device name
- Providing the Device name during registration

The CPM Client SHALL support interacting with the client components of the supporting Enablers.

The CPM Client SHALL support the following Media Plane communication related functionalities:

- Establishing and maintaining Media Plane connections
- Generating and receiving Media Plane requests and responses

5.3.1.2 Message Storage Client

The Message Storage Client manages CPM Messages, CPM File Transfer Histories, CPM Conversation Histories and CPM Session Histories including Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories stored at the Message Storage Server in the network as well as locally. The other functional components in the Device may communicate internally with the Message Storage Client. More detail on these functional components can be found in section 5.3.1.1.

NOTE: Specifying interfaces or protocols for internal communication within a Device is out of scope of the CPM Enabler.

NOTE: The management of the local storage of a Device is out-of-scope for standardisation.

The Message Storage Client SHALL support the following data management functionalities:

- Managing and consuming resources stored at the CPM User's own Message Storage Server
 - Retrieving CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories and Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories (or previews) from the Message Storage Server
 - Deleting CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, including any Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories, from the Message Storage Server
 - Uploading CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, including any Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories to the Message Storage Server

- Creating (and deleting) folders and subfolders
- Moving and copying CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, including any Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories between folders
- Creating and updating metadata of CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories and folders, including access policy settings
- Obtaining the list of stored CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories and folders residing in the Message Storage Server, with or without filtering criteria
- Synchronizing (e.g. periodically and/or partially) the resources in the local storage of a CPM User's Device(s) with the Message Storage Server, including proper handling of locally deleted items
- Managing and consuming (e.g. retrieving, previewing, viewing) resources from other CPM User's Message Storage Server based on received information and access rights to that data.
- Providing view of stored CPM Conversation Histories

The Message Storage Client also SHALL support local deletion, i.e. deletion of CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories and Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories from the local storage of the Device of a CPM User while retaining them on the Message Storage Server in the network.

5.3.1.3 CPM Participating Function

The CPM Participating Function is a user-specific functional component. Its functionalities are performed in a CPM User's home network and are invoked when the CPM User initiates an originating request or receives a terminating request. The CPM Participating Function handles CPM Sessions, CPM File Transfers, CPM Standalone Messages, and Media on behalf of the Participant it serves and enforces CPM User Preferences and service provider policies. Additionally, it can perform deferred messaging functionality by storing CPM Standalone Messages temporarily and deferring CPM Standalone Messages for later delivery depending on user preferences, availability and willingness of the CPM User, and it can perform conversation history functionality by recording a Participant's CPM Conversations and storing these recorded CPM Conversations into the Message Storage Server.

The CPM Participating Function SHALL support the following functionalities:

- Enforcing policies when handling originating and terminating CPM Session requests, CPM File Transfers and CPM Messages according to the CPM User preferences/service provider policies as follows:
 - Validating the service subscription of the CPM User.
 - Giving precedence to service provider policies over CPM User preferences, when applicable.
 - Providing support for Content Screening based on CPM User preferences and service provider policies.
- For CPM Session handling:
 - Managing CPM Session initiation, modification and termination for originating or terminating CPM Users.
 - Providing SIP session handling, such as SIP session origination, release, etc, on behalf of the involved CPM Client.
 - Providing SDP negotiation support.
 - Providing support for multiple SIP sessions handling in a single CPM Conversation.
 - Providing support for multiple CPM Sessions handling for multiple simultaneous CPM Conversations.

- For CPM File Transfer handling:
 - Managing CPM File Transfer initiation and termination for originating and terminating CPM Users.
 - Provide SIP session handling, such as SIP session origination, release, etc, on behalf of the involved CPM Client.
- For CPM Message handling:
 - Providing sending and receiving of CPM Messages for originating and terminating CPM Users.
 - Providing support for delivery notifications and read reports.
 - Deferring CPM Standalone Messages when the recipient is offline or not willing to receive them, based on the preferences of the CPM User.
 - Interacting with the Message Storage Server to handle a request from a CPM Client to send CPM Messages with referenced stored resources without prior download of the resource to the Device.
 - Interacting with the Message Storage Server to handle a request from a CPM Client to forward the referenced stored CPM Messages or CPM Session Histories without prior download.
- For performing deferred messaging functionalities:
 - Providing policy enforcement according to the CPM User preferences and service provider policies.
 - Temporarily keeping CPM Standalone Messages and associated metadata when the messages cannot be delivered.
 - Providing support for message delivery when the recipient becomes available and reachable. Both Push and Pull methods are supported as follows:
 - The Push method directly delivers the message to all or a subset of the Devices bound to the recipient CPM User's address(es).
 - The Pull method provides notifications with message information (e.g. message size, message sender's address, message sending time) to all or a subset of the Devices bound to the recipient CPM User's address(es), and then the recipient can retrieve the message when needed.
 - Managing the expiry time of a deferred message.
 - Overriding the expiry time provided by the sending CPM Client with the operator defined maximum expiry time when applicable.
 - Handling expired CPM Standalone Messages in accordance with the preferences of the recipient and the service provider policies (i.e. discard the message or store in the Message Storage Server).
 - Sending a non-delivery delivery notification in case of a discarded expired CPM Standalone Message.
- For multiple Devices handling:
 - Supporting multiple Devices handling based on Media characteristics, Communication Capabilities, CPM User preferences and/or service provider policies:
 - Supporting delivery of CPM Session Invitations to more than one Device for a CPM User.
 - Supporting delivery of CPM File Transfer initiation requests to more than one Device for a CPM User.
 - Supporting delivery of CPM Standalone Messages to more than one Device for a CPM User.
 - Supporting the notification of a deferred CPM Standalone Message to more than one Device, when notifications are enabled by the CPM User or service provider policies.

- Ensuring that only one delivery notification or read report is sent to the originator of the CPM Standalone Message in case the CPM Standalone Message was delivered to multiple Devices.
- Involving the Interworking Selection Function:
 - for terminating CPM Session Invitations and terminating CPM Standalone Messages that are to be routed towards Non-CPM Communication Services based on the preferences of the recipient and service provider policies;
 - for originating CPM Session Invitations and originating CPM Standalone Messages that are to be routed towards Non-CPM Communication Services based on service provider policies.
- For performing conversation history functionalities, i.e. recording of CPM Conversations persistently (based on either a CPM Client's request or CPM user preferences):
 - Storing conversation history into the Message Storage Server
 - Storing CPM Standalone Messages and Media Objects that are part of CPM Standalone Messages
 - Storing CPM Sessions as CPM Session Histories
 - Storing CPM File Transfers as CPM File Transfer Histories
 - Storing CPM Conversations as CPM Conversation Histories
 - Providing metadata associated with items to be stored to the Message Storage Server, e.g. identifier linking CPM Standalone Messages to the thread they belong to.

The CPM Participating Function interacts with the Message Storage Server for the storage of recorded CPM Conversations in the CPM User's own message storage area, and to store expired CPM Messages in the recipient's network storage, as determined by the preferences of the recipient and service provider policies.

The CPM Participating Function interacts with the following supporting Enablers:

- The XDM Enabler for the retrieval of user preferences, User Preferences Profiles, and user access policies.
- The Push Enabler to provide notification of deferred messages.

5.3.1.4 CPM Controlling Function

The CPM Controlling Function is a function specific to group communication. The function is either performed in the initiating CPM User's home network or in the remote network. The CPM Controlling Function terminates the incoming requests, and is typically able to distribute requests to multiple recipients and handle group communication. The CPM Controlling Function is provided for any authorized CPM User in a CPM group communication.

The CPM Controlling Function handles the negotiation of capability support for communications involving different CPM networks, and the interaction with local interworking functions.

The CPM Controlling Function SHALL support the following functionalities:

- For policy handling:
 - Performing policy enforcement according to the service provider policies.
 - Performing policy enforcement according to the group policies.
 - Giving precedence to service provider policies over group policies, when applicable.
- For CPM Standalone Message distribution handling (e.g. sending a CPM Standalone Message to a group):
 - Providing support for privacy and anonymity.

- Handling CPM Pre-defined Groups as well as CPM Ad-hoc Groups.
- For CPM Session handling:
 - Providing centralized CPM Session handling (e.g. session parameter negotiation, maintenance of CPM Group Session).
 - Providing centralized mixing and distribution of Media sent by Participants to all the other Participants.
 - Acting as conference focus.
 - Providing support for privacy and anonymity.
 - Providing support for CPM Pre-defined Groups and CPM Ad-hoc Groups.
 - Maintaining CPM Group Session state.
 - Distributing incoming requests to CPM Group Session Participants.
 - Responding to CPM Group Session control requests from Participants.
 - Providing, upon an Authorized Principal's request, information such as the Participants, the Media Stream Types of the used Media Streams and state information concerning the ongoing CPM Group Session
 - Providing negotiation of a Pseudonym unique to a CPM Group Session for a CPM User based on CPM Group policies and the service provider policies (e.g. rejecting or revising the Pseudonym(s) proposed by a CPM User which contain sensitive words).
- Involving the Interworking Selection Function for CPM Sessions and CPM Standalone Messages that are to be routed towards Non-CPM Communication Services, based on service provider policies

The CPM Controlling Function interacts with the following supporting Enablers:

- The XDM Enabler for the retrieval of group definitions and user lists.

5.3.1.5 Interworking Selection Function

The Interworking Selection Function identifies the Non-CPM Communication Service and selects the appropriate Interworking Function a CPM Session, a CPM File Transfer, a CPM Standalone Message, or a CPM Message disposition notification needs to be sent to.

The Interworking Selection Function SHALL support the following functionalities:

- Selecting the Interworking Function to which a CPM Session Invitation, a CPM File Transfer initiation request, a CPM Standalone Message or a CPM Message disposition notification needs to be sent, based on a number of criteria (e.g. message size, service provider policies). When operating in the terminating network, the Interworking Selection Function will also take the preferences of the recipient CPM User into account.
- Subsequently sending the CPM Session, CPM File Transfer, CPM Standalone Message or CPM Message disposition notification to the selected Interworking Function.

