



# OMA Converged IP Messaging System Description

## Candidate Version 2.0 – 11 Jun 2013

---

**Open Mobile Alliance**  
OMA-TS-CPM\_System\_Description-V2\_0-20130611-C

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.**

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.

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

# Contents

<b>1. SCOPE</b> .....	<b>7</b>
<b>2. REFERENCES</b> .....	<b>8</b>
2.1 <b>NORMATIVE REFERENCES</b> .....	<b>8</b>
2.2 <b>INFORMATIVE REFERENCES</b> .....	<b>9</b>
<b>3. TERMINOLOGY AND CONVENTIONS</b> .....	<b>10</b>
3.1 <b>CONVENTIONS</b> .....	<b>10</b>
3.2 <b>DEFINITIONS</b> .....	<b>10</b>
3.3 <b>ABBREVIATIONS</b> .....	<b>11</b>
<b>4. INTRODUCTION</b> .....	<b>13</b>
4.1 <b>CPM VERSION 1.0</b> .....	<b>13</b>
4.2 <b>CPM VERSION 2.0</b> .....	<b>14</b>
<b>5. CPM FUNCTIONAL CONCEPTS</b> .....	<b>16</b>
<b>5.1 GENERAL CONCEPTS</b> .....	<b>16</b>
5.1.1 Identification.....	16
5.1.2 Signalling over the SIP/IP Core.....	17
5.1.3 Security .....	18
5.1.4 Registration Event Information Feature.....	19
<b>5.2 CPM CONVERSATION</b> .....	<b>19</b>
5.2.1 CPM Conversation Concept.....	19
5.2.2 CPM Messaging Concepts.....	22
5.2.3 CPM Session Concepts.....	26
5.2.4 CPM File Transfer Concepts.....	40
5.2.5 Handling of Deferred CPM Messages .....	43
5.2.6 Disposition Notification.....	44
<b>5.3 CPM INTERWORKING</b> .....	<b>46</b>
5.3.1 Interworking from CPM to Non-CPM.....	46
5.3.2 Interworking from Non-CPM to CPM.....	50
<b>5.4 CPM USER PREFERENCES</b> .....	<b>51</b>
<b>5.5 CPM MESSAGE STORAGE</b> .....	<b>53</b>
5.5.1 Storage Model.....	53
5.5.2 Operations.....	57
<b>APPENDIX A. CHANGE HISTORY (INFORMATIVE)</b> .....	<b>64</b>
<b>A.1 APPROVED VERSION HISTORY</b> .....	<b>64</b>
<b>A.2 DRAFT/CANDIDATE VERSION 2.0 HISTORY</b> .....	<b>64</b>
<b>APPENDIX B. CPM CONVERSATION FLOWS (INFORMATIVE)</b> .....	<b>65</b>
<b>B.1 REGISTRATION</b> .....	<b>65</b>
B.1.1 CPM Client Registering.....	65
B.1.2 Registration Event Information Flows.....	65
B.1.2.1 CPM Participating Function Subscribing to Registration Event Information.....	65
<b>B.2 CPM SESSIONS</b> .....	<b>67</b>
B.2.1 CPM Session Initiation .....	67
B.2.1.1 CPM 1-1 Session Invitation .....	67
B.2.1.2 CPM Group Session Invitation .....	68
B.2.2 Adding CPM Users to an Existing CPM Session .....	71
B.2.2.1 Extending a CPM 1-1 Session to a CPM Group Session .....	71
B.2.2.2 Adding Participants to an Existing CPM Group Session .....	72
B.2.3 CPM Session Termination .....	72
B.2.3.1 CPM User Closing a CPM 1-1 Session.....	72
B.2.3.2 Participant Leaving a CPM Group Session.....	73
B.2.3.3 Removing Participants from a CPM Group Session .....	74
B.2.3.4 Participant Being Released from a CPM Group Session .....	75

- B.2.3.5 Long-lived CPM Group Session .....76
- B.2.3.5.1 Basic flow – Long-lived Group Session re-started with same focus.....76
- B.2.3.5.2 Basic flow – new focus .....77
- B.2.3.6 CPM User re-joins an active CPM Group Session .....79
- B.2.3.6.1 Client re-joins after the MSRP session was previously expired in the SIP/IP Core (timeout).....79
- B.2.3.7 Deferred delivery of CPM Group Session .....81
- B.2.3.7.1 Deferred delivery of CPM Group Session while session is inactive.....82
- B.2.3.7.2 PF initiates delivery of CPM Group Session, while the session is still on-going .....83
- B.2.3.7.3 CPM Client B requests to re-join the active CPM Group Session .....84
- B.2.3.7.4 Race conditions provisions .....86
- B.2.4 CPM Session Modification .....86
- B.2.4.1 Modifying Media Streams in a CPM 1-1 Session .....86
- B.2.4.2 Modifying Media Streams in a CPM Group Session .....87
- B.2.5 Participant Information .....89
- B.2.5.1 Participant Information Subscription .....89
- B.2.5.2 Participant Information Notification .....90
- B.3 CPM STANDALONE MESSAGES .....91**
- B.3.1 Pager Mode CPM Standalone Message related flows.....91
- B.3.1.1 Sending a Pager Mode CPM Standalone Message to a Single Recipient .....91
- B.3.1.2 Pager Mode CPM Standalone Message to a CPM Group.....96
- B.3.1.3 Forwarding/Including Data Stored in Message Storage.....99
- B.3.1.4 Indirect Media Delivery .....100
- B.3.2 Large Message Mode CPM Standalone Message Flows .....101
- B.3.2.1 Sending a Large Message Mode CPM Standalone Message .....101
- B.3.2.2 Extending to a Large Message Mode CPM Standalone Message after Fetching Media.....106
- B.3.3 Deferred CPM Message Related Flows .....107
- B.3.3.1 Deferred CPM Message Pushing .....107
- B.3.3.2 Deferred CPM Message Pulling .....108
- B.4 CPM FILE TRANSFER FLOWS .....109**
- B.4.1 Transferring a File.....109
- B.5 OUTBOUND INTERWORKING.....112**
- B.5.1 Pager Mode CPM Standalone Message Interworking in the Originating Network .....112
- B.5.2 Pager Mode CPM Standalone Message Interworking in the Terminating Network .....114
- B.5.3 Large Message Mode CPM Standalone Message Interworking .....116
- B.5.4 CPM File Transfer Interworking.....118
- B.5.4.1 File Transfer supported by Non-CPM Communication Service .....118
- B.5.4.2 File Transfer not supported by Non-CPM Communication Service .....119
- B.5.5 CPM Session Interworking .....120
- B.5.5.1 Interworking with a Single Interworking Function .....120
- B.5.5.2 Interworking with Multiple Interworking Functions.....121
- B.5.6 CPM Session Modification .....123
- B.5.6.1 Adding a New Media Stream.....123
- B.6 INBOUND INTERWORKING .....125**
- APPENDIX C. MESSAGE STORAGE FLOWS (INFORMATIVE) .....127**
- C.1 AUTHENTICATION .....127
- C.2 SELECTING THE ACTIVE FOLDER.....127
- C.3 STORING A CPM STANDALONE MESSAGE.....128
- C.4 RETRIEVING A STORED OBJECT.....128
- APPENDIX D. HANDLING OF ISF RELEVANT CRITERIA (INFORMATIVE) .....130**

## Figures

- Figure 1: Schematic to illustrate the modularity and extensibility of the CPM Enabler. ....13**

- Figure 2: Example of Default Folder Storage Model of the CPM 2.0 Message Storage.....54**

Figure 3: Example of Optional User Folder Storage Model of the CPM 2.0 Message Storage .....	55
Figure 4: CPM Client registration .....	65
Figure 5: CPM Participating Function subscribes to the Registration Event Information .....	66
Figure 6: CPM 1-1 Session, originating and terminating parts .....	67
Figure 7: CPM Group Session setup, originating part .....	69
Figure 8: CPM Group Session setup, terminating side, multiple CPM Clients .....	70
Figure 9: Extending a CPM 1-1 Session to a CPM Group Session.....	71
Figure 10: CPM Client adding users to an existing CPM Group Session.....	72
Figure 11: Closing a CPM 1-1 Session .....	73
Figure 12: CPM Client leaving a CPM Group Session .....	74
Figure 13: CPM Client removing other Participants from CPM Session.....	75
Figure 14: CPM Client being released from CPM Session .....	75
Figure 15: CPM Client re-starting a Long-lived CPM Group Session with same focus .....	77
Figure 16: CPM Client re-starting a Long-lived CPM Group Session with a new focus .....	78
Figure 17: CPM Client re-joining an active CPM Group Session – expiry at SIP/IP Core .....	80
Figure 18: Delivery, CPM Client re-joining an inactive CPM Group Session .....	82
Figure 19: CPM 1-1 Session modification.....	87
Figure 20: Modifying Media Streams in a CPM Group Session, originating side .....	88
Figure 21: Modifying Media Streams in a CPM Group Session, terminating side.....	89
Figure 22: CPM Client subscribes to Participant Information of a CPM Session.....	90
Figure 23: CPM Client receives Participant Information of a CPM Session .....	91
Figure 24: Sending a CPM Standalone Message to a CPM User (originating side) .....	92
Figure 25: Sending a CPM Standalone Message to a CPM User (terminating side) .....	94
Figure 26: Unsuccessful delivery of CPM Standalone Message with a delivery notification request .....	96
Figure 27: Pager Mode CPM Standalone Message to a CPM Group (originating).....	97
Figure 28: Pager Mode CPM Standalone Message to a CPM Pre-defined Group (terminating).....	98
Figure 29: Forwarding/including data stored in Message Storage Server .....	100
Figure 30: Indirect media delivery.....	101
Figure 31: Sending a CPM Standalone Message in Large Message Mode to a CPM User (originating side) .....	102
Figure 32: Sending a CPM Standalone Message in Large Message Mode to a CPM User (terminating side).....	104
Figure 33: Extending to a Large Message Mode CPM Standalone Message in CPM Participating Function.....	106
Figure 34: Deferred CPM Message Pushing .....	107

Figure 35: Deferred CPM Message pulling ..... 108

Figure 36: Sending a CPM File Transfer to a CPM User (originating side) ..... 109

Figure 37: Sending a File Transfer to a CPM User (terminating side) ..... 111

Figure 38: Interworking at the originating network ..... 113

Figure 39: Interworking based on recipient’s user preferences at the terminating network..... 115

Figure 40: Large Message Mode CPM Standalone Message Interworking ..... 117

Figure 41: CPM File Transfer Interworking to non-CPM file transfer ..... 118

Figure 42: CPM File Transfer Interworking to non-CPM message..... 119

Figure 43: CPM Session Interworking with a single Interworking Function ..... 120

Figure 44: CPM Session Interworking with multiple Interworking Function ..... 122

Figure 45: Adding a new Media Stream to the ongoing CPM Session..... 124

Figure 46: Non-CPM message to CPM User ..... 125

Figure 47: Authentication to Message Storage Server ..... 127

Figure 48: Selection the active folder on the Message Storage Server ..... 127

Figure 49: Storing a CPM Standalone Message on the Message Storage Server ..... 128

Figure 50: Retrieving a stored object from the Message Storage Server..... 129

## Tables

Table 1: Example routing table ..... 131

# 1. Scope

This document describes the system concepts of the CPM Enabler to fulfil the requirements outlined in the Converged IP Messaging requirements document [OMA-CPM-RD] for CPM V2.0 in compliance to the architecture described in [OMA-CPM-AD]. The description of the system concepts comprises the definition of central data elements of the CPM Enabler and the specification of general procedures for each of the functionalities supported by the CPM Enabler.

Additionally, this document describes a set of detailed call flows for the functionalities of the CPM Enabler. These call flows explain the system concepts in a graphical manner, and describe the relationships between the functional components of the CPM Enabler.

## 2. References

### 2.1 Normative References

- [3GPP TS24.229] "Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3", 3GPP, TS24.229, URL:<http://www.3gpp.org/>
- [3GPP TS33.203] "3G security; Access security for IP-based services", 3GPP, TS33.203, URL:<http://www.3gpp.org/>
- [3GPP2 S.R0086-0] "IMS Security Framework", 3GPP2, S.R0086-0, URL:<http://www.3gpp2.org/>
- [3GPP2 X.S0013-004] "All-IP Core Network Multimedia Domain - IP Multimedia Call Control Protocol Based on SIP and SDP Stage 3", 3GPP2, X.S0013-004-A, URL:<http://www.3gpp2.org/>
- [IETF-DRAFT-Chat] Multi-party Chat Using the Message Session Relay Protocol, Version 14, March 2, 2012, URL:<http://tools.ietf.org/html/draft-ietf-simple-chat-14>
- [OMA-CPM-AD] "Converged IP Messaging Architecture", Open Mobile Alliance™, OMA-AD-CPM-V2\_0, URL:<http://www.openmobilealliance.org/>
- [OMA-CPM-MSG] "CPM Message Storage", Open Mobile Alliance™, OMA-TS-CPM\_MessageStorage-V2\_0, URL:<http://www.openmobilealliance.org/>
- [OMA-CPM-RD] "Converged IP Messaging Requirements", Open Mobile Alliance™, OMA-RD-CPM-V2\_0, URL:<http://www.openmobilealliance.org/>
- [OMA-IMPS\_CSP-TS] "OMA IMPS Client-Server Protocol Transport Bindings", Open Mobile Alliance™, OMA-TS-IMPS\_CSP\_Transport-V1\_3, URL:<http://www.openmobilealliance.org/>
- [OMA\_SEC\_CF-AD] "Security Common Functions Architecture", Open Mobile Alliance™, OMA-AD-SEC\_CF-V1\_0, URL:<http://www.openmobilealliance.org/>
- [OMA-XDM-AD] "XML Document Management (XDM) Architecture", Open Mobile Alliance™, OMA-AD-XDM-V2\_1, URL:<http://www.openmobilealliance.org/>
- [OMA-XDM\_Core-TS] "XML Document Management (XDM) Specification", Open Mobile Alliance™, OMA-TS-XDM\_Core-V2\_1, URL:<http://www.openmobilealliance.org/>
- [OMA-XDM\_Group-TS] "Group XDM Specification", Open Mobile Alliance™, OMA-TS-XDM\_Group-V1\_1, URL:<http://www.openmobilealliance.org/>
- [OMA-XDM-Policy] "Policy XDM Specification"; Open Mobile Alliance™, OMA-TS-XDM\_Policy-V1\_1, URL:<http://www.openmobilealliance.org/>
- [OMA-XDM-UPP] "UPP Directory XDM Specification"; Open Mobile Alliance™, OMA-TS-XDM\_UPP\_Directory-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>
- [RFC2141] "URN Syntax", R. Moats, May 1997, URL:<http://www.ietf.org/rfc/rfc2141.txt>
- [RFC2246] "The TLS Protocol, version 1.0", T. Dierks et al, January 1999, URL:<http://www.ietf.org/rfc/rfc2246.txt>
- [RFC2368] "The mailto URL scheme", P. Hoffman et al, July 1998, URL:<http://www.ietf.org/rfc/rfc2368.txt>
- [RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee et al, August 1998, URL:<http://www.ietf.org/rfc/rfc2396.txt>
- [RFC3261] "SIP: Session Initiation Protocol", U. Rosenberg et al, June 2002, URL:<http://www.ietf.org/rfc/rfc3261.txt>
- [RFC3264] "An Offer/Answer Model with the Session Description Protocol (SDP)", J. Rosenberg et al, June 2002, URL:<http://www.ietf.org/rfc/rfc3264.txt>
- [RFC3320] "Signalling Compression(SigComp)", R. Price et al, January 2003, URL:<http://www.ietf.org/rfc/rfc3320.txt>
- [RFC3321] "Signaling Compression (SigComp) - Extended Operations", H. Hannu et al, January 2003, URL:<http://www.ietf.org/rfc/rfc3321.txt>
- [RFC3485] "The Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Static Dictionary for Signaling Compression (SigComp)", M. Garcia-Martin et al, February 2003,

- URL:<http://www.ietf.org/rfc/rfc3485.txt>
- [RFC3486] "Compressing the Session Initiation Protocol (SIP)", G. Camarillo, February 2003, URL:<http://www.ietf.org/rfc/rfc3486.txt>
- [RFC3711] "The Secure Real-time Transport Protocol (SRTP)", M. Baugher et al, March 2004, URL:<http://www.ietf.org/rfc/rfc3711.txt>
- [RFC3840] " Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", J. Rosenberg et al, August 2004, URL:<http://www.ietf.org/rfc/rfc3840.txt>
- [RFC3966] "The tel URI for Telephone Numbers", H. Schulzrinne, December 2004, URL:<http://www.ietf.org/rfc/rfc3966.txt>
- [RFC4122] "A Universally Unique Identifier (UUID) URN Namespace", P. Leach et al, July 2005, URL:<http://www.ietf.org/rfc/rfc4122.txt>
- [RFC4279] "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", P. Eronen et al, December 2005, URL:<http://www.ietf.org/rfc/rfc4279.txt>
- [RFC4483] "A Mechanism for Content Indirection in Session Initiation Protocol(SIP) Messages", E. Burger, May 2006, URL:<http://www.ietf.org/rfc/rfc4483.txt>
- [RFC4566] "SDP: Session Description Protocol", M. Handley et al, July 2006, URL:<http://www.ietf.org/rfc/rfc4566.txt>
- [RFC 4579] "Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", A. Johnston et al, August 2006, URL:<http://www.ietf.org/rfc/rfc4579.txt>
- [RFC5547] "A Session Description Protocol (SDP) Offer/Answer Mechanism to Enable File Transfer," M. Garcia-Martin et al, May 2009, URL:<http://tools.ietf.org/html/rfc5547>
- [RFC5627] "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", J. Rosenberg, October 2009, URL:<http://www.ietf.org/rfc/rfc5627.txt>
- [SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR\_Rules\_and\_Procedures, URL:<http://www.openmobilealliance.org/>

## 2.2 Informative References

- [OMA-DICT] "Dictionary for OMA Specifications", Open Mobile Alliance™, OMA-ORG-Dictionary-V2\_9, 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-SIMPLE-IM] "Instant Messaging", Open Mobile Alliance™, OMA-ERELED-SIMPLE\_IM-V1\_0, URL: <http://www.openmobilealliance.org/>

## 3. Terminology and Conventions

### 3.1 Conventions

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

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

### 3.2 Definitions

<b>CPM 1-1 Session</b>	A CPM Session established between two Participants
<b>CPM-based Service</b>	See [OMA-CPM-AD].
<b>CPM-based Service Identifier</b>	A unique identification of a CPM-based Service.
<b>CPM Ad-hoc Group</b>	See [OMA-CPM-RD].
<b>CPM Address</b>	See [OMA-CPM-RD].
<b>CPM Chat Message</b>	See [OMA-CPM-RD].
<b>CPM Client</b>	See [OMA-CPM-AD].
<b>CPM Closed Group Session</b>	See [OMA-CPM-RD].
<b>CPM Contribution Identity</b>	A unique identification of a CPM Standalone Message or of a CPM Session belonging to a CPM Conversation.
<b>CPM Controlling Function</b>	See [OMA-CPM-AD].
<b>CPM Conversation</b>	See [OMA-CPM-RD].
<b>CPM Conversation History</b>	See [OMA-CPM-RD].
<b>CPM Conversation Identity</b>	A unique identification of a CPM Conversation.
<b>CPM Feature Tag</b>	A piece of information in CPM service control messages indicating the usage of CPM Enabler services and capabilities of the addressed entity or entities.
<b>CPM File Transfer</b>	See [OMA-CPM-RD].
<b>CPM File Transfer History</b>	See [OMA-CPM-RD].
<b>CPM Group Session</b>	See [OMA-CPM-RD].
<b>CPM Group Session Identity</b>	A SIP URI identifying the CPM Group Session.
<b>CPM Long-lived Session</b>	See [OMA-CPM-RD].
<b>CPM Message</b>	See [OMA-CPM-RD].
<b>CPM Participating Function</b>	See [OMA-CPM-AD].
<b>CPM Pre-defined Group</b>	See [OMA-CPM-RD].
<b>CPM Session</b>	See [OMA-CPM-RD].
<b>CPM Session Invitation</b>	See [OMA-CPM-RD].
<b>CPM Standalone Message</b>	See [OMA-CPM-RD].
<b>CPM User</b>	See [OMA-CPM-RD].
<b>Deferred CPM Message</b>	See [OMA-CPM-RD].
<b>Device</b>	See [OMA-DICT].
<b>Enabler</b>	See [OMA-DICT].

<b>File Transfer Thumbnail</b>	A small image representing the contents of the CPM File Transfer and of CPM File Transfer History.  For example, the thumbnail for an audio clip might be album cover art; the thumbnail for a video clip could be a snapshot of the key frame at time mark 00:30 seconds; the thumbnail for a well known document type could be a related icon or graphic.
<b>Group XDMS</b>	See [OMA-XDM-AD].
<b>Group State Object</b>	Information related to a CPM Group Session containing Participants Information, type of CPM Group Session, and other relevant information.
<b>Instance ID</b>	See [3GPP TS24.229].
<b>Interworking Function</b>	See [OMA-CPM-AD].
<b>Interworking Selection Function</b>	See [OMA-CPM-AD].
<b>Join-in Group</b>	See [OMA-XDM_Group-TS].
<b>Large Message Mode</b>	See [OMA-CPM-AD].
<b>Media</b>	See [OMA-CPM-RD].
<b>Media Object</b>	See [OMA-CPM-AD].
<b>Media Plane</b>	See [OMA-CPM-AD].
<b>Media Stream</b>	As defined in [RFC3264].
<b>Media Stream Type</b>	See [OMA-CPM-AD].
<b>Media Type</b>	See [OMA-DICT].
<b>Message Storage Client</b>	See [OMA-CPM-AD].
<b>Message Storage Server</b>	See [OMA-CPM-AD].
<b>Non-CPM Communication Service</b>	See [OMA-CPM-RD].
<b>Non-CPM Communication Service Identifier</b>	An identifier identifying a Non-CPM Communication Service in the service provider network.
<b>Pager Mode</b>	See [OMA-CPM-AD].
<b>Participant</b>	See [OMA-DICT].
<b>Participant Information</b>	Information about the Participants of a CPM Session.
<b>Pre-arranged Group</b>	See [OMA-XDM_Group-TS]
<b>Pseudonym</b>	See [OMA-CPM-RD].
<b>Registration Event Information</b>	Information about registered clients and their user agent capabilities.
<b>Reply-to Indication</b>	An indication in a CPM Standalone Message determining to which CPM Address a potential reply to the CPM Standalone Message has to be sent to.
<b>Signalling Path</b>	Routing path of the SIP request or reverse routing path of the SIP request.  NOTE: The reverse routing path of the SIP request is used e.g. when an invited CPM Client releases a SIP session.
<b>Unique User Agent Identifier</b>	A globally unique persistent non routable instance identifier of a user agent.
<b>User</b>	See [OMA-DICT].

### 3.3 Abbreviations

<b>3GPP</b>	See [OMADICT]
<b>CAB</b>	See [OMADICT]

<b>CPM</b>	See [OMADICT]
<b>GRUU</b>	Globally Routable User agent URI
<b>IMS</b>	See [OMADICT]
<b>IP</b>	See [OMADICT]
<b>MMS</b>	See [OMADICT]
<b>OMA</b>	See [OMADICT]
<b>PSK</b>	Pre-Shared Key
<b>RFC</b>	See [OMADICT]
<b>SCR</b>	Static Conformance Requirements
<b>SIP</b>	See [OMADICT]
<b>SMS</b>	See [OMADICT]
<b>SRTP</b>	Secure Real-time Transport Protocol
<b>TLS</b>	See [OMADICT]
<b>UA</b>	User Agent
<b>URI</b>	See [OMADICT]
<b>URL</b>	See [OMADICT]
<b>URN</b>	See [OMADICT]
<b>XDM</b>	See [OMADICT]
<b>XML</b>	See [OMADICT]

Note: Abbreviations defined in the OMA Dictionary complements this section.

## 4. Introduction

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.

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 of these Non-CPM Communication Services 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 1 below shows how the CPM Enabler can be extended with CPM-based Services and Non-CPM Communication Services.

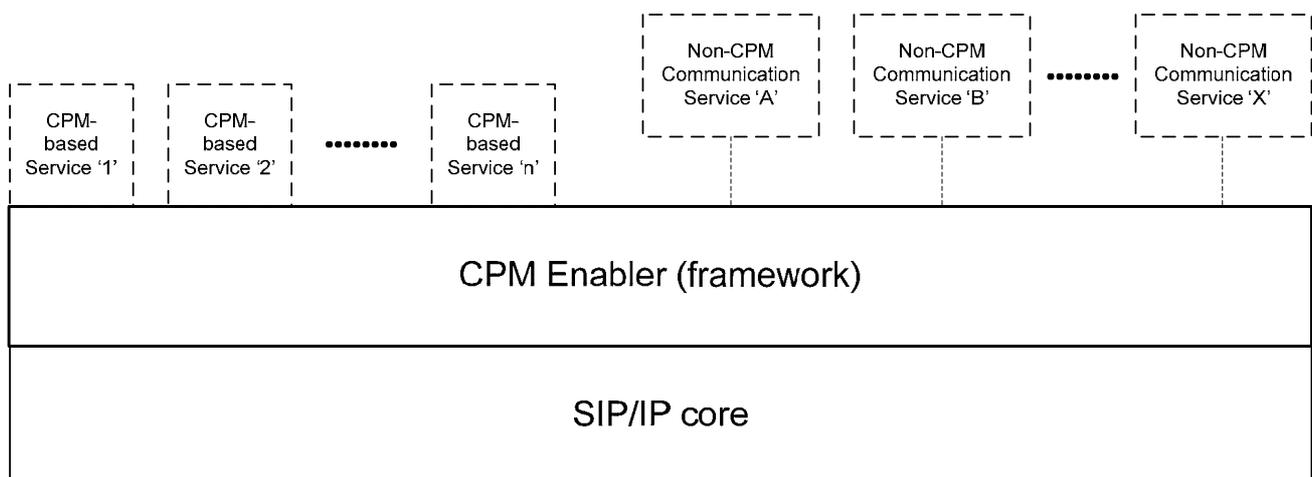


Figure 1: Schematic to illustrate the modularity and extensibility of the CPM Enabler.

### 4.1 CPM 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).
  - One-to-one and one-to-many CPM Conversations among CPM Users with the selection of any kind of Media (single or multiple).
  - 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, which can be modified during CPM Conversations.
- support for CPM Users to utilize several User Preferences Profiles like Office, Home, Meeting, etc. to which the preferences of the CPM User are associated.
- an environment supporting 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 Standalone Messages and CPM Session Histories, including any Media Objects attached to them. All these data can be synchronized to all the Devices of the CPM User. The storage capabilities are subject to the preferences of the CPM User and service provider policies.
- support for interoperability between multiple networks, i.e. CPM Conversations between Principals from different CPM service providers.

## 4.2 CPM Version 2.0

CPM Enabler version 2.0 adds the following features:

- Chat enhancements:
  - Support for notifications within the chat such as “isComposing” and disposition notifications (i.e. delivered, displayed), with a seamless delivery for interworking with SIMPLE IM V2.0
  - Support for storing chat messages and associated notifications while a chat participant is not available (e.g. lost coverage, not registered in IMS) or missed chat
  - Support for delivery of stored chat messages and disposition notifications once the recipient becomes available
  - Group chat features for support of CPM Closed Group Session, CPM Long-lived Session
  - Enhanced multi-device functionality for chat (forking in case of auto-answer)
  - Support for live recording of chat messages
  - Interworking over the NNI with SIMPLE IM V2.0 chat
- CPM File Transfer enhancements:
  - Support CPM File Transfer while having an ongoing CPM Session.
  - Support CPM File Transfer to off-line recipients
  - Support CPM File Transfer termination independently without impacting an ongoing CPM Session, or vice versa.
  - Support CPM File Transfer termination by either sender or recipient
  - Support the request by either sender or recipient to resume an interrupted CPM File Transfer
  - Support for sending a thumbnail image in the invitation of the content to be transferred
  - Explicit support for file size, file name, and type in a CPM File Transfer request.
  - Define a maximum file size policy.

- Support multi device CPM File Transfer request and termination, with cancellation to all remaining devices after one of the devices accepts
- Enhanced CPM Standalone Message support:
  - Enhanced multi-device functionality for sending delivery notifications and read reports towards the device that originated the CPM Standalone Message.
- CPM Message Store enhancements:
  - The location of the 'default' system folder can now be determined using a server annotation.
- Backward compatibility of version 2.0 with CPM Version 1.0 excludes support of the session history object as defined in CPM version 1.0. This has been replaced in CPM version 2.0 by the session history folder and session info object.

## 5. CPM Functional Concepts

### 5.1 General Concepts

#### 5.1.1 Identification

The objects described in this section can be addressed in the CPM Enabler.

##### 5.1.1.1 CPM User

A CPM User is identified by a CPM User address.

NOTE: A CPM User can have multiple CPM User addresses.

A CPM User address SHALL be in the format of

- a SIP URI as specified in [RFC3261]; or,
- a TEL URI as specified in [RFC3966].

##### 5.1.1.2 Non-CPM Principal

A non-CPM Principal is identified inside the CPM Enabler by a non-CPM address. When a non-CPM address is transported inside the CPM Enabler, the non-CPM address SHALL be in the URI format as specified in [RFC2396].

Examples of the possible formats are:

- a SIP URI as specified in [RFC3261];
- a TEL URI as specified in [RFC3966];
- a WV URI as specified in [OMA-IMPS\_CSP-TS];
- a MAILTO URI as specified in [RFC2368].

CPM requests from non-CPM Users MAY contain a Non-CPM Communication Service Identifier.

##### 5.1.1.3 CPM Client

A CPM Client is uniquely identified by an Instance ID defined in [3GPP TS 24.229]. An Instance ID distinguishes the CPM Client on a Device from other CPM Clients on other Devices used by the same CPM User, and it is recommended to be always used in a multi-device, or multi-client deployment context.

In each contact included in the SIP REGISTER request, the CPM Client SHOULD include an instance ID ("sip.instance" header field parameter), whose value is the Instance ID that identifies the User Agent instance being registered. As network support for GRUU is not mandatory, sip.instance can be used instead. A CPM Client will use GRUU if provided by the network.

##### 5.1.1.4 CPM Pre-defined Group

A CPM Pre-defined Group is uniquely identified by a CPM Pre-defined Group address. A CPM Pre-defined Group address SHALL be in the format of a SIP URI as specified in [RFC3261].

##### 5.1.1.5 CPM Group Session

A CPM Group Session is uniquely identified by a CPM Group Session Identity.

A CPM Group Session Identity SHALL be in the format of a SIP URI as specified in [RFC3261], and SHALL be based on Conference URI as specified in [RFC4579].

### 5.1.1.6 CPM-based Service

A CPM-based Service is uniquely identified by a CPM-based Service Identifier.

A CPM-based Service Identifier SHALL be in the format of a URN as specified in [RFC2141].

### 5.1.1.7 CPM Conversation Identity

A CPM Conversation is uniquely identified by a CPM Conversation Identity.

### 5.1.1.8 CPM Contribution Identity

A CPM Standalone Message, CPM File Transfer or CPM Session Invitation is uniquely identified by a CPM Contribution Identity.

## 5.1.2 Signalling over the SIP/IP Core

### 5.1.2.1 Registration / Authentication

The CPM Client SHALL, according to rules and procedures of [3GPP TS24.229] and the SIP/IP core,

- Perform SIP registration for the CPM Enabler to the SIP/IP core prior to using the CPM Enabler functionality. In a multi-device or multi-client context, the CPM Client also includes the Instance ID handling as per [3GPP TS24.229]
- Maintain the SIP registration for the CPM Enabler active by using a re-registration procedure while the CPM Enabler is being used; and,
- Terminate the SIP registration for the CPM Enabler after the CPM Enabler functionality is no longer used.

NOTE: The SIP registration can be shared with other SIP based Enablers at the Device hosting the CPM Client.

The CPM Client SHALL indicate its Instance ID at SIP registration. The CPM User MAY assign a name to his/her device equipped with a CPM Client. The CPM Client SHALL indicate the user assigned name of the CPM Client and its CPM Client user agent capabilities as defined by [RFC3840] in the SIP registration. The CPM Client MAY also update its capabilities by re-registration when the capabilities have changed.

The CPM Client SHALL register with at least one CPM User address.

On receiving the SIP registration request from a CPM Client, the SIP/IP core, according to [3GPP TS24.229]

- Authenticates the CPM User; and,
- Binds a CPM User address and one or more Instance IDs (i.e. one for each CPM User's Client); and,
- Returns the complete list of bindings of that CPM User address in the response to the registration, if successful; and,
- If GRUU is supported by the SIP/IP Core, it returns both a public GRUU and a temporary GRUU associated with the device of the CPM Client according to [3GPP TS24.229] and
- Keeps the binding information (e.g. CPM User address, Instance ID, expiration time, user agent capabilities, temporary GRUU) until de-registration is performed for each Instance ID bound to the CPM User address.

When the SIP/IP core corresponds with 3GPP/3GPP2 IMS then it is possible that the SIP/IP core informs the CPM Client about additional implicitly registered CPM User addresses in the SIP registration response.

The SIP/IP core informs the CPM Participating Function about the registering CPM Client. The CPM Participating Function SHALL subscribe to the Registration Event Information, as defined in section 0, upon reception of the SIP/IP core information about the registering CPM Client, if not subscribed yet.

The SIP/IP core provides an SIP authentication framework as defined in [RFC3261] to authenticate the CPM User. When the SIP/IP core corresponds with 3GPP/3GPP2, then the CPM Client SHALL consider IMS variants [3GPP TS33.178] and determine the appropriate authentication scheme according to [3GPP TS33.203] / [3GPP2 S.R0086-0] and [3GPP TS33.178].

### 5.1.2.2 SIP Signalling Compression

SIP signalling compression is performed in wireless environment on both sides of the wireless access to maximize the transport efficiency and to improve the service quality.

The Device and the SIP/IP core provide signalling compression to reduce the transmission delays according to [RFC3320], [RFC3485], and [RFC3486]. The Device SHALL indicate the signalling compression support as a CPM Client user agent capability in the SIP registration. Whether to use signalling compression is determined by the SIP/IP Core based on service provider policy.

To improve the compression efficiency, the Device and the SIP/IP core MAY support extended operations (e.g. dynamic compression) as defined in [RFC3321].

When the SIP/IP core corresponds with 3GPP/3GPP2 IMS, the Device and the SIP/IP core follow the procedures defined in [3GPP TS24.229]/[3GPP2 X.S0013-004].

### 5.1.2.3 Originator Address

The CPM Client MAY include the address of the sending CPM User when sending CPM requests and CPM responses.

The CPM Enabler expects that the SIP/IP core serving the CPM Client provides an authenticated address of the CPM User in CPM requests and responses.

## 5.1.3 Security

### 5.1.3.1 Secured Client Registration and CPM Service Usage Authentication

The SIP security measures applied to client registration and service usage authentication as described in section 5.1.2.1 also apply to the CPM Client registration and authentication of the services used by the CPM User according to [RFC3261], [3GPP TS33.203] / [3GPP2 S.R0086-0].

### 5.1.3.2 Media Plane Data Integrity and Confidentiality

If integrity and confidentiality of Media Plane data are requested by the CPM User and this is allowed by service provider policies, TLS/PSK-TLS or SRTP will be performed during CPM Media Plane communication in accordance with [RFC2246], [RFC4279], [OMA-SEC\_CF-AD] and [RFC3711].

### 5.1.3.3 Interworking Security

For protection of data integrity and confidentiality of the Media Plane, the Interworking Function SHALL apply the same level of security measures as described in sub-section 5.1.3.2 to the communication received from and sent to non-CPM networks.

### 5.1.3.4 Message Storage and Communication Security

On receiving a request from a Message Storage Client to access CPM Standalone Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, Media Objects and folders stored in the Message Storage Server, TLS/PSK-TLS can be used according to [RFC2246], [RFC4279], [OMA-SEC\_CF-AD], [3GPP TS33.210] and [RFC3501] as following, if requested by the user and allowed by service provider policies.