The Interworking Selection Function MAY support the following functionalities:

- In case of interworking failure with the previously selected Interworking Function, selecting another available Interworking Function based on a number of criteria (e.g. message size, service provider policies).

Additionally, the Interworking Selection Function SHALL also be able to send a CPM Session, a CPM File Transfer, a CPM Standalone Message or a CPM Message disposition notification to a functionality in the SIP/IP core (e.g. MGCF, IP-SM-GW) that performs interworking.

NOTE: The Interworking Selection Function will only send the SIP signalling to the selected Interworking Function (or the selected functionality in the SIP/IP core), and will not insert itself in the Media path.

The Interworking Selection Function interacts with the XDM Enabler for the retrieval of user preferences and User Preferences Profiles.

5.3.1.6 Interworking Function

The Interworking Function provides interworking capabilities between CPM-based Services and Non-CPM Communication Services.

NOTE: Within the CPM architecture it is possible that multiple instances of the Interworking Function exist at the same time, e.g. one per Non-CPM Communication Service.

The Interworking Function SHALL support the following functionalities:

- CPM Message handling:
 - Forwarding a CPM Message to a Non-CPM Communication Service as a message specific for that Non-CPM Communication Service.
 - Receiving a message from the Non-CPM Communication Service and forwarding it as a CPM Message.
- CPM File Transfer handling:
 - Forwarding a CPM File Transfer initiation request to a Non-CPM Communication Service as a file transfer initiation request specific for that Non-CPM Communication Service, when the Non-CPM Communication Service supports file transfers.
 - Forwarding the response (e.g. accept, deny, timeout) to a file transfer initiation request from a Non-CPM Communication Service.
 - Accepting the CPM File Transfer on behalf of the non-CPM User and sending the file in a message towards the Non-CPM Communication Service, when the Non-CPM Communication Service does not support file transfers, but does support messages.
 - Rejecting the CPM File Transfer when the Non-CPM Communication Service does not support file transfers, nor supports messages.
 - Forwarding the actual file of a CPM File Transfer to the Non-CPM Communication Service.
 - Forwarding a file transfer initiation request from the Non-CPM Communication Service as a CPM File Transfer initiation request.
 - Forwarding the response (e.g. accept, deny, timeout) to a CPM File Transfer initiation request to a Non-CPM Communication Service as the response to a file transfer initiation request.
 - Forwarding the actual file of a file transfer received from a Non-CPM Communication Service towards the CPM environment as part of the CPM File Transfer.
- CPM Session signalling handling:
 - Forwarding a CPM Session Invitation to a Non-CPM Communication Service as a session invitation or a non-CPM message inviting a User specific for that Non-CPM Communication Service, when the Non-CPM Communication Service supports sessions.
 - Forwarding a CPM Session Invitation to a Non-CPM Communication Service as a non-CPM message inviting a User specific for that Non-CPM Communication Service, when the Non-CPM Communication Service does not support sessions.

- Forwarding the response (e.g. accept, deny, timeout) to a session invitation or a non-CPM message inviting a User from a Non-CPM Communication Service.
- Forwarding the response (e.g. accept, deny, timeout) to a non-CPM message inviting a User from a Non-CPM Communication Service.
- Forwarding a session invitation from the Non-CPM Communication Service as a CPM Session Invitation.
- Forwarding the response (e.g. accept, deny, timeout) to a CPM Session Invitation to a Non-CPM Communication Service as the response to a session invitation.
- Forwarding requests to terminate a CPM Session to the Non-CPM Communication Service to which the original CPM Session Invitation was forwarded.
- Forwarding requests to terminate a session from the Non-CPM Communication Service as a CPM Session termination request.
- Media Stream handling:
 - Forwarding Media of used Media Streams received from a CPM functional component to a Non-CPM Communication Service supporting the Media Stream Type of the Media Stream in the format specific for that Non-CPM Communication Service.
 - Forwarding Media of used Media Streams data received from a Non-CPM Communication Service towards the CPM functional components (with appropriate format changes).
- Delivery notification and read reports handling:
 - Forwarding delivery notifications and read reports to a Non-CPM Communication Service.
 - Forwarding delivery notifications and read reports from a Non-CPM Communication Service.

5.3.1.7 Message Storage Server

The following resources are stored in various folders on the Message Storage Server and they can be accessed individually for fulfilment of specific resource management actions:

- CPM Messages (including CPM Messages that are part of a CPM Session History) including any Media Objects attached to them
- CPM File Transfer Histories including any Media Objects attached to them
- CPM Session Histories including any Media Objects attached to them
- Standalone Media Objects consisting of any media wrapped, indexed and handled as standalone stored objects.
- CPM Conversation Histories

The Message Storage Server SHALL support the following functionalities:

- Authentication of the locally recognized User address;

NOTE: Locally recognized User address is any address known by the Message Storage Server and is independent of the access method.

- Storage of CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, including any Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories
- Synchronization of stored resources with the Message Storage Client's local storage including synchronization management under scenarios with multiple Devices, according to the CPM User's preferences and/or the service provider's policy

- Support of management of access control lists over folders and notification of changes to the resources, when notifications are enabled by the CPM User or service provider policies
- Management (e.g. copy, delete, ...) of stored resources and access to those resources
- Searching stored resources
- Enforcement of quota and notification of overflows, when notifications are enabled by the CPM User or service provider policies

The stored resources can be subject to access control lists of Principals associated with folders, defined and managed by the CPM User, which associate Principals with all or a subset of the following privileges:

- copy
- move between folders
- rename
- read / access
- write / modify
- delete

Folders are subject to an additional privilege that allows their creation and the listing of their content, with or without filtering criteria.

Privileges associated to a resource can be subject to an expiration time after which the privilege is revoked.

The Message Storage Server MAY support content adaptation (i.e. for previewing) of stored Media Objects attached to CPM Messages, CPM File Transfer Histories, CPM Session Histories and of CPM Messages (e.g. character set conversion) before delivering to the Device's local storage, based on recipient's preferences, Communication Capabilities, and/or service provider policies

The Message Storage Server interacts with other CPM functional components such as:

- the CPM Participating Function:
 - for the composition of CPM Messages without requiring download of selected stored resources to the Message Storage Client
 - for storage of CPM Messages, CPM File Transfers, CPM Sessions, CPM Conversations and Media Objects that are part of CPM Messages, CPM File Transfers or CPM Session Histories
- the Message Storage Client for
 - Synchronization (e.g. periodically and/or partially) with the local storage on the Device(s) of a CPM User
 - Uploading a CPM Message, A CPM File Transfer History, a CPM Session History, a CPM Conversation History, including any Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories or a set of those from the Device's local storage
 - Downloading a CPM Message, a CPM File Transfer History, a CPM Session History, a CPM Conversation History, a Media Object that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories (or a preview of a Media Object) or a set of those to the Device's local storage
 - Management of stored resources and folders (e.g. copy, move between folders, rename, read / access, write / modify, delete, manage access control lists)

- Listing stored CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, and/or Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories residing in the Message Storage Server, with or without filtering criteria
- Providing references to a CPM Message, a CPM File Transfer History, a CPM Session History, a CPM Conversation History, and/or a Media Object that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories residing in the Message Storage Server (e.g. for forwarding a CPM Message without downloading to the Device's local storage)

The Message Storage Server interacts with the following supporting Enablers:

- The Push Enabler for notifications delivery (e.g. quota overflows or arrival of a new message).

5.3.2 External Functional Components

The CPM Enabler interacts with several OMA and non-OMA functional components.

The following subsections describe the external functional components with which the CPM Enabler interacts directly in the fulfillment of the CPM requirements.

5.3.2.1 SIP/IP Core

The SIP/IP core includes a number of SIP proxies and SIP registrars as defined in [RFC3261]. The SIP/IP core performs the following functions that are needed to support the CPM Enabler:

- Routes the SIP signalling.
- Provides address resolution services.
- May provide charging information.
- Provides SIP compression.
- Performs authentication and authorization of the CPM User at the CPM Client based on the CPM User's service subscription.
- Provides and maintains registration of CPM Clients and their associated CPM Address(es).
- Provides support for the CPM User address anonymity in the SIP signalling as described in [RFC3325].
- Provides list of registered Devices and their capabilities.
- Provides support for NAT traversal

When the SIP/IP core is based on the 3GPP IMS or 3GPP2 MMD, the SIP/IP core architecture is specified in 3GPP [3GPP TS23.228] or 3GPP2 [3GPP2 X.S0013-002], respectively.

Each of the interfaces exposed by the SIP/IP core is an I2 interface in the OSE context. These interfaces are used by all the CPM functional components requiring SIP/IP core functionality. When the SIP/IP core is based on the 3GPP IMS or 3GPP2 MMD, the utilization of these interfaces complies with the rules described in [OMA-IMS-AD].

5.3.2.2 Presence Enabler

The Presence Enabler provides users with the possibility to store and manipulate Presence Information. It also allows users or entities to obtain the Presence Information of other users. The functionalities provided by the Presence Enabler are specified in [OMA-PRS-AD].

The CPM Enabler uses the functionalities of the Presence Enabler to obtain Presence Information.

The CPM functional components interact with the Presence Enabler via the PRS-2 reference point defined by the Presence Enabler.

5.3.2.3 XML Document Management Enabler

The XDM Enabler allows users and other Enablers to store and manage XML documents [OMA-XDM-AD]. The functionalities provided by the XDM Enabler are specified in [OMA-XDM-AD].

For the CPM Enabler, the XDM Enabler provides XDM operations for:

- CPM User preferences and User Preferences Profiles
- access policies
- user lists
- group definitions

The CPM functional components interact with the XDM Enabler via the XDM-4i interface.

5.3.2.4 Device Management Enabler

The Device Management Enabler can perform the following functions that are needed for the support of the CPM Enabler:

- Provisioning and management of configuration parameters necessary for the CPM Client.
- Provisioning and management of configuration parameters necessary for the Message Storage Client.

The functionalities provided by the Device Management Enabler are as described in [OMA-DM-PRO].

5.3.2.5 Push Enabler

The Push Enabler can perform the following functions that are needed to support the CPM Enabler:

- Delivery of notifications towards the Device(s) of the CPM User.

The functionalities provided by the Push Enabler are as described in [OMA-PUSH-AD].

The CPM functional components interact with the Push Enabler via the Push Access Protocol (PAP) reference point defined by the Push Enabler.

5.3.2.6 Non-CPM Communication Services

The Non-CPM Communication Services represent the other communication services the CPM Enabler connects to via the Interworking Function.

The CPM Enabler interacts with these Non-CPM Communication Services to:

- Send and receive messages to and from non-CPM Users.
- Send and receive delivery notification and/or read reports to and from non-CPM Users.
- Initiate and receive file transfers to and from non-CPM Users.
- Send and receive session invitations to and from non-CPM Users.
- Exchange Media within Media Streams with non-CPM Users.

The CPM functional components interface to the Non-CPM Communication Services via the interfaces exposed by the Non-CPM Communication Service server, to inject and receive messages, session invitations, and Media Streams into the Non-CPM Communication Service. The CPM functional components use these interfaces without requiring changes to them.

5.3.2.7 Remote CPM Environment

The remote CPM environment is the CPM environment residing in another (remote) network.

This remote CPM environment is a mirror of the environment described in this document, and can contain the full set of CPM functional components described in this document or a subset thereof.

The CPM Enabler interacts with the remote CPM environment to allow CPM Users of the local CPM environment to interact with CPM Users of the other (remote) network.

5.3.3 CPM Interfaces

5.3.3.1 CPM-PF1

The CPM-PF1 interface is exposed by the CPM Participating Function to allow other functional components (such as the CPM Client) to access / trigger the functionalities of the CPM Participating Function.

Supported functionalities include:

- CPM Session signalling
- CPM File Transfer
- CPM Message transfer
- Media transfer within Media Streams
- Control of Media transfers within Media Streams
- Delivery of stored deferred messages

The protocols used for the CPM-PF1 interface are SIP, SDP, MSRP, and RTP/RTCP. SIP is used for CPM Session signalling, for CPM File Transfer signalling and for discrete Pager Mode CPM Standalone Message transfer. SDP is used to describe the set of Media Streams, codecs, and other Media related parameters supported during CPM Session set up and for describing file characteristics during CPM File Transfer initiation. MSRP is used for the transfer of Large Message Mode CPM Standalone Messages, for the exchange of CPM Chat Messages, both small and large, and for the Media transfer of a CPM File Transfer. RTP is used for continuous Media transport and RTCP supports for the exchange of information needed to control RTP sessions.

NOTE: The exact network path used for the actual Media transfers (i.e. MSRP and RTP/RTCP protocols) will be negotiated via the SIP signalling part of this interface. For example, it is possible that direct client-to-client Media transfers are negotiated, or a direct Media transfer between a client and an Interworking Function.

The signalling part of the CPM-PF1 interface is dependent on an underlying SIP/IP core infrastructure.

5.3.3.2 CPM-PF2

The CPM-PF2 interface is exposed by the CPM Participating Function to allow other network-side functional components (such as another CPM Participating Function in a remote CPM environment or a CPM Controlling Function) to access / trigger the functionalities of the CPM Participating Function.

The CPM-PF2 interface enables the communication between a home CPM Participating Function and a CPM Conversation requestor located in a remote CPM environment. The CPM Conversation requestor can be a remote CPM Participating Function or a remote CPM Controlling Function. For this purpose, the interface relies on the communication between the different networks via SIP/IP cores.

NOTE: The CPM-PF2 interface can also be used by remote SIP based networks for CPM-compatible SIP services.

In addition, the CPM-PF2 interface enables the communication between a CPM Controlling Function and a CPM Participating Function that is located in the same network as the CPM Controlling Function, e.g. when one of the recipients of a group communication belongs to the same network as the originator and the group's CPM Controlling Function forwards the message to the recipient's CPM Participating Function via the CPM-PF2 interface.

Supported functionalities include:

- CPM Session signalling
- CPM File Transfer
- CPM Message transfer
- Media transfer within Media Streams
- Control of Media transfers within Media Streams

The protocols used for the CPM-PF2 are SIP, SDP, MSRP, and RTP/RTCP. SIP is used for CPM Session signalling, for CPM File Transfer signalling and for discrete Pager Mode CPM Standalone Message transfer. SDP is used to describe the set of Media Streams with codecs, and other Media related parameters supported during CPM Session set up and for describing file characteristics during CPM File Transfer initiation. MSRP is used for the transfer of Large Message Mode CPM Standalone Messages, for the exchange of CPM Chat Messages, both small and large, and for the Media transfer of a CPM File Transfer. RTP is used for continuous Media transport and RTCP supports for the exchange of information needed to control the RTP sessions.

NOTE: The exact network path used for the actual Media transfers (i.e. MSRP and RTP/RTCP protocols) will be negotiated via the SIP signalling part of this interface. For example, it is possible that a CPM Participating Function has a direct Media transfer with a CPM Client in a remote CPM environment.

The signalling part of the CPM-PF2 interface is dependent on an underlying SIP/IP core infrastructure.

5.3.3.3 CPM-CF

The CPM-CF interface is exposed by the CPM Controlling Function to allow entities to support exchange of CPM Messages and CPM Sessions involving group functionalities.

Supported functionalities include:

- CPM Session signalling
- CPM Message transfer
- Media transfer within Media Streams
- Control of Media transfers within Media Streams

The protocols used for the CPM-CF interface are SIP, SDP [RFC4566], MSRP [RFC4975], and RTP/RTCP [RFC3550]. SIP is used for CPM Session signalling and for discrete Pager Mode CPM Standalone Message transfer. MSRP is used for the transfer of Large Message Mode CPM Standalone Messages, as well as for the exchange of CPM Chat Messages, both small and large. RTP is used for continuous Media transport and RTCP supports for the exchange of information needed to control RTP sessions.

NOTE: The exact network path used for the actual Media transfers (i.e. MSRP and RTP/RTCP protocols) will be negotiated via the SIP signalling part of this interface. For example, it is possible that a CPM Controlling Function has a direct Media transfer with a CPM Client in a remote CPM environment.

The signalling part of the CPM-CF interface is dependent on an underlying SIP/IP core infrastructure.

5.3.3.4 CPM-IW1

The CPM-IW1 interface is exposed by the Interworking Selection Function to allow other functional components (such as the CPM Controlling Function or the CPM Participating Function) to send CPM Messages, Media Streams, and CPM Sessions towards an Interworking Function that takes care of the interworking towards Non-CPM Communication Services.

Supported functionalities include:

- CPM Session signalling

- CPM File Transfer
- CPM Message transfer

The protocols used for the CPM-IW1 interface are SIP and SDP. SIP is used for CPM Session signalling, for CPM File Transfer signalling and for discrete Pager Mode CPM Standalone Message transfer. SDP is used to describe the set of Media Streams with codecs, and other Media related parameters supported during CPM Session set up and for describing file characteristics during CPM File Transfer initiation.

5.3.3.5 CPM-IW2

The CPM-IW2 interface is exposed by the Interworking Function to allow other functional components (such as the Interworking Selection Function) to perform interworking functionality of CPM Messages, Media Streams, and CPM Sessions towards Non-CPM Communication Services.

Supported functionalities include:

- CPM Session signalling
- CPM File Transfer
- CPM Message transfer
- Continuous Media transfer
- Control of continuous Media transfers

The protocols used for the CPM-IW2 interface are SIP, SDP, MSRP and RTP/RTCP. SIP is used for CPM Session signalling, for CPM File Transfer signalling and for discrete Pager Mode CPM Standalone Message transfer. SDP is used to describe the set of Media Streams with codecs, and other Media related parameters supported during CPM Session set up and for describing file characteristics during CPM File Transfer initiation. MSRP is used for the transfer of Large Message Mode CPM Standalone Messages, for the exchange of CPM Chat Messages, both small and large, and for the Media transfer of a CPM File Transfer. RTP is used for continuous Media transport and RTCP supports the exchange of information needed to control RTP sessions.

5.3.3.6 CPM-MSG

The CPM-MSG interface is exposed by the Message Storage Server to allow other functional components (such as the Message Storage Client, the CPM Participating Function or a remote environment trusted server) to access the Message Storage Server.

Supported functionalities include:

- Authentication of the locally recognized User address.
- Uploading and downloading of (parts of) CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, and Media Objects attached to them.
- Retrieving a preview of a Media Object attached to a CPM Message, a CPM File Transfer History, or a CPM Session History.
- Deletion of (parts of) CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, and Media Objects attached to them from the Message Storage Server.
- Renaming and copying of CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, and Media Objects attached to them in a folder in the Message Storage Server.
- Creation, deletion, moving, and renaming of folders in the Message Storage Server.
- Moving and copying of CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, and Media Objects attached to them between folders in the Message Storage Server.

- Searching within accessible folders in the Message Storage Server.
- Synchronisation between the Message Storage Client's local storage and the Message Storage Server.
- Obtaining a reference to a stored object, or obtaining a stored object using a reference in the Message Storage Server (e.g. to be used for "forward without download" functionality).
- Management of access permissions to specific folders in the Message Storage Server.

The protocol used for the CPM-MSG interface is IMAP4.

5.4 Flows

A comprehensive set of flows for CPM can be found in [OMA-CPM-SD].

Appendix A. Change History

(Informative)

A.1 Approved Version 1.0 History

Reference	Date	Description
OMA-AD-CPM-V1_0-20120612-A	12 Jun 2012	Status changed to Approved by TP: OMA-TP-2012-0221R01-INP_CPM_V1_0_ERP_for_Final_Approval

Appendix B. Deployment Scenarios (Informative)

B.1 Deployment Configurations

The architecture presented in section 5.2 shows a full deployment in which all CPM functional components are deployed. It is also mentioned in section 5.2 that this does not always have to be the case, and that, depending on the required functionality, a deployment can leave out functional components, at the expense of not being able to offer the functionalities associated with the not deployed functional components in the CPM specified manner.

The following subsections show a number of deployment configurations in which not all of the CPM functional components are deployed.