- a Message Storage Server authenticates a Message Storage Client by checking the Principal's address/certificate or by using pre-shared key authentication mechanisms.
- a Message Storage Client authenticates a Message Storage Server by checking the Message Storage Server's certificate or by using pre-shared key authentication mechanisms.

If integrity and confidentiality of the message are requested by the Principal and subject to service provider policies, the messages are encrypted with integrity protection before transporting them between a Message Storage Client and a Message Storage Server.

In addition to the use of TLS/PSK-TLS, the Message Storage Server checks against the access control list if a Principal has the right to access the resources stored in the Message Storage Server.

## 5.1.4 Registration Event Information Feature

The CPM Participating Function SHALL support the Registration Event Information feature provided by the SIP/IP core.

In order to receive the Registration Event Information of a CPM User, a requestor SHALL send a Registration Event Information subscription request targeted to the CPM User address towards the SIP/IP core.

NOTE: A Registration Event Information subscription can be shared with other SIP-based Enablers at the Device hosting the CPM Client.

When receiving a Registration Event Information subscription request, the SIP/IP core serving the CPM User authorizes the Registration Event Information subscription request. If successfully authorized, the SIP/IP core serving the CPM User:

- responds with a Registration Event Information subscription successful response;
- sends an initial Registration Event Information notification using the reverse path of the Registration Event Information subscription request; and,
- continues sending Registration Event Information notifications using the reverse path of the Registration Event Information subscription request whenever the Registration Event Information changes until the Registration Event Information subscription becomes invalid.

## 5.2 CPM Conversation

### 5.2.1 CPM Conversation Concept

#### 5.2.1.1 CPM Conversation Indications

In order to link CPM Standalone Messages, CPM File Transfers and CPM Sessions to a CPM Conversation the CPM Client, the CPM Participating Function and the CPM Controlling Function SHALL support CPM Conversation Identities.

In order to support creation of a threaded view of CPM Standalone Messages, CPM File Transfers and CPM Sessions belonging to a CPM Conversation the CPM Client, the CPM Participating Function and the CPM Controlling Function SHALL support CPM Contribution Identities and the indications of the CPM Contribution Identity being replied to.

NOTE 1: A threaded view allows CPM Users to view the items in the CPM Conversation History in the relationship of contributions to a CPM Conversation and replies to them. It is possible to present the items in the CPM Conversation History in other views as well - such as views based on sender, subject, sending time or any other information that is available from the CPM Conversation History.

Upon a CPM User's request to send:

- a message, or
- a file, or
- a chat invitation, or
- disposition notifications,

as part of an existing CPM Conversation or to start a new CPM Conversation, the CPM Client SHALL indicate the CPM Conversation Identity in the CPM Standalone Message, the CPM File Transfer, the CPM Session Invitation or the disposition notification sent after CPM Session closure, using the following rules:

- a newly generated CPM Conversation Identity, if the CPM Standalone Message, the CPM File Transfer or the CPM Session Invitation does not relate to any other CPM Standalone Message, CPM File Transfer or CPM Session; or
- the CPM Conversation Identity of a related CPM Standalone Message, a related CPM File Transfer or a related CPM Session.

NOTE 2: The CPM Client can get the CPM Conversation Identity from a stored CPM Conversation History or a received CPM Standalone Message or a received CPM File Transfer or a received CPM Session Invitation.

In addition to indicating the CPM Conversation Identity, the CPM Client SHALL:

- generate and indicate the CPM Contribution Identity of the CPM Standalone Message, the CPM File Transfer or the CPM Session Invitation being sent, or the disposition notification for CPM Chat Message(s) that are sent after CPM Session termination; and,
- indicate the CPM Contribution Identity of the CPM Standalone Message, the CPM File Transfer or the CPM Session being replied to, if replying to a CPM Standalone Message, a CPM File Transfer or a CPM Session.

### 5.2.1.2 Recording a CPM Conversation

When recording a CPM Conversation History, the CPM Participating Function SHALL record and upload the following items to the Message Storage Server:

- CPM Standalone Messages associated to a CPM Conversation to be recorded; and,
- information about successful CPM File Transfers associated to a CPM Conversation to be recorded, including:
  - CPM File Transfer information:
    - when a CPM File Transfer is established:
      - date and time of the CPM File Transfer establishment; and,
      - address of the inviting User; and,
      - addresses of the invited Users.
    - files sent within a CPM File Transfer associated to a CPM Conversation to be recorded;
  - information about successful CPM Sessions associated to a CPM Conversation to be recorded, including:
    - CPM Session information:
      - when a CPM Session is established:
        - date and time of the CPM Session establishment; and,
        - address of the inviting User; and,
        - addresses of the invited Users.
      - CPM Chat Messages sent within a CPM Session associated to a CPM Conversation to be recorded; and,
      - for a CPM Group Session, the Group State Object, and,
      - Disposition notifications sent for the CPM Chat Messages exchanged during the CPM Session; and
      - Media sent within a CPM Session associated to a CPM Conversation to be recorded.

NOTE: CPM Chat Messages, dispositions notifications and Media sent or received by the CPM User in the corresponding CPM Session can be recorded in the CPM User's Message Storage Server

When recording a CPM Conversation History, a CPM Participating Function SHALL include

- the received CPM Conversation Identities; and
- the received CPM Contribution Identities; and,
- the received indications of the CPM Contribution Identity being replied to

in the recorded CPM Standalone Messages, the recorded CPM File Transfer Invitations or the recorded CPM Session Invitations, CPM Chat Messages and disposition notifications exchanged, as part of the CPM Conversation History in the Message Storage Server.

The CPM Participating Function SHALL remain in the media path for the CPM Sessions and CPM File Transfers associated to a CPM Conversation to be recorded.

### 5.2.1.2.1 Recording a CPM Standalone Message

Upon receiving a CPM Standalone Message, a CPM Participating Function SHALL check the user preferences of the CPM User it serves. If the user preferences indicate that recording the CPM Standalone Messages is enabled, the CPM Participating Function SHALL upload the received CPM Standalone Message to the Message Storage Server of the CPM User.

NOTE: This applies to both the originating CPM Participating Function and the terminating CPM Participating Function.

If the CPM Standalone Message is a large message and is delivered in chunks, the CPM Participating Function SHALL wait for all the chunks to be received and reassemble them to one storable object prior to uploading it.

When user preference provides filtering criteria for CPM Standalone Message to be recorded, the CPM Participating Function SHALL apply the filtering criteria to the received CPM Standalone Message and record the filtered output.

Upon completion of recording the CPM Standalone Message or when user preference was set not to record CPM Standalone Messages, the CPM Participating Function SHALL deliver the received CPM Standalone Message as described in section 5.2.2.1.

### 5.2.1.2.2 Recording a CPM File Transfer

Upon receiving a CPM File Transfer session invitation, a CPM Participating Function SHALL check the user preferences of the CPM User it serves. If user preferences indicate that recording of the CPM File Transfer is enabled and if service provider policies allow the CPM Participating Function to stay on the Media path, the CPM Participating Function SHALL stay in the Media path of the CPM File Transfer session.

Upon completion of the CPM File Transfer, the CPM Participating Function SHALL upload the CPM File Transfer information and the received file(s) to the Message Storage Server of the CPM User. If a file is transferred in chunks, the CPM Participating Function SHALL wait for all the chunks to be received and reassemble them to one storable object prior to uploading it.

NOTE 1: If the service provider policies do not allow the CPM Participating Function to stay on the Media path, then the CPM File Transfer will not be recorded irrespective of user preferences.

NOTE 2: This applies to both the originating CPM Participating Function and the terminating CPM Participating Function.

### 5.2.1.2.3 Recording a CPM Session

Upon receiving a CPM Session Invitation, a CPM Participating Function SHALL check the user preferences of the CPM User it serves. If the user preferences indicate that recording the CPM Session is enabled and if service provider policies allow the CPM Participating Function to stay on the Media path, the CPM Participating Function SHALL stay in the Media path of the CPM Session.

The CPM Session recording includes CPM Chat Messages and their disposition notifications. The CPM Session details are stored in the session info object within the session history folder corresponding to the session being recorded, either upon completion of the CPM Session, or as CPM Chat Messages are sent and received by the CPM Participating Function, subject to service provider policies. For CPM Group Sessions the CPM Participating Function SHALL also store the Group State Object.

The CPM Participating Function SHALL upload the CPM Session information, the received Media, CPM Chat Messages and the dispositions notifications sent or received to the Message Storage Server of the CPM User into the session history folder which corresponds to the session being recorded.

NOTE 1: If the service provider policies do not allow the CPM Participating Function to stay on the Media path, then the CPM Session will not be recorded irrespective of user preferences.

NOTE 2: This applies to both the originating CPM Participating Function and the terminating CPM Participating Function.

## 5.2.2 CPM Messaging Concepts

### 5.2.2.1 CPM Standalone Message Delivery

#### 5.2.2.1.1 Message Handling in the Originating Network

The CPM Client SHALL include the address of the target CPM User or the target non-CPM Principal when sending a CPM Standalone Message to a single recipient. The CPM Client SHALL use the address of the target CPM User or of the target non-CPM Principal as the target address of the CPM Standalone Message.

When sending a CPM Standalone Message to a CPM Ad-hoc Group, the CPM Client SHALL include the addresses of the target CPM Users and the target non-CPM Principals when sending the CPM Standalone Message. The CPM Client SHALL use the address of the CPM Controlling Function provisioned to the CPM Client as the target address of the CPM Standalone Message.

When sending a CPM Standalone Message to a CPM Pre-defined Group, the CPM Client SHALL include the address of the target CPM Pre-defined Group when sending the CPM Standalone Message. The CPM Client SHALL use the address of the CPM Pre-defined Group as the target address of the CPM Standalone Message.

NOTE: The address of a CPM Pre-defined Group comprises the address of a CPM Controlling Function. When a CPM Client sends a CPM Standalone Message to a CPM Pre-defined Group, the Controlling Function hosting this CPM Pre-defined Group may be different from the CPM Controlling Function provisioned to the CPM Client.

Upon request of a CPM User, the CPM Client SHALL include a Reply-to Indication in a CPM Standalone Message containing the CPM Address where replies to the CPM Standalone Message have to be sent to, instead of sending those replies to the CPM Address that was used to send the CPM Standalone Message from.

Upon request from a CPM User to send a CPM Standalone Message, the CPM Client SHALL send the CPM Standalone Message towards the target address via the SIP/IP core. The CPM Client SHALL indicate a corresponding CPM Feature Tag in the CPM Standalone Message.

NOTE: The CPM Feature Tag is used by the SIP/IP core to route the CPM Standalone Message to the CPM Participating Function.

The SIP/IP core serving the sender routes the CPM Standalone Message to the CPM Participating Function serving the sender based on the included CPM Feature Tag.

The CPM Participating Function serving the sender SHALL authorize the sender and if authorized, the CPM Participating Function SHALL send the CPM Standalone Message towards the target address via the SIP/IP core.

The SIP/IP core serving the sender forwards the CPM Standalone Message towards the target address.

#### 5.2.2.1.2 Message Handling in the CPM Controlling Function

The SIP/IP core in the controlling network routes the CPM Standalone Message to the CPM Controlling Function based on the target address of the CPM Standalone Message.

Upon receiving a CPM Standalone Message targeted to a CPM Pre-defined Group, the CPM Controlling Function SHALL fetch the CPM Pre-defined Group members as well as the policies from the Group XDMS. Subsequently the CPM Controlling Function SHALL determine whether to allow the CPM Standalone Message request or not based on the policies stored as part of the CPM Pre-defined Group definition. These conditions include criteria such as whether the request is allowed for the group, whether the requesting CPM User is allowed to initiate the service, etc.

When the CPM Controlling Function receives a CPM Standalone Message targeted to a CPM Ad-hoc Group, the CPM Controlling Function SHALL determine whether to allow the request or not based on the service provider policies.

When the CPM Controlling Function allows the request, the CPM Controlling Function SHALL respond with a CPM Standalone Message successful response and SHALL send the CPM Standalone Message towards each CPM Group member via the SIP/IP core; otherwise, the CPM Controlling Function SHALL reject the request with an appropriate error response.

When distributing the CPM Standalone Message to the members of a CPM-Pre-defined Group, the CPM Controlling Function SHALL ensure that the CPM Standalone Message contains both the CPM Address of the sender of the CPM Message and the address of the CPM Pre-defined Group.

The CPM Controlling Function SHALL indicate the CPM Feature Tag in the CPM Standalone Messages it sends towards the members of the CPM Group.

For CPM Standalone Messages sent by the CPM Controlling Function to non-CPM Principal recipients with a valid SIP address, the procedures described in sections 5.3.1.1.2, 5.3.1.1.5, and 5.3.1.1.6 apply.

For CPM Standalone Messages sent by the CPM Controlling Function to non-CPM Principal recipients with an address that is not routable in the SIP/IP core, the procedures in section 5.3.1.1.7 applies.

For CPM Standalone Messages sent to CPM User recipients, the SIP/IP core associated with the CPM Controlling Function and the SIP/IP core serving the respective recipient route the CPM Standalone Message to the CPM Participating Function serving the respective recipient based on the included CPM Feature Tag.

### 5.2.2.1.3 Message Handling in Terminating Networks

The SIP/IP core serving the recipient routes the CPM Standalone Message to the CPM Participating Function serving the recipient based on the included CPM Feature Tag.

The CPM Participating Function serving the recipient SHALL handle the CPM Standalone Message in one of the following ways:

- reject the CPM Standalone Message, if user preferences of the recipient indicate rejection for the received CPM Standalone Message.  
NOTE 1: The user preferences can depend on e.g. originator address, undisclosed sender identity, or message type/content.
- hand over the CPM Standalone Message to the Interworking Selection Function, if user preferences of the recipient indicate interworking
- deliver the CPM Standalone Message to the Message Storage Server, if user preferences of the recipient indicate delivery to the Message Storage Server  
NOTE 2: delivery to the Message Storage Server is realized as a storage operation and not to be confused with recording copies of incoming messages in the Message Storage Server (which are also realized as storage operations). In cases where both scenarios apply (delivery and recording to Message Storage Server) only one storage operation is triggered.
- forward the CPM Standalone Message to a different address, if user preferences indicate forwarding to a specific forward-to address;  
NOTE 3: if user preferences indicate one of the above (rejection, interworking, delivery to the Message Storage Server, or forwarding), a parallel user preference setting stating to “do not disturb” has no effect. The respective CPM Standalone Message will be rejected, interworked, delivered to the Message Storage Server, or forwarded. Implementations may simply choose to not offer do not disturb settings then.
- defer the CPM Standalone Message and proceed as specified in section 5.2.5, if the CPM Standalone Message was neither rejected, interworked, delivered to the Message Storage Server nor forwarded, and if
  - user preferences of the recipient indicate deferring the CPM Standalone Message; or,
  - user preferences of the recipient indicate delivery of the CPM Standalone Message, but the user has also indicated not to be disturbed; or
  - user preferences of the recipient indicate delivery of the CPM Standalone Message, but no suitable CPM Client of the recipient is registered  
NOTE 4: A CPM Client is suitable if user preferences, device capabilities, device connectivity and server provider policies indicate delivering the CPM Standalone Message

- send the CPM Standalone Message via the SIP/IP Core towards each suitable registered CPM Client of the recipient;

Subsequently the SIP/IP core serving the recipient routes the CPM Standalone Message to the addressed recipient's CPM Client(s).

NOTE 5: CPM Clients of the recipient for which user preferences or other factors (e.g. service provider policies, device connectivity) prevent delivery will not receive the CPM Standalone Message. User preferences may be specified per device only for the delivery of CPM Standalone Messages.

NOTE 6: If the CPM Standalone Message is recorded in the Message Storage Server as specified in 5.2.1.2, any Device of the recipient can get the CPM Standalone Message by synchronizing with the Message Storage Server.

If supported by the SIP/IP core serving the recipient, the SIP/IP core inserts the target CPM Address into the CPM Standalone Message. If this is not supported by the SIP/IP core serving the recipient, the CPM Participating Function serving the recipient SHALL insert the target CPM Address into the CPM Standalone Message.

Upon receiving the CPM Standalone Message, the recipient's CPM Client(s) SHALL respond with a CPM Standalone Message successful response.

#### 5.2.2.1.4 Replying to a CPM Standalone Message

When replying to a received CPM Standalone Message that was sent to a CPM Group, the CPM Client SHALL give the CPM User the option to either reply to only the sender of the CPM Standalone Message, or send the reply to the whole group. When the CPM User chooses to send the reply only to the sender of the CPM Standalone Message, the CPM Client SHALL send the reply CPM Standalone Message just towards the sender of the CPM Standalone Message, otherwise the CPM Client SHALL send the reply CPM Standalone Message as a CPM Group CPM Standalone Message towards the whole group.

When replying to a received CPM Standalone Message, the CPM Client SHALL use the CPM Address indicated in the Reply-to Indication of the received CPM Standalone Message as the target address of the original sender, when the Reply-to Indication is present in the received CPM Standalone Message, unless the CPM User selects to use the address of the originator of the received CPM Standalone Message. The use of the Reply-to Indication is not supported in case of replying to a CPM Pre-defined Group. When no Reply-to Indication is present in the received CPM Standalone Message the target address of the original sender SHALL be set to the address of the originator of the received CPM Standalone Message.

When replying to a received CPM Standalone Message from a non-CPM Principal, the CPM Client SHALL accompany the target non-CPM address with the Non-CPM Communication Service Identifier, if included in the received CPM Standalone Message.

NOTE 1: The inclusion of the Non-CPM Communication Service Identifier allows for the CPM Standalone Message to be delivered through the same Non-CPM Communication Service from where the original CPM Standalone Message was received.

NOTE 2: The Non-CPM Communication Service Identifier is not supposed to be visible to the CPM User.

#### 5.2.2.1.5 Large Message Mode Delivery

When a CPM functional component A wants to send or forward a CPM Standalone Message to another CPM functional component B and the CPM Standalone Message is too large (i.e. larger than 1300 bytes) to be delivered as a Pager Mode CPM Standalone Message, the CPM functional component A SHALL establish a SIP session with a recipient CPM functional component B, indicating MSRP as the Media Stream. CPM functional component A SHALL include the size of the CPM Standalone Message and the content type(s) used in the CPM Standalone Message in the SIP session invitation. CPM functional component A SHALL indicate a corresponding CPM Feature Tag in the SIP session invitation.

NOTE: The CPM Feature Tag is used by the SIP/IP core to route the SIP session invitation to the next CPM functional component receiving it.

Upon establishing a SIP session for Large Message Mode delivery, the direction of media stream SHALL be set uni-directional. After the SIP session is established, the CPM Standalone Message is transferred via MSRP.

When the CPM Message delivery with Large Message Mode is completed, the originating CPM functional component SHALL close the SIP session.

If an error (e.g. temporary loss of connection with a mobile CPM Client and that mobile CPM Client is the only registered device or if an error is returned from the registered CPM Clients) prevents the delivery of the Large Message Mode CPM Message in the terminating network, the terminating CPM Participating Function SHALL defer the Large Message Mode CPM Standalone Message and further handle the Deferred CPM Message as described in section 5.2.5.

#### 5.2.2.1.6 Forwarding/Including Stored Data without Downloading to the CPM Client

When a CPM User sends a CPM Standalone Message with attached Media Objects, CPM Messages, CPM Session Histories or CPM Conversation History stored in the Message Storage Server, the originating CPM Client SHALL interact with the Message Storage Client to obtain a reference to the stored data from the Message Storage Server respectively and include this reference to the stored data in the CPM Standalone Message instead of the stored data itself.

Upon receiving a CPM Standalone Message containing a reference to stored data, the originating CPM Participating Function SHALL interact with the Message Storage Server to fetch the Media Objects, CPM Messages, CPM Session Histories or CPM Conversation History indicated by the reference and, if successful, the originating CPM Participating Function SHALL replace the reference to the stored data in the CPM Message with the fetched data. The CPM Standalone Message containing the data instead of the reference is routed to the target address via the SIP/IP core following the procedures described in 5.2.2.1.3.

If, after adding the fetched data, the size of the CPM Standalone Message exceeds the maximum size for a Pager Mode CPM Standalone Message of 1300 bytes, the originating CPM Participating Function SHALL send the message as a Large Message Mode CPM Standalone Message.

If fetching of the data from the Message Storage Server is unsuccessful, the CPM Participating Function SHALL send an error response to the CPM Standalone Message sender indicating that the supplied references cannot be found.

#### 5.2.2.1.7 CPM Standalone Message Delivery with a Reference to the Media

NOTE: This functionality is only available for CPM Users who have access to a Message Storage Server.

Upon receiving a CPM Standalone Message containing attachments (i.e., Media Objects, CPM Messages, CPM File Transfer Histories, CPM Session Histories or CPM Conversation Histories), the terminating CPM Participating Function SHALL, in accordance with the recipient's preferences and service provider policies, determine whether to deliver the CPM Standalone Message including the attachments or with references to those attachments. If it is determined that references to the attachments are to be delivered, the CPM Participating Function SHALL store the CPM Standalone Message including all attachments in the Message Storage Server, request references (generated by the Message Storage Server) indicating the respective locations of the attachments according to rules in [RFC4483] and, if successful, route the CPM Standalone Message with references to the attachments to the target address via the SIP/IP core following the procedures in 5.2.2.1.3.

NOTE: The stored attachments are removed when the CPM User removes the stored CPM Standalone Message.

When a received CPM Standalone Message contains references indicating the respective locations of attachments, the recipient's CPM Client SHALL interact with the Message Storage Client to fetch specific attachments (e.g., when requested by the CPM User) from the referenced locations in the Message Storage Server.

NOTE: Referenced attachments may be fetched at any point in time.

#### 5.2.2.1.8 CPM Standalone Message End-to-end Indications

The CPM Client SHALL indicate the following end-to-end indications in the sent CPM Standalone Message:

- date and time of sending;
- CPM Conversation Identity;
- CPM Contribution Identity;
- indication of the CPM Contribution Identity being replied to;

In addition, the CPM Client MAY indicate the following end-to-end indications in the sent CPM Standalone Message:

- subject;
- that the message is of a sensitive nature;

- priority;
- content disposition of each Media embedded in the CPM Standalone Message;

When forwarding a CPM Standalone Message, the CPM Participating Function and the CPM Controlling Function SHALL preserve the end-to-end indications of the received CPM Standalone Message.

## 5.2.3 CPM Session Concepts

### 5.2.3.1 CPM 1-1 Sessions

The following subsections describe the high level procedures for creating, closing and modifying CPM 1-1 Sessions, whereby CPM 1-1 Session requests are routed between the network serving the inviting CPM User and the network serving the invited CPM User without involvement of a controlling network.

#### 5.2.3.1.1 Session Initiation

##### 5.2.3.1.1.1 Network Serving the Inviting CPM User

In order to establish a CPM 1-1 Session, the inviting CPM Client SHALL send a CPM Session Invitation to the SIP/IP core serving the inviting CPM Client. The CPM Client SHALL indicate a corresponding CPM Feature Tag in the CPM Session Invitation.

NOTE: The CPM Feature Tag is used by the SIP/IP core to route the CPM Session Invitation to the CPM Participating Function.

The CPM Client SHALL include the address of the invited CPM User or the invited non-CPM Principal in the CPM Session Invitation.

The CPM Client SHALL include the Instance ID in the CPM Session Invitation. If GRUU is supported and anonymity is requested, then the temporary GRUU associated with the device SHALL be included; otherwise the public GRUU associated with the device SHALL be included.

The SIP/IP core serving the inviting CPM Client routes the CPM Session Invitation to the CPM Participating Function serving the inviting CPM Client based on the included CPM Feature Tag.

Upon receiving a CPM Session Invitation, the CPM Participating Function SHALL:

- determine whether or not to continue CPM Session establishment; and
- route the CPM Session Invitation to the invited CPM User through the SIP/IP core.
- if interworking is to occur as described in section 5.3.1.1, send the CPM Session Invitation to the Interworking Selection Function.

The CPM Participating Function SHALL decide whether or not it remains in the media path for the CPM Chat Messages exchanged during the CPM Session (e.g. if the CPM session needs to be recorded), subject to user preferences and according to service provider policies.

##### 5.2.3.1.1.2 Network Serving the Invited CPM User

The SIP/IP core serving the invited CPM User routes the CPM Session Invitation to the CPM Participating Function serving the invited CPM User based on the included CPM Feature Tag.

Upon receiving a CPM Session Invitation, the CPM Participating Function SHALL:

- reject the CPM Session Invitation, if
  - user preferences of the recipient indicate rejecting the received CPM Session Invitation; or,

NOTE: The user preferences can depend on e.g. originator address, undisclosed sender identity, or requested Media Streams.

- the invited CPM User has indicated to not disturb; or,

- no CPM Client of the recipient is registered and the user preferences of the recipient indicate rejecting incoming CPM Session Invitations when no CPM Client is registered;
- if one or more CPM Clients are registered and the CPM Session Invitation is not rejected according to preceding bullet:
  1. determine which CPM Clients are not to receive the CPM Session Invitation based on criteria such as offered Media Streams in the CPM Session Invitation, the capabilities of CPM Clients of the invited CPM User, invited CPM User's preferences and service provider policies.
  2. indicate in the CPM Session Invitation the Unique User Agent Identifier(s) of the CPM Client(s) not eligible for reception of the CPM Session Invitation.
  3. decide whether or not to remain in the media path for this session according to service provider policies. When the CPM Participating Function detects that the CPM Client involved in the CPM Session has become unreachable (e.g. offline), the CPM Participating Function SHALL act as a B2BUA by keeping the missed CPM Chat Messages and corresponding disposition notifications for a deferred delivery to the CPM Client.

**Editor's Note:** check if a bullet item needs to be added to cover the case of recipient never accepting the invitation, as PF needs to act also as B2BUA in this case.

4. determine if the CPM Session needs to be recorded, subject to user preferences and service provider policies. If the CPM Session needs to be recorded the CPM Participating Function remains in the media path. Recording of CPM Chat Messages exchanged during the CPM Session is performed as described in sect. 5.2.1.2.3 "Recording a CPM Session".
  5. route the CPM Session Invitation to the invited CPM Client(s) via the SIP/IP core.
- If disposition notifications are requested in the CPM Chat Message, the CPM Participating Function SHALL:
    1. handle them as described in sect. 5.2.6 "Disposition Notification" when the disposition notifications are delivered after the termination of the CPM Session, or
    2. handle them in the Media Plane when received over MSRP during the CPM Session.
  - if interworking is to occur as described in section 5.3.1.1, send the CPM Session Invitation to the Interworking Selection Function.

If supported by the SIP/IP core serving the recipient, the SIP/IP core inserts the received CPM Address of the called party into the CPM Session Invitation. If this is not supported by the SIP/IP core serving the recipient, the CPM Participating Function serving the recipient SHALL insert the received CPM Address of the called party into the CPM Session Invitation.

Subsequently the SIP/IP core serving the recipient routes the CPM Session Invitation to the CPM Clients of the invited CPM User apart from those listed as not eligible for reception of the CPM Session Invitation.

### 5.2.3.1.2 Closing a Session

In order to close a CPM Session, the CPM Client SHALL send a CPM Session leaving request to the SIP/IP core serving the CPM Client.

The SIP/IP core routes the CPM Session leaving request according to the Signalling Path used at the CPM Session establishment.

The CPM Participating Function of the CPM User closing the CPM Session SHALL route the CPM Session leaving request according to the Signalling Path used at the CPM Session establishment.

The SIP/IP core routes the CPM Session leaving request according to the Signalling Path used at the CPM Session establishment.

The CPM Participating Function of the peer CPM User SHALL route the CPM Session leaving request according to the Signalling Path used at the CPM Session establishment.

The CPM Client receiving the CPM Session leaving request SHALL close the CPM Session and release all resources associated with that CPM Session.

### 5.2.3.1.3 Session Modification

CPM Session modification allows a CPM Client participating in a CPM Session to add new Media Streams, to remove existing Media Streams or to change the characteristics of existing Media Streams.

NOTE 1: CPM Session modification can be initiated based on the CPM User input or for other reasons, e.g. due to moving to lower bandwidth IP network.

The CPM Participating Function SHALL support CPM Session modification. A CPM Client that supports CPM Session establishment SHALL support CPM Session modification.

The CPM Client MAY initiate modification of a CPM Session or MAY create an additional CPM Session for the additional Media Stream.

NOTE 2: Using more than one CPM Session in a scenario that constitutes one logical real-time exchange to the user has impacts: when the user, for instance, requests the current list of Participants, this will be valid only for one of the CPM Sessions, and the user needs to be aware that the other CPM Sessions might have divergent sets of Participants. Also, when one of the CPM Sessions is terminated, the other CPM Sessions are not affected. The Media Streams used in the different CPM Sessions will not be correlated, mixed, etc. In this case, the correlation is left to implementation and may use information available at CPM Conversation level, e.g., parallel CPM Sessions carrying the same CPM Conversation Identity.

In order to modify a CPM Session the CPM Client SHALL send a CPM Session modification request according to the Signalling Path used at the CPM Session establishment.

The SIP/IP core routes the CPM Session modification request according to the Signalling Path used at the CPM Session establishment.

The CPM Participating Function of the originating CPM User SHALL route a CPM Session modification request according to the route determined during the CPM Session establishment, if the CPM Session modification request satisfies the originating user preferences and service provider policies.

The CPM Participating Function of the terminating CPM User SHALL route a CPM Session modification request according to the Signalling Path used at the CPM Session establishment, if the CPM Session modification request satisfies the terminating user preferences and service provider policies.

### 5.2.3.2 CPM Group Sessions

#### 5.2.3.2.1 CPM Group Session Initiation

A CPM Session Invitation is routed through the network serving the inviting CPM User, through the network hosting the CPM Group, and the network serving the invited CPM User. The following sections describe the handling of the CPM Session Invitation in each of those networks.

##### 5.2.3.2.1.1 Network Serving the Inviting CPM User

When initiating a CPM Group Session for a CPM Ad-hoc Group, the CPM Client SHALL include the addresses of the target CPM Users and the target non-CPM Principals in a CPM Session Invitation. The CPM Client SHALL use the address of the CPM Controlling Function provisioned to the CPM Client as the target address of the CPM Session Invitation.

When initiating a CPM Group Session for a CPM Pre-defined Group, the CPM Client SHALL include the address of the target CPM Pre-defined Group in a CPM Session Invitation. The CPM Client SHALL use the address of the CPM Pre-defined Group as the target address of the CPM Session Invitation.

NOTE 1: The address of a CPM Pre-defined Group comprises the address of a CPM Controlling Function. When a CPM Client sends a CPM Group Session invitation to a CPM Pre-defined Group, the Controlling Function hosting this CPM Pre-defined Group may be different from the CPM Controlling Function provisioned to the CPM Client.

In order to establish a CPM Group Session, the inviting CPM Client SHALL send a CPM Session Invitation to the SIP/IP core serving the inviting CPM Client. The CPM Client SHALL indicate a corresponding CPM Feature Tag in the CPM Session Invitation.

NOTE 2: The CPM Feature Tag is used by the SIP/IP core to route the CPM Session Invitation to the CPM Participating Function.

The SIP/IP core serving the inviting CPM Client routes the CPM Session Invitation to the CPM Participating Function serving the inviting CPM Client based on the included CPM Feature Tag.

Upon receiving a CPM Session Invitation, the CPM Participating Function SHALL:

- determine whether or not to continue the CPM Session establishment; and,
- route the CPM Session Invitation to the CPM Controlling Function hosting the CPM Session via the SIP/IP core.

The CPM Participating Function SHALL indicate the CPM Feature Tag in the CPM Session Invitation.

The CPM Participating Function SHALL decide whether or not it remains in the media path for the CPM Chat Messages exchanged during the CPM Group Session (e.g. if the CPM session needs to be recorded), subject to user preferences and according to service provider policies.

#### 5.2.3.2.1.2 Network Hosting the CPM Group Session

The SIP/IP core of the network hosting the CPM Group Session routes the CPM Session Invitation to the CPM Controlling Function hosting the CPM Group Session based on the included CPM Feature Tag.

When the CPM Controlling Function receives a CPM Session Invitation targeted to a CPM Pre-defined Group, the CPM Controlling Function SHALL fetch the CPM Pre-defined Group members as well as the policies from the Group XDMS and SHALL determine whether to allow the request or not based on the policies stored as part of the CPM Pre-defined Group definition such as whether the service requested is allowed for the group, whether the requesting CPM User is allowed to initiate the service, etc.

When the CPM Controlling Function receives a CPM Session Invitation targeted to a CPM Ad-hoc Group, the CPM Controlling Function SHALL determine whether to allow the request or not based on the service provider policies related to CPM Group Sessions such as the type of CPM Ad-hoc Group Session (e.g. Closed), media types offered, etc

If determined to continue the CPM Group Session establishment, the CPM Controlling Function SHALL send a CPM Session Invitation to the invited CPM Users via the SIP/IP core from which the CPM Session Invitation was received. In case of a CPM Ad-hoc Group, the invited CPM Users are determined from the received CPM Session Invitation, in case of a CPM Pre-defined Group which is a Pre-arranged Group, the invited CPM Users are the members of the CPM Pre-defined Group indicated in the received CPM Session Invitation.

NOTE: No CPM Session Invitations are sent for a CPM Pre-defined Group which is a Join-in Group.

The CPM Controlling Function SHALL indicate the CPM Feature Tag in the CPM Session Invitations it sends.

The CPM Controlling Function SHALL include into the CPM Session Invitations it sends:

- the address of the inviting CPM User or the inviting non-CPM Principal; and,
- the address of the invited CPM User or the invited non-CPM Principal; and,
- the addresses of the target CPM Users and the target non-CPM Principals if the CPM Session Invitation is for a CPM Ad-hoc Group; and,
- the CPM Pre-defined Group address if the CPM Session Invitation is for a CPM Pre-defined Group; and,
- the CPM Group Session Identity, and
- The CPM Conversation Identity, the CPM Contribution Identity and the CPM Contribution Identity being replied to received in the CPM Session Invitation.

If the received CPM Session Invitation included a CPM Conversation Identity and a CPM Contribution Identity, then the CPM Controlling Function SHALL store the CPM Conversation Identity and the CPM Contribution Identity for the duration

of the CPM Group Session. If the received Session Invitation also included an indication of a CPM Contribution Identity being replied to, then the CPM Controlling Function SHALL store this CPM Contribution Identity being replied to as well for the duration of the CPM Group Session. The CPM Controlling Function SHALL include those values in the response to the CPM Session Invitation.

The CPM Controlling Function SHALL include the CPM Group Session Identity into the CPM Session Invitation response.

If interworking is to occur as described in section 5.3.1.1, the CPM Controlling Function SHALL send the CPM Session Invitation to the Interworking Selection Function.

#### 5.2.3.2.1.3 Network Serving the Invited CPM User

Apply the same mechanism described for routing a CPM Session Invitation for a CPM 1-1 Session in section 5.2.3.1.1.2.

#### 5.2.3.2.1.4 CPM Long-lived Group Sessions

A CPM Long-lived Group Session SHALL be re-started by the CPM Client with the same CPM Group Session Identity, the same Conversation-ID and the same Contribution-ID as the original CPM Group Session.

A CPM Long-lived Group Session is re-started with a re-join request. If the request for a CPM Group Session initiation is a re-join of an inactive CPM Session, the CPM Controlling Function SHALL check if there is a match with a previously used CPM Group Session Identity, Conversation-ID, and Contribution-ID, in order to determine if this is a re-start of a CPM Long-lived Group Session. If a match is found, the CPM Controlling Function SHALL retrieve the Participants Information and validate that the CPM User is among the Participants.

In case of a mismatch, of either CPM Group Session Identity, or of the subsequent check on the CPM User against the Participants Information, the CPM Controlling Function SHALL reject the request.

The CPM Controlling Function SHALL re-start the CPM Group Session using the same CPM Group Session type (e.g. CPM Closed Group Session) as the previous CPM Group Session of this CPM Group Session Identity.

The criteria for CPM Long-lived Group Session termination is subject to service provider policies (e.g. originator leaving the CPM Session, or other criteria). Once it was terminated, the CPM Group Session Identity is no longer kept and the CPM Group Session cannot be re-started at a later time as a CPM Long-lived Group Session by the CPM Controlling Function.

When a CPM Client that has re-started a CPM Long-lived Group Session receives the 404 NOT FOUND error, the CPM Client SHALL initiate a new CPM Group Session using the same Conversation-ID and Contribution-ID values as the original one of the CPM Long-lived Group Session.

Note: The value of the Contribution-ID associated with the re-started CPM Group Session can be the existing Contribution-ID of the CPM Session being re-started, or a new value.

### 5.2.3.2.2 CPM Client (Re-)Joining an Existing CPM Group Session

A CPM Session Invitation is routed through the network serving a (re-)joining CPM User to the network hosting the CPM Session. The following sections describe the processing in each of those networks.

#### 5.2.3.2.2.1 Network Serving the (Re-)Joining CPM User

In order to join an existing CPM Group Session, the (re-)joining CPM Client SHALL send a CPM Session Invitation targeted to the CPM Group Session Identity associated with the CPM Group Session to the SIP/IP core serving the (re-)joining CPM Client. The CPM Client SHALL indicate a corresponding CPM Feature Tag in the CPM Session Invitation. The CPM Client SHALL re-use the CPM Conversation Identity from the previously received CPM Session Invitation from that particular CPM Group Session Identity, and generate a new CPM Contribution Identity as is required for any new CPM Session Invitation or SHALL use a new CPM Conversation Identity and CPM Contribution identity if joining an ongoing dial-in CPM Session.

NOTE: The CPM Feature Tag is used by the SIP/IP core to route the CPM Session Invitation to the CPM Participating Function.

When the CPM Client receives a response to the CPM Session Invitation it sent, the CPM Client SHALL replace the previously self-generated CPM Conversation Identity and CPM Contribution Identity with the CPM Conversation Identity and the CPM Contribution Identity included in the CPM Session Invitation response and SHALL store the CPM Contribution Identity being replied to if received in the response.

The SIP/IP core serving the (re-)joining CPM Client routes the CPM Session Invitation to the CPM Participating Function serving the (re-)joining CPM Client based on the included CPM Feature Tag.

Upon receiving the CPM Session Invitation, the CPM Participating Function SHALL:

- determine whether or not to continue the CPM Session establishment; and,
- route the CPM Session Invitation to the CPM Controlling Function hosting the CPM Group Session via the SIP/IP core. The CPM Participating Function SHALL indicate the CPM Feature Tag in the CPM Session Invitation.

#### 5.2.3.2.2.2 Network Hosting the CPM Group Session

The SIP/IP core of the network hosting the CPM Group Session routes the CPM Session Invitation to the CPM Controlling Function hosting the CPM Group Session based on the included CPM Group Session Identity.

When the CPM Controlling Function receives a CPM Session Invitation targeted to a CPM Group Session Identity, the CPM Controlling Function SHALL determine whether the requesting CPM User is allowed to be a Participant in the CPM Group Session based on the policies associated with the CPM Group Session.

In an active Closed CPM Group Session, the CPM Controlling Function SHALL reject re-join requests from CPM Clients (i.e. the cases in which the CPM Client explicitly left the CPM Session).

If CPM Long-lived Group Session is allowed by service provider policies, and if the CPM Group Session Identity is still valid (i.e. not expired by service provider policies):

- when the CPM User requests to re-start an inactive CPM Group Session, then the CPM Controlling Function SHALL check that the CPM User is among the Participants list available in the latest Participants Information of that CPM Group Session. If a match is found, then the CPM Long-lived Group Session is re-started, otherwise is rejected.

If the CPM User is allowed to be a Participant in the CPM Group Session the CPM Controlling Function SHALL accept the CPM Session Invitation and add the requesting CPM User to the CPM Group Session; otherwise the CPM Controlling Function SHALL respond with an appropriate error.

The CPM Controlling Function SHALL include the stored CPM Conversation Identity and the CPM Contribution Identity in the CPM Session Invitation response as described in section **Error! Reference source not found.** If a CPM Contribution Identity being replied to is also stored, then the CPM Controlling Function SHALL include the CPM Contribution Identity being replied to in the CPM Session Invitation response.

The CPM Controlling Function SHALL add the CPM User that has successfully re-joined the CPM Group Session to the Participants Information of that CPM Group Session and send updates to all participants that subscribe to the Participants Information.

#### 5.2.3.2.3 CPM Client Leaving a CPM Group Session

In order to leave a CPM Group Session in which a CPM Client participates, the CPM Client SHALL send a CPM Session leaving request to the SIP/IP core serving the CPM Client.

The SIP/IP core routes the CPM Session leaving request according to the Signalling Path used at CPM Session establishment.

The CPM Participating Function of the leaving CPM User SHALL route the CPM Session leaving request according to the Signalling Path used at CPM Session establishment.

When receiving a CPM Session leaving request the CPM Controlling Function SHALL:

- stop sending Media belonging to the CPM Group Session to the CPM Client and discard any Media belonging to the CPM Group Session received from the CPM Client; and,

- if the CPM Group policies indicate that the CPM Session should be released when this CPM Client leaves the CPM Session, then stop handling of the Media belonging to the CPM Group Session of the other CPM Clients participating in the CPM Group Session and release the other CPM Clients participating in the CPM GroupSession as specified in the subsection **Error! Reference source not found.**; and release the resources associated with the CPM Group Session.
- remove the CPM User from the Participants Information of the CPM Group Session and send updates to all participants that subscribe to the Participants Information.

#### 5.2.3.2.4 CPM Client Inviting CPM Users to an Existing CPM Group Session

In order to add CPM Users to an existing CPM Group Session in which the CPM Client participates, the CPM Client SHALL send a CPM Session adding request to the SIP/IP core serving the CPM Client. The CPM Client SHALL indicate the newly invited CPM User(s) in the CPM Session adding request.

The SIP/IP core routes the CPM Session adding request towards the CPM Participating Function of the adding CPM User based on the included CPM Feature Tag.

The CPM Participating Function of the adding CPM User SHALL route the CPM Session adding request towards the CPM Controlling Function hosting the CPM Group Session based on the CPM Group Session Identity.

When receiving a CPM Session adding request the CPM Controlling Function SHALL authorize the request according to the type of CPM Group Session and to the CPM Group policies (e.g. policies may specify whether the requesting CPM User is authorized to add new Participants and whether the invited users are allowed to be part of the CPM Group Session based on criteria such as age, or they cannot be added twice) and service provider policies (e.g. service provider policies may impose rules such as the maximum number of Participants for a CPM Group Session) and if authorized, the CPM Controlling Function SHALL invite each CPM User listed in the CPM Session adding request as specified in the subsection 0.

When a CPM Client wants to add new Participants to a Closed CPM Group Session, the CPM Controlling Function SHALL reject the request.

The CPM Controlling Function SHALL send a response to the inviting CPM User with either the successful or unsuccessful result of the request.

#### 5.2.3.2.5 CPM Client Removing Participants from a CPM Group Session

In order to remove Participants from a CPM Group Session in which the CPM Client participates, the CPM Client SHALL send a CPM Session removal request to the SIP/IP core serving the CPM Client. The CPM Client SHALL indicate the addresses of the Participants to be removed in the CPM Session removal request.

The SIP/IP core routes the CPM Session removal request towards the CPM Participating Function of the removing CPM Participant based on the included CPM Feature Tag.

The CPM Participating Function of the removing Participant SHALL route the CPM Session removal request towards the CPM Controlling Function hosting the CPM Group Session based on the CPM Group Session Identity.

When receiving a CPM Session removal request the CPM Controlling Function SHALL authorize the request according to the CPM Group policies and service provider policies and if authorized, the CPM Controlling Function SHALL release the Participants listed in the CPM Session removal request from the CPM Session as specified in the subsection **Error! Reference source not found.** and send updates to all participants that subscribe to the Participants Information.

#### 5.2.3.2.6 CPM Client Being Released from a CPM Group Session

The CPM Controlling Function can release a Participant from a CPM Group Session, e.g. when originator of the CPM Session leaves, when an Authorized Principal removes the Participant from the CPM Session or for other reasons.

The CPM Controlling Function SHALL end the CPM Group Session if the release of the Participant leaves the number of remaining Participants in the CPM Group Session as many as, or less than, the specified Participants left as indicated by “number-of-remaining-participants” operator configurable variable (as defined in sect. 9.2.9 of [CPM\_TS\_Conv]).

In order to release a participating CPM Client from a CPM Group Session, the CPM Controlling Function SHALL:

- stop sending Media belonging to this CPM Group Session to the CPM Client and discard any Media belonging to the CPM Group Session received from the CPM Client; and,
- send a CPM Session releasing request towards the CPM Client according to the Signalling Path used at CPM Session establishment.

The SIP/IP core routes the CPM Session releasing request according to the Signalling Path used at CPM Session establishment.

The CPM Participating Function of the released CPM User SHALL route the CPM Session releasing request according to the Signalling Path used at the CPM Session establishment.

#### 5.2.3.2.7 CPM Session Modification

CPM Session modification allows a CPM Client participating in a CPM Group Session or the CPM Controlling Function to add new Media Streams, to remove existing Media Streams or to change the characteristics of existing Media Streams.

NOTE 1: CPM Session modification can be initiated based on the CPM User input or for other reasons, e.g. due to moving to lower bandwidth IP network.

The CPM Participating Function and the CPM Controlling Function SHALL support CPM Session modification. A CPM Client that supports CPM Session establishment SHALL support CPM Session modification.

The CPM Client MAY initiate modification of a CPM Session or MAY create an additional CPM Session for the additional Media Stream.

NOTE 2: Using more than one CPM Session in a scenario that constitutes one logical real-time exchange to the user has impacts: when the user, for instance, requests the current list of Participants, this will be valid only for one of the CPM Sessions, and the user needs to be aware that the other CPM Sessions might have divergent sets of Participants. Also, when one of the CPM Sessions is terminated, the other CPM Sessions are not affected. The Media Streams used in the different CPM Sessions will not be correlated, mixed, etc. In this case, the correlation is left to implementation and may use information available at CPM Conversation level, e.g., parallel CPM Sessions carrying the same CPM Conversation Identity.

The CPM Controlling Function MAY initiate modification of a CPM Session according to group and service provider policies.

NOTE 3: CPM Session modification can be initiated based on a CPM Session modification request received from a CPM Client.

The SIP/IP core routes the CPM Session modification request according to the Signalling Path used at CPM Session establishment.

##### 5.2.3.2.7.1 CPM Session Modification Initiated by a CPM Client

In order to modify a CPM Session the CPM Client that wants to modify a CPM session SHALL send a CPM Session modification request according to the Signalling Path used at CPM Session establishment.

The CPM Participating Function of the originating CPM User SHALL route the CPM Session modification request according to the Signalling Path used at CPM Session establishment, if the CPM Session modification request satisfies the originating user preferences and service provider policies. Otherwise, the CPM Participating Function SHALL reject the session modification request.

When receiving a CPM Session modification request the CPM Controlling Function SHALL authorize the request according to the CPM Group policies and service provider policies and if authorized, the CPM Controlling Function SHALL initiate a CPM Session modification towards those CPM Clients participating in the CPM Session, which are affected by the changes introduced by the originating CPM Client. If not authorized, the CPM Controlling Function SHALL reject the CPM Session modification request.

#### 5.2.3.2.7.2 CPM Session Modification Initiated by the CPM Controlling Function

In order to modify a CPM Session with a participating CPM Client the CPM Controlling Function SHALL send a CPM Session modification request according to the Signalling Path used at CPM Session establishment for the CPM Client.

The CPM Participating Function of the terminating CPM User SHALL route the CPM Session modification request according to the Signalling Path used at CPM Session establishment, if the CPM Session modification request satisfies the terminating user preferences and service provider policies. Otherwise, the CPM Participating Function SHALL reject the CPM Session modification request.

#### 5.2.3.3 Extending a CPM 1-1 Session to a CPM Group Session

In order to extend an existing CPM 1-1 Session to a CPM Group Session (e.g. to add Participants to a CPM 1-1 Session), the extending CPM Client initiates the creation of a CPM Group Session with the list of Participants, including the Instance ID or the GRUU of the other Participant in the CPM 1-1 Session. In order to establish the CPM Group Session, the extending CPM Client SHALL send a CPM Session Invitation to the SIP/IP core serving the extending CPM Client, including in the CPM Session Invitation the address of the CPM Controlling Function provisioned to the CPM Client. The extending CPM Client SHALL indicate that for the other Participant of the CPM 1-1 Session, the existing session is to be replaced by the new session.

The CPM Participating Function SHALL treat the CPM Session Invitation as described in section **Error! Reference source not found.**

The CPM Controlling Function SHALL treat the CPM Session Invitation as described in section **Error! Reference source not found.** and include in the CPM Session Invitation to the original Participant of the CPM 1-1 Session the indication to replace the existing session with the new session.

Upon receiving a CPM Session Invitation for a CPM Group Session, each CPM Participating Function serving an invited CPM Client SHALL determine whether to reject the CPM Session Invitation based on service provider policies and user preferences, and SHALL either forward the CPM Session invitation to the invited CPM Client or reject the CPM Session Invitation.

Upon receiving a CPM Session Invitation for a CPM Group Session, the original participating CPM Client of the CPM 1-1 Session SHALL replace the existing CPM 1-1 Session with the new CPM Group Session, if accepted by the CPM Client, and SHALL end the CPM 1-1 Session by sending a session leaving request to the other CPM Client in the original CPM 1-1 Session using the same Signalling Path of the original CPM 1-1 Session.

Upon receiving a CPM Session Invitation for a CPM Group Session, the other invited CPM Clients SHALL either accept or reject the CPM Session Invitation based on user preferences.

Upon receiving at least one OK response the CPM Controlling Function SHALL set up the CPM Group Session with the CPM Clients that accepted the invitation.

#### 5.2.3.4 CPM Ad-hoc and Pre-defined Groups

A CPM Ad-hoc Group is a list of addresses created by the CPM User dynamically.

NOTE: CPM Ad-hoc Groups are not stored in the network.

The CPM User can populate the addresses in a CPM Ad-hoc Group in different ways (e.g. enter them manually or select them from his address book). For CPM Standalone Messages and CPM Session Invitations, the list of addresses constituting the CPM Ad-hoc Group is inserted in the body of the CPM Standalone Message or the CPM Session Invitation.

When the CPM Controlling Function receives in a CPM Standalone Message or a CPM Session Invitation a list of addresses, the CPM Controlling Function uses the list to send the CPM Standalone Message or the CPM Session Invitation to each member of the list.

A CPM Pre-defined Group is either a Pre-arranged Group or a Join-in Group. The owner of a CPM Pre-defined Group can add/delete members to/from the CPM Pre-defined Group. Policies are attached to the CPM Pre-defined Group to regulate the

allowed services and other membership rules for the CPM Pre-defined Group. CPM Pre-defined Groups are stored as defined in the Shared Group XDM specification [OMA-XDM\_Group-TS].

#### 5.2.3.4.1 Closed CPM Group Sessions

A CPM Ad-hoc Group Session is a CPM Closed Group Session when:

- the CPM Client populates the SDP attribute *a=chatroom* defined in [IETF-DRAFT-Chat] with the CPM reserved chat-token value of *'org.openmobilealliance.groupchat.closed'* to indicate the CPM Closed Group Session in the SIP INVITE. The SDP attribute value SHALL be: *"a=chatroom:org.openmobilealliance.groupchat.closed"*.

When the CPM Controlling Function determines that a CPM Ad-hoc Group Session is a CPM Closed Group Session, it prevents new participants to be added to the on-going CPM Session.

Current participants can choose to explicitly leave the session (i.e. SIP BYE), in which case they cannot re-join it at a later time, or when the Closed CPM Group Session is re-started.

#### 5.2.3.4.2 Long-lived CPM Group Sessions

The CPM Group Sessions are handled as CPM Long-lived Group Sessions, according to service provider policies.

When CPM Long-lived Group Sessions are re-started by any of the Participants allowed to do so by the policies of the service provider hosting the CPM Long-lived Group Session, the re-started CPM Long-lived Group Session preserves the same features (e.g. a Closed Session remains Closed whenever it is re-started).

The CPM Controlling Function keeps metadata of the CPM Long-lived Group Sessions, such as: CPM Conversation Identity, CPM Session Identity, list of participants to the CPM Group Session at the time of CPM Session inactivity timeout, or the indication of the type of CPM Group Session (e.g. Closed, Ad-hoc, Pre-defined).

If a CPM Message Storage exists for a Participant, the CPM Participating Function of that Participant stores collected metadata into the corresponding session history folder as:

- a session info object, stored at the beginning of the CPM Session, and
- a CPM Group State Object, stored at the end of the CPM Group Session.

#### 5.2.3.5 Participant Information

The CPM Client SHALL support requesting Participant Information.

The CPM Participating Function SHALL support forwarding a Participant Information subscription request from the CPM Client to the CPM Controlling Function and Participant Information notifications from the CPM Controlling Function to the CPM Client.

CPM Client that is reconnected by the serving CPM Participating Function to an active CPM Group Session after being unavailable (e.g. connection loss) SHALL re-refresh the subscription request to obtain the latest Participant Information or SHALL submit a new subscription to the conference focus.

If the subscription was successful the CPM Client SHALL use this latest Participant Information when re-connected into an active CPM Group Session.

The Participants Information is stored in the CPM Group State Object information by the CPM Participating Function at the termination of a CPM Group Session.

If the subscription resulted in an 404 NOT FOUND error, it means that the CPM Group Session is not longer available. In this case, if CPM Message Storage exists for a Participant, the CPM Client SHALL use the CPM Group State Object to retrieve the latest Participants Information.

The CPM Controlling Function SHALL support handling of requests for Participant Information.

In order to receive Participant Information a CPM Client SHALL send a Participant Information subscription request with a CPM Group Session Identity and a corresponding CPM Feature Tag to the SIP/IP core serving the CPM Client.

The SIP/IP core serving the CPM Client routes the Participant Information subscription request to the CPM Participating Function serving the CPM Client.

NOTE 1: The CPM Feature Tag is used by the SIP/IP core to route the Participant Information subscription request to the CPM Participating Function.

When receiving a Participant Information subscription request, the CPM Participating Function serving the CPM Client SHALL forward the Participant Information subscription request to the SIP/IP core of the network hosting the CPM Group Session identified by the CPM Group Session Identity via the SIP/IP core from which the Participant Information subscription request was received

NOTE 2: The CPM Group Session Identity is used by the SIP/IP core serving the CPM Client to route the Participant Information subscription request to the SIP/IP core of the network hosting the CPM Group Session.

The SIP/IP core of the network hosting the CPM Group Session routes the Participant Information subscription request to the hosting CPM Controlling Function.

NOTE 3: The CPM Feature Tag is used by the SIP/IP core of the network hosting the CPM Group Session to route the Participant Information subscription request to the hosting CPM Controlling Function

When receiving a Participant Information subscription request, the hosting CPM Controlling Function SHALL check that it hosts the targeted CPM Group Session and SHALL check the Participant Information subscription request against the CPM Group policies and service provider policies. If successfully authorized, the CPM Controlling Function SHALL

- respond with the Participant Information subscription successful response; and,
- send the initial Participant Information notification to the CPM Client using the reverse path of the Participant Information subscription request; and,
- continue sending the Participant Information notifications to the CPM Client using the reverse path of the Participant Information subscription request whenever the Participant Information changes until the Participant Information subscription becomes invalid.

Otherwise, if the Participant Information subscription request is not authorized, the CPM Controlling Function SHALL send back an error response.

### 5.2.3.6 Anonymous Participation

When an inviting CPM User wants to initiate a CPM Session anonymously, the CPM Client SHALL:

- include an anonymous CPM User identity; and,
- change the display name of the sender to an anonymous name; and,
- include an anonymity indication; and,
- include a temporary GRUU

in the CPM Session Invitation. In a CPM Session, the CPM Client using an anonymous CPM User identity SHALL NOT include any disposition notifications requests in the CPM Chat Messages sent.

The SIP/IP core removes the true identity of an inviting CPM User before forwarding the CPM Session Invitation to the invited CPM Client or to a not trusted SIP/IP core.

Upon receiving a CPM Session Invitation with an anonymity indication, the originating CPM Participating Function SHALL reject the CPM Session Invitation if service provider policies indicate that anonymity is not allowed.

Upon receiving a CPM Session Invitation without the true identity of the originating CPM User, the CPM Controlling Function SHALL reject the CPM Session Invitation if service provider policies indicate that participation in a CPM Group Session without a known true identity is not allowed.

Upon receiving a CPM Session Invitation with an anonymity indication, the CPM Controlling Function SHALL reject the CPM Session Invitation if service provider policies or the CPM Pre-defined Group policies indicate that anonymity is not allowed.

Upon receiving a CPM Session Invitation without the true identity of the originating CPM User, the terminating CPM Participating Function SHALL reject the CPM Session Invitation if service provider policies indicate that CPM Sessions without knowing the true identity of the originator are not allowed.

Upon receiving a CPM Session Invitation with an anonymity indication, the terminating CPM Participating Function SHALL reject the CPM Session Invitation if service provider policies indicate that anonymity is not allowed.

When an invited CPM User wants to accept a CPM Session Invitation anonymously, the CPM Client SHALL change the display name of the invited CPM User to an anonymous name and SHALL include an anonymity indication in the CPM Session Invitation successful response.

Upon receiving a CPM Session Invitation successful response with an anonymity indication, the terminating and originating CPM Participating Functions SHALL reject the CPM Session Invitation if service provider policies indicate that anonymity is not allowed and SHALL release the CPM User from the CPM Session.

Upon receiving a CPM Session Invitation successful response with an anonymity indication, the CPM Controlling Function SHALL release the CPM User from the CPM Session if service provider policies or the Pre-defined CPM Group policies, indicate that anonymity is not allowed.

The CPM Controlling Function SHALL include in the Participant Information the CPM User identities as anonymous only for CPM Users which requested anonymity in the CPM Session establishment.

If anonymity is allowed, the CPM Controlling Function SHALL either allow the anonymous display name of the requesting CPM User to be sent to other Participants in the CPM Session, or MAY assign a Pseudonym to the requesting CPM User.

### 5.2.3.7 Pseudonym Negotiation

A CPM User can negotiate a unique Pseudonym when requesting to join anonymously in a CPM Group Session.

When an inviting CPM User wants to initiate a CPM Group Session anonymously, the CPM Client MAY include a Pseudonym specified by the CPM User in the CPM Session Invitation.

Upon receiving a CPM Session Invitation with a Pseudonym, the CPM Participating Function SHALL reject the CPM Session Invitation if the service provider policies indicate that the use of Pseudonyms is not allowed or the requested Pseudonym is not allowed.

Upon receiving a CPM Session Invitation with a Pseudonym, the CPM Controlling Function SHALL:

- reject the CPM Session Invitation if service provider policies or the CPM Pre-defined Group policies indicate that the use of Pseudonyms is not allowed or the requested Pseudonym is not allowed according to the service provider policies and Pre-defined CPM Group policies; or,
- assign a different Pseudonym if the requested Pseudonym is not unique for the CPM Group Session.

When an invited CPM User wants to accept a CPM Session Invitation anonymously, the CPM Client MAY include a Pseudonym specified by the CPM user in the CPM Session Invitation successful response.

Upon receiving a CPM Session Invitation successful response with a Pseudonym, the terminating CPM Participating Function SHALL reject the CPM Session Invitation if the service provider policies indicate that the use of Pseudonyms is not allowed or the requested Pseudonym is not allowed and SHALL release the CPM User from the CPM Group Session.

Upon receiving a CPM Session Invitation successful response with a Pseudonym, the CPM Controlling Function SHALL:

- release the CPM User from the CPM Group Session if the service provider policies or the CPM Pre-defined Group policies indicate that the use of Pseudonyms is not allowed or the requested Pseudonym is not allowed. or
- assign a different Pseudonym if the requested Pseudonym is not unique for the CPM Group Session.

NOTE: When a different Pseudonym is assigned, the CPM User can get to know this assigned Pseudonym via subscription to the Participant Information.

### 5.2.3.8 Media Streams Usage in CPM Sessions

A CPM Session can consist of one or more Media Streams, each of a specific Media Stream Type e.g. audio, video, or message. The Media Streams are negotiated in the CPM Session initiation procedure and can be later on re-negotiated using the CPM Session modification procedure.

Each offered and accepted Media Stream contains the characteristics of the Media Stream Type. The characteristics of each Media Stream are negotiated during the CPM Session establishment and these characteristics can be modified during the CPM Session by the CPM Session modification procedure.

#### 5.2.3.8.1 Media Streams Usage in CPM Session Initiation

CPM Clients, CPM Participating Functions and CPM Controlling Functions SHALL include the offered Media Streams in CPM Session Invitations they send.

When sending a CPM Session Invitation, the inviting CPM Client SHALL offer one or more Media Streams of the Media Stream Types supported by the CPM Client.

NOTE 1: The CPM Client can offer the Media Streams based on the CPM User's input.

When sending a CPM Session Invitation, the CPM Participating Function SHALL offer all or a non-empty subset of the Media Streams offered in the received CPM Session Invitation according to the service provider policies.

NOTE 2: The CPM Participating Function can restrict the offered Media Streams, e.g. because of the allowed Media Stream Types in service provider policies.

NOTE 3: If no Media Streams are authorized, the CPM Session Invitation is rejected.

When sending a CPM Session Invitation, the CPM Controlling Function SHALL offer all or a non-empty subset of the Media Streams offered in the received CPM Session Invitation according to the hosted CPM Group policies and service provider policies.

NOTE 4: The CPM Controlling Function can restrict the offered Media Streams, e.g. because of the allowed Media Stream Types in the CPM Group policies.

NOTE 5: If no Media Streams are authorized, the CPM Session Invitation is rejected.

If a CPM Session Invitation is accepted by a CPM Client, the CPM Client SHALL accept all or a non-empty subset of the offered Media Streams.

NOTE 6: Those Media Streams, which are not accepted, are rejected.

NOTE 7: The CPM Client can accept the offered Media Streams based on the CPM User decision.

NOTE 8: If no Media Streams are acceptable, the CPM Session Invitation is rejected.

CPM Clients, CPM Participating Functions and CPM Controlling Functions SHALL include the accepted Media Streams in the successful CPM Session Invitation response.

When sending a successful CPM Session Invitation response, the invited CPM Client SHALL accept all or a non-empty subset of the offered Media Streams supported by the CPM Client.

When sending a successful CPM Session Invitation response, the CPM Participating Function SHALL accept the Media Streams accepted in the received CPM Session Invitation response.

When sending a successful CPM Session Invitation response, the CPM Controlling Function SHALL accept the Media Streams accepted in the received CPM Session Invitation response.

The handling of Media Streams in a CPM Session Invitation by the Interworking Function is detailed in section 5.3.1.3.

### 5.2.3.8.2 Media Streams Usage in CPM Session Modification

CPM Clients, CPM Participating Functions and CPM Controlling Functions SHALL include the offered Media Streams in a CPM Session modification request and SHALL include the accepted Media Streams in a successful CPM Session modification response.

When sending a CPM Session modification request the originating CPM Client SHALL offer one or more Media Streams of the Media Stream Types supported by the originating CPM Client.

NOTE 1: The offered Media Streams include all existing Media Streams which the CPM Client wishes to continue using and all new Media Streams which the CPM Client wishes to start using . The Media Streams which the CPM Client currently uses and wishes to not use any longer are not offered.

NOTE 2: Characteristics of an offered Media Stream can be the same or different from the currently used characteristics of the Media Stream.

When sending a CPM Session modification request, the CPM Participating Function of the originating CPM Client SHALL offer all or a non-empty subset of the Media Streams offered in the received CPM Session modification request according to service provider policies.

NOTE 3: If no Media Streams are authorized, the CPM Session modification request is rejected.

When receiving a CPM Session modification request the CPM Controlling Function SHALL send a CPM Session modification request to the other participating CPM Clients, if

- there is at least one offered Media Stream which is not currently used in the CPM Session by any participating CPM Client and the originating CPM Client is authorized to add such a Media Stream to the CPM Session according to the hosted CPM Group policies and service provider policies; or,
- there is at least one Media Stream currently used by the originating CPM Client which is not offered by the originating CPM Client and the originating CPM Client is authorized to remove this Media Stream from all Participants according to the hosted CPM Group policies and service provider policies.

When sending a CPM Session modification request towards a terminating CPM Client the CPM Controlling Function SHALL offer:

- the Media Streams currently used by the terminating CPM Client not including the Media Streams currently used by the originating CPM Client which were not offered anymore by the originating CPM Client in the received CPM Session modification request and which the originating CPM Client is authorized to remove from all Participants according to the hosted CPM Group policies and service provider policies; and,
- the Media Streams offered by the originating CPM Client which are not currently used in the CPM Session by any participating CPM Client, if the originating CPM Client is authorized to add such Media Streams to the CPM Session according to the hosted CPM Group policies and service provider policies.

When handling the Media Streams offered by the originating CPM Client, the CPM Participating Function SHALL ignore offered Media Streams with no characteristics supported by the CPM Participating Function.

When sending a CPM Session modification request the CPM Participating Function of the terminating CPM Client SHALL offer all or a non-empty subset of the Media Streams offered in the received CPM Session modification request according to the user preferences of the terminating CPM User and the service provider policies.

NOTE 4: If no Media Streams are authorized, the CPM Session modification request is rejected.

When sending a successful CPM Session modification response the terminating CPM Client SHALL accept all or a non-empty subset of the offered Media Streams supported by the terminating CPM Client and accepted by the CPM User.

When sending a successful CPM Session modification response the CPM Participating Function of the terminating CPM Client SHALL accept the Media Streams accepted in the received CPM Session modification response.

When sending a successful CPM Session modification response the CPM Controlling Function SHALL accept:

- the Media Streams offered in the received CPM Session modification request which are currently used in the CPM Session by any participating CPM Client; and,

- the Media Streams offered in the received CPM Session modification request which are not used yet by any participating CPM Client and which the originating CPM Client is authorized to add according to the hosted CPM Group policies and service provider policies.

NOTE 5: Service provider policy determines how to handle session modification when only a subset of the participants accepts the session modification. E.g., the session modification may be accepted or rejected based on the number of participants that accept the addition or the removal of a specific Media Type.

When sending a successful CPM Session modification response the CPM Participating Function of the originating CPM Client SHALL accept the Media Streams accepted in the received CPM Session modification response.

When receiving a failure CPM Session modification response for a CPM Session modification request not offering a Media Stream currently used by the terminating CPM Client the CPM Controlling Function SHALL stop forwarding to the terminating CPM Client the Media of the Media Streams, which are currently used by the terminating CPM Client and which were not offered to the terminating CPM Client. The modification is forced on the terminating CPM Client even if the terminating CPM Client does not accept the modification, since the originating CPM Client will stop sending this Media Stream.

### 5.2.3.9 CPM Session Media Plane Handling

The CPM Client MAY send Media within the negotiated Media Streams.

The CPM Controlling Function SHALL

- for each full duplex Media Stream of a real time Media Stream Type
  - send the mixed Media received from the other participating CPM Clients to each participating CPM Client which uses the Media Stream;
- for each simplex Media Stream of a real time Media Stream Type
  - send the Media received from the sending participating CPM Client to each receiving participating CPM Client which uses the Media Stream;
- for each Media Stream of a non real time Media Stream Type
  - send the Media received from the other participating CPM Client to each participating CPM Client which uses the Media Stream; and,

A CPM Participating Function MAY stay in the Media path.

When staying in the Media path, the CPM Participating Function for a CPM Client SHALL relay any Media received as part of a CPM Session to and from the CPM Client.

### 5.2.3.10 Deferring CPM Chat Messages

If the CPM User has the CPM Message deferral enabled, the CPM Participating Function triggers the deferring of the received CPM Chat Messages and their respective disposition notifications, in the following cases:

- when a CPM User previously engaged in the CPM Session loses connection, or
- when a CPM User has neither accepted nor rejected a CPM Group Session invitation.

In such cases, CPM Participating Function stays in the media path on behalf of the CPM Client. When the CPM Client becomes available, the deferred delivery of the missed CPM Chat Messages is performed according to the procedures described in sect. 5.2.5 “*Handling of Deferred CPM Messages*”.

Note: if the CPM Participating Function must secure deferral of the CPM Chat Messages, then the CPM Participating Function SHALL remain in the CPM Group Session on behalf of the CPM Client. Hence, the CPM Controlling Function will assume that the Participant is available,

## 5.2.4 CPM File Transfer Concepts

CPM File Transfer allows transferring one or more files with discrete Media contents between Users, where the recipient explicitly has to acknowledge his/her willingness to receive the file(s) before any data is transferred. Depending on Service Provider policies, the CPM Enabler MAY support initiating and handling of CPM File Transfers to off line subscribers, either by temporarily storing the file, or by delivering the file to the CPM User's CPM Message Store, subject to recipient User preferences. Because a CPM File Transfer activity could be deferred, the sending CPM User may request delivery notification of when the recipient CPM Client has completed the transfer.

Use of the File Transfer Thumbnail can be controlled and provisioned per CPM User by the service provider, independently of the support for CPM File Transfer. There are NNI considerations for this feature.

### 5.2.4.1 CPM File Transfer to a Single Recipient

CPM File Transfers to a single recipient follow the same procedures as defined for CPM 1-1 Sessions in section **Error! Reference source not found.** except for the differences noted in the subsections below.

#### 5.2.4.1.1 CPM File Transfer Initiation

CPM File Transfer initiations towards a single recipient follow the same procedures as defined in section **Error! Reference source not found.** for initiating a CPM 1-1 Sessions. When setting up the session for a CPM File Transfer, the CPM Client SHALL:

- include in the invitation only one Media Stream Type; and
- SHALL include a filename for each of the files to be transferred; and
- MAY include a thumbnail for each of the files to be transferred; and
- SHALL specify the Media Stream(s) as unidirectional from the sender to the recipient; and
- MAY include a delivery notification request.

Once the session for the CPM File Transfer has been established, the file(s) are transferred via the Media Plane.

#### 5.2.4.1.2 Closing a CPM File Transfer

Closing a CPM File Transfer to a single recipient follows the same procedures as defined in section **Error! Reference source not found.** for closing a CPM 1-1 Session. Either the sender or the receiver may close the CPM File Transfer at any time. It is expected that the sender closes the CPM File Transfer after sending the last file.

#### 5.2.4.1.3 No Answer from Recipient

In situations where the recipient CPM User fails to respond to the CPM File Transfer invitation within a service provider or terminating network specified time period, either the originating or terminating CPM Participating Function MAY accept the invitation on behalf of the recipient CPM Client for subsequent delivery.

The decision whether the originating or the terminating CPM Participating Function accepts the CPM File Transfer invitation depends on service provider policies. The default behaviour SHOULD be that the terminating CPM Participating Function accepts the invitation when there is no timely response from the recipient.

### 5.2.4.2 CPM File Transfer to a CPM Group

The procedures used for CPM File Transfer to a CPM Group are a subset of the procedures defined in section **Error! Reference source not found.** for CPM Group Sessions. The procedures used and any differences to them are described in the subsections below.

#### 5.2.4.2.1 CPM File Transfer Initiation

CPM File Transfer initiation towards a CPM Group follows the same procedures as defined in section **Error! Reference source not found.** for initiating a CPM Group Session. When setting up the session for a CPM File Transfer, the CPM Client SHALL:

- include in the invitation only on Media Stream Type; and

- SHALL include a filename; and
- MAY include a thumbnail; and
- SHALL specify the Media Stream(s) as unidirectional from the sender to the recipient; and
- MAY include a delivery notification request.

Once the session for the CPM File Transfer has been established, the file is transferred via the Media Plane.

#### 5.2.4.2.2 CPM Client Leaving a CPM File Transfer to a CPM Group

Closing a CPM File Transfer to a CPM Group follows the same procedures as defined in section **Error! Reference source not found.** for closing a CPM Group Session. The CPM Client that initiated the CPM File Transfer is expected to close it once it has sent one file. The leg to each recipient is torn down once the file has finished being transferred.