B.1.1 Deployment without Network Storage

This subsection shows a deployment configuration for a Service Provider that doesn't want to offer network storage functionalities.

For this deployment configuration the CPM Client, CPM Participating Function, CPM Controlling Function, Interworking Selection Function, and Interworking Function(s) functional components are deployed. The other functional components of the CPM Architecture (Message Storage Client and Message Storage Server) are not deployed. Figure 5 shows this deployment configuration.

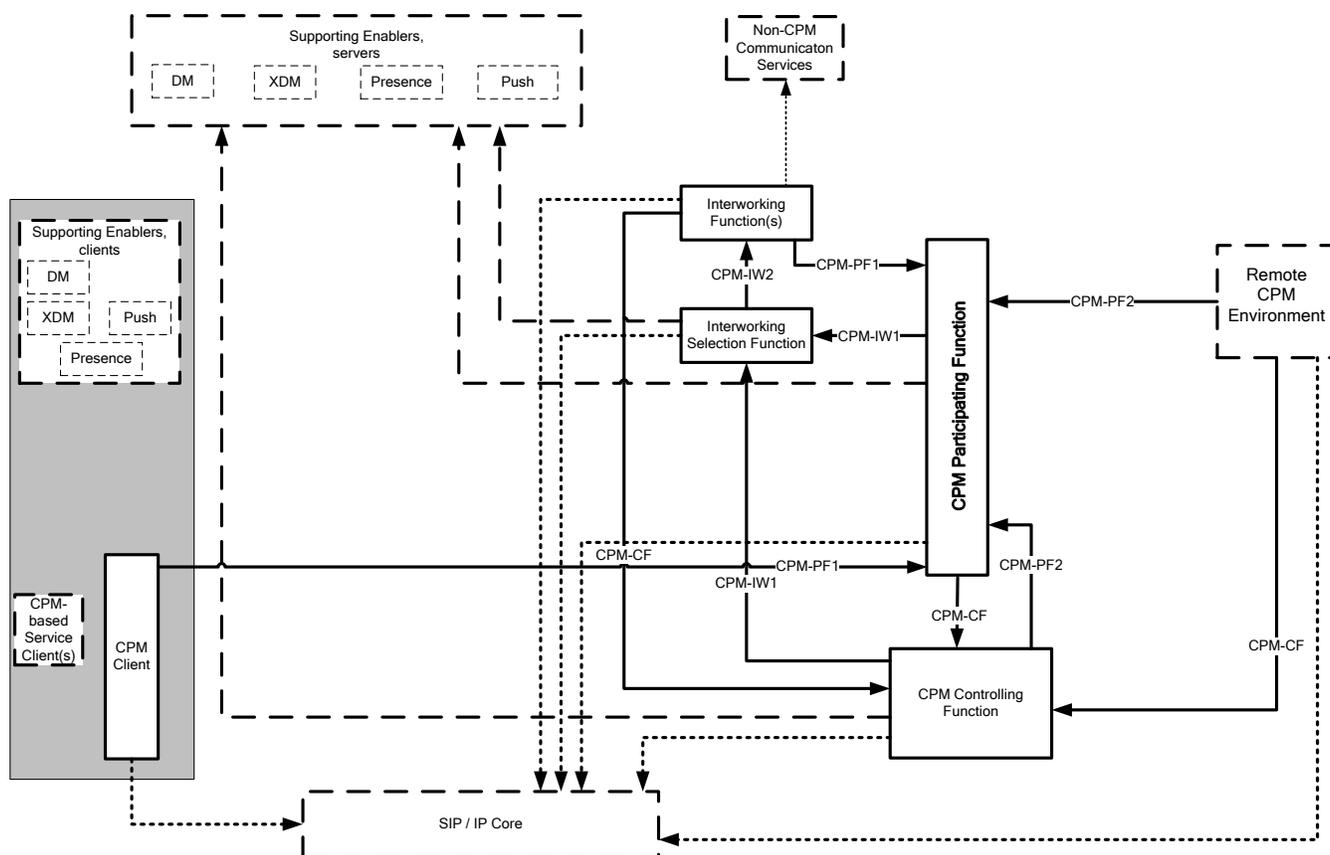


Figure 5: Deployment configuration without network storage and VAS Applications

B.1.2 Deployment for Basic Communications Service

This subsection shows a minimal deployment configuration for an environment that wants to provide real-time messaging and continuous Media exchange capabilities only, without the need for network-based storage, connectivity towards VAS Applications, or interworking functionality.

For this deployment configuration only the CPM Client, CPM Participating Function, and CPM Controlling Function, functional components are deployed. All other functional components of the CPM Architecture are not deployed. Figure 6 shows this deployment configuration.

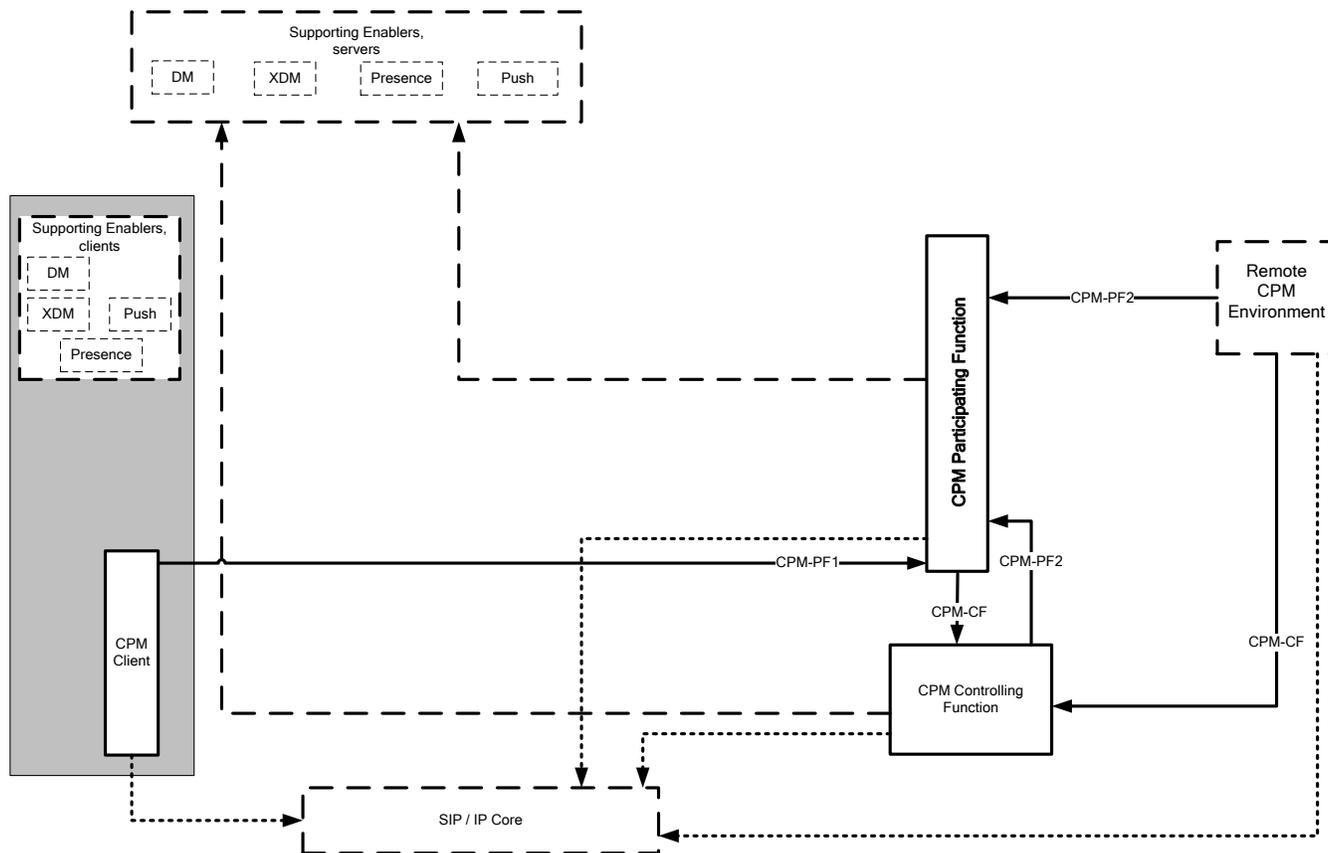


Figure 6: Deployment configuration for a basic communications service

B.2 Co-location Deployment Scenarios

The architecture presented in section 5.2 shows all functional components being deployed separately. This does not always have to be the case. Some functional components can easily be deployed co-located with one of the other functional components in the CPM architecture. Most notably this is true for the following functional components:

1. The Interworking Selection Function
2. The CPM Controlling Function and CPM Participating Function

The following subsections provide more details for the co-located deployments for each of these functional components.

B.2.1 Interworking Selection Function

The CPM architecture as depicted in Figure 3 shows the Interworking Selection Function as an independent logical functional component. It is assumed, however, that the Interworking Selection function is usually deployed co-located with another logical functional component of the CPM architecture, most notably either the CPM Participating Function or the

Interworking Function. This section describes the various deployment scenarios that exist for the deployment of the Interworking Selection Function:

1. The Interworking Selection Function is deployed co-located with the CPM Participating Function
2. The Interworking Selection Function is deployed co-located with the Interworking Function
3. The Interworking Selection Function is deployed as a separate physical component

The following sub-sections detail each of these deployment scenarios in more detail.

B.2.1.1 ISF Co-located with the CPM Participating Function

Figure 7 shows the scenario where the Interworking Selection Function is deployed co-located with the CPM Participating Function.

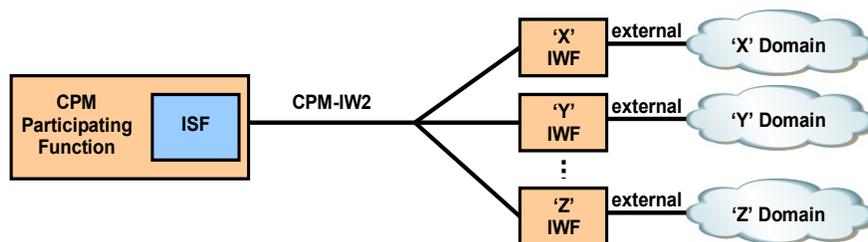


Figure 7: ISF co-located with the CPM Participating Function.