#### 5.2.4.3 CPM Ad-hoc and Pre-defined Groups in a CPM File Transfer

CPM File Transfers to a CPM Group can be set up using CPM Ad-Hoc Groups and CPM Pre-defined Groups. These two types of CPM Groups are as described in section **Error! Reference source not found.**

#### 5.2.4.4 Anonymous Participation in a CPM File Transfer

For anonymous participation in a CPM File Transfer, the same mechanisms as described in section **Error! Reference source not found.** for anonymous participation in a CPM Group Session apply.

#### 5.2.4.5 CPM File Transfer to an off-line recipient

The CPM Enabler MAY support, based on Service Provider policies, that the sending CPM User initiates a CPM File Transfer session even if the intended recipient CPM User is off-line. Either the originating or terminating CPM Participating Function may accept the invitation on behalf of the recipient's CPM Client.

The decision whether the originating or the terminating CPM Participating Function accepts the CPM File Transfer invitation depends on Service Provider policies. The default behaviour SHOULD be that the terminating CPM Participating Function accepts the invitation. If the originating CPM Participating Function supports delivery to off-line recipients, it MAY trigger Store & Forward.

**Editor's Note: It is FFS how the S&F capability is communicated from the originating CPM Client (ref. RCS 5.1 approach)**

#### 5.2.4.6 Resuming an interrupted CPM File Transfer

During the CPM File Transfer session, it is the responsibility of the CPM Clients to keep track of how much of the data has been sent or received (as applicable). If the File Transfer session is interrupted, the affected CPM Client MAY request to resume the interrupted file transfer from the point of interruption, without having to transfer the entire file again. When requesting to resume a file transfer, the same mechanisms as described in this section are used, with the clarification that appropriate attributes and values are used according to [RFC5547] to indicate the point at which the CPM File Transfer session activity is to resume.

#### 5.2.4.7 Deferred and subsequent delivery of a CPM File Transfer

The deferral and subsequent delivery may take place in one of the following ways:

- When the terminating CPM Participating Function accepted the CPM File Transfer session:
  1. in the case of an expired CPM File Transfer invitation, if the CPM User indicates to the CPM Client to request the retrieval of the missed File Transfer, the CPM Client SHALL send a new CPM File Transfer invitation to the CPM Participating Function from which the missed invitation was last received, but with the direction being "receive only", i.e. a pull request. The CPM Participating Function SHALL accept the session and establish an appropriate Media Plane session to transfer the file to the recipient; or
  2. based on the 3<sup>rd</sup> party register of the recipient, the terminating CPM Participating Function SHALL re-send the CPM File Transfer invitation, as if it came from the originator.

- When the originating CPM Participating Function accepted the CPM File Transfer session:
  1. If the originator and terminating CPM Users are in the same network, and the originating CPM Participating Function receives the 3<sup>rd</sup> party register of terminating CPM User re-send the CPM File Transfer invitation, as if it came from the originator; or
  2. If, the CPM Participating Function cannot receive the 3<sup>rd</sup> party registration of the recipient CPM User, it will periodically attempt to re-send CPM File Transfer invitations. The timing and frequency of such retries are subject to Service Provider policies.

In all cases when handling deferred or subsequent delivery of CPM File Transfers, the invitation from the terminating CPM Participating Function SHALL NOT contain any delivery notification request, and the CPM Participating Function which sent the deferred CPM File Transfer invitation SHALL ensure that any request for delivery notification contained in the original invitation is delivered back to the originator once the recipient has completed the transfer.

## 5.2.5 Handling of Deferred CPM Messages

As described in section 5.2.2.1.3 for CPM Standalone Message and 5.2.3.10 for the CPM Session, the CPM Participating Function may decide to defer a CPM Message for later delivery. For the delivery of Deferred CPM Messages, the CPM Participating Function SHALL support three options, to be taken depending on user preferences:

1. If user preferences indicate to send a notification and wait for the CPM Client(s) of the recipient CPM User to take action for the delivery of the CPM Standalone Messages, the CPM Participating Functions proceeds as specified in section 5.2.5.1;
2. If user preferences indicate to push Deferred CPM Messages once one of the CPM Clients of the recipient CPM User becomes available, the CPM Participating Function proceeds as specified in section 5.2.5.2;
3. If user preferences indicate to hand over Deferred CPM Messages to the Message Storage Server, the CPM Participating Function proceeds as specified in section 5.2.5.3.

NOTE 1: In the case that the CPM User has not set one of the above three preferences, server provider policies guide which option to take.

NOTE 2: In the first two options, if a Deferred CPM Message expires before it is delivered, the CPM Participating Function SHALL handle the Deferred CPM Message based on user preferences and service provider policies. The result will be that an expired Deferred CPM Message is deleted and not delivered to the CPM User or it is delivered to the Message Storage Server.

### 5.2.5.1 Sending Notifications and Awaiting CPM Client Action

If the recipient does not want to be disturbed and this procedure is entered, the CPM Participating Function SHALL NOT send notifications to the recipient's CPM Client(s) until the user preference to not disturb has changed.

Notifications about Deferred CPM Standalone Messages can be sent in-band or out-of-band, depending on the CPM registration state of the corresponding CPM Client. In-band notifications are sent via the SIP/IP core while out-of-band notifications are handed over to the Push Enabler for delivery.

When no suitable CPM Client is registered, the CPM Participating Function MAY send an out-of-band notification to all or some of the CPM User's devices. The corresponding CPM Clients MAY then register themselves to the CPM service.

When the CPM Participating Function wishes to send a notification about Deferred CPM Standalone Messages and at least one suitable CPM Client is registered, the CPM Participating Function SHALL send an in-band notification to these CPM Clients including information about the Deferred CPM Standalone Messages. This notification SHALL include CPM Standalone Message related information, such as subject, originator address, date, size, etc.

The CPM Client SHALL allow the CPM User to determine whether none, some or all of the Deferred CPM Standalone Messages are to be retrieved, or if other actions would be more appropriate (like interworking, discarding, delivering to the Message Storage Server or keeping deferred). If any Deferred CPM Standalone Messages are selected by the CPM User to be retrieved, interworked, discarded, or delivered to the Message Storage Server, the CPM Client SHALL establish a session with the CPM Participating Function and SHALL include the identity of the Deferred CPM Standalone Messages and for each of the identified Deferred CPM Standalone Messages the action to be performed. Once the actions on the identified

Deferred CPM Standalone Messages have been completed and all Deferred CPM Standalone Messages that were to be retrieved have been retrieved, the CPM Client SHALL tear down the session.

NOTE: This session is NOT a CPM Session

After retrieval, interworking, discarding, or delivering to the Message Storage Server of the selected Deferred CPM Standalone Messages, the CPM Participating Function SHALL remove these Deferred CPM Standalone Messages from the deferred message queue. Deferred CPM Standalone Messages indicated by the CPM User to be handled differently (i.e. interworking, discarding or delivering to the Message Storage Server) will be handled accordingly.

### 5.2.5.2 Pushing Deferred CPM Messages

If the recipient does not want to be disturbed and this procedure is entered, the CPM Participating Function SHALL NOT send notifications or push messages to the recipient's CPM Client(s) until the user preference to not disturb has changed. If there is no suitable CPM Client registered, the CPM Participating Function MAY send an out-of-band notification (as described in section 5.2.5.1) to all known, unregistered clients. The CPM Participating Function SHALL wait until at least one CPM Client has registered (e.g., when the CPM Participating Function receives a Registration Event Information about one of the CPM Clients of the CPM User). Then:

- the CPM Participating Function SHALL deliver all Deferred CPM Message(s) for which suitable CPM Client(s) are registered.
- a CPM Session (include disposition notifications) is delivered to any one of CPM Clients on CPM User's devices that becomes available.
- CPM disposition notifications that are received at a later time, for Chat Message(s) of a CPM Group Session for which deferred delivery already occurred, will be delivered by the CPM Participating Function through one of the following:
  - in a dedicated session for the disposition notifications towards the originator's CPM Client that sent the respective CPM Chat Message(s), or
  - as separate messages, following the delivery procedures of the Pager Mode CPM Standalone Messages.

NOTE: It might happen that a subset of the deferred messages is pushed to one device and another subset is pushed to another device. The union of all these subsets is considered delivered afterwards.

If not all Deferred CPM Messages could be delivered, the CPM Participating Function MAY send a new out-of-band notification (as described in section 5.2.5.1) to all known, unregistered clients. The CPM Participating Function SHALL wait until at least one such CPM Client has registered (e.g., when the CPM Participating Function receives a Registration Event Information about one of the CPM Clients of the CPM User). After such registration, the CPM Participating Function SHALL repeat the above steps of message delivery.

After message delivery, the CPM Participating Function SHALL remove those Deferred CPM Messages from the deferred message queue that were delivered to the registered CPM Client(s).

### 5.2.5.3 Handing over Deferred CPM Messages to the Message Storage Server

The CPM Participating Function SHALL:

1. Re-set the expiry time of the CPM Message according to service provider policies, and store the CPM Message in the Message Storage Server,
2. Respond with positive delivery notification to the delivery notification request for the Deferred CPM Message as delivered.

Upon receiving a Deferred CPM Message, the Message Storage Server SHALL:

1. Store the Deferred CPM Message as a temporary stored object,
2. Notify the Message Storage Client of the arrival of the new message, and
3. Monitor the message's expiry time and:

- a. Either discard the Deferred CPM Message if the expiry is reached before the CPM User accessed the message, or
- b. Store the Deferred CPM Message as a stored object.

## 5.2.6 Disposition Notification

When a CPM User wants to receive the disposition-state of a CPM Message he/she sends, the CPM Client SHALL request a disposition notification when sending the CPM Message. Disposition notifications for CPM are delivery notifications and read reports. In the case that a CPM Message containing a disposition notification request is targeted at a CPM Pre-defined Group or a CPM Ad-hoc Group or multiple different types of disposition notifications are requested for the same CPM Standalone Message, the originating user MAY receive aggregated disposition notifications based on service provider policies. Aggregating disposition notifications MAY be performed by the originating CPM Participating Function or CPM Controlling Function.

NOTE: It is not possible to request for disposition notifications when sending CPM Messages anonymously.

When sending a disposition notification to a CPM User, a CPM functional component SHALL target the disposition notification towards the CPM Address of the CPM User from which the original CPM Message was sent.

When sending a disposition notification to a non-CPM Principal, the CPM Client SHALL, in the disposition notification, accompany the target non-CPM address with the Non-CPM Communication Service Identifier included in the received CPM Message.

In case of delivery of a CPM Message to multiple devices, the terminating CPM Participating Function SHALL, for each disposition notification type (i.e. delivery notification and read report), forward the first disposition notification received from one of the devices that the CPM Message was delivered to the originator of the CPM Message and SHALL suppress the forwarding of subsequent disposition notifications received from the other devices that the CPM Message was delivered to.

In case aggregated disposition notifications are supported, the CPM Participating Function serving the CPM User may aggregate the received disposition notifications.

### 5.2.6.1 Delivery Notification

There are two kinds of delivery notifications:

- Positive-delivery notifications, which indicate that a CPM Message was successfully delivered to the intended recipient, and
- Negative-delivery notifications, which indicate that a CPM Message could not be delivered to the intended recipient.

A positive-delivery notification indicates that the requested CPM Message was either received by at least one of the CPM Clients or non-CPM clients on the devices of the recipient or delivered to the Message Storage Server of the recipient. A negative-delivery notification indicates that the requested CPM Message was neither received by one of the CPM Clients or non-CPM clients on the devices of the intended recipient nor delivered to the Message Storage Server of the recipient.

When requesting a delivery notification, the CPM Client SHALL request a positive-delivery notification, a negative-delivery notification or both of them in a CPM Standalone Message.

When a CPM Client receives a CPM Message containing a positive-delivery notification request, the CPM Client SHALL send a CPM Message delivery notification to notify the originating CPM User of the successful delivery of the CPM Message, if allowed by the recipient CPM User.

In the case that a CPM Participating Function decides that interworking is to occur for a CPM Message containing a delivery notification request, the procedures described in section 5.3.1.3 apply for the creation of delivery notifications.

In the case that a terminating CPM Participating Function delivers a CPM Message with a positive-delivery notification request to the Message Storage Server, the terminating CPM Participating Function SHALL remove the positive-delivery notification request from the CPM Message and SHALL notify the originating CPM User of the successful delivery of the message unless service provider policies prevent this.

In the case that a terminating CPM Participating Function fails to deliver a CPM Message containing a negative-delivery notification request to the recipient, the CPM Participating Function SHALL notify the originating CPM User of the failure of the message delivery unless service provider policies prevent this.

In the case that the originating CPM Participating Function fails to deliver a CPM Message containing a negative-delivery notification request, the CPM Participating Function SHALL NOT generate a negative-delivery notification since the originating CPM User is already aware of the failure.

When receiving a CPM Message containing a delivery notification request, the CPM Controlling Function SHALL insert in the CPM Message an indication that indicates the corresponding CPM Message delivery notification should be routed back through the CPM Controlling Function unless service provider policies prevent this.

When receiving a CPM Message delivery notification, the CPM Controlling Function hosting a CPM Pre-defined Group or the CPM Participating Function serving the CPM User who requested the CPM Message delivery notification MAY aggregate the received CPM Message delivery notification to the CPM User originating the CPM Message unless service provider policies prevent this.

### 5.2.6.2 Read Report

A read report indicates whether the CPM Message was read or the contents were otherwise rendered to the intended recipient. When a receiving CPM client receives a CPM Message containing a read report request and the contents enclosed in its body was displayed or otherwise rendered to the recipient, the CPM client SHALL, if authorized by the recipient, generate a read report and send it to the originating CPM User.

NOTE 1: The authorization by the recipient is governed by a setting in his/her CPM Client, and may involve interaction with the recipient in order to authorize generation of read reports on a case-by-case basis.

NOTE 2: It is CPM Client implementation-specific to determine when a CPM Message is considered to have been read and may be based on rendering of the CPM Message to the CPM User.

## 5.3 CPM Interworking

### 5.3.1 Interworking from CPM to Non-CPM

This subsection describes routing of CPM Session Invitations, CPM File Transfer requests, CPM Standalone Messages and disposition notifications to non-CPM destinations.

NOTE: Signalling responses to a CPM Session Invitation, a CPM File Transfer request, a CPM Standalone Message or a disposition notification follow the reverse path of the CPM Session Invitation, the CPM File Transfer request, the CPM Standalone Message or the disposition notification.

The general interworking process of CPM consists of three separate steps:

1. Interworking decision; this is the process performed by the CPM Participating Function or the CPM Controlling Function to determine that a CPM Standalone Message, a CPM File Transfer request, a CPM Session Invitation, or a disposition notification is to be interworked towards a Non-CPM Communication Service.
2. Interworking selection; this is the process performed by the Interworking Selection Function to select to which Non-CPM Communication Service to interwork to.
3. Interworking execution; this is the process, performed by the Interworking Function selected in the previous step, to translate the CPM request into a request for the selected Non-CPM Communication Service and subsequent routing of the non-CPM request to the selected Non-CPM Communication Service.

For a CPM originator to be able to receive replies from non-CPM recipients, a CPM originator must already have or be assigned during interworking a routable address in the respective Non-CPM Communication Service domain.

No CPM originator user preferences are involved in the interworking decisions, neither in the decision to interwork, nor in the decision towards which domain to interwork.

### 5.3.1.1 Interworking Decision

The decision to interwork happens in the originating network, the controlling network, or the terminating network.

There are different scenarios in which the CPM Participating Function and the CPM Controlling Function decide whether interworking is to occur. These scenarios are detailed in each of the following subsections.

#### 5.3.1.1.1 Not Registered CPM User

When the terminating CPM Participating Function receives a CPM Session Invitation, a CPM File Transfer request or a CPM Standalone Message and the target CPM User is currently not registered for the CPM service, the terminating CPM Participating Function SHALL determine how to handle the received CPM request based on the service provider policies and user preferences of the target CPM User. If the decision is to deliver the CPM request via interworking, the terminating CPM Participating Function SHALL send it to the Interworking Selection Function.

NOTE: The user preferences can indicate other methods of handling the CPM Standalone Message, the CPM File Transfer request or the CPM Session Invitation, e.g. to defer the CPM Message.

#### 5.3.1.1.2 User Preferences Indicate Interworking

When the terminating CPM Participating Function receives a CPM Session Invitation, a CPM File Transfer request or a CPM Standalone Message and the user preferences of the target CPM User indicate that the CPM User wishes to have this CPM request be interworked to an address specified in his user preferences, the terminating CPM Participating Function SHALL send the CPM request to the Interworking Selection Function.

#### 5.3.1.1.3 Service Provider Policies Indicate Interworking upon Non-Delivery

When the terminating CPM Participating Function receives a CPM Session Invitation, a CPM File Transfer request or a CPM Standalone Message and the CPM request cannot be handled inside the CPM System (e.g. a message delivery failed, no CPM Client matches the delivery criteria), the terminating CPM Participating Function SHALL inspect the service provider policies. If the service provider policies indicate interworking for this scenario, the terminating CPM Participating Function SHALL send the CPM request to the Interworking Selection Function.

#### 5.3.1.1.4 Non-CPM Principal with a Valid SIP Address

When the terminating network receives a CPM Standalone Message, CPM File Transfer request or a CPM Session Invitation targeted to a non-CPM subscriber, it will determine handling of the CPM Standalone Message, CPM File Transfer request or CPM Session Invitation based on the user's provisioned service profile.

NOTE: The terminating network may deliver the CPM Standalone Message, CPM File Transfer request or CPM Session Invitation to another application server or directly to the user or it may return an error message.

When the originating CPM Participating Function or the CPM Controlling Function receive an error response, based on service provider policies, they either

- SHALL return an error response back to the originating CPM User; or,
- SHALL send the CPM Session Invitation, the CPM File Transfer request or the CPM Standalone Message to the Interworking Selection Function.

#### 5.3.1.1.5 Missing CPM Support in the Controlling or Terminating Network

When the originating CPM Participating Function or the CPM Controlling Function receive an error response stating that a CPM Session Invitation, CPM File Transfer request or CPM Standalone Message failed due to no CPM Enabler support in the target network, based on service provider policies, they either

- SHALL return an error response back to the originating CPM User; or,
- SHALL send the CPM Session Invitation, the CPM File Transfer request or the CPM Standalone Message to the Interworking Selection Function.

### 5.3.1.1.6 Missing SIP/IP Core Support in the Controlling or Terminating Network

When the originating CPM Participating Function or the CPM Controlling Function receive an error response stating that a CPM Session Invitation, CPM File Transfer request or CPM Standalone Message failed due to no SIP/IP core support in the target network, based on service provider policies, they either

- SHALL return an error response back to the originating CPM User; or
- SHALL send the CPM Session Invitation, the CPM File Transfer request or the CPM Standalone Message to the CPM Interworking Selection Function.

### 5.3.1.1.7 Recipient not Routable via SIP/IP Core

When the originating CPM Participating Function or the CPM Controlling Function receive a CPM Session Invitation, CPM File Transfer request or CPM Standalone Message targeted to an address whose URI scheme is not routable in SIP they either:

- SHALL reject the CPM Session Invitation, the CPM File Transfer request or the CPM Standalone Message; or,
- SHALL send the CPM Session Invitation, the CPM File Transfer request or the CPM Standalone Message to the Interworking Selection Function.

NOTE: The wv: URI scheme or mailto: URI scheme are examples of the URI schemes not routable in SIP.

## 5.3.1.2 Interworking Selection

Interworking selection happens in the network where the decision to interwork is made.

Upon receiving a CPM Standalone Message, a CPM File Transfer request, a CPM Session Invitation, or a disposition notification, the Interworking Selection Function SHALL select an Interworking Function. The Interworking Selection Function MAY base its selection on the following criteria:

- The Non-CPM Communication Service Identifier that accompanies the non-CPM address, if present.
- The service provider policies.
- The target user preferences, when operating in the terminating network.
- The target user's presence information, if available.
- Message characteristics (such as message size and content).
- File transfer characteristics (such as file size and content).
- Session characteristics (such as Media used in the session).

An example implementation of this decision making process is described in Appendix E.

When the Interworking Selection Function receives a disposition notification targeted to a non-CPM address accompanied with a Non-CPM Communication Service Identifier, the Interworking Selection Function SHALL bypass the selection process described above, and SHALL select the Interworking Function corresponding to the Non-CPM Communication Service Identifier. If the received disposition notification does not contain a Non-CPM Communication Service Identifier, the Interworking Selection Function SHALL reject the disposition notification.

When selecting a Non-CPM Communication Service, the Interworking Selection Function SHALL take into account only those Non-CPM Communication Services, in which the CPM originator already has or can be assigned during interworking a routable Non-CPM Communication System address.

The Interworking Selection Function SHALL send the CPM Session Invitation, the CPM File Transfer request, the CPM Standalone Message or the disposition notification to the selected Interworking Function. The Interworking Selection Function SHALL indicate in the CPM request the address to be used for routing in the Non-CPM Communication Service. The address MAY be the one received in the CPM Standalone Message, the CPM File Transfer request, the CPM Session Invitation or the disposition notification, or, when interworking occurs in the terminating network, the address MAY be taken from the target user preferences. Other means (e.g. fixed address mappings) to determine the address MAY be used.

NOTE 1: If the address for the recipient is not valid for the selected Non-CPM Communication Service, the Interworking Selection Function does not attempt interworking and instead rejects the CPM request with an error response.

If the Interworking Selection Function receives a failure response for a CPM Standalone Message, a CPM File Transfer request or a CPM Session Invitation sent towards a Non-CPM Communication Service via an Interworking Function, the Interworking Selection Function MAY attempt to deliver the CPM Standalone Message, CPM File Transfer request or CPM Session Invitation to the non-CPM recipient using a different Non-CPM Communication Service via the corresponding Interworking Function, in which the CPM originator already has or can be assigned a routable Non-CPM Communication System address.

NOTE 2: Interworking selection re-attempt for disposition notifications will not be performed.

In the above case of attempting to deliver to a different Non-CPM Communication Service, the same selection process as described above is applied (potentially repeatedly) to identify the most appropriate Non-CPM Communication Service among the other Non-CPM Communication Services.

If the Interworking Selection Function did not attempt to deliver to a different Non-CPM Communication Service or the last attempt to deliver to a different Non-CPM Communication Service ended with a failure response, the Interworking Selection Function SHALL forward the failure response towards the CPM originator.

If there is no routable address assigned to the CPM originator for any Non-CPM Communication Service, the Interworking Selection Function SHALL reject the CPM request with an error response.

### 5.3.1.3 Interworking Execution

Upon receiving a CPM Session Invitation, a CPM File Transfer request or a CPM Standalone Message, the selected Interworking Function SHALL convert the CPM Session Invitation, the CPM File Transfer request or the CPM Standalone Message to a non-CPM session invitation, a non-CPM file transfer request or a non-CPM message according to the Non-CPM Communication Service procedures and send the non-CPM session invitation, the non-CPM file transfer request or the non-CPM message towards the address provided by the Interworking Selection Function.

The Interworking Function MAY respond to the CPM Session Invitation, the CPM File Transfer request or the CPM Standalone Message prior to receiving a response from the Non-CPM Communication Service (e.g., per service provider policies, to prevent the risk of timeout).

Upon receiving a CPM Session modification request, the Interworking Function SHALL check if the offered Media Stream is supported by itself. If supported, the Interworking Function SHALL convert the CPM Session modification request to a non-CPM session modification or a non-CPM message according to the Non-CPM Communication Service procedures and send the non-CPM session modification or the non-CPM message towards the same non-CPM address as used at CPM Session establishment. Otherwise, the Interworking Function SHALL reject the CPM Session modification request with an appropriate error response.

Upon receiving a disposition notification, the Interworking Function SHALL convert the disposition notification to a non-CPM disposition notification and route the non-CPM disposition notification towards the address provided in the received non-CPM message.

When interworking a CPM Standalone Message with a disposition notification request, depending on the Non-CPM Communication Service, the selected Interworking Function SHALL store all information from the CPM Standalone Message necessary to create a disposition notification (message identifier, address information, disposition notification request information) together with identifying information of the corresponding message for the Non-CPM Communication Service.

NOTE: The exact definition of identifying information depends upon the target non-CPM Communication Service. For example, identifying information can be a numeric id, user address, time stamp or combination of these parameters.

When a session between the CPM User and the non-CPM Principal has been established, the Interworking Function SHALL convert the Media exchanged via the session.

#### 5.3.1.3.1 CPM Conversation Interworking

The Interworking Function SHOULD support maintaining information about CPM Conversations during interworking with Non-CPM Communication Services. This section describes CPM Conversation interworking from CPM to Non-CPM Communication Services, while section 5.3.2.1 describes CPM Conversation interworking for the reverse direction. When

CPM Conversation indications are included in a received CPM Session Invitation, a received CPM File Transfer request or a received CPM Standalone Message, the Interworking Function SHALL extract from the CPM Session Invitation, the CPM File Transfer request or the CPM Standalone Message:

- the received CPM Conversation Identity indication; and,
- the received CPM Contribution Identity indication

and SHALL maintain a correlation between this information and the corresponding non-CPM conversation identifying information for the target Non-CPM Communication Service.

NOTE: The exact definition of non-CPM conversation identifying information depends upon the target Non-CPM Communication Service. For example, this can be a numeric id, originator and recipient address, subject or a combination of these parameters etc.

If not already stored and in order to maintain conversations between CPM and non-CPM Principals, the Interworking Function SHALL store the information above for a period of time defined by the service provider. All CPM Standalone Messages, CPM Session Invitations and CPM File Transfers exchanged within this period of time and carrying the same CPM Conversation Identity as the stored information will be considered as belonging to the same CPM Conversation. The Interworking Function MAY restart the timer associated with this conversation information each time a new CPM or non-CPM message containing the related information is received, and delete the stored information after the timer expires.

### 5.3.1.4 Interworking of a CPM Session with Multiple Media Streams

#### 5.3.1.4.1 Session Initiation with Multiple Media Streams

When the CPM Participating Function or the CPM Controlling Function involved in making interworking decision receives a SIP 200 “OK” response from the Interworking Function, they SHALL check whether all Media Streams offered in the CPM Session Invitation were accepted. If not, based on service provider policy, the involved CPM entity:

- SHALL send the SIP 200 “OK” response towards the originating CPM Client; or, optionally
- SHALL generate a new CPM Session Invitation offering the rejected Media Stream(s) and send it to the Interworking Selection Function.

#### 5.3.1.4.2 Session Modification Adding a New Media Stream

When the CPM Participating Function or the CPM Controlling Function receives an error response to a CPM Session modification request for adding new Media Stream(s) from an Interworking Function, based on the error response and service provider policy, the involved entity:

- SHALL send the error response towards the originating CPM Client; or, optionally
- SHALL send a CPM Session Invitation offering the new Media Stream(s) to the Interworking Selection Function.

### 5.3.2 Interworking from Non-CPM to CPM

Upon receiving a non-CPM session invitation, a non-CPM file transfer request or a non-CPM message from a Non-CPM Communication Service, the respective Interworking Function SHALL convert the non-CPM session invitation to a CPM Session Invitation, the non-CPM file transfer request to a CPM File Transfer request or the non-CPM message to a CPM Standalone Message, include the Non-CPM Communication Service Identifier, and send the CPM Session Invitation, CPM File Transfer request or the CPM Standalone Message to either the CPM Participating Function or the CPM Controlling Function via the SIP/IP core.

NOTE 1: CPM Session Invitations, CPM File Transfer requests and CPM Standalone Messages directed to a single CPM User will be routed towards the CPM Participating Function, and CPM Session Invitations, CPM File Transfer requests and CPM Standalone Messages directed to a CPM Group are routed towards the CPM Controlling Function.

NOTE 2: Signalling responses to a CPM Session Invitation, a CPM File Transfer request or a CPM Standalone Message follow the reverse path of the CPM Session Invitation, the CPM File Transfer request or the CPM Standalone Message.

NOTE 3: Inclusion of the Non-CPM Communication Service Identifier allows CPM-side replies and CPM Standalone Message disposition notifications to be delivered to the same Non-CPM Communication Service.

If a CPM Message converted from a received non-CPM message has a size less than or equal to 1300 bytes, the respective Interworking Function SHALL send the message as a Pager Mode CPM Standalone Message; otherwise the Interworking Function SHALL send the message as a Large Message Mode CPM Standalone Message.

Upon receiving a disposition notification from a Non-CPM Communication Service, if the disposition notification includes the identifying information of a non-CPM message stored in the respective Interworking Function, the respective Interworking Function SHALL use the corresponding CPM Contribution Identity for converting the disposition notification. The Interworking Function SHALL include the corresponding stored address in the CPM Standalone Message disposition notification. The respective Interworking Function SHALL convert the non-CPM disposition notification to a CPM Standalone Message disposition notification and route the CPM Standalone Message disposition notification to the CPM Participating Function or the CPM Controlling Function via the SIP/IP core.

When a session between the non-CPM Principal and the CPM User has been established, the Interworking Function SHALL convert the Media exchanged via the session.

### 5.3.2.1 CPM Conversation Interworking

The Interworking Function SHOULD support maintaining information about CPM Conversations during interworking with Non-CPM Communication Services.

Upon receiving a non-CPM session invitation, a non-CPM file transfer request or a non-CPM message from a Non-CPM Communication Service, the Interworking Function SHALL extract the non-CPM conversation identifying information and SHALL maintain a correlation between this information and particular CPM Conversation indications (CPM Conversation Identity, CPM Contribution Identity) for a period of time defined by the service provider.

NOTE 1: The exact definition of the non-CPM conversation identifying information depends upon the target Non-CPM Communication Service. For example, conversation identifying information can be a numeric id, originator and recipient address, subject or a combination of these parameters etc.

If the Interworking Function associates the received invitation or message to a new CPM Conversation (e.g. no corresponding correlation information was found), it SHALL generate and include a new CPM Conversation Identity indication and a new CPM Contribution Identity indication in the converted CPM Session Invitation, the converted CPM File Transfer request or the converted CPM Standalone Message routed towards either the CPM Participating Function or the CPM Controlling Function.

If the Interworking Function associates the received invitation, file transfer request or message to an existing CPM Conversation (e.g. corresponding correlation information was stored), it SHALL extract and include:

- the associated CPM Conversation Identity indication
- the associated CPM Contribution Identity indication as the indication indicating the CPM Contribution Identity being replied to

and construct and include a new CPM Contribution Identity indication, in the converted CPM Session Invitation, the converted CPM File Transfer request or the converted CPM Standalone Message routed towards either the CPM Participating Function or the CPM Controlling Function. Depending on the Non-CPM Communication Service interworked with, the Interworking Function MAY reset the validity period associated with the extracted and/or constructed indications and information.

NOTE 2: The validity period (e.g. defined by the service provider) provides the period of time during which a CPM Conversation is considered ongoing and its associated indications and information are used by the Interworking Function.

If the validity period associated with a stored set of CPM Conversation Identity indication, CPM Contribution Identity indication and conversation identifying information is reached, the Interworking Function MAY remove these stored indications and information.

## 5.4 CPM User Preferences

The CPM Enabler SHALL allow Users to set preferences that the CPM Enabler takes into account when determining how to handle communications for the CPM User.

A subset of the CPM User Preferences is applicable to all CPM User's devices, e.g., message and session handling preferences. Another subset of the CPM User preferences specific to devices is grouped in a collection known as a User Preferences Profile, e.g., message delivery preferences. The User Preference Profile can be created using any of the CPM User's devices. One particular User Preference Profile is designated as the active profile at any given time for a specific device. Each User Preference Profile has a unique identifier and a human readable display name.

The CPM User preferences stored in [OMA-XDM-Policy], User Preferences Profiles stored in [OMA-XDM-Policy] and active User Preference Profile indication stored in [OMA-XDM-UPP] can be accessed and managed using common procedures as specified in [OMA-XDM\_Core-TS].

The CPM Participating Function and Interworking Selection Function SHALL enforce user preferences according to their functionalities.

The CPM Participating Function and Interworking Selection Function SHALL retrieve the list of defined User Preferences Profile(s) and the active User Preferences Profile indication from [OMA-XDM-UPP].

The CPM Participating Function and Interworking Selection Function SHALL retrieve the user preferences from [OMA-XDM-Policy].

The CPM Participating Function SHALL take the following User Preferences into consideration for determining how to handle communications for the CPM User:

A) General request Handling:

1. Do not disturb flag for incoming CPM Standalone Messages, CPM Standalone Message disposition notifications, CPM File Transfers and CPM Sessions.

B) 1-1 CPM Session, CPM File Transfer and CPM Standalone Message Handling:

1. Reject CPM communication (i.e. Standalone Messages, 1-1 CPM Session, CPM File Transfer) based on:
  - Blacklist (i.e. list of senders whose messages, 1-1 CPM Session or file transfer requests would be rejected)
2. Handling of CPM Messages:
  - Defer, Deliver to Message Storage Server, Forward, Deliver using Non-CPM Communication service
  - Deliver the whole message, Deliver by providing a link to stored media instead of the media itself

C) Handling of expired Deferred CPM Messages

- Discard, deliver to Message Storage Server

D) Session handling

1. Automatically accept/reject specific types of media during a session
2. Handling of session invitation when CPM user not available:
  - Reject session, Deliver session invitation using interworking and auto-accept CPM Session invitations on behalf of the CPM User.

E) Recording handling

1. Automatically record/not record:
  - CPM Standalone Messages, CPM Sessions, CPM Conversations, media

F) Multi-Device handling

1. Deliver message to all or subset of user's devices
  2. Deliver session invitations to all or a subset of user's devices
  3. Deliver based on active User Preference Profile, message size, content type
- G) File transfer handling
1. Store and Forward functionality enabled / or not
  2. Automatically accept File Transfer session invitations / or not

The Interworking Selection Function SHALL retrieve the CPM User Preferences from the XDMS and SHALL take the following User Preferences into consideration as follows:

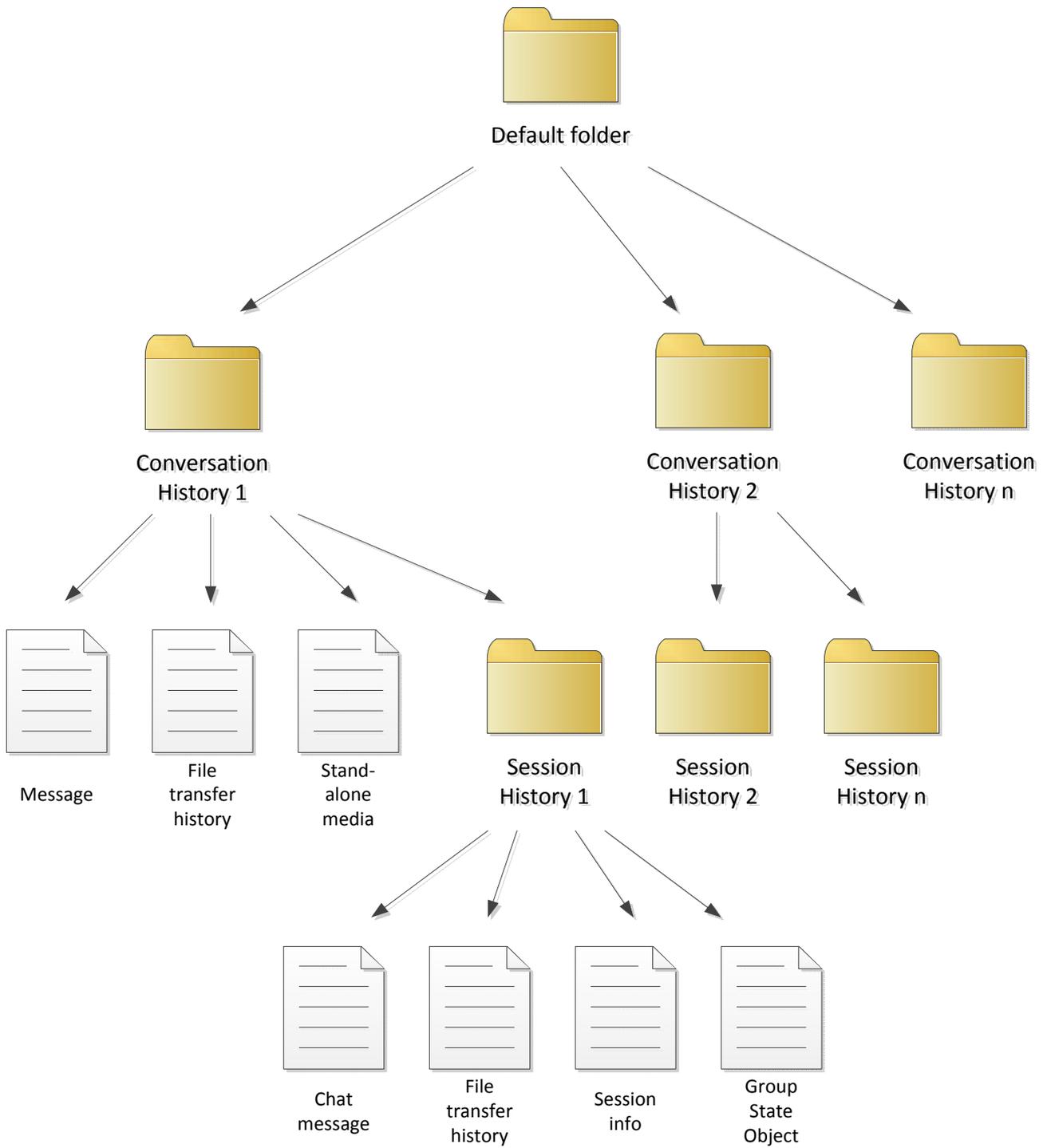
- A) Interworking selection
1. Non-CPM Communication Services to be interworked and their priorities

## 5.5 CPM Message Storage

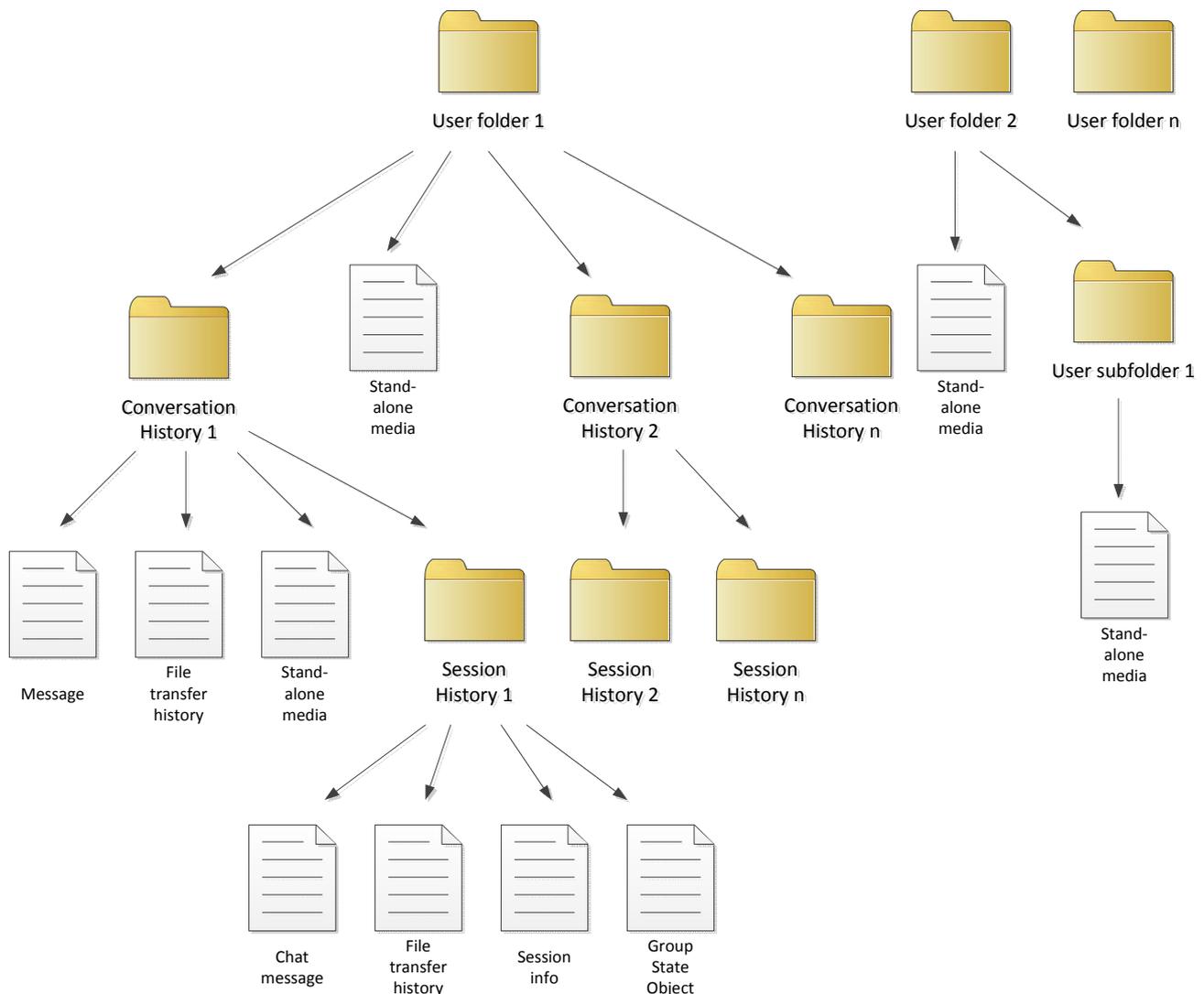
### 5.5.1 Storage Model

The CPM Message Storage SHALL allow for the storage of CPM Standalone Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, and any potential Media objects either stand-alone or attached to CPM Standalone Messages and CPM Session Histories. A Group State Object is stored in relation to the CPM Session Histories and it contains the last information related to the CPM Group Session. The storage model of the CPM Message Storage closely resembles the storage models that are exposed by IMAP mailboxes, and containing folders and objects stored as IMAP messages.

Figure 2 and Figure 3 show a schematic overview of the storage model exposed by the CPM Message Storage.



**Figure 2: Example of Default Folder Storage Model of the CPM 2.0 Message Storage**



**Figure 3: Example of Optional User Folder Storage Model of the CPM 2.0 Message Storage**

### 5.5.1.1 Folders

Within this storage model, the following types of folders can be distinguished:

1. the “default” system folder<sup>1</sup>, which is the root folder for all CPM Message Store folders and objects. The initial location of the “default” system folder is determined by the service provider. While the “default” system folder may

<sup>1</sup> Note: in practice, since the CPM Message Store is based on IMAP message storage principles, the folders shown in the model in [Figure 1] are mapped to IMAP mailboxes. The IMAP specification dictates that before any operation can be performed on objects in the message store, a mailbox must be selected using the IMAP SELECT operation, which is analogous to “changing directory” in a computer file system. There is no “default” system folder in IMAP. This specification uses the “default” system folder as a logical reference to pinpoint a specific mailbox in the storage hierarchy, which serves as an “entry point” - a root folder - for all CPM-related storage objects that are managed by the system (e.g. the participating function).

be located along a different folder path for each user, there SHALL be only one such folder per CPM User. The “default” system folder SHALL only contain Conversation History Folders.

The ‘DefaultFolderLocation’ server annotation (see appendix “CPM METADATA annotations” in [OMA-CPM-MSG]) SHALL be set, and its value SHALL refer to the location of the “default” system folder.

The CPM Message Storage Client and the CPM Participating Function (i.e. all CPM functional components that are accessing the CPM Message Storage Server one way or another) SHALL determine the location of the “default” system folder by retrieving the ‘DefaultFolderLocation’ server annotation.

2. user folders MAY be created and accessed by the CPM Enabler. They MAY contain any number of sub-folders. User folders MAY contain stand-alone media objects, and conversation history folders which have been moved from the “default” system folder by the CPM User. Once they have been moved, conversation history folders SHALL NOT be moved back to the “default” system folder. There MAY be any number of such folders per CPM User.
3. conversation history folder, which is used to store all items related to a single CPM Conversation. Conversation history folders MAY be moved, in their entirety, to a user folder in the user folders hierarchy. Conversation history folders SHALL contain at least one of the following items:
  - a. message object
  - b. file transfer history object
  - c. stand-alone media object
  - d. session history folder
4. session history folder, which is used to store all items related to a single CPM Session. Session history folders SHALL contain a session info object and at least one of the following items:
  - a. message object
  - b. file transfer history object
  - c. group state object
  - d. session info object

### 5.5.1.2 Objects

Within the folders of the CPM Message Storage the following types of stored objects can be distinguished:

1. A message object, which is a container for a CPM Standalone Message (including any Media objects attached to the CPM Standalone Message).
2. A file transfer history object, which is a container for a CPM File Transfer History (including the file transferred in the CPM File Transfer).
3. A stand-alone media object, which is stand-alone media wrapped as a MIME formatted object with no requirement to be [RFC2822] conforming, but it may include an indication to its content type.
4. A message object, which is a container for a CPM Chat Message exchanged within a CPM Session. This object may only appear in a session history folder.
5. A group state object, which is a container for the CPM Group State Object. This object MAY only appear once in a session history folder related to a CPM Group Session.
6. A session info object, which is a container for the session information when recording the session history.

NOTE 1: In this release of the CPM Enabler only complete conversation history folder can be moved between folders. The individual components of conversation histories folder such as threaded message objects, file transfer history objects or session history folder cannot be moved separately.

NOTE 2: Message objects, file transfer history objects and session info objects can only be so-called “leaf-objects”, meaning that they SHALL NOT contain other message objects, file transfer history objects, session info objects, conversation history objects or folder objects as their sub-objects within the storage hierarchy.

Metadata is associated with the message objects, the file transfer history objects, the session info objects and the folder objects. This metadata is used to store data that is associated with the content / folder, but is not part of the real contents of the content / folder. The following are examples of the metadata that is supported:

- Flags associated with the message object (e.g. read-flag, answered-flag, forwarded-flag)
- Access control lists to determine who has access to a folder and who can modify the folder.

NOTE 3: Access control lists SHALL be only associated with folder objects, not with individual message objects, file transfer history objects or session info objects.

## 5.5.2 Operations

The Message Storage Server SHALL support the following operations:

- authenticate;
- set active folder;
- object store;
- object fetch;
- object copy;
- object remove;
- folder create;
- list folders;
- folder move;
- folder remove;
- search;
- metadata update;
- metadata fetch;
- set access control list;
- get access rights;
- delete access control list;
- get access control list;
- generate reference;
- fetch by reference;
- synchronize with client.

NOTE 1: In the above operations an object can either be a CPM Standalone Message, a CPM Session History, a CPM File Transfer History or a standalone Media Object, which are not to be confused with a Media Object attached to the Message.

NOTE 2: The folder move operation is also used for renaming a folder.

The Message Storage Server MAY support the following operations:

- preview fetch

Upon enabling notifications by the CPM User, the Message Storage Server SHALL provide notifications of changes to the stored resources to the Message Storage Client. It includes all changes that occurred since the client's last de-registration.

The communication between a requestor, such as the Message Storage Client, and the Message Storage Server is based on transport-level (i.e. TCP) connections. This means that the requestor needs to establish a transport-level connection with the Message Storage Server, and that all operations are requested within the scope of that transport-level connection. A requestor will have to authenticate before being able to make other requests within a transport-level connection.

Upon request from a CPM User, the Message Storage Client SHALL send a request corresponding to the requested operation to the Message Storage Server.

Upon receiving a request, the Message Storage Server SHALL

- check that the requestor has already successfully authenticated itself; and,
- validate authorization of the request and deny the request if the request is determined to not been authorized; and
- handle the request as described in the subsections for the individual operations below.

### 5.5.2.1 Authenticate Operation

The authenticate operation allows a requestor, such as the Message Storage Client, to initiate the authentication process for a transport-level connection. Several authentication options are possible, including an option for the requestor to authenticate himself to the Message Storage Server, and an option that allows for mutual authentication between the requestor and the Message Storage Server. The authentication result is valid for the duration of the transport-level connection, i.e. a requestor is considered to be authenticated for all subsequent operations sent within the transport-level connection.

A requestor SHALL include in the authenticate request:

- a CPM User address; and,
- the type of authentication requested; and,
- authentication information.

Upon receiving an authenticate request, the Message Storage Server SHALL check the authentication information and SHALL respond with the outcome of the authentication check. If requested by the requestor, the Message Storage Server SHALL include its own authentication information in the response. If the outcome of the authentication check was successful, then the Message Storage Server SHALL consider the requestor to be authenticated for the remainder of the transport-level connection.

Upon receiving an authenticate response including authentication information of the Message Storage Server, a requestor SHALL check the authentication information. If the authentication information is determined to not be valid, the requestor SHALL stop the current transport-level connection, otherwise it SHALL continue with the transport-level connection.

### 5.5.2.2 Set Active Folder Operation

The set active folder operation allows a requestor, such as the Message Storage Client, to set one of the folders of the CPM User as the active one within the transport-level connection.

A requestor SHALL include the name of the folder that is to become active in the set active folder request.

Upon receiving a set active folder request, the Message Storage Server SHALL set the indicated folder as the active folder for the current transport-level connection. If successful, the Message Storage Server SHALL respond with an OK response, otherwise it SHALL respond with an appropriate error.

### 5.5.2.3 Object Store Operation

The object store operation allows a requestor, such as the Message Storage Client or the CPM Participating Function, to store a CPM Standalone Message or a CPM Session History in a particular folder of the CPM User on the Message Storage Server.

A requestor SHALL include the CPM Standalone Message / CPM Session History to be stored in the object store request and the name of the folder in which the CPM Standalone Message / CPM Session History is to be stored.

Upon receiving an object store request, the Message Storage Server SHALL store the CPM Standalone Message / CPM Session History in the indicated folder. If successful, the Message Storage Server SHALL respond with an OK response, otherwise it SHALL respond with an appropriate error.

### 5.5.2.4 Object Fetch Operation

The object fetch operation allows a requestor, such as the Message Storage Client, to retrieve a CPM Standalone Message or a CPM Session History, including the metadata associated with it, from the active folder on the Message Storage Server. The

requestor can indicate which parts of the CPM Standalone Message or CPM Session History it wants to receive (e.g. only the metadata or only a particular attachment).

A requestor SHALL include the identifier of the CPM Standalone Message / CPM Session History to be fetched and an indication of the required information in the object fetch request.

Upon receiving an object fetch request, the Message Store Server SHALL retrieve the requested parts of the indicated CPM Standalone Message / CPM Session History. If successful, the Message Storage Server SHALL respond with an OK response carrying the requested parts of the CPM Standalone Message / CPM Session History corresponding with the identifier provided in the request, otherwise it SHALL respond with an appropriate error.

### 5.5.2.5 Object Copy Operation

The object copy operation allows a requestor, such as the Message Storage Client, to copy a CPM Standalone Message or a CPM Session History to a particular folder of the CPM User on the Message Storage Server.

A requestor SHALL include the identifier of the CPM Standalone Message / CPM Session History that needs to be copied in the object copy request and the name of the folder to which the CPM Standalone Message / CPM Session History is to be copied.

Upon receiving an object copy request, the Message Storage Server SHALL copy the identified CPM Standalone Message / CPM Session History to the identified folder. If successful, the Message Storage Server SHALL respond with an OK response, otherwise it SHALL respond with an appropriate error.

### 5.5.2.6 Object Remove Operation

The object remove operation allows a requestor, such as the Message Storage Client, to remove a CPM Standalone Message or a CPM Session History from the active folder on the Message Storage Server.

A requestor SHALL include the identifier of the CPM Standalone Message / CPM Session History to be removed from the active folder in the object remove request.

Upon receiving an object remove request, the Message Store Server SHALL remove the identified CPM Standalone Message / CPM Session History object from the active folder. If successful, the Message Storage Server SHALL respond with an OK response, otherwise it SHALL respond with an appropriate error.

### 5.5.2.7 Preview Fetch Operation

The preview fetch operation allows a requestor, such as the Message Storage Client, to retrieve preview of a CPM File Transfer History, a CPM Standalone Message or a CPM Session History from the active folder on the Message Storage Server.

A requestor SHALL include in the preview fetch request:

- The identifier of the CPM File Transfer History / CPM Standalone Message / CPM Session History for which preview is being fetched.
- The identification of the parts of the CPM Standalone Message / CPM Session History for which a preview is being requested.
  - Note: if the request is for a CPM File Transfer History preview, requesting this part is not needed since the part being fetched is actually the File Transfer Thumbnail.
- The format and dimensions the preview should adhere to.

Upon receiving a preview fetch request, the Message Store Server SHALL generate the preview of the identified CPM Transfer History / CPM Standalone Message / CPM Session History according to the supplied format and dimensions. If successful, the Message Storage Server SHALL respond with an OK response carrying the preview of the attachment, otherwise it SHALL respond with an appropriate error.

A “Preview Fetch” operation MAY involve a server-side content adaptation in response to the client request for the stored object in a compacted or digested form rather than in its original full size and shape.

### 5.5.2.8 Folder Create Operation

The folder create operation allows a requestor, such as the Message Storage Client, to create a folder on the Message Storage Server.

A requestor SHALL include the name of the folder to be created in the folder create request.

Upon receiving a folder create request, the Message Storage Server SHALL create the folder with the requested name. If successful the Message Storage Server SHALL respond with an OK response, otherwise it SHALL respond with an appropriate error.

### 5.5.2.9 List Folders Operation

The list folders operation allows a requestor, such as the Message Storage Client, to list the folders of the CPM User on the Message Storage Server.

A requestor SHALL send a list folders request to get a list of the folders.

Upon receiving a list folders request, the Message Storage Server SHALL determine the names of all folders of the CPM User on the Message Storage Server. If successful, the Message Storage Server SHALL respond with an OK response carrying the names of all the folders, otherwise it SHALL respond with an appropriate error.

### 5.5.2.10 Folder Move Operation

The folder move operation allows a requestor, such as the Message Storage Client, to rename a folder of the CPM User on the Message Storage Server or move a folder of the CPM User to a new place in the folder hierarchy.

A requestor SHALL include the name of a folder to be renamed / moved and a new name for the folder in the folder move request.

Upon receiving a folder move request, the Message Storage Server SHALL move the indicated folder to its new location (as indicated by the new name). If successful, the Message Storage Server SHALL respond with OK response, otherwise it SHALL respond with an appropriate error.

### 5.5.2.11 Folder Remove Operation

The folder remove operation allows a requestor, such as the Message Storage Client, to delete a folder of the CPM User on the Message Storage Server.

A requestor SHALL include the name of the folder to be removed in the folder remove request.

Upon receiving a folder remove request, the Message Storage Server SHALL remove the indicated folder. If successful, the Message Storage Server SHALL respond with OK response, otherwise it SHALL respond with an appropriate error.

### 5.5.2.12 Search Operation

The search operation allows a requestor, such as the Message Storage Client, to search for CPM Standalone Messages and CPM Session Histories in the active folder, according to particular search criteria.

The Message Storage Client SHALL include the search criteria in the search request.

Upon receiving a search request, the Message Storage Server SHALL find in the active folder all CPM Standalone Messages / CPM Session Histories that match the search criteria. If successful, it SHALL respond with OK response carrying the identifiers of CPM Standalone Messages and CPM Session Histories stored in the active folder that satisfy the search condition, otherwise it SHALL respond with an appropriate error.

### 5.5.2.13 Metadata Update Operation

The metadata update operations allows a requestor, such as the Message Storage Client, to update the metadata associated with the server (i.e. server annotations) or a particular message object (e.g. a CPM Standalone Message, a CPM Standalone Media object, CPM Session Info object, etc.) or folder object (e.g. the “default” system folder, user folders, CPM Session

History folder, etc.) in the active folder on the Message Storage Server. The metadata available for these objects is defined in section **Error! Reference source not found.** “*Error! Reference source not found.*” of [OMA-CPM-MSG].

NOTE: The metadata update operation allows addition, updating, and removal of metadata.

A requestor SHALL include an identifier of the object and the metadata modifications in the metadata update request.

Upon receiving a metadata update request, the Message Storage Server SHALL update the metadata for the indicated object. If successful, the Message Storage Server SHALL respond with an OK response, otherwise it SHALL respond with an appropriate error.

#### 5.5.2.14 Metadata Fetch Operation

The metadata fetch operations allow a requestor, such as the Message Storage Client, to retrieve the metadata associated with the server (i.e. server annotations) or a particular message object (e.g. a CPM Standalone Message, a CPM Standalone Media object, CPM Session Info object, etc.) or folder object (e.g. the “default” system folder, user folders, CPM Session History folder, etc.) in the active folder on the Message Storage Server. The metadata available for these objects is defined in section **Error! Reference source not found.** “*Error! Reference source not found.*” of [OMA-CPM-MSG].

A requestor SHALL include an identifier of the object and the type of the metadata required in the metadata fetch request.

Upon receiving a metadata fetch request, the Message Storage Server SHALL retrieve the metadata for the indicated object. If successful, the Message Storage Server SHALL return the requested metadata and respond with an OK response, otherwise it SHALL respond with an appropriate error.

#### 5.5.2.15 Set Access Control List Operation

The set access control list operation allows a requestor, such as the Message Storage Client, to set the access control list associated with a Principal for a particular folder of the CPM User on the Message Storage Server.

A requestor SHALL include in the set access control list request:

- the name of a folder;
- an address of a Principal;
- the access rights to be associated with the indicated Principal.

Upon receiving an access control list management request, the Message Storage Server SHALL update the stored access control list for the indicated Principal in the indicated folder. If successful, the Message Storage Server SHALL respond with an OK response, otherwise it SHALL respond with an appropriate error.

#### 5.5.2.16 Get Access Rights Operation

The get access rights operation allows a requestor, such as the Message Storage Client, to get the access rights associated with a Principal for a particular folder of the CPM User on the Message Storage Server.

The Message Storage Client SHALL include a name of a folder and an address of a Principal in the get access rights request.

Upon receiving a get access rights request, the Message Storage Server SHALL retrieve the access rights for the indicated Principal for the indicated folder. If successful, the Message Storage Server SHALL respond with an OK response containing the retrieved access rights, otherwise it SHALL respond with an appropriate error.

#### 5.5.2.17 Delete Access Control List Operation

The delete access control list operation allows a requestor, such as the Message Storage Client, to delete the access control list associated with a Principal for a particular folder of the CPM User on the Message Storage Server.

The Message Storage Client SHALL include request a name of a folder and an address of a Principal in the delete access control list.

Upon receiving a delete access control list request, the Message Storage Server SHALL remove the stored access control list for the indicated Principal for the indicated folder. If successful, the Message Storage Server SHALL respond with an OK response, otherwise it SHALL respond with an appropriate error.

#### 5.5.2.18 Get Access Control List Operation

The get access control list operation allows a requestor, such as the Message Storage Client, to get the set of access control lists that are set for a particular folder of the CPM User on the Message Storage Server.

The Message Storage Client SHALL include a name of a folder whose access control lists are requested in the get access control list request.

Upon receiving a get access control list request, the Message Storage Server SHALL retrieve the stored access control lists for the indicated folder. If successful, the Message Storage Server SHALL respond with an OK response containing the list of access control lists associated with the indicated folder, otherwise it SHALL respond with an appropriate error.

#### 5.5.2.19 Generate Reference Operation

The generate reference operation allows a requestor, such as the Message Storage Client, to obtain a reference towards (a part of) a CPM Standalone Message / CPM Session History.

The Message Storage Client SHALL include an identifier of a CPM Standalone Message / CPM Session History and optionally an indication of a part of the CPM Standalone Message / CPM Session History in the generate reference request.

Upon receiving a generate reference request, the Message Storage Server SHALL generate the reference for the identified (part of the) CPM Standalone Message / CPM Session History. If successful, the Message Storage Server SHALL respond with an OK response containing the generated reference, otherwise it SHALL respond with an appropriate error.

#### 5.5.2.20 Fetch by Reference Operation

The fetch by reference operation allows a requestor, such as the CPM Participating Function, to obtain the data pointed to by a reference.

A requestor SHALL include a reference previously generated by the Message Storage Server in the fetch by reference request.

Upon receiving a fetch by reference request, the Message Store Server SHALL retrieve the (parts of a) CPM Standalone Message / CPM Session History that the reference points to. If successful, the Message Storage Server SHALL respond with an OK response carrying the retrieved data, otherwise it SHALL respond with an appropriate error.

#### 5.5.2.21 Synchronization

Based on CPM User's preferences and service provider policies, the CPM Standalone Messages, CPM File Transfer Histories, CPM Session Histories and/or standalone Media Objects within the Message Storage Client's local storage SHALL be synchronized with the stored objects in the CPM Message Storage Server. This means that any changes to the Message Storage Server (e.g., arrival of new or deletion of old CPM Standalone Messages, CPM File Transfer Histories, CPM Session Histories and/or standalone Media objects) will be reflected in the Message Storage Client's local storage and, similarly, any changes to the Message Storage Client's local storage will be reflected in the Message Storage Server, in accordance with user preferences and service provider policies.

Synchronization SHALL always be initiated by the Message Storage Client, e.g. upon the CPM User's request, a client setting to synchronize periodically, or upon detection of changes to the Message Storage Client's local storage. The Message Storage Server MAY trigger the Message Storage Client to initiate a synchronization by sending notifications of changes done to the CPM User's stored resources in the Message Storage Server.

Upon initiation, the synchronization process itself will go through a combination of the operations described above. In general, the Message Storage Client SHALL:

1. Request the lists of folders on the Message Storage Server, as described in section 5.5.2.9.

2. Determine the changes to the list of folders with respect to its local storage and creates or removes the appropriate folders in its local storage and creates or removes the appropriate folders in the Message Storage Server, as described in sections 5.5.2.8 and 5.5.2.11.
3. For each of the resulting list of folders:
  - a. Set the folder as the active folder, as described in section 5.5.2.2, and determines, based on the response, if there are new objects stored in the folder on the Message Storage Server,
  - b. Fetch all new objects from the Message Storage Server, as described in section 5.5.2.4,
  - c. Remove from local storage all the objects that have been removed from the Message Storage Server,
  - d. Store all new objects from the local storage to the Message Storage Server, as described in 5.5.2.3,
  - e. Remove from the Message Storage Server all objects that have been removed from the local storage, as described in section 5.5.2.6, and
  - f. Update the metadata of the resources stored in the local storage based on the Message Storage Server, and vice-versa, as described in sections 5.5.2.4 and 5.5.2.13.  
NOTE: In this step, only metadata is fetched instead of the whole message

## Appendix A. Change History

(Informative)

### A.1 Approved Version History

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

### A.2 Draft/Candidate Version 2.0 History

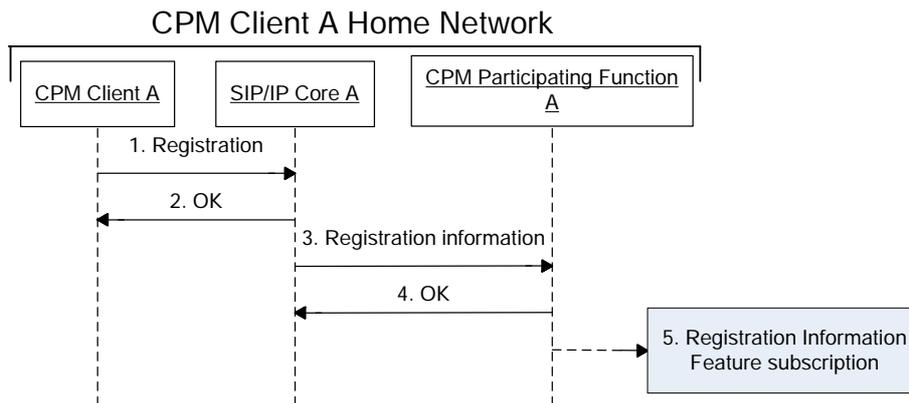
Document Identifier	Date	Sections	Description
Draft Versions OMA-TS-CPM_System_Description-V_2_0	20 Apr 2012	All	Baseline Creation: Content from v1.0
	09 Jul 2012	4.2	CPM v2.0 features from RCS 5.0 and RCS 5.1.
	16 Jul 2012	5.2	Chat notifications
	15 Aug 2012	2.1, 3.2, 5.2.4, 5.4, 5.5.1, 5.5.2.7	CR 28R01 incorporated
	05 Sep 2012	3.2, 4.2, 5.2	CR 47R01 incorporated
	25 Sep 2012	2.1, 5.2.3, 5.4, 5.5	CR 48R01 incorporated
	09 Oct 2012	5.5.1	CR 54R03 and CR 64 incorporated
	22 Nov 2012	4.2, 5.5.1	CR 72 and CR 76R01 incorporated
	17 Dec 2012	B.2.3.4, B.2.3.5, B.2.3.6	Incorporated CR: OMA-COM-CPM-2012-0070R03-CR_GC_Long_lived_flows Editorial changes
	20 Dec 2012	B.2.3.7	Incorporated CR: OMA-COM-CPM-2012-0087R04-CR_GC_Deferred_delivery_flows
	21 Dec 2012	B.2.3.8	Incorporated CR: OMA-COM-CPM-2012-0090-CR_GC_active_case_delivery_flow
	11 Jan 2013	B.2.3.8	Editorial fixes in numbering of new appendix flows: removed section B.2.3.8, so the next sub-sections can follow under B.2.3.7. Removed the prefix "Scenario B.2" in B.2.3.8.2
	25 Feb 2013		Comments from CONRR incorporated. CRs incorporated: CR0020R01
	04 Apr 2013		Incorporated CRs: CR0056, CR0057, CR0059R01, CR0061, CR0077R01
	17 Apr 2013		Incorporated CRs: CR0073R01, CR0075R01
	25 Apr 2013		Incorporated CRs: CR0052R01, CR0053, CR0084
	05 May 2013		Incorporated CRs: CR0063R01, CR105R02
09 May 2013		Editorial fixes	
22 May 2013	3.2, 4.2, 5.1.1.3, 5.1.2.1, 5.2.3.1.1.1, 5.2.3.2.1.4, 5.2.3.3, 5.2.3.4.1, 5.2.4.7, 5.2.5, 5.2.5.2, B.2.2.1, B.2.3.5.1, B.2.3.5.2	Incorporated CRs: OMA-COM-CPM-2013-0046R02-CR_SD_sip_instance_AI003 OMA-COM-CPM-2013-0110-CR_SD_GC_Contribution_ID OMA-COM-CPM-2013-0113R01-CR_SD_CloseAI014 OMA-COM-CPM-2013-0116-CR_SD_Closed_GC-Token Editorial changes	
Candidate Version OMA-TS-CPM_System_Description-V_2_0	11 Jun 2013	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2013-0163R01- INP_Converved_IP_Messaging_2.0_ERP_and_ETR_Candidate_Approval

## Appendix B. CPM Conversation Flows (Informative)

### B.1 Registration

#### B.1.1 CPM Client Registering

This section describes the case where a CPM Client registers.



**Figure 4: CPM Client registration**

1. CPM Client A sends a registration request to SIP/IP Core A.
2. SIP/IP Core A registers CPM Client A and informs CPM Client A about the result.
 

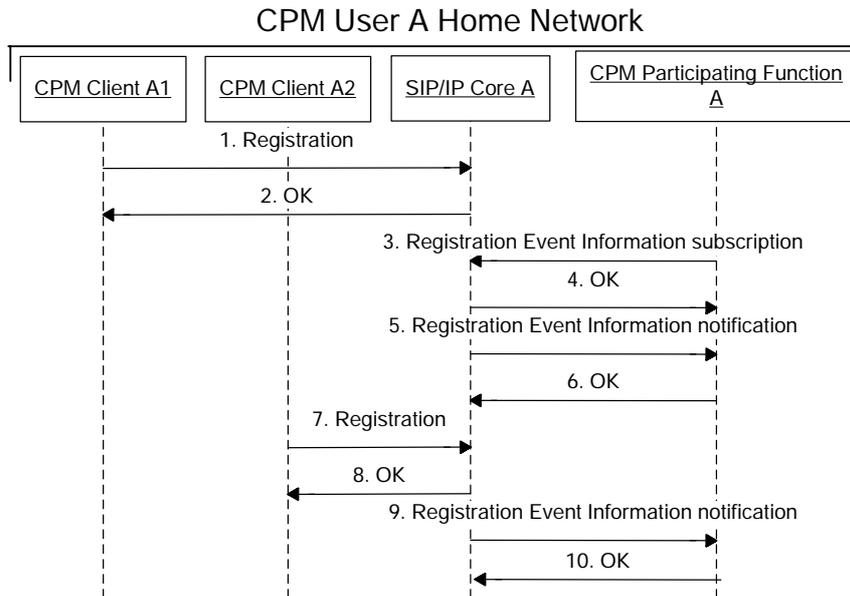
NOTE: Steps 3, 4, and 5 only happen when SIP/IP Core A supports informing SIP application servers of CPM Client registrations.
3. SIP/IP Core A informs CPM Participating Function A about the registration of CPM Client A.
4. CPM Participating Function A confirms delivery of the registration information by sending an OK response.
5. CPM Participating Function A subscribes to Registration Event Information, as shown in section B.1.2.1.

### B.1.2 Registration Event Information Flows

#### B.1.2.1 CPM Participating Function Subscribing to Registration Event Information

This section describes the case where a CPM Participating Function gets a list of CPM Clients used by the same CPM User with a possibility to later get notifications of the changes of the registered CPM Clients used by the same CPM User.

NOTE 1: A CPM Participating Function performs these procedures when being notified by the SIP/IP core of a registration (when the SIP/IP core supports informing a SIP application server of CPM Client registrations, as described in section B.1.1). Alternatively, when the SIP/IP core does not support informing a SIP application server of CPM Client registrations, a CPM Participating Function performs these procedures whenever it needs up-to-date registration status information of a CPM User (e.g. when having to deliver a CPM Standalone Message or a CPM Session Invitation towards a CPM User) and it doesn't already have this information.



**Figure 5: CPM Participating Function subscribes to the Registration Event Information**

1. CPM Client A1 registers to SIP/IP Core A.

2. SIP/IP Core A registers CPM Client A1 and responds with an OK response.

NOTE 2: Steps 1 and 2 above are shown to illustrate the scenario of one CPM Client of CPM User A already being registered. The following steps of CPM Participating Function A subscribing to Registration Information are not caused by steps 1 and 2 but by causes explained above.

3. CPM Participating Function A sends a Registration Event Information subscription request for CPM User A to SIP/IP Core A.

4. SIP/IP Core A authorizes the Registration Event Information subscription request and if successfully authorized, SIP/IP Core A responds with an OK response.

5. SIP/IP Core A sends a Registration Event Information notification to CPM Participating Function A, informing CPM Participating Function A about the currently registered clients of CPM User A (in this case only CPM Client A1).

6. CPM Participating Function A confirms delivery of the Registration Event Information notification by sending an OK response.

7. Sometime later, CPM Client A2 registers to SIP/IP Core A.

8. SIP/IP Core A registers CPM Client A2 and responds with an OK response.

9. SIP/IP Core A sends a Registration Event Information notification to CPM Participating Function A, informing CPM Participating Function A that CPM Client A2 has registered.

10. CPM Participating Function A confirms delivery of the Registration Event Information notification by sending an OK response.

## B.2 CPM Sessions

### B.2.1 CPM Session Initiation

#### B.2.1.1 CPM 1-1 Session Invitation

This section describes the case where an inviting CPM User sends a CPM Session Invitation to another CPM User without involving the CPM Controlling Function, and where the CPM Participating Function may or may not stay in the Media path.

The reasons for which a CPM Participating Function may be configured to stay in the Media path include:

- Charging based on Media;
- History recording done in the network;
- Handling of multiple devices per CPM Session;
- Lawful Intercept (however, it is expected that Lawful Intercept is provided by the SIP/IP core).

The reasons for which a CPM Participating Function may be configured not to stay in the media path include:

- To avoid bottleneck effect, caused by forcing Media through intermediate nodes;
- Charging is based on factors other than Media;
- Multiple devices are not involved in this CPM 1-1 Session;
- Lawful Intercept is handled by other nodes in the network.