In this deployment scenario the ISF is an internal component of the CPM Participating Function. This means that the CPM-IW1 interface is internal to the CPM Participating Function.

In this deployment scenario the CPM Participating Function takes the decision that interworking is to occur, involves the internal ISF to decide to which Interworking Function a CPM Message, a CPM File Transfer initiation request or CPM Session Invitation needs to be sent, and then sends (via the CPM-IW2 interface) the CPM Message, CPM File Transfer initiation request or CPM Session Invitation towards the selected Interworking Function, which takes care of the protocol adaptation to the Non-CPM Communication Service.

B.2.1.2 ISF Co-located with the Interworking Function

Figure 8 shows the scenario where the Interworking Selection Function is deployed co-located with the Interworking Function.

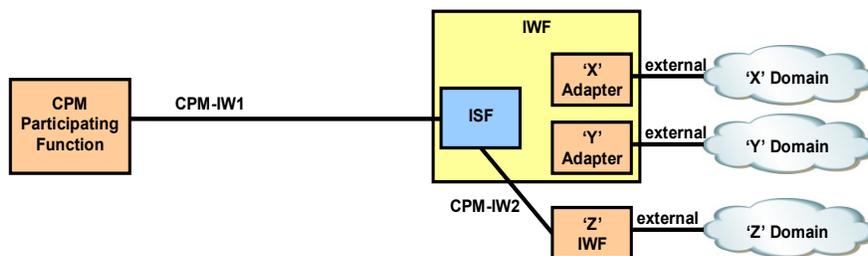


Figure 8: ISF co-located with the Interworking Function.

In this deployment scenario the ISF is an internal component of the 'primary' Interworking Function. This means that the CPM-IW2 interface is internal to the Interworking Function for the adapters that the Interworking Function implements by itself. The ISF internal to the 'primary' Interworking Function still is able to relay CPM Messages, CPM File Transfer initiation requests and CPM Sessions Invitations (via the CPM-IW2 interface) to a secondary, external, Interworking Function, which specializes in a certain Non-CPM Communication Service.

In this deployment scenario the CPM Participating Function takes the decision that interworking is to occur, and sends (via the CPM-IW1 interface) the CPM Message, CPM File Transfer initiation request or CPM Session Invitation to the ISF of the 'primary' Interworking Function. The ISF internal to the 'primary' Interworking Function selects the Non-CPM Communication Service that needs to be interworked with, and either involves the appropriate internal adapter to take care of the protocol adaptation to the Non-CPM Communication Service, or (via the CPM-IW2 interface) to a secondary, external, Interworking Function, which specializes in a certain Non-CPM Communication Service, to take care of the protocol adaptation towards that specific Non-CPM Communication Service.

B.2.1.3 ISF as a Separate Physical Component

Figure 9 shows the scenario where the Interworking Selection Function is deployed as a separate physical component.

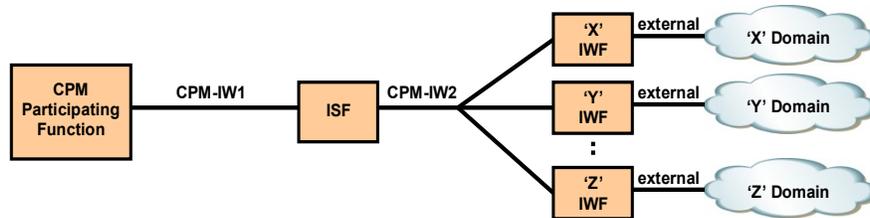


Figure 9: ISF as a separate physical component.

This deployment scenario realizes the CPM architecture where all 3 logical functional components (CPM Participating Function, Interworking Selection Function, and Interworking Function) are deployed separate from each other.

In this deployment scenario the CPM Participating Function takes the decision that interworking is to occur, and sends (via the CPM-IW1 interface) the CPM Message, CPM File Transfer initiation request or CPM Session Invitation to the ISF. The ISF selects which Interworking Function needs to handle a CPM Message, CPM File Transfer initiation request or CPM Session Invitation, and then sends (via the CPM-IW2 interface) that CPM Message, CPM File Transfer initiation request or CPM Session Invitation towards the selected Interworking Function, which takes care of the protocol adaptation to the Non-CPM Communication Service.

B.2.2 CPM Controlling Function and CPM Participating Function

The CPM architecture as depicted in Figure 2 shows the CPM Controlling Function and the CPM Participating Function as independent logical functional components. In certain deployments and/or implementations, the two functions may co-exist within a physical component. This section depicts two scenarios:

1. The CPM Controlling Function and the CPM Participating Function deployed as separate physical components;
2. The CPM Controlling Function and the CPM Participating Function deployed co-located in a physical component.

B.2.2.1 CPM Controlling Function and CPM Participating Function deployed separately

In this scenario, the CPM Controlling Function and CPM Participating Function are deployed separately in the home network and communicate using the CPM-CF interface.

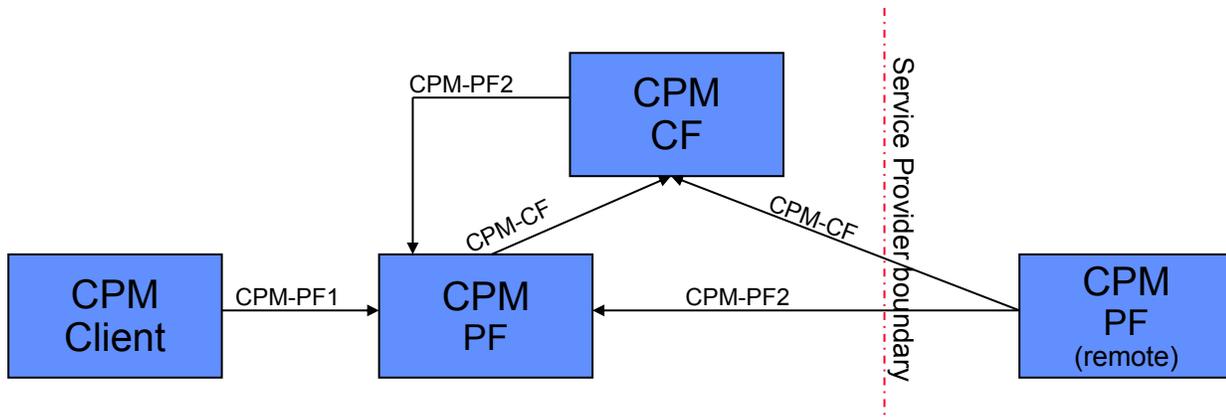


Figure 10: CPM Participating Function and CPM Controlling Function deployed separately.

B.2.2.2 CPM Controlling Function and CPM Participating Function deployed co-located

In this deployment, the CPM Controlling Function and the CPM Participating Function are co-located on a physical server in the home network. In this deployment, it is left at the implementer’s discretion to decide how to implement the “internal” CPM-CF and CPM-PF2 interfaces between the home CPM Participating Function and the CPM Controlling Function.

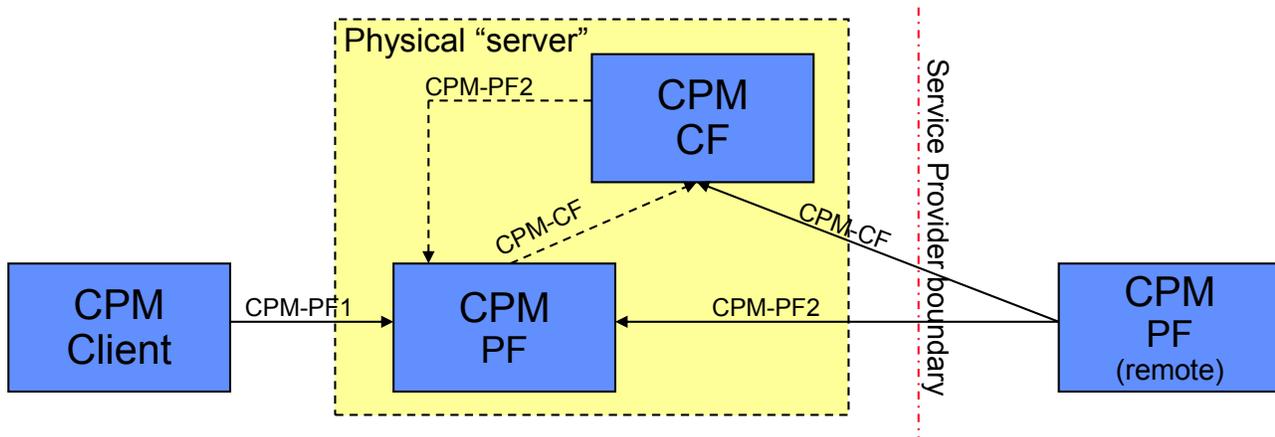


Figure 11: CPM Participating Function Co-located with the CPM Controlling Function.

Appendix C. Architecture for Future CPM Functionality (Informative)

During the course of the development of the CPM Enabler release version 1.0, additional functionality was discussed that, due to time constraints, could not be finalized for the first release, and hence was descope. To give the reader an impression of what this additional functionality comprises, and what future releases of CPM are likely to provide, this informative appendix was created. It must be noted though that although this appendix represents agreed architectural approaches, these approaches were not investigated and backed up by detailed technical work and hence the below functionalities may change once a future CPM Enabler release has created the detailed technical specifications for these future functionalities.

C.1 Additional Security Considerations

C.1.1 Content Storage Security

The Content Storage Server shall authenticate the Principal's address supplied by the Content Storage Client prior to granting access to the standalone Media Objects and folders stored in the Content Storage Server, when a Content Storage Client directly accesses a Content Storage Server. The Content Storage Client shall authenticate the Content Storage Server.

NOTE: Any other untrusted functional component could be authenticated using the same authentication principles.

The Content Storage Server shall authenticate a requesting Content Storage Server from a remote CPM environment prior to granting access to the standalone Media Objects and folders stored in the Content Storage Server. The requesting Content Storage Server from a remote CPM environment shall authenticate the Content Storage Server.

The Content Storage Server shall authorize the Principal prior to granting access to the resources stored in the Content Storage Server, i.e. validate that the Principal is either the owner of the resource being accessed, or has been given access to it by the owner of the resource being accessed.

In order to protect the interaction between a Content Storage Client and a Content Storage Server and the interaction between two Content Storage Servers against eavesdropping, modification and spoofing, the interaction should be protected to support its integrity and confidentiality, if requested by the Principal and subject to service provider policies.

C.2 Additional Dependencies

The following technology dependencies might be added in future versions of the CPM Enabler:

- Web Distributed Authoring and Versioning (WebDAV) protocol as described in [RFC4918]

The following external OMA Enablers might be added as dependency in future versions of the CPM Enabler:

- Charging technology : Charging Enabler as described in [OMA-CHRG-AD]
- Synchronisation technology: DS Enabler as described in [OMA-DS]

C.3 Evolved Architecture Diagram

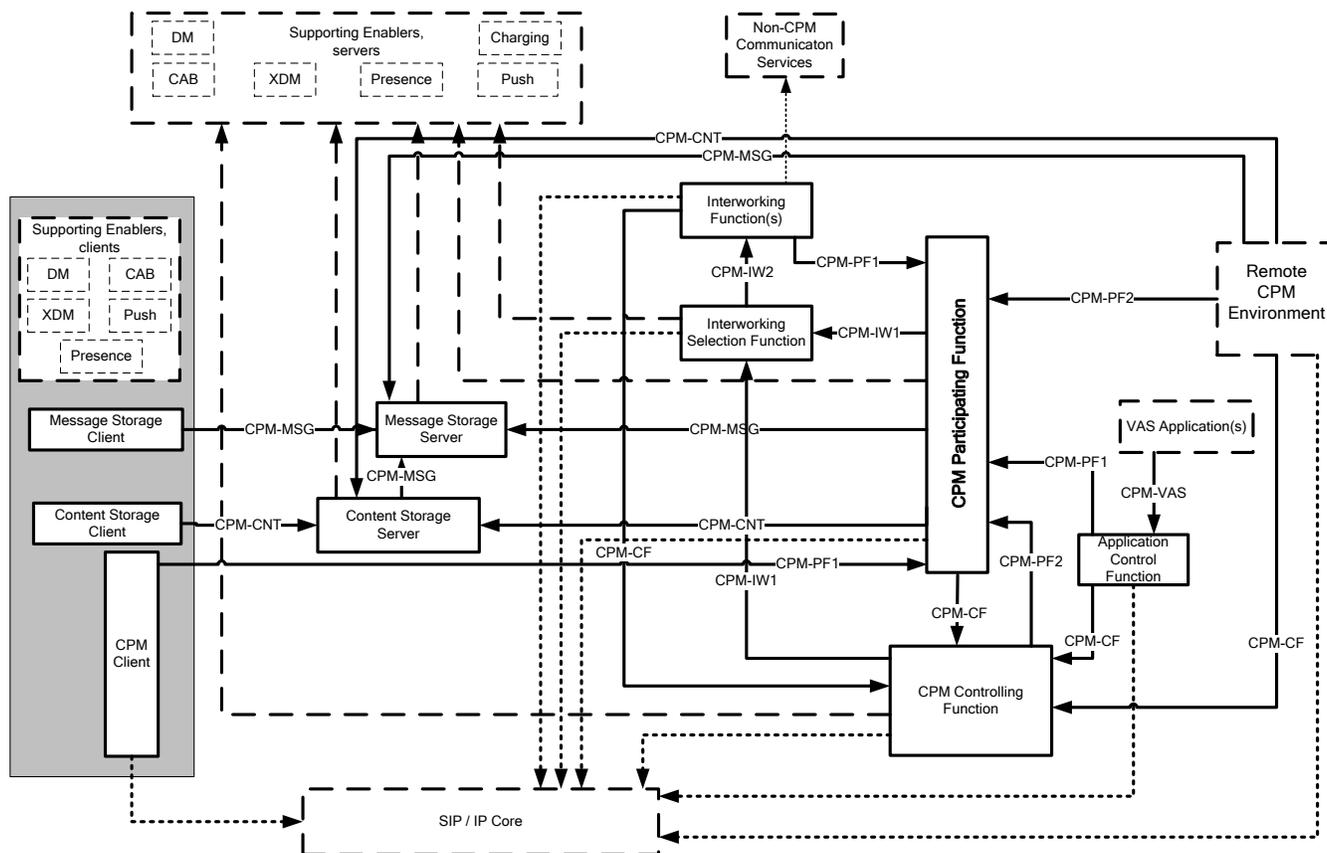


Figure 12: A possible future CPM architecture diagram.

Figure 12 gives a representation of a possible future CPM architecture model, when all future requirements of [OMA-CPM-RD] are taken into account.

Within this future architecture model of the CPM Enabler the following additional functional components can be identified:

- The Application Control Function, which provides access to functionalities of the CPM Controlling Function and of the CPM Participating Function towards external functional components, for example Applications. See section C.5.1.3 for details of this functional component.
- The Content Storage Client, which manages standalone Media Objects stored at Content Storage Server in the network as well as at the Device’s local storage. See section C.5.1.1 for details of this functional component.
- The Content Storage Server, which provides management and storage functions for standalone Media Objects in the network. See section C.5.1.2 for details of this functional component.

The CPM functional components might expose their functionalities to the following additional external functional components:

- VAS Application(s), which are applications using the CPM Enabler to deliver value-added services. See section C.7.2 for details of this external functional component.

The CPM functional components might interact with the following additional external functional components to be able to provide the functionalities of the CPM Enabler:

- Supporting Enablers, which are other OMA Enablers being used to support the CPM Enabler. The following supporting Enablers might be used by the CPM Enabler:
 - The Charging Enabler for the submission of charging information.
 - The CAB Enabler for complementing the CPM Enabler with address book functionality

See sections C.6 and C.7.2 for details on why and how the CPM Enabler uses these supporting Enablers.

Figure 13 shows the details of the interactions between the functional components of the CPM Enabler and the supporting Enablers when the future CPM architecture model is taken into account.

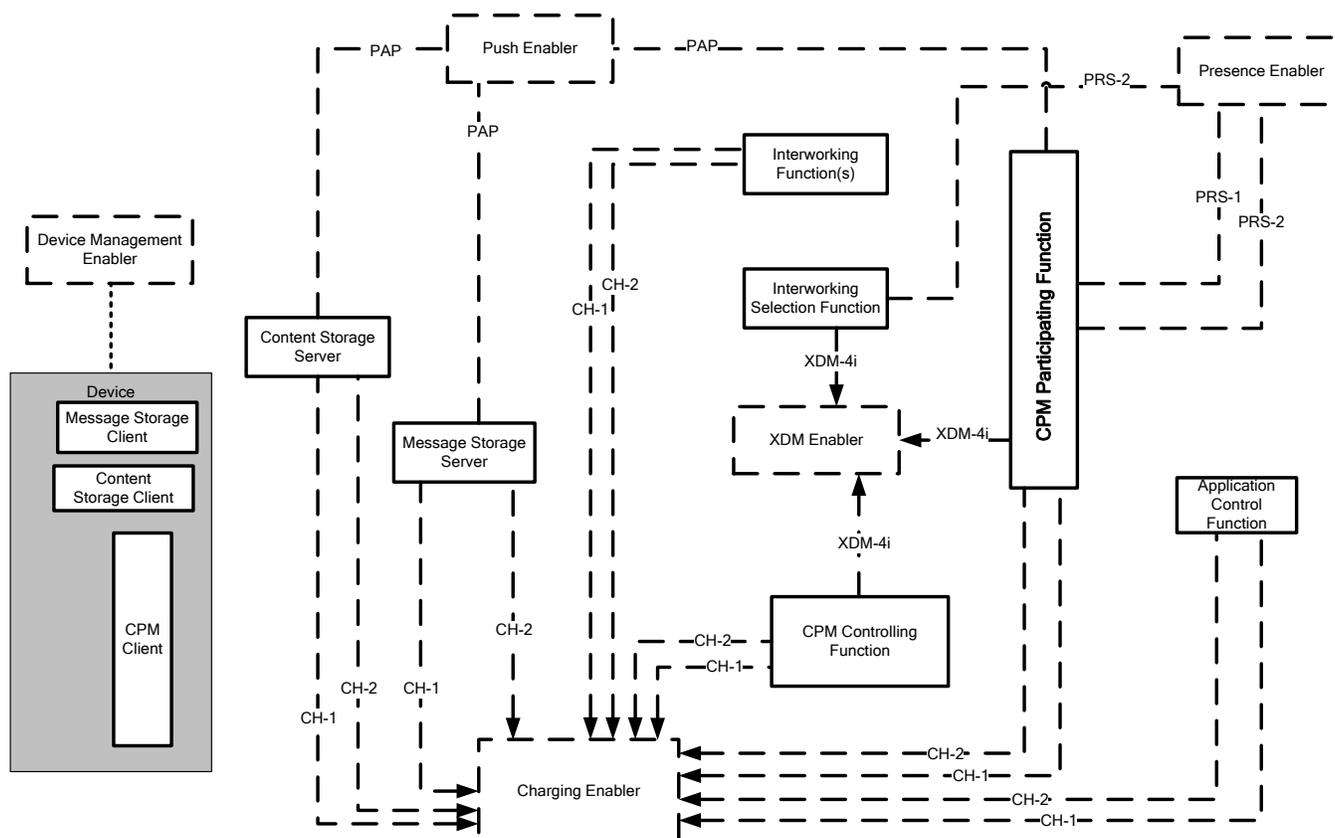


Figure 13: Interactions between the CPM Enabler functional components and the supporting Enablers.

C.4 Additions to Existing CPM Enabler Functional Components

The following subsections define additional functionalities that might be supported by the existing CPM Enabler functional components in future versions of the CPM Enabler.