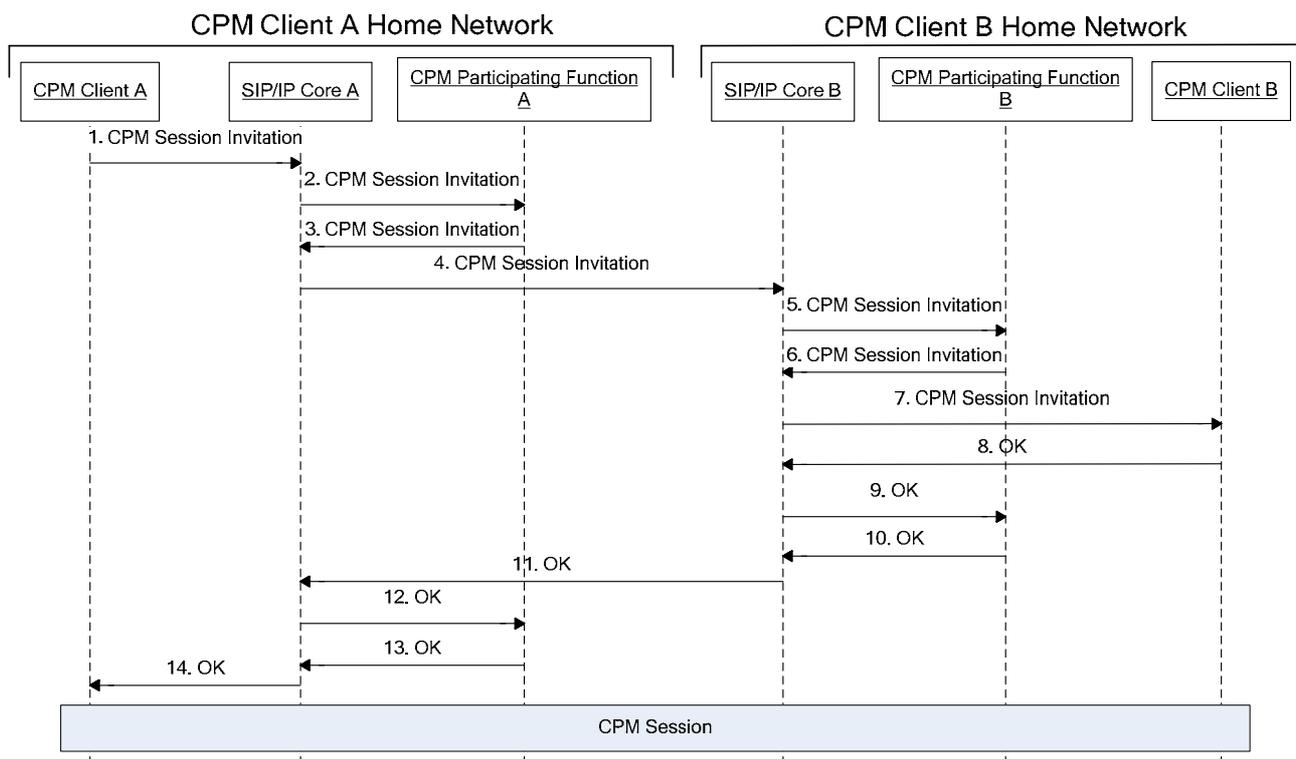


Figure 6: CPM 1-1 Session, originating and terminating parts

1. CPM Client A sends a CPM Session Invitation to CPM Client B.

2. SIP/IP Core A routes the CPM Session Invitation to CPM Participating Function A triggered on the CPM Feature Tag and the CPM Address. If CPM Participating Function A wants to stay in the Media path for this CPM Session, it will act as a back to back user agent for the rest of the CPM Session. If it does not want to stay in the Media path, it will act as a proxy for the rest of the CPM Session.
3. CPM Participating Function A executes the necessary originating service control and then sends the CPM Session Invitation to SIP/IP Core A.
4. SIP/IP Core A routes the CPM Session Invitation to the SIP/IP core of CPM User B's network.
5. SIP/IP Core B routes the CPM Session Invitation to CPM Participating Function B triggered on the CPM Feature Tag and the CPM Address.
6. CPM Participating Function B executes the necessary terminating service control and then sends the CPM Session Invitation to SIP/IP Core B. If CPM Participating Function B wants to stay in the media path for this CPM Session, it will act as a back to back user agent for the rest of the CPM Session. If it does not want to stay in the media path, it will act as a proxy for the rest of the CPM Session
7. SIP/IP Core B sends the CPM Session Invitation to CPM Client B.
8. CPM Client B accepts the CPM Session Invitation with an OK response to SIP/IP core B.
9. SIP/IP Core B sends the response to CPM Participating Function B.
- 10-14. The response is conveyed back to CPM Client A using the reverse routing of the CPM Session Invitation.

### **B.2.1.2 CPM Group Session Invitation**

This section describes the case where a CPM Group Session is set up towards a CPM Group (either a CPM Pre-defined Group or a CPM Ad-hoc Group).

Figure 6 shows the originating side of the overall flow, where an inviting CPM User sends a CPM Session Invitation to a CPM Group.

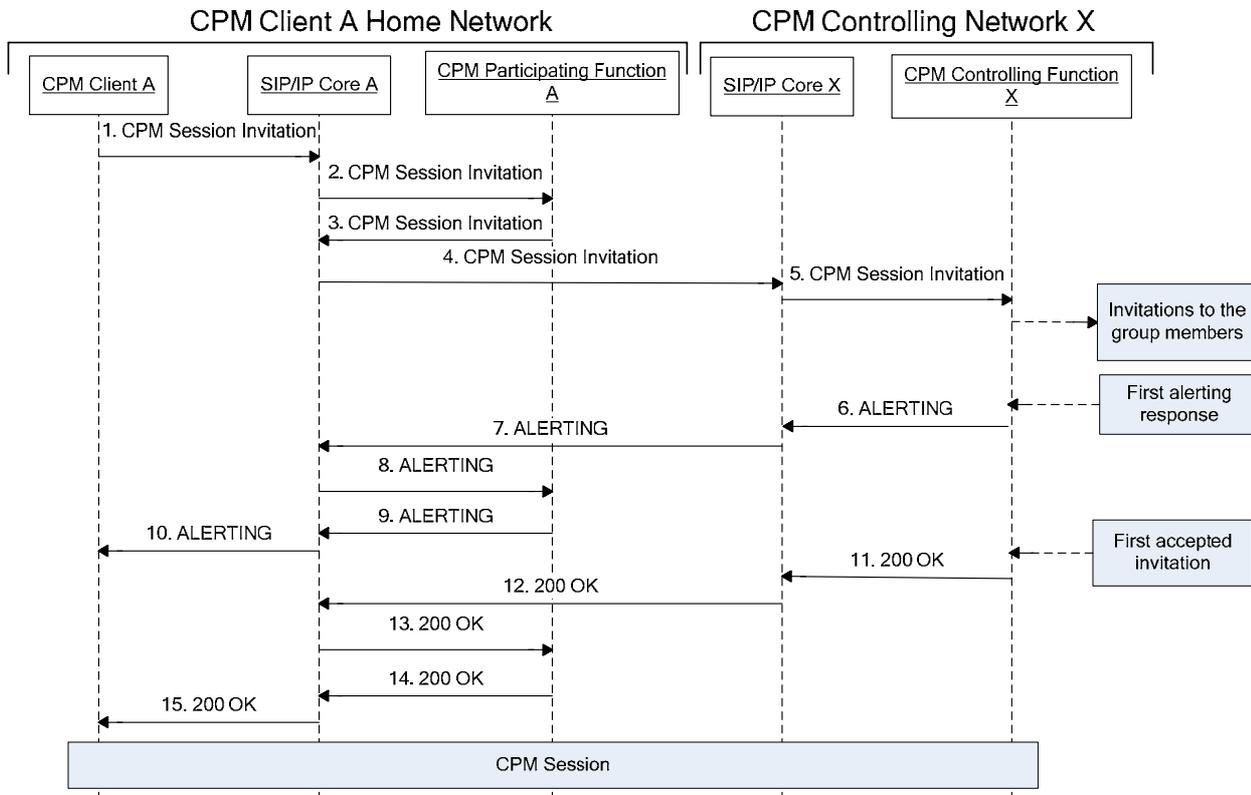
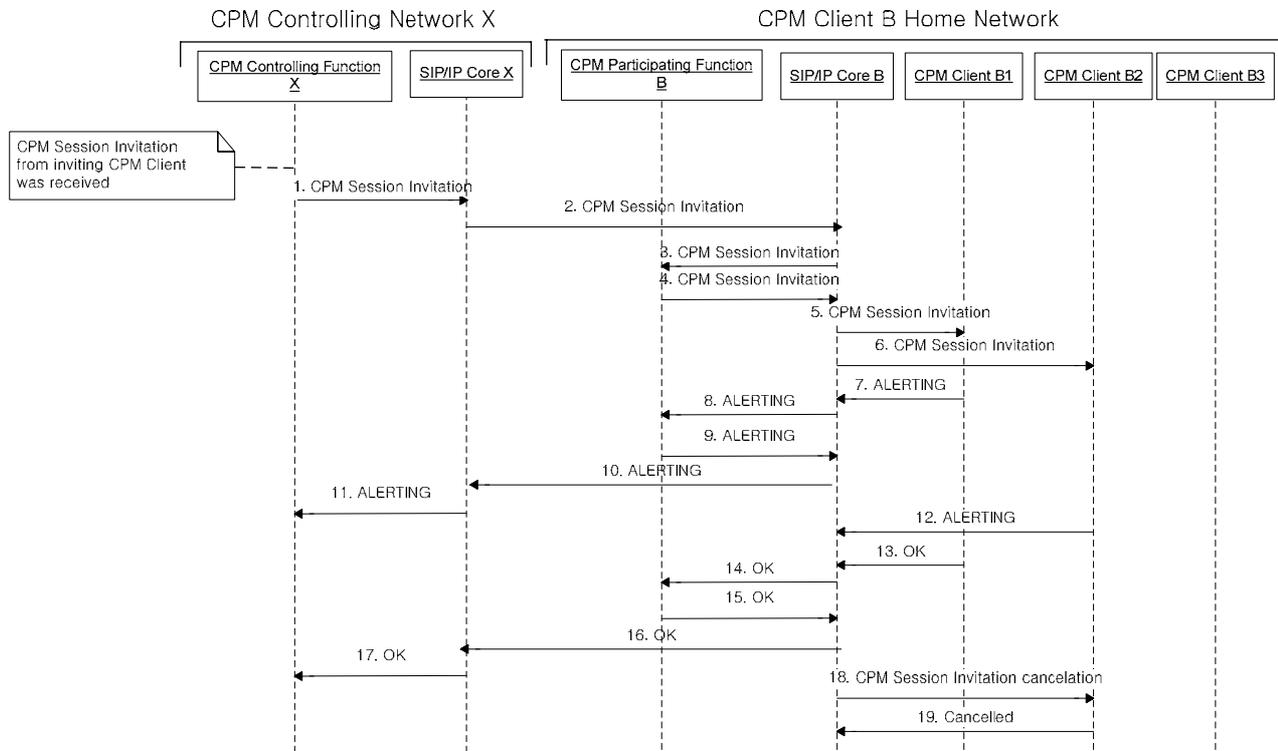


Figure 7: CPM Group Session setup, originating part

1. CPM Client A sends a CPM Session Invitation to establish a CPM Group Session. If the CPM Group Session is for a CPM Pre-defined Group the CPM Client includes the identity of the CPM Pre-defined Group in the CPM Session Invitation. If the CPM Group Session is for a CPM Ad-hoc Group the CPM Client includes the list of group members in the CPM Session Invitation.
2. SIP/IP Core A routes the CPM Session Invitation to CPM Participating Function A triggered on the CPM Feature Tag and the CPM Address.
3. As the CPM Session Invitation is for a CPM Group, CPM Participating Function A sends the CPM Session Invitation to SIP/IP Core A.
4. SIP/IP Core A routes the CPM Session Invitation to SIP/IP Core X.
5. SIP/IP Core X routes the CPM Session Invitation to CPM Controlling Function X based on the target CPM Address. CPM Controlling Function X determines whether or not to allow the request based on the service provider policy, and in case of CPM Pre-defined group, in addition based on CPM Pre-defined Group policy, and if establishment of the CPM Group Session is allowed, the CPM Controlling Function X invites the other members to the CPM Group Session as described in Figure 7.
- 6-10. When the first ALERTING response is received from an invited CPM Client CPM Controlling Function X sends an ALERTING response towards CPM Client A.
- 11-13. When the first CPM Client accepts the CPM Session Invitation for the CPM Group Session, CPM Controlling Function X sends an OK response to CPM Participating Function A along the same Signalling Path.
- 14-15. CPM Participating Function A sends an OK response to CPM Client A along the same Signalling Path.

Figure 7 describes the terminating side of the flow where the CPM Session Invitation is delivered towards an invited CPM User.



**Figure 8: CPM Group Session setup, terminating side, multiple CPM Clients**

1. CPM Controlling Function X sends a CPM Session Invitation to SIP/IP Core X
2. SIP/IP Core X routes the CPM Session Invitation to the home network of invited CPM User B.
3. SIP/IP Core B routes the CPM Session Invitation to CPM Participating Function B based on the invited CPM User address and the CPM Feature Tag.
4. CPM Participating Function B executes the necessary terminating service control and determines to which CPM Clients of CPM User B to deliver/not to deliver the CPM Session Invitation as described in section 5.2.3.1.1.2. In this flow, CPM Participating Function B determines to route the CPM Session Invitation to CPM Clients B1 and B2 and not to route the CPM Session Invitation to CPM Client B3. CPM Participating Function B includes in the CPM Session Invitation an indication not to route to CPM Client B3 and sends the CPM Session Invitation to SIP/IP Core B.
- 5 – 6. SIP/IP Core B routes the CPM Session Invitation request to all CPM Clients of invited CPM User B apart from CPM Client B3 indicated in the received CPM Session Invitation, i.e. to CPM Client B1 and CPM Client B2.
- 7 – 11. When CPM Client B1 receives the CPM Session Invitation, CPM Client B1 responds with an ALERTING response. The ALERTING response is routed to CPM Controlling Function X through the Signalling Path.
12. When CPM Client B2 receives the CPM Session Invitation, CPM Client B2 responds with an ALERTING response. Since an ALERTING response has already been sent to CPM Controlling Function X, CPM Participating Function B does not route the ALERTING response towards CPM Controlling Function X.
- 13 – 17. CPM Client B1 responds with an OK response (e.g. after CPM User B accepted the CPM Session Invitation manually or after CPM Client B1 accepted the CPM Session Invitation automatically). The OK response is routed to CPM Controlling Function X through the Signalling Path.

18. Since CPM Client B1 has already accepted the CPM Session Invitation, SIP/IP Core B cancels the CPM Session Invitation to CPM Client B2 using the CPM Session Invitation cancellation request.

19. CPM Client B2 confirms reception of the CPM Session Invitation cancellation request.

## B.2.2 Adding CPM Users to an Existing CPM Session

### B.2.2.1 Extending a CPM 1-1 Session to a CPM Group Session

This section describes transforming a CPM 1-1 Session into a CPM Group Session.

NOTE 1: To reduce the complexity of the figure no SIP/IP core elements are shown. However, the SIP/IP cores are being used to relay the SIP messages between networks and elements within those networks.

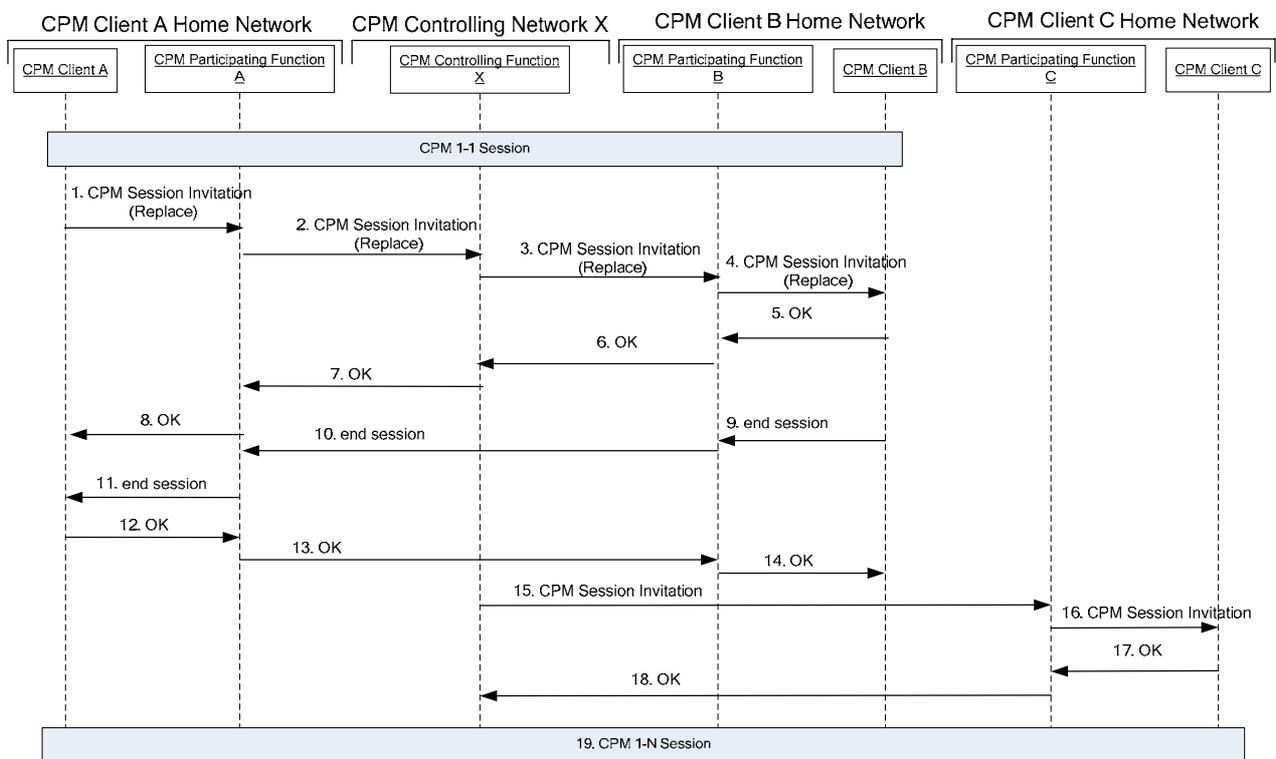


Figure 9: Extending a CPM 1-1 Session to a CPM Group Session

1-2. CPM Client A sends a CPM Session Invitation towards CPM Controlling Function X with a list of new invitees and the GRUU or the Instance ID of the original CPM Client B. An indication is added for Client B to replace the existing CPM 1-1 Session with the new CPM Group Session.

3-4. CPM Controlling Function X sends a CPM Session Invitation with the “Replace” indication to CPM Client B.

5-6. CPM Client B sends an OK response.

7-8 CPM Controlling Function X responds with an OK response to CPM Client A

NOTE 2: CPM Controlling Function X responds with an OK response to Client A when at least one invitation has been accepted (not necessarily from CPM Client B).

9-14. CPM Client B ends the original CPM 1-1 Session with CPM Client A

15-16. CPM Controlling Function X sends a CPM Session Invitation to CPM Client C.

17-18. CPM Client C sends an OK response.

NOTE 3: Steps 15-18 are repeated for each newly invited CPM Client.

NOTE 4: Steps 3-14 can be executed before or after steps 15-18.

19. CPM Controlling Function X has established a CPM Group Session with all invited CPM Clients who accepted the invitation.

### B.2.2.2 Adding Participants to an Existing CPM Group Session

This section describes the case where a CPM Client participating in a CPM Group Session adds further Participants to the CPM Session.

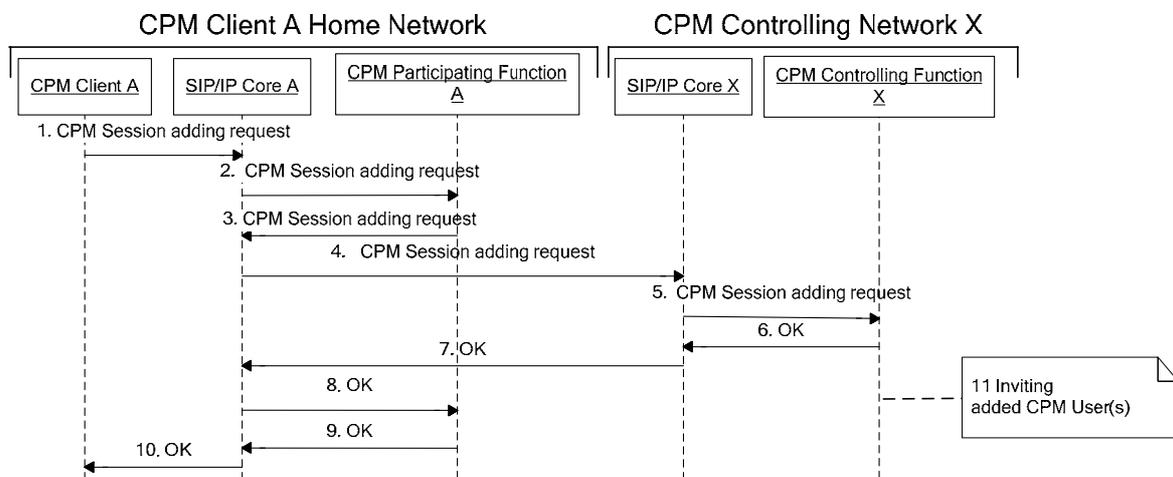


Figure 10: CPM Client adding users to an existing CPM Group Session

1-2. CPM Client A sends a CPM Session adding request containing the addresses of the Participants to be added towards CPM Participating Function A.

3-5. CPM Participating Function A forwards the CPM Session adding request towards CPM Controlling Function X.

6. CPM Controlling Function X checks the CPM Session adding request against the CPM Group policies (if the CPM Group Session is for a CPM Pre-defined Group) and/or service provider policies and if authorized CPM Controlling Function X confirms the reception of the CPM Session adding request by sending an OK response.

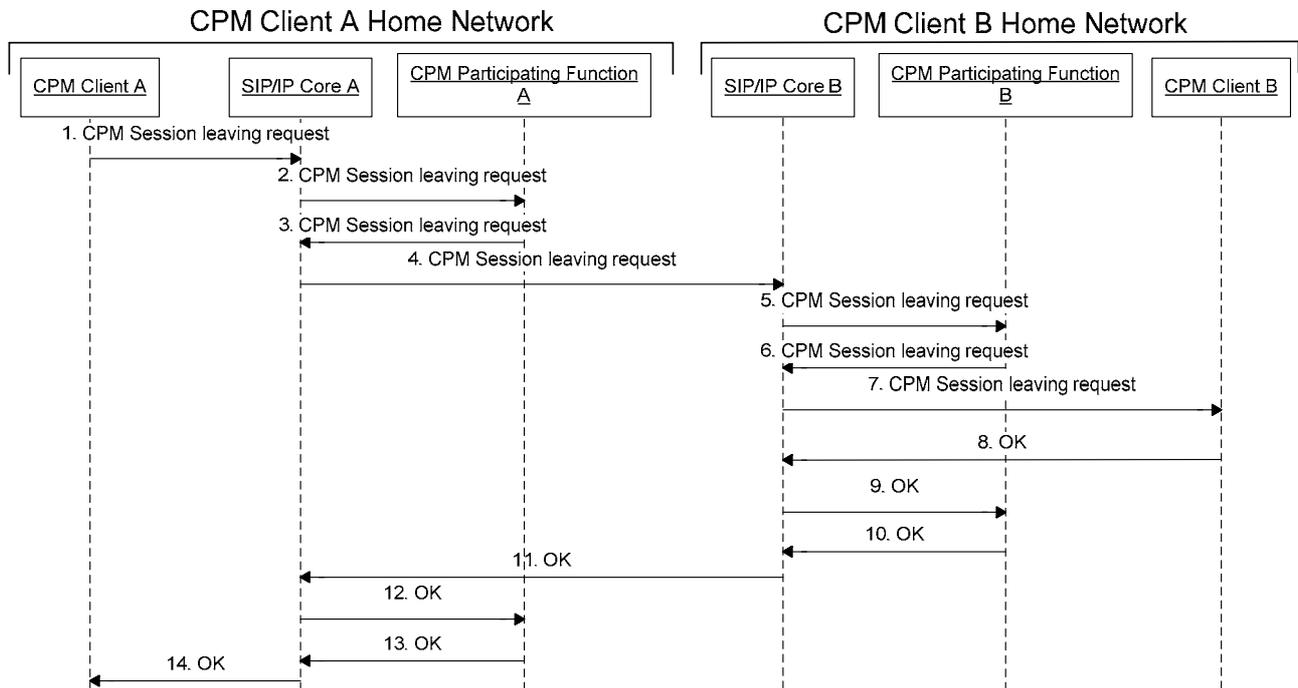
7-10. The OK response is routed towards CPM Client A.

11. CPM Controlling Function X invites each Participant listed in the CPM Session adding request to the CPM Session as described in Figure 7.

## B.2.3 CPM Session Termination

### B.2.3.1 CPM User Closing a CPM 1-1 Session

This section describes the case where a CPM User closes a CPM Session that he/she has with a peer user.

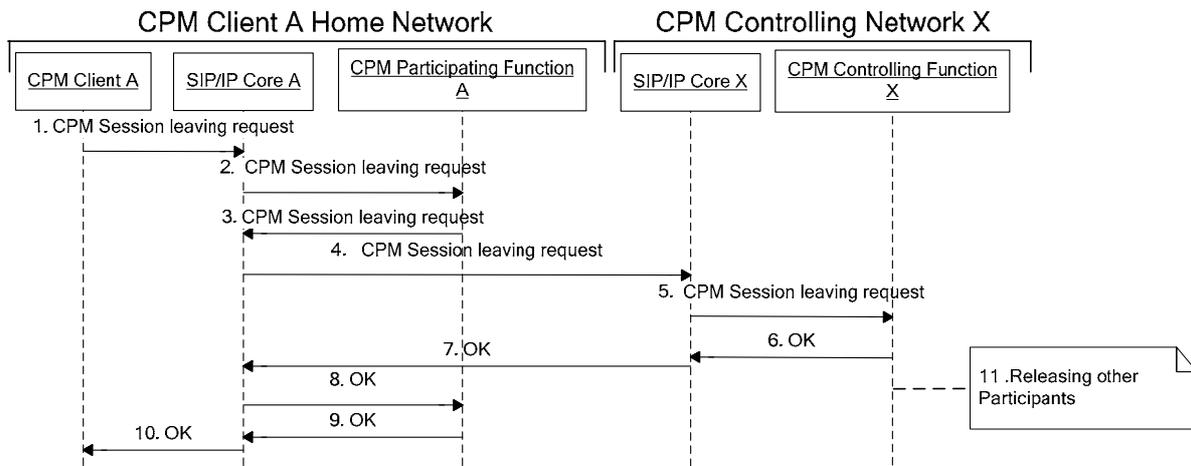


**Figure 11: Closing a CPM 1-1 Session**

1. CPM Client A sends a CPM Session leaving request to CPM Client B.
2. SIP/IP Core A routes the CPM Session leaving request to CPM Participating Function A.
3. CPM Participating Function A executes the necessary originating service control and then sends the CPM Session leaving request to SIP/IP Core A.
4. SIP/IP Core A routes the request to the SIP/IP Core of CPM User B’s network.
5. SIP/IP Core B routes the request to CPM Participating Function B.
6. CPM Participating Function B executes the necessary terminating service control and then sends the CPM Session leaving request to SIP/IP Core B.
7. SIP/IP Core B sends the CPM Session leaving request to CPM Client B.
8. CPM Client B accepts the CPM Session leaving request with an OK response to SIP/IP core B.
- 9-14. The response is conveyed back to CPM Client A using the reverse routing of the CPM Session leaving request.

**B.2.3.2 Participant Leaving a CPM Group Session**

This section describes the case where a CPM Client participating in a CPM Group Session decides to leave the CPM Group Session.



**Figure 12: CPM Client leaving a CPM Group Session**

1-2. CPM Client A sends a CPM Session leaving request according to the Signalling Path used at CPM Group Session establishment.

3-5. CPM Participating Function A forwards the CPM Session leaving request according to the Signalling Path used at CPM Group Session establishment.

6. CPM Controlling Function X starts discarding Media received from CPM Client A, stops sending Media received from other Participants towards CPM Client A and confirms the reception of the CPM Session leaving request by sending an OK response.

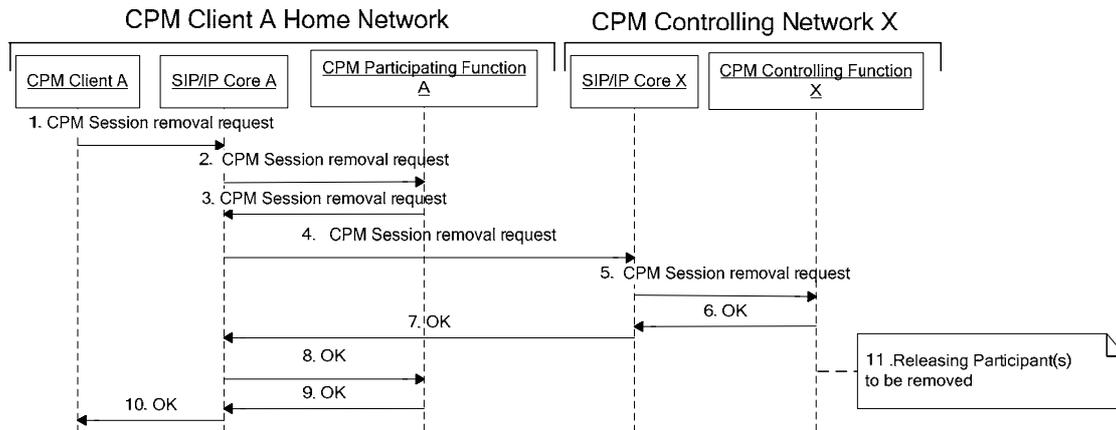
NOTE: This affects only Media that belong to this CPM Session. CPM Client A might be in other CPM Sessions at this time. Media of such CPM Sessions are not affected.

7.-10. The OK response is routed towards CPM Client A.

11. If the whole CPM Session is to be terminated according to the CPM Group policies and service provider policies, CPM Controlling Function X stops handling of the Media of all the other Participants and releases all the other Participants from the CPM Session as described in section B.2.3.4.

### B.2.3.3 Removing Participants from a CPM Group Session

This section describes the case where a Participant participating in a CPM Group Session removes other Participant(s) from the CPM Session.



**Figure 13: CPM Client removing other Participants from CPM Session**

1-2. CPM Client A sends a CPM Session removal request containing the addresses of the Participants to be removed.

3-5. CPM Participating Function A forwards the CPM Session removal request.

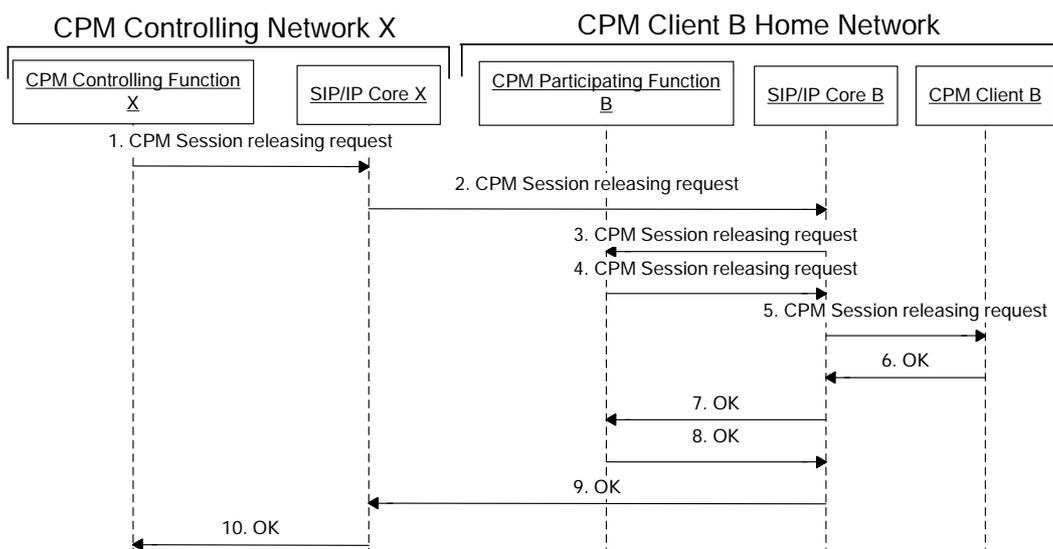
6. CPM Controlling Function X checks the CPM Session removal request against the CPM Group policies and/or service provider policies and if authorized CPM Controlling Function X confirms the reception of the CPM Session removal request by sending an OK response.

7-10. The OK response is routed towards CPM Client A.

11. CPM Controlling Function X releases the Participants listed in the CPM Session removal request from the CPM Session as described in the section B.2.3.4.

### B.2.3.4 Participant Being Released from a CPM Group Session

This section describes the case where a CPM Client participating in a CPM Group Session is released from the CPM Session.



**Figure 14: CPM Client being released from CPM Session**

1-3. CPM Controlling Function X starts discarding Media received from CPM Client B; stops sending Media received from other Participants towards CPM Client B; and sends a CPM Session releasing request towards CPM Client B.

4-5. CPM Participating Function B forwards the CPM Session releasing request according to the Signalling Path used at the CPM Session establishment.

6. CPM Client B confirms the reception of the CPM Session releasing request by sending an OK response.

7-10. The OK response sent by CPM Client B is routed towards CPM Controlling Function X.

The CPM Controlling Function removes CPM User B from the list of participants and sends a SIP NOTIFY to all current subscribers to this conference focus to inform of this update.

NOTE: When a CPM Group Chat that is also a Long-lived CPM Group Chat is released, the CPM Controlling Function X keeps locally the data related to the Long-lived CPM Group Chat, such as the information defined in the Group State Object data. This Long-lived CPM Group Chat information will be used by the CPM Controlling Function X when a request to re-start the CPM Controlling Function X is received.

### **B.2.3.5 Long-lived CPM Group Session**

This section describes the main flow of the Long-lived CPM Group Session.

Pre-requisites for all flows:

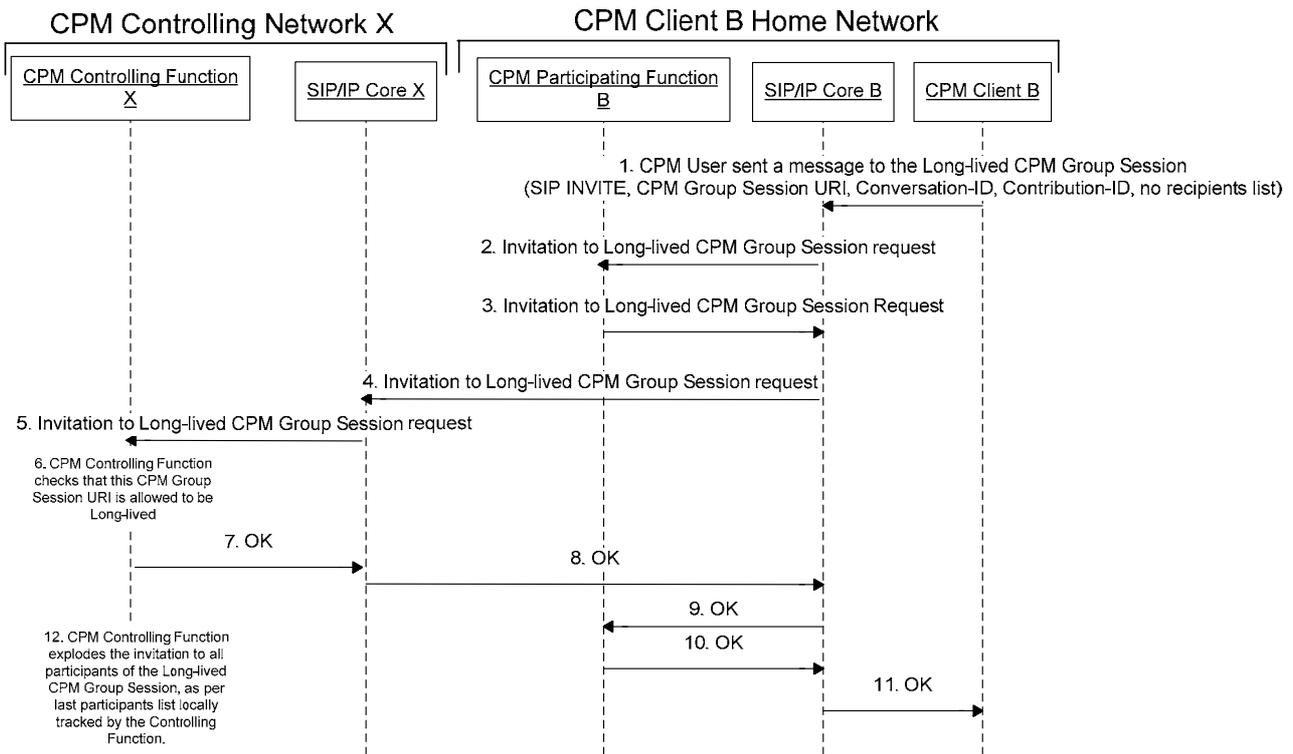
- the CPM Group Session was terminated due to inactivity (i.e. no participant has sent anymore messages) and,
- the Long-lived CPM Session is allowed and the CPM Client B is permitted to re-start the CPM Group Session, according to Service Provider policies, and

the CPM Client B has the latest participants information in the CPM Group Session that is being re-started.