C.4.1 CPM Client

In addition to the registration-related functionalities described in section 5.3.1.1, in the future the CPM Client may support the following registration-related functionalities:

- Retrieval of registration information

In addition to the client-side CPM service logic related functionalities described in section 5.3.1.1, in the future the CPM Client may support the following client-side CPM service logic related functionalities:

- Sending a CPM Standalone Message with more than one CPM Addresses to be used for reply.
- Activating, splitting, merging, and transferring CPM Sessions between Devices
- Accessing the list of the registered Devices (i.e. the list of names of his/her Devices)

In addition to the Media Plane communication related functionalities described in section 5.3.1.1, in the future the CPM Client may support the following Media Plane communication related functionalities:

- Handling Media Plane floor control requests when applicable

C.4.2 CPM Participating Function

In addition to the functionalities described in section 5.3.1.3, in the future the CPM Participating Function may support the following functionalities:

- Providing support for Online and Offline Charging functions for CPM Sessions, CPM File Transfers and CPM Standalone Messages via the Charging Enabler
- For CPM Message handling:
 - Interacting with the Content Storage Server to handle a request from a CPM Client to send CPM Messages with referenced stored resources without prior download of the resource to the Device.
 - Supporting content adaptation.
 - Re-directing an incoming CPM Standalone Message to an alternate address based on Media Types and/or content adaptation, Communication Capabilities, preferences of the CPM User and/or service provider policies.
- For performing deferred messaging functionalities:
 - Extension of the methods to handle expired CPM Standalone Messages in accordance with the preferences of the recipient and the service provider policies with the possibility to extend the expiry time.
- For multiple Devices handling:
 - Supporting delivery of Media Streams to more than one Device for a CPM User. If there are multiple Media Streams, each Media Stream can be delivered to the same or different set of Devices.
 - Supporting the transfer of a CPM Session between Devices of a CPM User.
 - Supporting the transfer of a subset of the Media Streams of a CPM Session to a different Device of a CPM User.
 - Supporting request from a CPM Client to access information related to the Devices of the CPM User e.g. to determine the list of CPM Sessions per registered Device.
 - Providing information about ongoing CPM Sessions (e.g. the Media Stream Types of the used Media Streams) of the CPM Clients of the same CPM User
- Providing support for capabilities for lawful interception
- Acting as Presence Source and Presence Watcher.
- Providing support for Malware detection

In addition to the supporting Enablers described in section 5.3.1.3, in the future the CPM Participating Function may interact with the following supporting Enablers:

- The Charging Enabler for the submission of Charging Events for both Online and Offline Charging.

- The Presence Enabler for the retrieval of a CPM User's Presence Information and the publishing of Presence Information on behalf of a CPM User.

C.4.3 CPM Controlling Function

In addition to the functionalities described in section 5.3.1.4, in the future the CPM Controlling Function may support the following functionalities:

- Providing support for Online and Offline Charging functions for CPM Sessions and CPM Standalone Messages via the Charging Enabler.
- For session handling:
 - Providing support for Media Stream control handling and Media Stream control negotiation on a per Media Stream basis (floor control) for selected Media Stream.
- Providing criteria-based search for ongoing CPM Group Sessions.
- Providing "hidden mode" capability in a CPM Group Session based on service provider policies.

In addition to the supporting Enablers described in section 5.3.1.4, in the future the CPM Participating Function interacts with the following supporting Enablers:

- The Charging Enabler for the submission of Charging Events for both Online and Offline Charging.

C.4.4 Message Storage Server

In addition to the functionalities described in section 5.3.1.7, in the future the Message Storage Server may support the following functionalities:

- Authentication of the requesting remote environment trusted server;
- Support of notification of changes to the privileges associated with the folders, when notifications are enabled by the CPM User or service provider policies
- Recording actions (e.g. uploaded/downloaded/modified/removed specific items such as CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories and Media Objects that are part of CPM Messages, CPM File Transfer Histories or CPM Session Histories) being performed on a Principal's network-based storage, based on service provider policies and CPM User preferences
- Routing requests from served home CPM environment Message Storage Clients for access to resources of a remote CPM environment to the remote CPM environment Message Storage Servers.

In addition to the supporting Enablers described in section 5.3.1.7, in the future the Message Storage Server may interact with the following supporting Enablers:

- The Charging Enabler for the submission of Charging Events for both Online and Offline Charging.

C.5 Additional CPM Enabler Functional Components

The CPM Enabler functional components described in the following sub-sections may be added to the CPM architecture model to satisfy the future requirements.

C.5.1.1 Content Storage Client

The Content Storage Client manages standalone Media Objects stored at Content Storage Server in the network as well as stored locally. The other functional components in the Device may communicate internally with the Content Storage Client; see sections 5.3.1.1 and 5.3.1.2.

NOTE: The management of the local storage of a Device is out-of-scope for standardisation.

The Content Storage Client may support the following data management functionalities:

- Managing and consuming resources stored at the CPM User's own Content Storage Server
 - Retrieving and deleting standalone Media Objects (or previews) from the Content Storage Server
 - Uploading standalone Media Objects to the Content Storage Server
 - Creating (and deleting) folders and subfolders
 - Moving and copying standalone Media Objects between folders
 - Creating and updating metadata of standalone Media Objects and folders, including access policy settings
 - Obtaining the list of stored standalone Media Objects and folders residing in the Content Storage Server, with or without filtering criteria
 - Synchronizing (e.g. periodically and/or partially) the resources in the local storage of a CPM User's Device(s) with the Content Storage Server, including proper handling of locally deleted items
 - Indicating to Content Storage Server to download and store a Media Object attached to a CPM Message, a CPM File Transfer History or a CPM Session History from a Message Storage Server
- Managing and consuming (e.g. retrieving, previewing, viewing) resources from other CPM User's Content Storage Server based on received information and access right to that data.

The Content Storage Client also may support local deletion, i.e. deletion of standalone Media Objects from the local storage of the Device of a CPM User while retaining them on the Content Storage Server in the network.

C.5.1.2 Content Storage Server

The following resources, stored on the Content Storage Server, can be accessed individually for fulfillment of specific resource management actions:

- Folders
- Standalone Media Objects

The Content Storage Server may support the following functionalities:

- Authentication of the locally recognized User address;
NOTE: Locally recognized User address is any address known by the Content Storage Server and is independent of the access method.
- Authentication of the requesting remote environment trusted server;
- Storage of standalone Media Objects
- Synchronization of stored standalone Media Objects with the Content Storage Client's local storage of the standalone Media Objects including synchronization management under scenarios with multiple Devices, according to the CPM User's preferences and/or the service provider's policy
- Support of management of access control lists over stored resources and notification of changes to the resources or to the privileges associated with the resources, when notifications are enabled by the CPM User or service provider policies
- Management (e.g. copy, delete, ...) of stored resources and access to those resources
- Searching stored resources
- Recording actions (e.g. uploaded/downloaded/modified/removed standalone Media Objects) being performed on a Principal's network-based storage, based on service provider policies and CPM User preferences

- Enforcement of quota and notification of overflows, when notifications are enabled by the CPM User or service provider policies
- Routing requests from served home CPM environment Content Storage Clients for access to resources of a remote CPM environment to the remote CPM environment Content Storage Servers.
- Downloading and storing of a Media Object attached to a CPM Message, a CPM File Transfer History or a CPM Session History from a Message Storage Server

The stored resources can be subject to access control lists of Principals, defined and managed by the CPM User, which associate Principals with all or a subset of the following privileges:

- copy
- move between folders
- rename
- read / access
- write / modify
- delete

Folders are subject to an additional privilege that allows their creation and the listing of their content, with or without filtering criteria.

Privileges associated to a resource can be subject to an expiration time after which the privilege is revoked.

The Content Storage Server may support content adaptation (i.e. for previewing) of stored standalone Media Objects before delivering to the Device's local storage, based on recipient's preferences, Communication Capabilities, and/or service provider policies

The Content Storage Server interacts with other CPM functional components such as:

- the CPM Participating Function:
 - for the composition of CPM Messages without requiring download of selected stored resources to the Content Storage Client (via the CPM Participating Function)
- the Content Storage Client for
 - Synchronization (e.g. periodically and/or partially) with the local storage on the Device(s) of a CPM User
 - Uploading a standalone Media Object or a set of those from the Device's local storage
 - Downloading a standalone Media Object (or a preview of a standalone Media Object) or a set of those to the Device's local storage
 - Management of stored resources and folders (e.g. copy, move between folders, rename, read / access, write / modify, delete, manage access control lists)
 - Listing stored standalone Media Objects residing in the Content Storage Server, with or without filtering criteria
 - Providing references to a standalone Media Object residing in the Content Storage Server
- the Message Storage Server for
 - Downloading and storing of a Media Object attached to a CPM Message, a CPM File Transfer History or a CPM Session History

The Content Storage Server interacts with the following supporting Enablers:

- The Charging Enabler for the submission of Charging Events for both Online and Offline Charging.
- The Push Enabler for notifications delivery (e.g. quota overflows).

C.5.1.3 Application Control Function

The Application Control Function provides access to functionalities of the CPM Controlling Function and the CPM Participating Function towards external functional components, for example VAS Applications.

The Application Control Function may support the following functionalities on behalf of the CPM Controlling Function and the CPM Participating Function:

- Exposure of the functionalities provided by the CPM Controlling Function and the CPM Participating Function;
- Authentication of the functional components requesting usage of exposed functionalities;
- Authorization of the functional components requesting usage of exposed functionalities according to service provider policies;
- Relaying of events generated by the CPM Controlling Function and the CPM Participating Function towards the VAS Applications, if applicable;
- Providing anonymity to a CPM User's identity from specially identified VAS Applications, if applicable;
- Handling and relay of VAS Application requests to the CPM Controlling Function and the CPM Participating Function.
- Protocol translation between CPM-VAS and CPM-PF1 interfaces.
- Protocol translation between CPM-VAS and CPM-CF interfaces.

The Application Control Function interacts with the Charging Enabler for the submission of Charging Events for both Online and Offline Charging.

C.6 Additions to Existing External Functional Components

C.6.1 Presence Enabler

In addition of the functionalities described in section 5.3.2.2, in the future the CPM Enabler may also use the functionalities of the Presence Enabler to publish Presence Information.

In addition to the referene points described in section 5.3.2.2, in the future the CPM functional components may also interact with the Presence enabler via the PRS-1 reference point defined by the Presence Enabler.

C.7 Additional External Functional Components

The external functional components described in the following sub-sections may be added to the CPM architecture model to satisfy the future requirements.

C.7.1 Charging Enabler

The Charging Enabler can perform the following functions that are needed for the support of the CPM Enabler:

- Dealing with Charging Events created by CPM Enabler for different charging models

The Charging Enabler provides the functionality as specified in [OMA-CHRG-AD].

The CPM functional components interact with the Charging Enabler via the CH-1 and CH-2 interfaces defined by the Charging Enabler.

C.7.2 Converged Address Book Enabler

The CAB Enabler provides users with the possibility to store and manipulate address book information. The functionalities provided by the CAB Enabler are specified in [OMA-CAB-AD].

C.7.3 VAS Applications

The VAS Applications are applications that are resident in the network, and interact with the CPM Controlling Function or the CPM Participating Function in order to access functionalities and related information in the CPM Enabler.

C.8 Additions to Existing Interfaces

The following subsections define functionalities of existing and possibly new CPM Enabler interfaces to satisfy future requirements of the CPM Enabler.

C.8.1 CPM-MSG

The supported functionalities of the CPM-MSG interface, in the future, might also include:

- Authentication of the requesting remote environment trusted server.

C.9 Additional Interfaces

C.9.1 CPM-CNT

The CPM-CNT interface might be exposed by the Content Storage Server to allow other functional components (such as the Content Storage Client, the CPM Participating Function or a remote environment trusted server) to access the Content Storage Server.

Supported functionalities include:

- Authentication of the locally recognized User address.
- Authentication of the requesting remote environment trusted server.
- Uploading and downloading of standalone Media Objects.
- Retrieving a preview of a standalone Media Object.
- Deletion of standalone Media Objects from the Content Storage Server.
- Renaming and copying of standalone Media Objects in a folder in the Content Storage Server.
- Creation, deletion, moving, and renaming of folders in the Content Storage Server.
- Moving and copying of standalone Media Objects between folders in the Content Storage Server.
- Searching within accessible folders in the Content Storage Server.
- Synchronisation between the Content Storage Client's local storage and the Content Storage Server.
- Obtaining a reference to a stored object, or obtaining a stored object using a reference in the Content Storage Server (e.g. to be used for "forward without download" functionality).
- Management of access permissions to specific items in the Content Storage Server.

The protocols used for the CPM-CNT interface might be WebDAV and OMA DS. WebDAV may be used as the base protocol for the individual Media Object manipulation operations. OMA DS may be used for synchronisation between the Content Storage Client's local storage and the Content Storage Server.

C.9.2 CPM-VAS

The CPM-VAS interface might be exposed by the Application Control Function to allow CPM Participating Function and CPM Controlling Function functionality to be accessible by external functional components (such as VAS Applications).

Supported functionalities include:

- Authentication
- CPM Conversation management (e.g. replaying the recent history of a conversation)
- CPM Session management (e.g. starting/stopping a CPM Session, listing /searching ongoing CPM Sessions and associated Participants, add / remove Media Streams to / from a CPM Session), including on behalf of a CPM User
- CPM File Transfers inside/outside a CPM Session, including on behalf of a CPM User
- CPM Message exchanges inside/outside a CPM Session, including on behalf of a CPM User
- Media moderation controlled by the external functional component during a CPM Session
- Sending of event notification with relevant information (e.g. User causing the event, type of event)
- Anonymity indication of a CPM User.

The CPM-VAS interface might be based on web services, and may comply with the rules for web-services based interfaces within OMA as described in [OMA-OWSER].

C.10 Additional Deployment Scenarios

C.10.1 Additional Deployment Configuration

C.10.1.1 Deployment without Content Storage

This subsection shows a deployment configuration for a Service Provider that doesn't want to provide network-based content storage to its users, or wants to offer network-based content storage functionality via a non-CPM content storage solution.

This deployment configuration is nearly the same as the architecture shown in section 5.2; however, the Content Storage Client and the Content Storage Server functional components (and their associated interfaces) are not deployed. **Error! Reference source not found.**Figure 14 shows this deployment configuration.

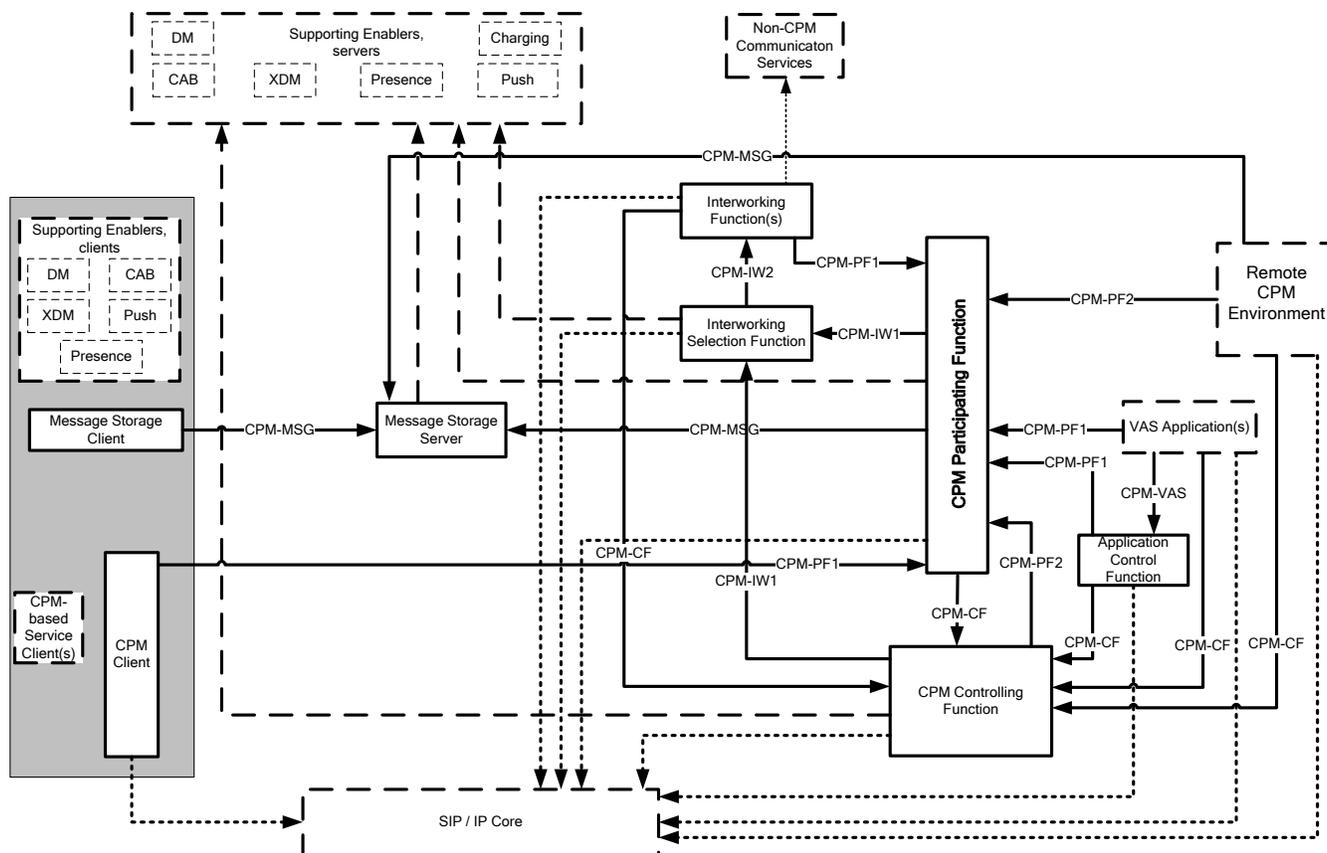


Figure 14: Deployment configuration without Content Storage

C.10.2 Additional Co-Location Deployment Scenarios

C.10.2.1 Application Control Function

The CPM architecture as depicted in Figure 3 shows the Application Control Function as an independent logical functional component. However this is not the only deployment scenario. This section describes the deployment scenarios for the deployment of the Application Control Function:

1. The Application Control Function is deployed as a separate physical component
2. The Application Control Function is deployed co-located with the CPM Participating Function

For simplicity, the following text assumes that the CPM Controlling Function is deployed co-located with the CPM Participating Function. When CPM Controlling Function and CPM Participating Function are not co-located, the below scenarios unfold further.

C.7.2.1.1 Application Control Function as a Separate Physical Component

In this deployment the Application Control Function is deployed as a separate physical component. It provides the CPM-VAS interface to external VAS Applications for bidirectional communication with applications. The Application Control Function uses the CPM-PF1 interface offered by the CPM Participating Function for relaying the events to this functional component.



Figure 15: Application Control Function deployed as a Separate Physical Component.

C.7.2.1.2 Application Control Function Co-located with the CPM Participating Function

In this deployment the Application Control Function is part of the CPM Participating Function. In this case the external VAS Applications interact directly with the CPM Participating Function (which includes the Application Control Function) via the CPM-VAS interface.

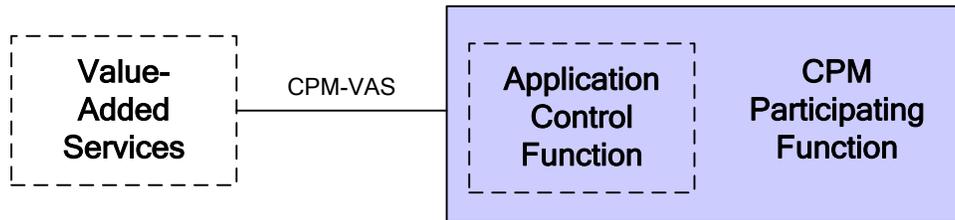


Figure 16: Application Control Function Co-located with the CPM Participating Function.