#### **B.2.3.5.1 Basic flow – Long-lived Group Session re-started with same focus**

Pre-requisite:

- the focus of the Long-lived CPM Group Session is same as previously used by the CPM Group Session with same CPM Group Session URI.



**Figure 15: CPM Client re-starting a Long-lived CPM Group Session with same focus**

1-5. CPM User B selects the CPM Group Session and sends a message to the group. The CPM Client B sends a re-join invitation using the same Conversation-ID and Contribution-ID, CPM Group Session URI as the original CPM Group Session, and does not populate any recipients in the recipient list.

6. The Controlling Function X checks successfully that the CPM Group Session URI is valid and that the CPM Client B is allowed to re-start this CPM Group Session.

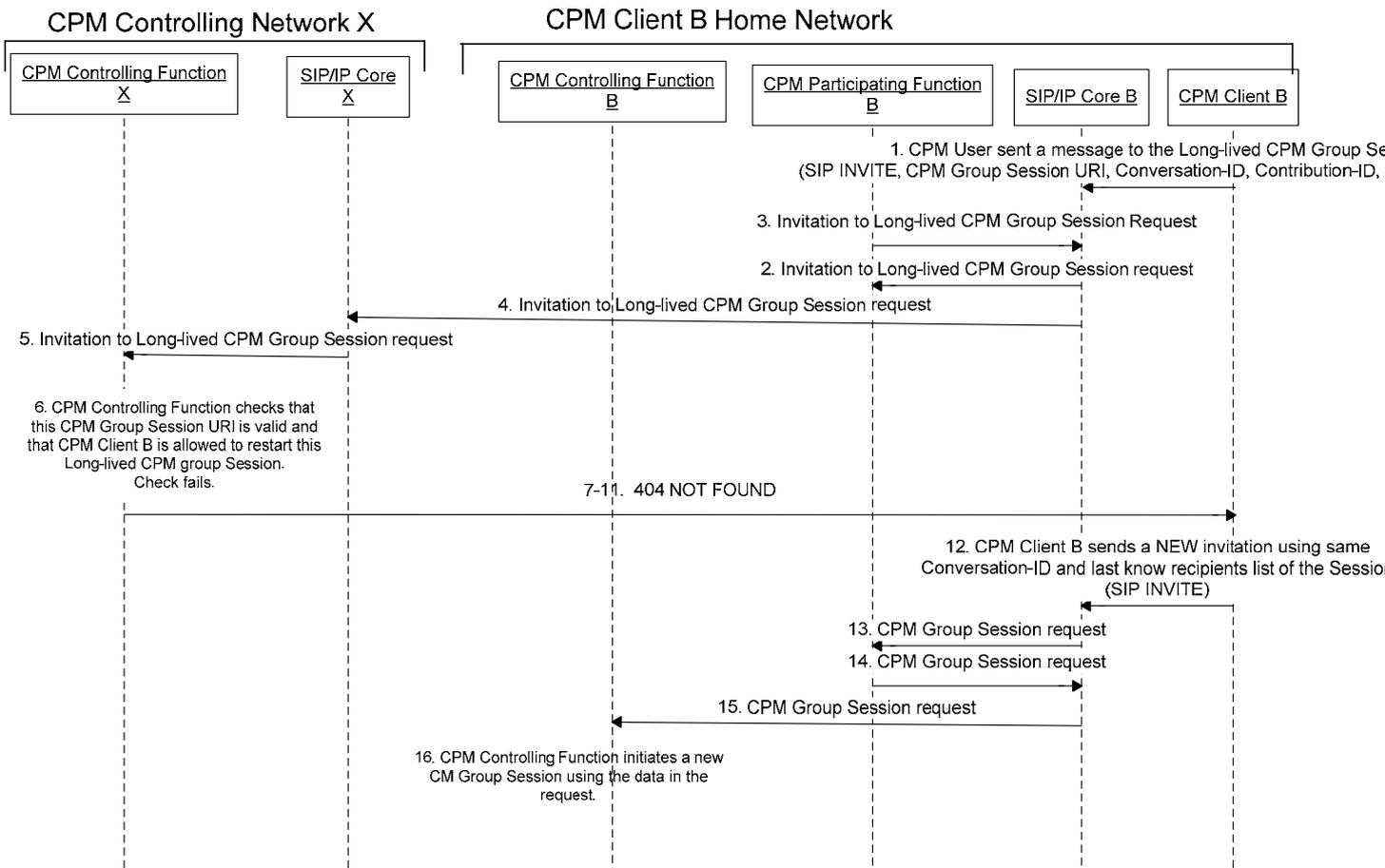
7-11. The Controlling Function X sends the OK response back to CPM Client B.

12. The CPM Controlling Function X explodes the invitation to all the last known participants in the CPM Group Session, including the recipients list, as per normal CPM Group Session establishment.

**B.2.3.5.2 Basic flow – new focus**

Pre-requisites:

- the focus of the Long-lived CPM Group Session is different than previously used (i.e. the Long-lived CPM Group Session is restarted using Controlling Function B, not necessarily the same Controlling Function X that handled a previous CPM Group Session of the same CPM Group Session URI).



**Figure 16: CPM Client re-starting a Long-lived CPM Group Session with a new focus**

1-5. CPM User B selects the CPM Group Session and sends a message to the group. The Client B sends a re-join invitation using the same Conversation-ID, CPM Group Session URI of the original CPM Group Session, and does not populate any recipients in the recipient list.

6. The Controlling Function X performs the following checks:

- that the CPM Group Session URI is valid and,
- that the CPM Client B is allowed to re-start a Long-lived CPM Group Session with CPM Controlling Function X ,

Either of these checks fail. This example flow shows the failure case due to conference URI not found, hence a 404 –NOT FOUND error with reason code “123 Session does not exist” is returned. (Alternatively, if the CPM User B is not allowed to re-start the CPM Group Session, a 403-FORBIDDEN would have been returned).

7-11. The Controlling Function X sends the 404 NOT FOUND response back to CPM Client B.

12-16. The CPM Client B sends a new invitation for CPM Group Session using similar procedure as in flow B.2.1.2. For this the Client is using the same data as the Long-lived CPM Group Session (Conversation-ID, Contribution-ID and populates recipients in the recipient list with last available information).

17. The Controlling Function B sends the invitation for new CPM Group Session to all participants as received in the recipients list of the request for the new CPM Group Session.

Alternative flows:

Depending on service provider policies, a possible scenario is that CPM Participating Function does not permit CPM User B to re-start a Long-lived CPM Group Session with the CPM Controlling Function X, so it returns a 404 NOT FOUND to the CPM User B. The flow in this case would continue with step 12.

### **B.2.3.6 CPM User re-joins an active CPM Group Session**

Pre-requisite:

- the CPM (Long-lived or not) Group Session is established and active throughout the entire flow,
- the recording and/or deferral of the CPM Group Chat are enabled,

The CPM Client B lost connection while the CPM Group Session was on-going. A network element of the SIP/IP Core (SBC) expires the MSRP session and sends a SIP BYE to the CPM Participating Function (and to CPM Client B) with.

#### **B.2.3.6.1 Client re-joins after the MSRP session was previously expired in the SIP/IP Core (timeout)**

This flow describes the behaviour in case the SIP/IP Core (e.g. SBCs) expires the MSRP session due to lack of response from Client B.

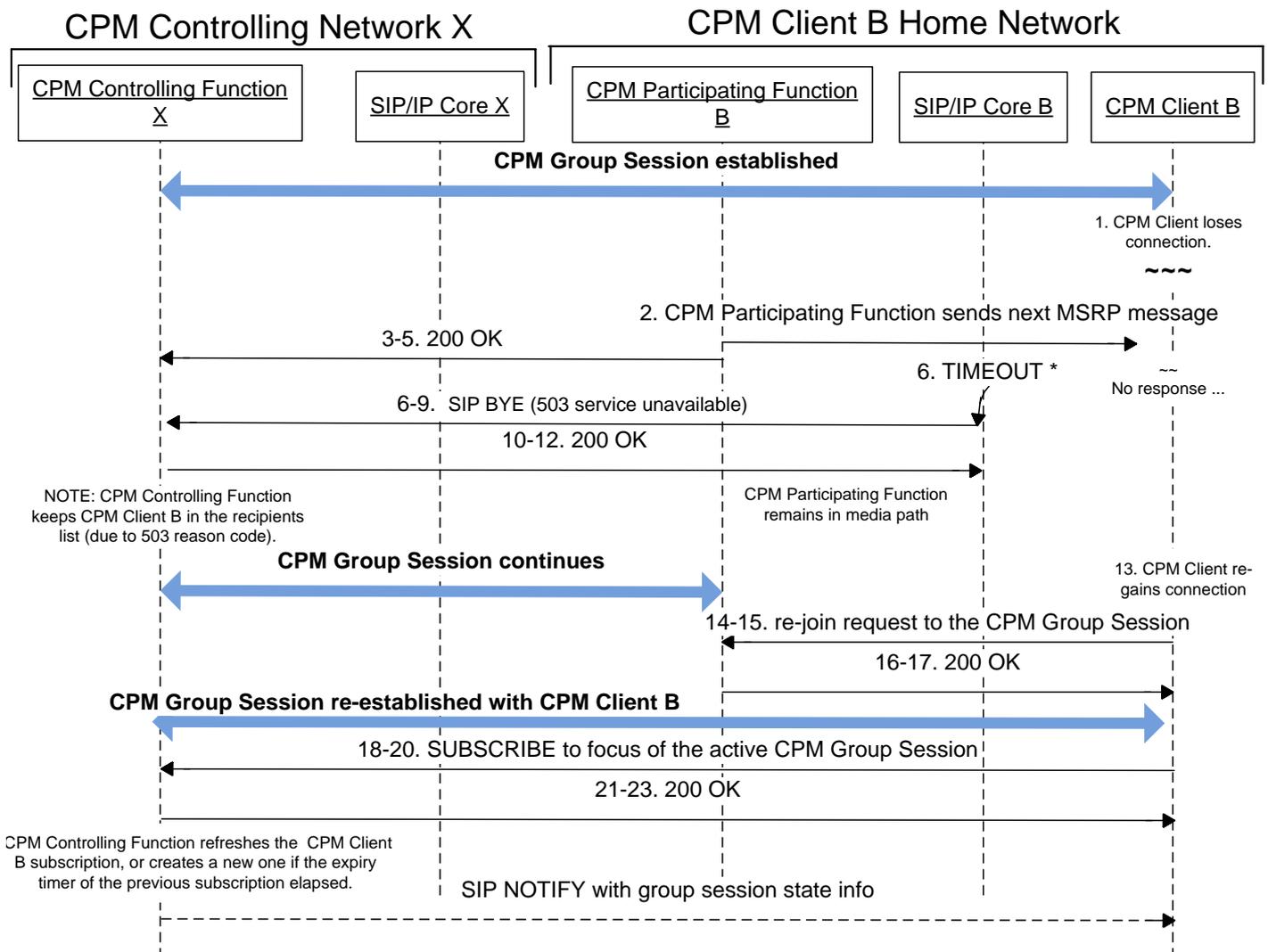


Figure 17: CPM Client re-joining an active CPM Group Session – expiry at SIP/IP Core

1. CPM Client B loses connection.
2. The CPM Participating Function B forwards the next MSRP chat message to CPM Client B.
- 3-5. The CPM Participating Function B sends back 200 OK towards the Controlling Function X.
- 6-9. A network entity within SIP/IP Core (e.g. SBC) expires the MSRP session due to no response received on MSRP leg from the CPM Client B and sends SIP BYE with a Reason header field with the cause parameter set to 503 (Service Unavailable) as per [3GPP TS 24.229] section. 5.2.8.1.2 “Release of an existing session” to the CPM Participating Function B. The SIP BYE is forwarded to CPM Controlling Function X via the SIP/IP Cores.
- 10-12. The CPM Controlling Function X acknowledges the SIP BYE with the SIP cause 503 and responds with a 200 OK, without removing the CPM Client B from the participants list nor terminating the leg towards CPM Participating Function B as per indication received in the Reason header (as opposed to a SIP BYE generated by the CPM User explicitly leaving the CPM Session that carries a different value e.g. “200 OK”).

CPM Participating Function B remains in media path on behalf of the CPM User B.

NOTE 1: If deferred delivery of the CPM Group Session is enabled, the CPM Participating Function B will also buffer all missed messages, disposition notifications and participants information received.

NOTE 2: If the recording of the CPM Group Session is enabled, the CPM Participating Function B stores all missed Group Session messages, disposition notifications and Group State Object in the CPM Message Storage of CPM User B.

13. CPM Client B regains connectivity.

14-15. The CPM Client B sends a re-join request for the CPM Group Session via the CPM Participating Function B.

16-17. CPM Participating Function B re-connects the legs and returns a 200 OK to the CPM Client B.

The CPM Group Session is now re-established with the CPM Client B.

18-20. The CPM Client B subscribes to the CPM Group Session focus, to retrieve the latest list of participants.

21-23. The Controlling Function X either refreshes the existing subscription, or creates a new one for CPM Client B if the previous one was expired. The Controlling Function X sends the 200 OK response back to CPM Client B. A SIP NOTIFY carrying the latest group session state also follows immediately towards the CPM Client B.

### **B.2.3.7 Deferred delivery of CPM Group Session**

This section describes the main flows of the Deferred delivery of CPM Group Sessions.

There are several cases in which a CPM Group Session is deferred by a CPM Participating Function and then delivered later on when the CPM User becomes available.

In all cases, the CPM Participating Function remains in the media path to intercept all CPM Group Session messages, disposition notifications and conference focus updates, in order to provide them in the delivery to the CPM Client when the CPM Client becomes available.

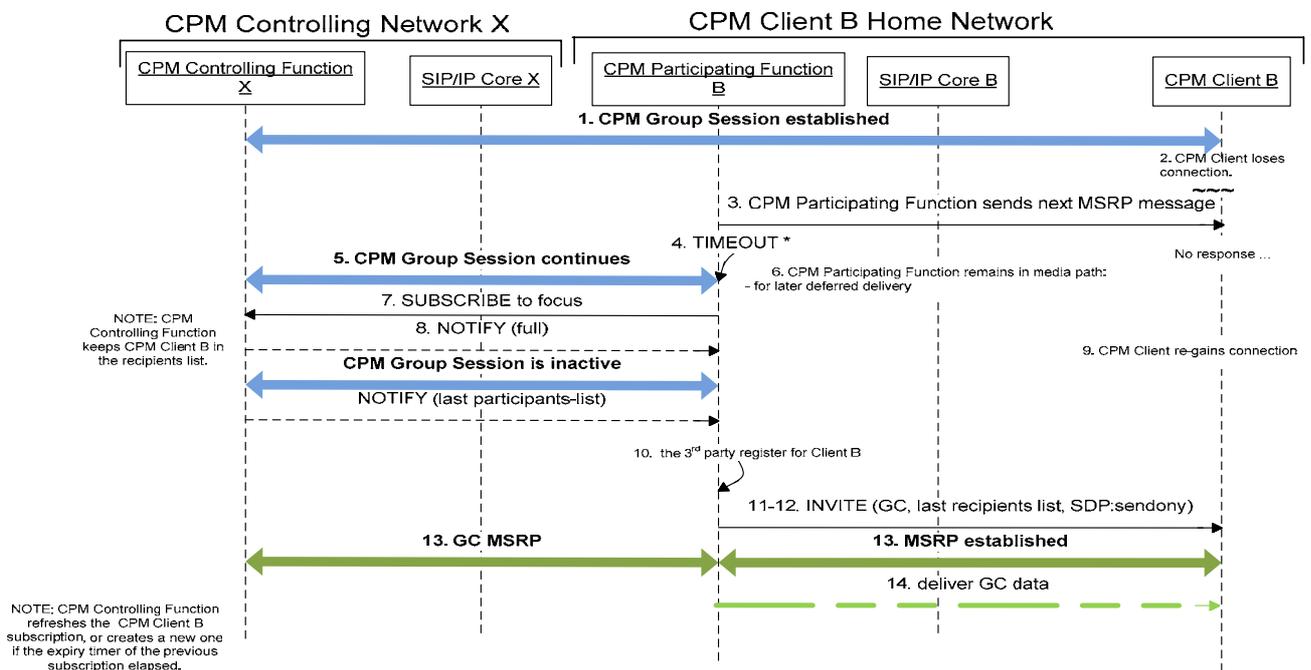
If the CPM Client B has already been involved in the CPM Group Session, the delivery of the CPM Group Session is made always to CPM Client B. For cases where the CPM Group Session was never accepted by the CPM User on any of his/her CPM Clients, the delivery can be made to any one of the CPM User B devices.

The following scenarios are possible:

- A) The CPM Group Session is inactive at the time when the CPM Client becomes available; the delivery of the CPM Group Session is initiated by the CPM Participating Function;
- B) The CPM Group Session is still on-going at the time when the CPM Client becomes available; the delivery of the CPM Group Session is initiated by the CPM Participating Function, which then re-joins the CPM Client with the on-going conference.

Scenario A is described in the example below.

### B.2.3.7.1 Deferred delivery of CPM Group Session while session is inactive



**Figure 18: Delivery, CPM Client re-joining an inactive CPM Group Session**

1. CPM Group Session is established, CPM Client B is a participant.
  2. CPM Client B loses connection.
  3. The CPM Participating Function B forwards the next MSRP chat message to CPM Client B.
  - 4-6. Due to no answer received, the CPM Participating Function B sends back 200 OK towards the Controlling Function X and stays in the media path, to buffer all further messages and IMDNs for this CPM Group Session.
  - 7-8. CPM Participating Function B subscribes to the conference focus (on behalf of CPM Client B), and retains the received information about the participants.
- NOTE: The CPM Group Session continues until it becomes inactive, and then the last participants list is provided to the CPM Participating Function in the last SIP NOTIFY.
9. CPM Client B regains connectivity.
  10. CPM Participating Function B is notified about the registration event of the CPM Client B.
  - 11-12. CPM Participating Function B sends an invitation to the inactive CPM Group Session to the CPM Client B, containing all the information from the original INVITE except for:
    - the SDP attribute containing 'sendonly' (i.e. indicates delivery purposes only) and,

- the participants list, for which the CPM Participating Function B uses the last participants list received in last notification from the CPM Controlling Function X.

13. CPM User B accepts the invitation and then the MSRP session is established.

14. The CPM Participating Function B delivers any pending CPM Messages for the CPM Group Session to the CPM Client B.

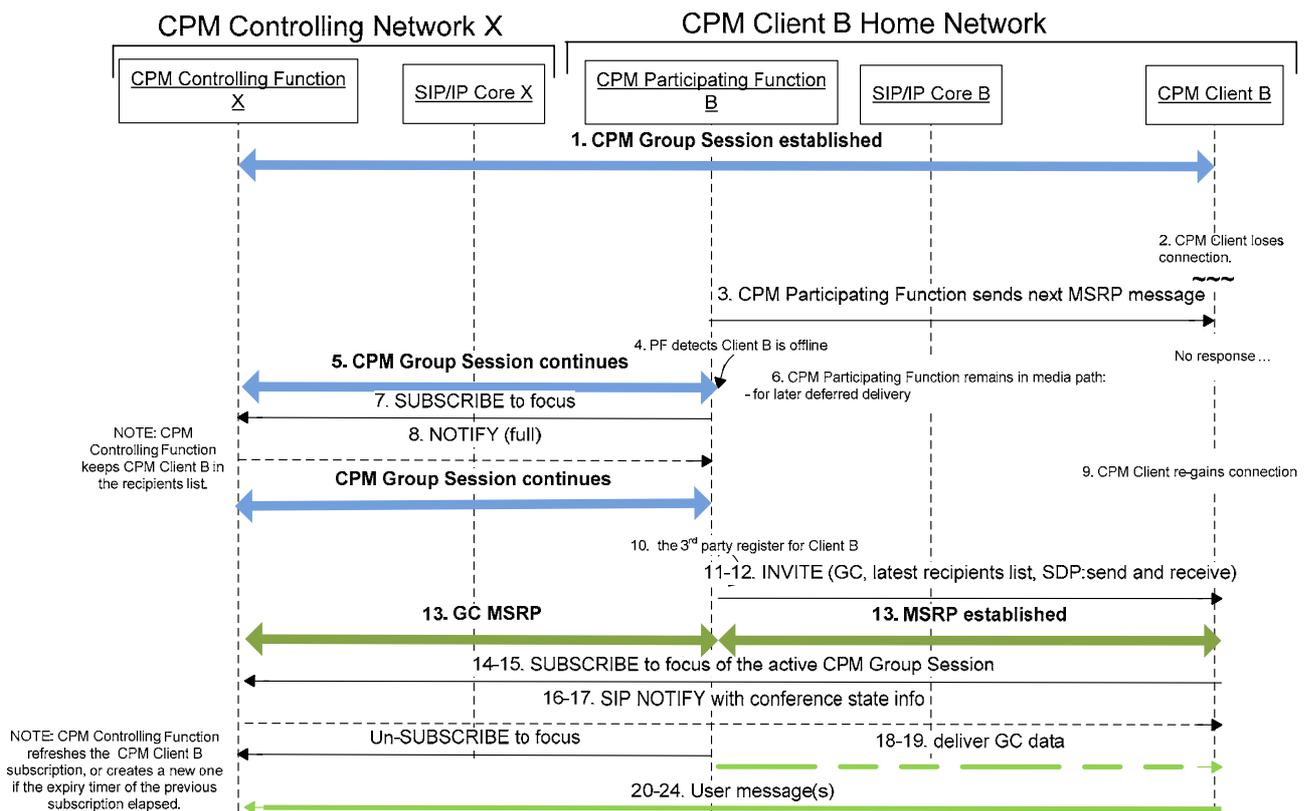
### B.2.3.7.2 PF initiates delivery of CPM Group Session, while the session is still on-going

In CPM v2.0, the CPM Group Session is delivered to the same device that accepted the original CPM Group Session. If the user never accepted the original CPM Group Session, then delivery can be done on any device of CPM User B.

#### Pre-conditions:

- CPM User B was a participant in the CPM Group Session, before losing connectivity;
- CPM User B does not take any action related to this CPM Group Session e.g. user doesn't send any message, etc (CPM Client B does not send a re-join).

After receiving the 3<sup>rd</sup> party register, the CPM PF invites the Client B back into the CPM Group Session.



1. CPM Group Session is established, CPM Client B is a participant.

2. CPM Client B loses connection.

3. The CPM Participating Function B forwards the next MSRP chat message to CPM Client B.

4-6. Due to no answer received, the CPM Participating Function B sends back 200 OK towards the Controlling Function X and stays in the media path, to buffer all further messages and IMDNs for this CPM Group Session.

7-8. CPM Participating Function B subscribes to the conference focus (on behalf of CPM Client B), and retains the received information about the participants changes.

9. CPM Client B regains connectivity.

10. CPM Participating Function B is notified about the registration event of the CPM Client B.

11-12. CPM Participating Function B sends an invitation to the active CPM Group Session to the CPM Client B, containing all the information from the original INVITE except for the participants list, for which the CPM Participating Function B uses the latest participants list received in notifications from the CPM Controlling Function X.

NOTE: as the CPM Group Session is active at the time of re-connecting the CPM Client B into the session, the INVITE contains a send and receive SDP indication (as opposed to the case where the CPM Group Session is inactive, where a sendonly session is initiated by the CPM Participating Function in view of delivery of the messages).

13. CPM User B accepts the invitation and then the MSRP session is established.

14-15. The CPM Client B subscribes to the conference focus of this CPM Group Session, to retrieve the latest list of participants.

NOTE: Depending on the timing and data of the subscribe request, the Controlling Function X either refreshes the existing subscription of the CPM Client B, or creates a new one.

16-17. The CPM Controlling Function X sends a SIP Notification with the latest conference state to CPM Client B.

NOTE: the CPM Participating Function B may unsubscribe from the conference focus at any time after the CPM Group Session is re-established.

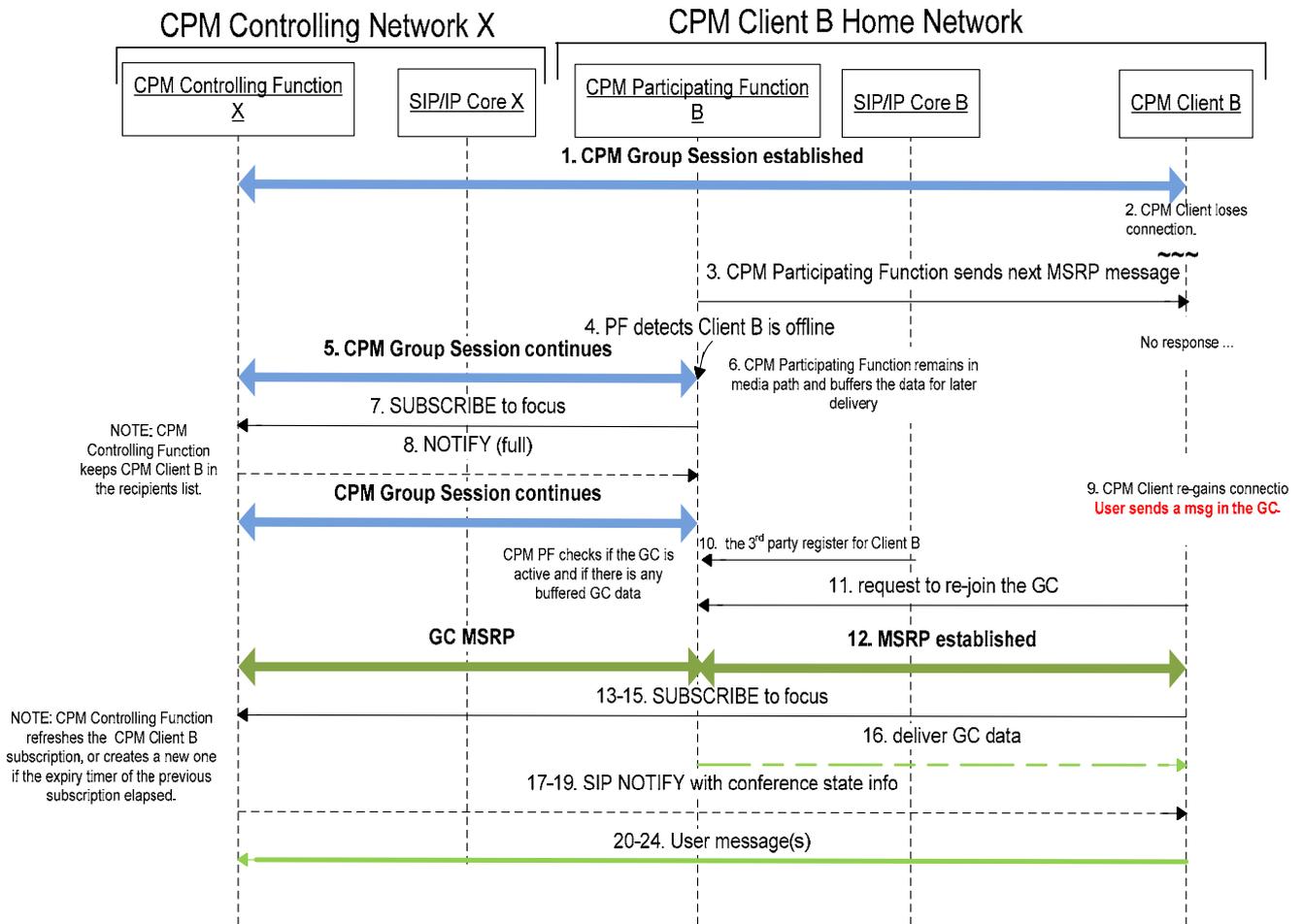
18-19. the PF delivers any pending CPM Messages for the CPM group Session to the CPM Client B.

20-24. Any live user messages from CPM Client B are handled within the session as per regular CPM Group Session procedures.

### **B.2.3.7.3 CPM Client B requests to re-join the active CPM Group Session**

#### **Pre-conditions:**

- CPM Client B determines that the CPM User B decided to re-join the CPM Group Session right after getting back online (e.g. types a message, opens this CPM Group Session window, etc – the action is dependent on the Client UI design which is out of scope of this specification)
- CPM Client B sends a re-join to the CPM PF, which will then proceed with the delivery of the buffered CPM Group Session data in this MSRP session.



1. CPM Group Session is established, CPM Client B is a participant.
2. CPM Client B loses connection.
3. The CPM Participating Function B forwards the next MSRP chat message to CPM Client B.
- 4-6. Due to no answer received, the CPM Participating Function B sends back 200 OK towards the Controlling Function X and stays in the media path, to buffer all further messages and IMDNs for this CPM Group Session.
- 7-8. CPM Participating Function B subscribes to the conference focus (on behalf of CPM Client B), and retains the received information about the participants changes.
9. CPM Client B regains connectivity.
10. CPM Participating Function B is notified about the registration event of the CPM Client B.
11. The CPM Client B sends a re-join request for the CPM Group Session to the CPM Participating Function B, before the CPM Participating Function B initiates the delivery invitation.
12. The session is established, including the MSRP.
- 13-15. The CPM Client B subscribes to the conference focus of this CPM Group Session, to retrieve the latest list of participants.

NOTE: Depending on the timing and data of the subscribe request, the Controlling Function X either refreshes the existing subscription of the CPM Client B, or creates a new one.

16. the PF delivers any pending CPM Messages for the CPM group Session to the CPM Client B.

17-19. The CPM Controlling Function X sends a SIP Notification with the latest conference state to CPM Client B.

NOTE: the CPM Participating Function B may unsubscribe from the conference focus at any time after the CPM Group Session is re-established.

20-24. Any live user messages from CPM Client B are handled within the session as per regular CPM Group Session procedures.

#### **B.2.3.7.4 Race conditions provisions**

If the CPM PF sends out the INVITE to CPM Group Session to the CPM Client B at the same time as the CPM Client B generates the re-join to the CPM PF B, then the CPM PF B shall handle it as follows:

- If the CPM Group Session is still active, then the CPM PF will only keep one session active with the CPM Client B for the same CPM Group Session (i.e. it will reject the re-join that comes in after the INVITE was already sent out by the PF, or it will suppress generating the INVITE to the Client if the re-join from Client was already received for that CPM Group Session).
- If the CPM Group Session is inactive, then the CPM PF will complete the delivery of the CPM Group Session data and then in parallel it shall forward the CPM Client B re-join request towards a CPM Controlling Function to re-start the CPM Group Session.

### **B.2.4 CPM Session Modification**

#### **B.2.4.1 Modifying Media Streams in a CPM 1-1 Session**

This section describes the case where a CPM Client participating in a CPM Session modifies used Media Streams such as adding/removing Media Stream(s) in a CPM Session or modifying the characteristics of already existing Media Streams.

In this scenario, CPM Participating Functions A and B decide not to stay in the Media path.

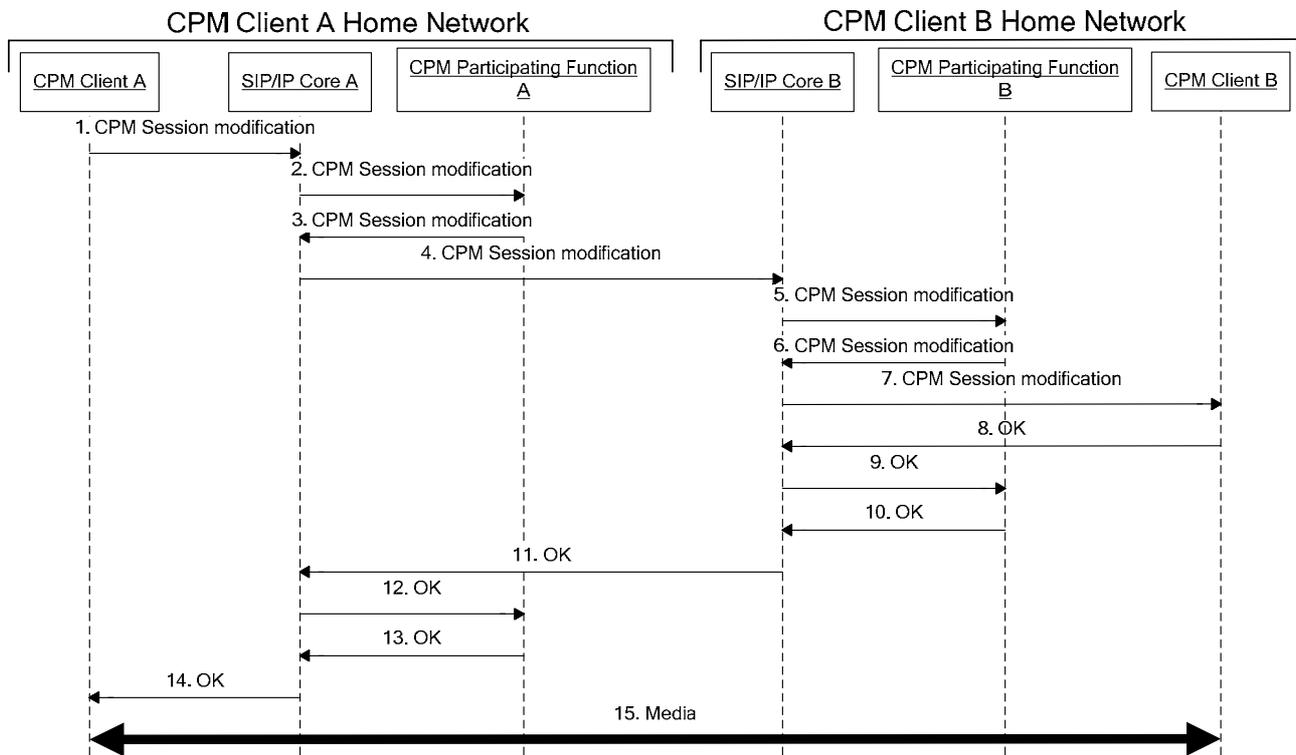


Figure 19: CPM 1-1 Session modification

1-2. CPM Client A sends a CPM Session modification request according to the Signalling Path used at CPM Session establishment. The CPM Session modification request contains the currently used Media Streams and additional Media Streams.

3-5. CPM Participating Function A checks the CPM Session modification request against service provider policies (e.g. authorization of the additional Media Streams not already present) and if successfully authorized it forwards the CPM Session modification request according to the Signalling Path used at CPM Session establishment.

6-7. CPM Participating Function B checks the CPM Session modification request against service provider policies (e.g. authorization of the additional Media Streams not already present) and if successfully authorized it forwards the CPM Session modification request according to the Signalling Path used at CPM Session establishment.

8-14. The OK response is routed towards CPM Client A.

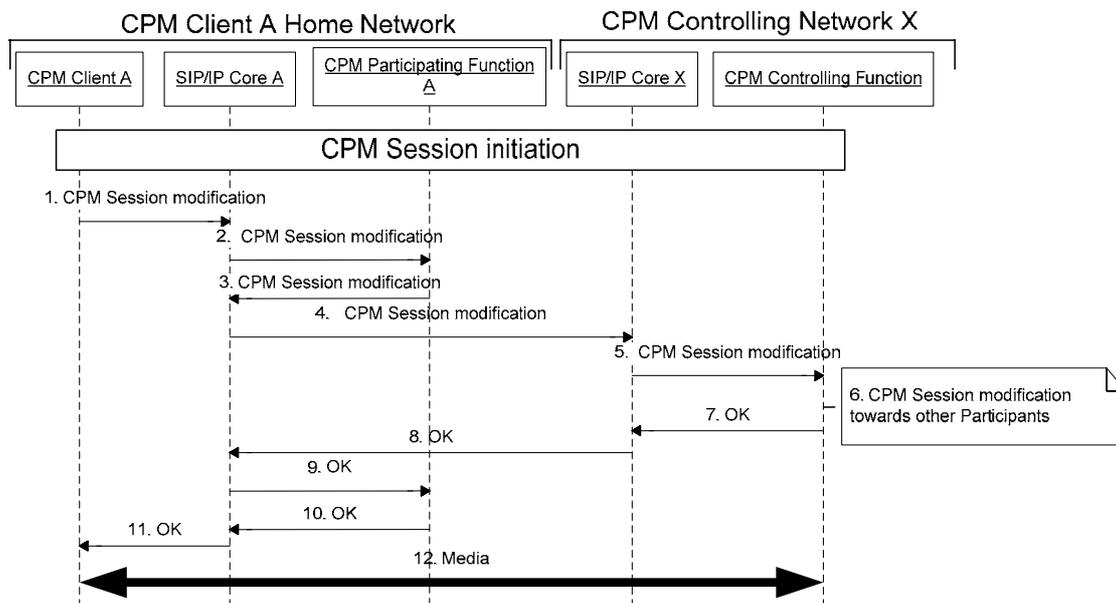
15. CPM Client A and CPM Client B exchange the Media within the newly negotiated Media Streams.

### B.2.4.2 Modifying Media Streams in a CPM Group Session

This section describes the case where a CPM Client participating in a CPM Group Session modifies used Media Streams such as adding/removing Media Stream(s) in a CPM Group Session or modifying the characteristics of already existing Media Streams.

CPM Participating Function A and CPM Participating Function B decide not to stay in the Media path.

Figure 15 shows the originating side of the flow.



**Figure 20: Modifying Media Streams in a CPM Group Session, originating side**

1-2. CPM Client A sends a CPM Session modification request according to the Signalling Path used at CPM Group Session establishment. The CPM Session modification request contains the changes requested as compared to the currently used Media Streams.

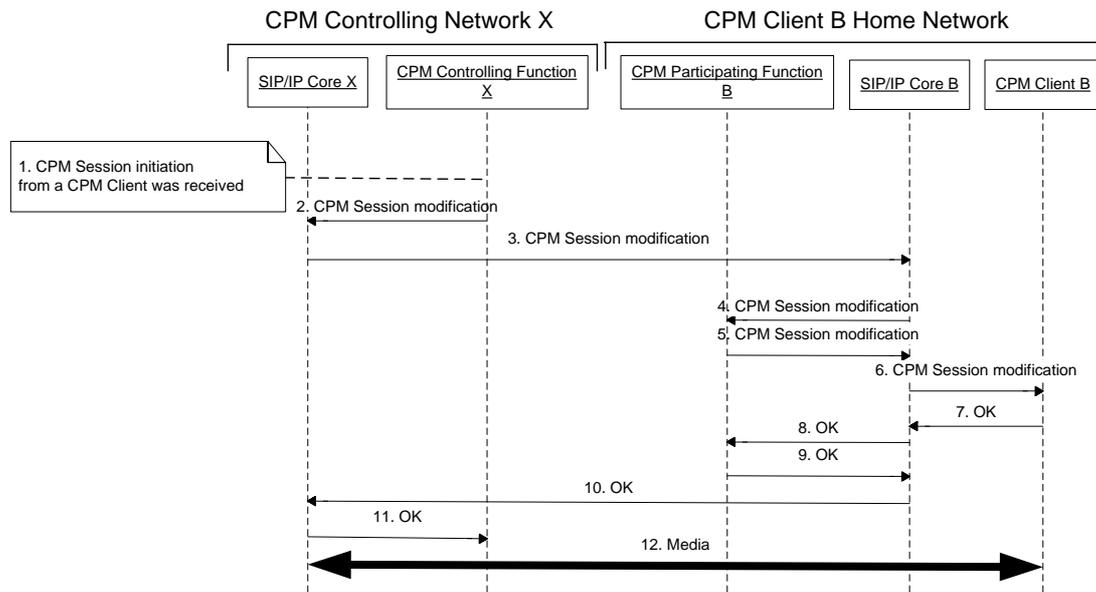
3-5. CPM Participating Function A checks the CPM Session modification request against service provider policies (e.g. authorization of the additional Media Streams not already present) and if successfully authorized CPM Participating Function A forwards the CPM Session modification request according to the Signalling Path used at CPM Group Session establishment.

6. CPM Controlling Function X checks the CPM Session modification request against the CPM Group policies and/or service provider policies (e.g. authorization of the originator of the CPM Session modification request and authorization of the additional Media Streams not used yet) and if successfully authorized CPM Controlling Function X updates the Media Plane configuration towards CPM Client A so that the additional Media Streams start to be used (or that removed Media Streams are not used anymore or that Media Streams with updated characteristics are used accordingly). CPM Controlling Function X initiates CPM Session modification towards those affected CPM Users as described in Figure 16 offering the Media Streams currently used by each affected CPM Client and the additional/removed/updated Media Streams as added/removed/updated by each affected CPM Client.

7-11. An OK response is routed towards CPM Client A.

12. CPM Client A and CPM Controlling Function X exchange the Media within the newly negotiated Media Streams. CPM Client A sends to CPM Controlling Function X the Media acquired at the Device and CPM Client A renders the Media received from CPM Controlling Function X. CPM Controlling Function X sends to CPM Client A the mixed Media received from other participating CPM Clients and CPM Controlling Function X mixes the Media received from CPM Client A into the Media sent towards the other participating CPM Clients.

Figure 16 shows the terminating side of the flow.



**Figure 21: Modifying Media Streams in a CPM Group Session, terminating side**

1. CPM Controlling Function X receives a CPM Session modification request from a CPM Client to modify Media Streams of the CPM Group Session as described in section **Error! Reference source not found.**.

2-4. CPM Controlling Function X sends a CPM Session modification request towards CPM Client B offering the Media Streams currently used by CPM Client B and the added/removed/modified Media Streams as requested by CPM Client A.

5-6. CPM Participating Function B checks the CPM Session modification request against the service provider policies and if successfully authorized, CPM Participating Function B forwards the CPM Session modification request according to the Signalling Path used at the CPM Session establishment.

7. CPM Client B may present the updated Media Stream set to the CPM User and accepts all or non-empty subset of the offered Media Streams according to the CPM User decision. CPM Client B responds with an OK response containing the accepted Media Streams.

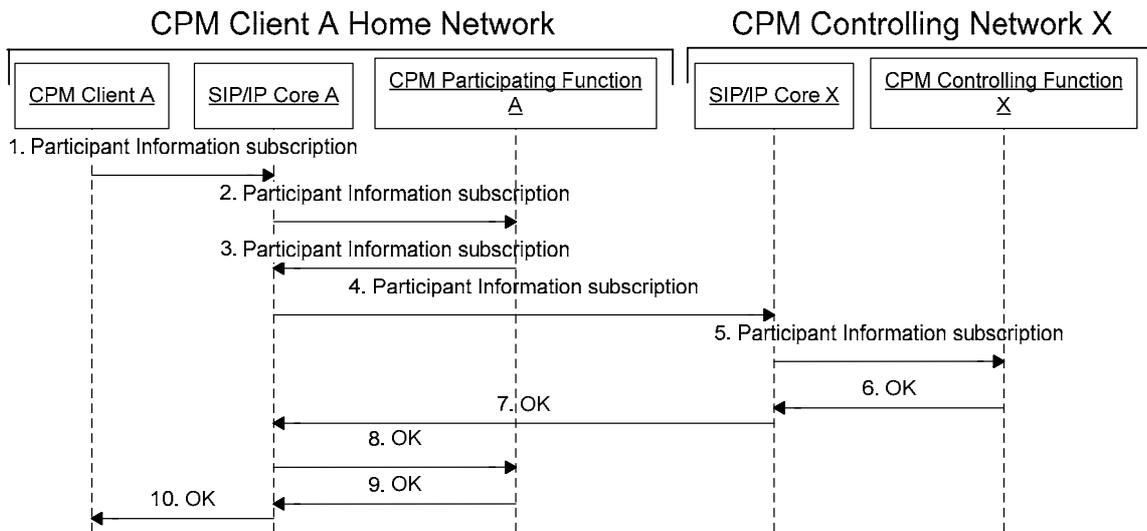
8-11. The OK response is routed towards CPM Controlling Function X.

12. CPM Client B and the CPM Controlling Function X exchange the Media within the newly negotiated Media Streams. CPM Client B sends to CPM Controlling Function X the Media acquired at the Device and CPM Client B renders the Media received from CPM Controlling Function X. CPM Controlling Function X sends to CPM Client B the mixed Media received from other participating CPM Clients and CPM Controlling Function X mixes the Media received from CPM Client B into the Media sent towards the other participating CPM Clients.

## B.2.5 Participant Information

### B.2.5.1 Participant Information Subscription

This section describes the case where a CPM Client subscribes to Participant Information of a CPM Group Session.

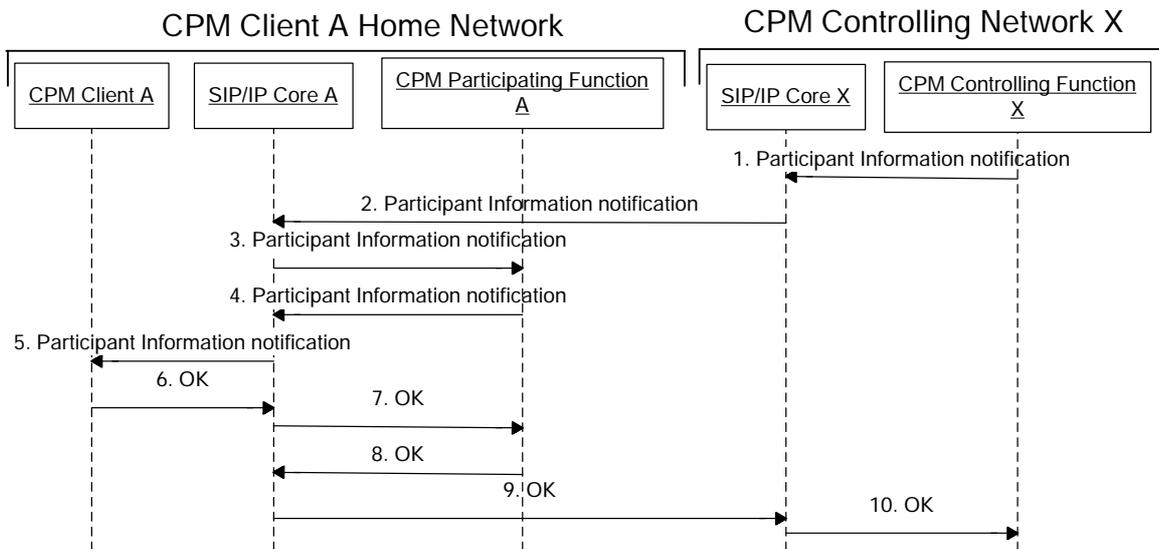


**Figure 22: CPM Client subscribes to Participant Information of a CPM Session**

1. CPM Client A sends a Participant Information subscription request towards CPM Participating Function A, including the CPM Group Session Identity of the CPM Group Session.
2. SIP/IP Core A routes the Participant Information subscription request to CPM Participating Function A.
3. CPM Participating Function A forwards the Participant Information subscription request towards the CPM Controlling Function hosting the CPM Group Session identified by the CPM Group Session Identity through SIP/IP Core A.
4. SIP/IP Core A routes the Participant Information subscription request to SIP/IP Core X.
5. SIP/IP Core X routes the Participant Information subscription request to CPM Controlling Function X hosting the CPM Group Session identified by the CPM Group Session Identity.
6. CPM Controlling Function X checks the Participant Information subscription request against the CPM Group policies and service provider policies and if successfully authorized, CPM Controlling Function X responds with an OK response. After sending the OK response, CPM Controlling Function X sends the Participant Information notification as described in section B.2.5.2.
- 7-10. The OK response is forwarded towards CPM Client A.

### B.2.5.2 Participant Information Notification

This section describes the case where a CPM Controlling Function notifies a CPM Client subscribed to Participant Information as described in section B.2.5.1 about the initial state of the Participant Information and later changes of the Participant Information.



**Figure 23: CPM Client receives Participant Information of a CPM Session**

1. Upon reception of a Participating Information subscription request and subsequently upon detecting changes in the Participant Information, CPM Controlling Function X sends a Participant Information notification to CPM Client A.
- 2-5. The Participant Information notification is forwarded using the reverse path of the Participant Information subscription request sent by CPM Client A.
6. CPM Client A confirms delivery of the Participant Information notification by sending an OK response.
- 7-10. The OK response is forwarded towards CPM Controlling Function X.

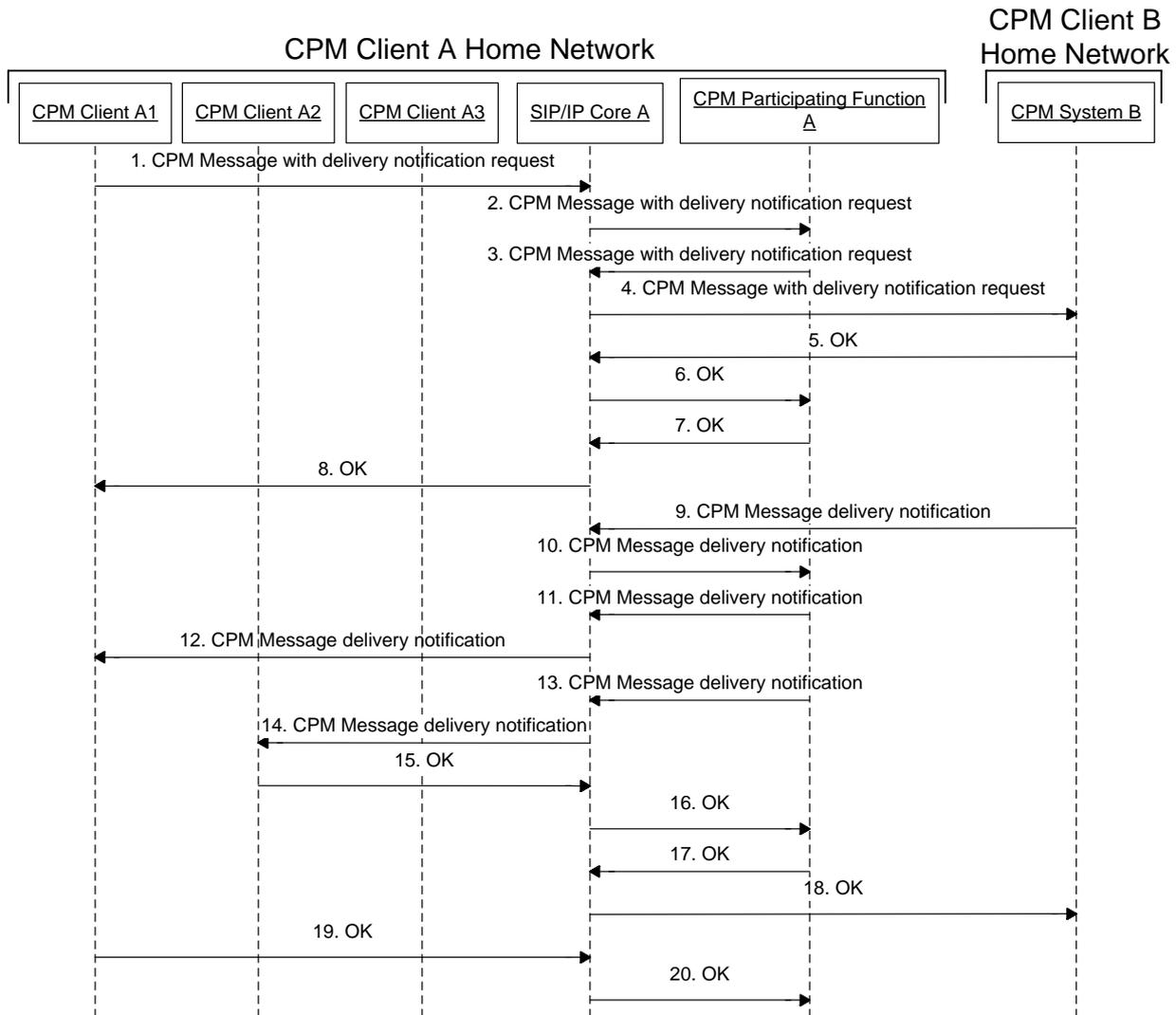
## B.3 CPM Standalone Messages

### B.3.1 Pager Mode CPM Standalone Message related flows

#### B.3.1.1 Sending a Pager Mode CPM Standalone Message to a Single Recipient

This section describes the case where a CPM User sends a Pager Mode CPM Standalone Message to another CPM User. The flow includes the handling of requests for disposition notifications. The flow describes the case where the originating CPM User has multiple devices, however, in the case that the originating CPM User has only one device then a similar procedure (in that procedure only the single CPM Client of the originator would be taken into account for the delivery of the CPM Standalone Message delivery notification) would be executed. The flow also includes the handling of requests for disposition notifications.

Figure 19 shows the originating side of the flow.



**Figure 24: Sending a CPM Standalone Message to a CPM User (originating side)**

1. CPM Client A1 sends a CPM Standalone Message towards a recipient. CPM Client A1 may request a CPM Standalone Message delivery notification and/or read report. In this flow CPM Client A requests a CPM Standalone Message delivery notification.
2. SIP/IP Core A routes the CPM Standalone Message to CPM Participating Function A triggered on the CPM Feature Tag and CPM User A’s CPM Address.
3. CPM Participating Function A performs originating CPM services and then sends the CPM Standalone Message to SIP/IP Core A.
4. SIP/IP Core A routes the CPM Standalone Message to CPM System B.
5. CPM System B returns an OK response after delivering the CPM Standalone Message.
6. The OK response is routed to CPM Participating Function A along the same Signalling Path.
- 7-8. CPM Participating Function A sends an OK response to CPM Client A along the same Signalling Path.

9. CPM System B generates a CPM Standalone Message delivery notification based on user preferences and service provider policies and then routes the CPM Standalone Message delivery notification to SIP/IP Core A.

10. SIP/IP Core A routes the CPM Standalone Message delivery notification to CPM Participating Function A triggered on the CPM Feature Tag.

11-14. CPM Participating Function A performs terminating CPM services and decides, based on registration status, user preferences and service provider policies, to which CPM Clients the CPM Standalone Message delivery notification is to be delivered (via SIP/IP Core A). In this case the CPM Standalone Message delivery notification is delivered to CPM Clients A1 and A2.

NOTE: CPM Standalone Message delivery notifications are treated like any CPM Standalone Message.

15-16. CPM Client A2 acknowledges the reception of the CPM Standalone Message delivery notification with an OK response.

17-18. The OK response is routed by CPM Participating Function A to CPM System B along the same Signalling Path.

19-20. CPM Client A1 acknowledges the reception of the CPM Standalone Message delivery notification with an OK response. This OK response is not relayed further upstream by CPM Participating Function A.

Figure 20 shows the terminating side of the flow for the case that the Pager Mode CPM Standalone Message is delivered successfully. The flow describes the case where the recipient CPM User has multiple devices, however, in the case that the recipient CPM User has only one device then a corresponding procedure would be executed. The flow also includes the handling of requests for disposition notifications.

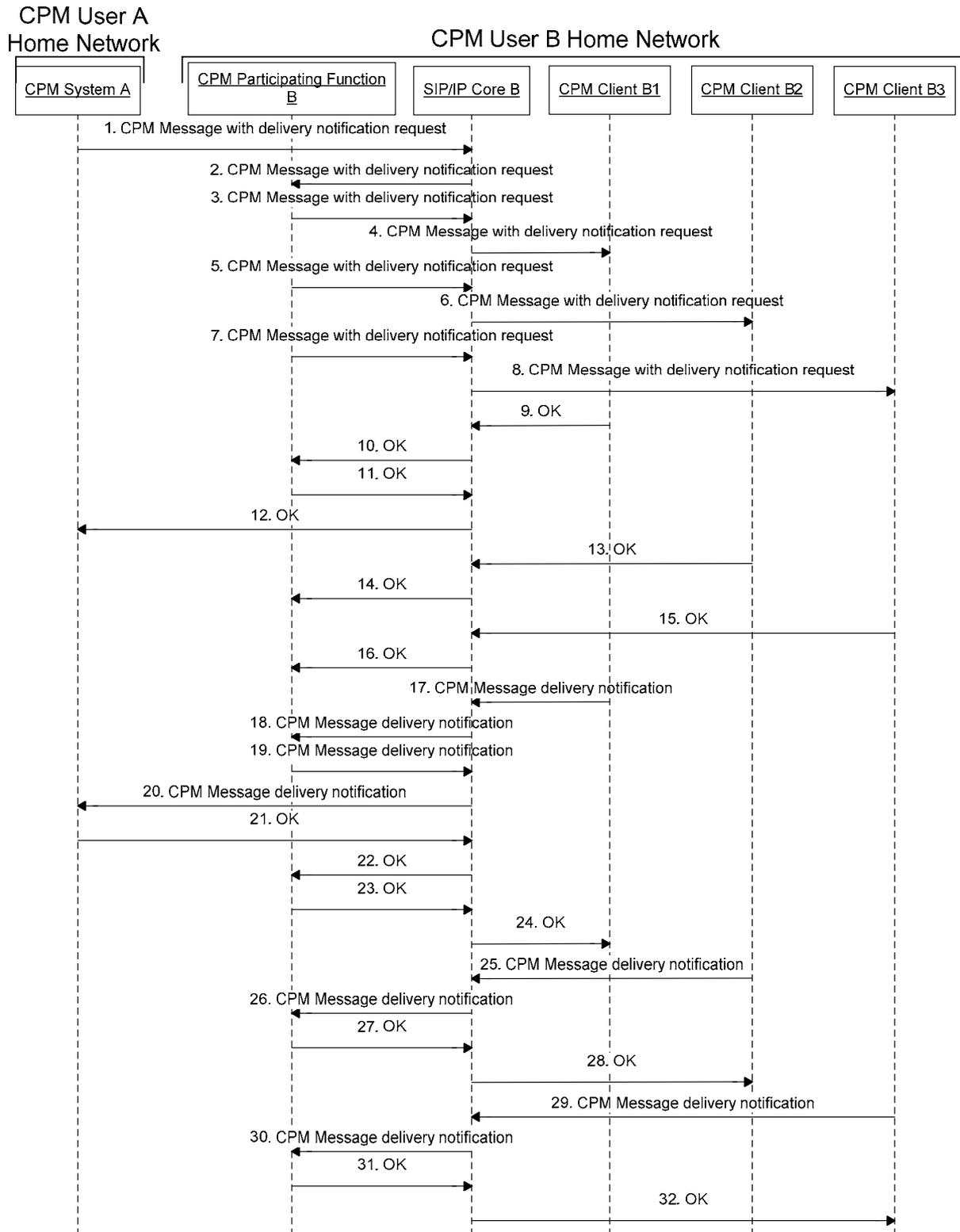
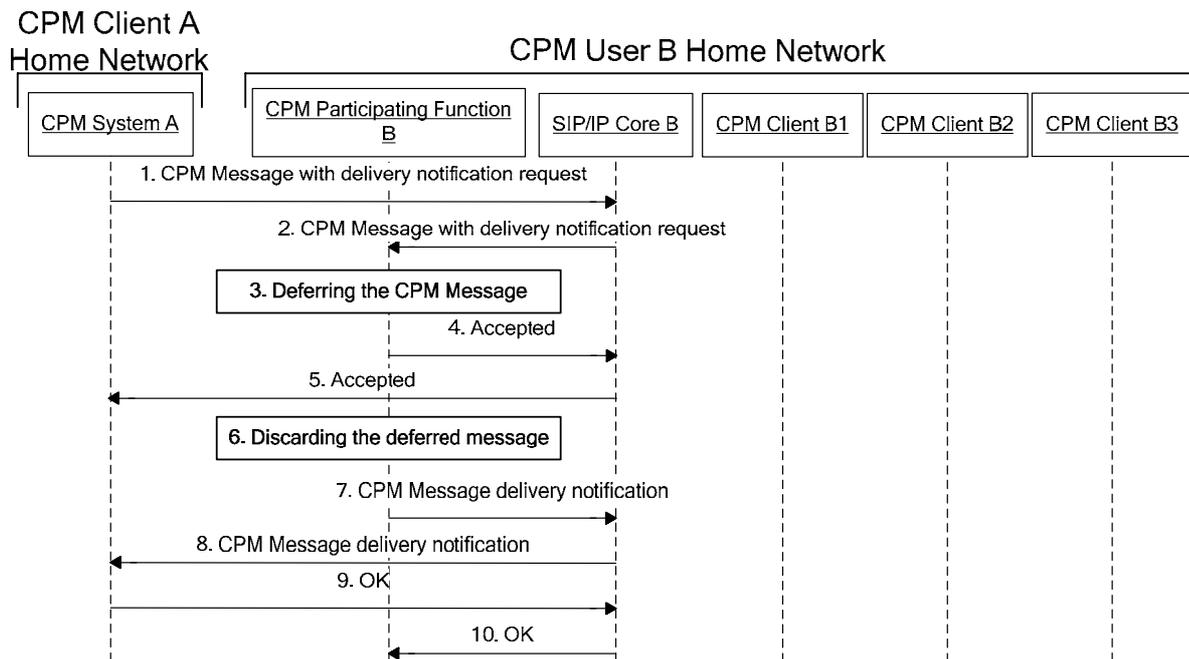


Figure 25: Sending a CPM Standalone Message to a CPM User (terminating side)

1. CPM System A sends a CPM Standalone Message with delivery notification request that arrives at SIP/IP Core B.
2. SIP/IP Core B routes the CPM Standalone Message to CPM Participating Function B based on the CPM Feature Tag and CPM User B's CPM Address.
- 3-8. CPM Participating Function B executes the necessary terminating service control and if successful, sends the CPM Standalone Message via SIP/IP Core B towards CPM Client B1, CPM Client B2 and CPM Client B3.  
NOTE 1: It is up to implementation in which order to serve the different clients.
- 9-10. CPM Client B1 acknowledges the reception of the CPM Standalone Message with an OK response.
- 11-12. The OK response is relayed back to CPM System A.
- 13-16. CPM Client B2 and CPM Client B3 acknowledge the reception of the CPM Standalone Message with an OK response each. These OK responses are not relayed further upstream by CPM Participating Function B.
- 17-18. CPM Client B1 generates a CPM Standalone Message delivery notification and sends the CPM Standalone Message delivery notification to Participating Function B, via SIP/IP Core B.
- 19-20. CPM Participating Function B forwards the CPM Standalone Message delivery notification towards CPM System A.
- 21-22. CPM System A acknowledges the reception of the CPM Standalone Message delivery notification with an OK response.
- 23-24. The OK response is relayed to CPM Client B1.
- 25-32. CPM Client B2 and CPM Client B3 each generate a CPM Standalone Message delivery notification and send the respective CPM Standalone Message delivery notification to CPM Participating Function B, via SIP/IP Core B. As a CPM Standalone Message delivery notification was already sent to CPM System A, these CPM Standalone Message delivery notifications are not forwarded by CPM Participating Function B, but just acknowledged with an OK response.  
NOTE 2: In this example, it is CPM Client B1 that was first in sending the OK response on the received CPM Standalone Message (step 9) and it was also first in sending a CPM Standalone Message delivery notification (step 17). It is not necessarily the case though that the device that received the message first, would also send an OK response first. Also, the order of sending CPM Standalone Message delivery notifications is independent of the order of having received the corresponding message.

Figure 21 shows the terminating side of the flow for the case that a Pager Mode CPM Standalone Message with delivery notification request fails to reach the target. In the flow it is assumed that the recipient's user preferences are to defer the CPM Standalone Message when none of the terminating CPM User's CPM Clients are available to receive the CPM Standalone Message and that the CPM Standalone Message cannot be delivered before it expires and that the user preference for expired messages is to discard the expired Deferred CPM Messages.



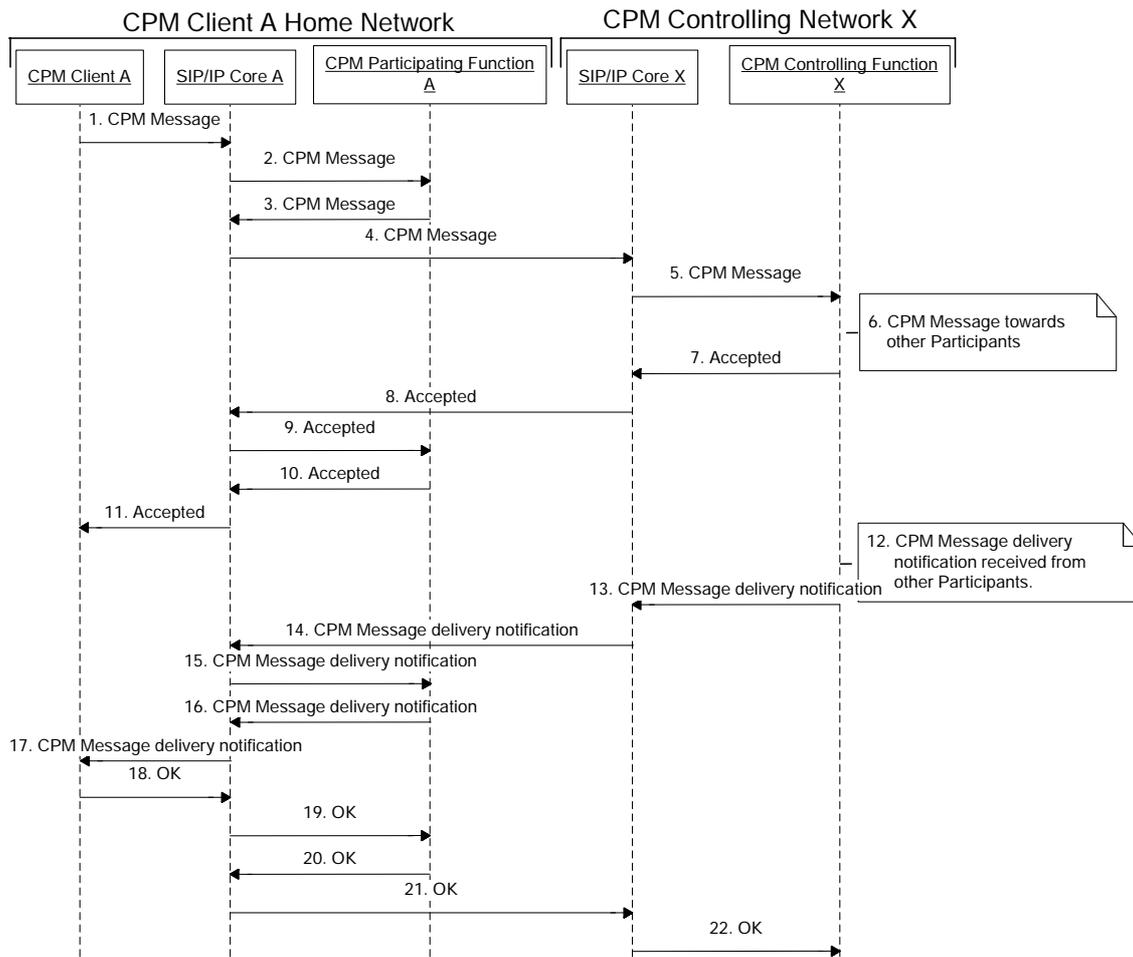
**Figure 26: Unsuccessful delivery of CPM Standalone Message with a delivery notification request**

1. CPM System A routes a CPM Standalone Message with delivery notification request to SIP/IP Core B.
2. SIP/IP Core B routes the CPM Standalone Message to CPM Participating Function B.
3. CPM Participating Function B identifies that the recipient is not available to receive the message and defers the CPM Standalone Message based on user preferences.
4. CPM Participating Function B returns an Accepted response to SIP/IP Core B.
5. SIP/IP Core B routes the Accepted response to CPM System A.
6. CPM Participating Function B discards the Deferred CPM Message for some reason (e.g. expiration or user request).
7. CPM Participating Function B generates a CPM Standalone Message delivery failure notification (depending on service provider policies) and then sends the CPM Standalone Message delivery failure notification to SIP/IP Core B.
8. SIP/IP Core B routes the CPM Standalone Message delivery failure notification to CPM System A.
9. CPM System A returns an OK response to SIP/IP Core B.
10. SIP/IP Core B routes the OK response to the CPM Participating Function B.

### B.3.1.2 Pager Mode CPM Standalone Message to a CPM Group

This section describes the originating case where a CPM User sends a Pager Mode CPM Standalone Message to a CPM Group.

Figure 22 shows the originating side of the flow.



**Figure 27: Pager Mode CPM Standalone Message to a CPM Group (originating).**

1. CPM Client A sends a CPM Standalone Message to a CPM Pre-defined Group identified by a CPM Pre-defined Group address or to a CPM Ad-hoc Group. In case of a CPM Ad-hoc Group, the list of recipients is included in the CPM Standalone Message, and the CPM Standalone Message is destined to the CPM Controlling Function. In case of a CPM Pre-defined Group, the list of recipients is determined by the CPM Pre-defined Group definition and the CPM Standalone Message is destined to the CPM Pre-defined Group address. CPM Client A may request a CPM Standalone Message delivery notification and/or read report. In this flow CPM Client A requests a CPM Standalone Message delivery notification.
2. SIP/IP Core A routes the CPM Standalone Message to CPM Participating Function A triggered on the CPM Feature Tag and CPM Client A's CPM Address.
3. CPM Participating Function A performs originating CPM services and then sends the CPM Standalone Message to SIP/IP Core A.
4. SIP/IP Core A routes the CPM Standalone Message to SIP/IP Core X.
5. SIP/IP Core X routes the CPM Standalone Message to CPM Controlling Function X based on either the CPM Pre-defined Group address or the CPM Controlling Function address and the CPM Feature Tag.
6. CPM Controlling Function X performs the necessary group and service control policies (e.g. authorizes the CPM Client A), and if the CPM Address of the CPM Client A is authorized successfully, CPM Controlling Function X sends the CPM Standalone Message to the members of the CPM Group as described in Figure 23.

- 7-11. An Accepted response is forwarded to CPM Client A.
- 12. CPM Controlling Function X receives the CPM Standalone Message delivery notification which is produced as described in section **Error! Reference source not found.**
- 13. CPM Controlling Function X may incorporate the received CPM Standalone Message delivery notifications into a new, aggregated, CPM Standalone Message delivery notification, and sends the CPM Standalone Message delivery notification to SIP/IP Core X.
- 14. SIP/IP Core X routes the CPM Standalone Message delivery notification to SIP/IP Core A.
- 15. SIP/IP Core A routes the CPM Standalone Message delivery notification to CPM Participating Function A triggered on the CPM Feature Tag and CPM Client A's CPM Address.
- 16. CPM Participating Function A performs terminating CPM services and then sends the CPM Standalone Message delivery notification to SIP/IP Core A.
- 17. The CPM Standalone Message delivery notification is routed to CPM Client A.
- 18-22. An OK response is routed to CPM Participating Function A and then to CPM Controlling Function X along the same Signalling Path.

Figure 23 shows the terminating side of the flow.

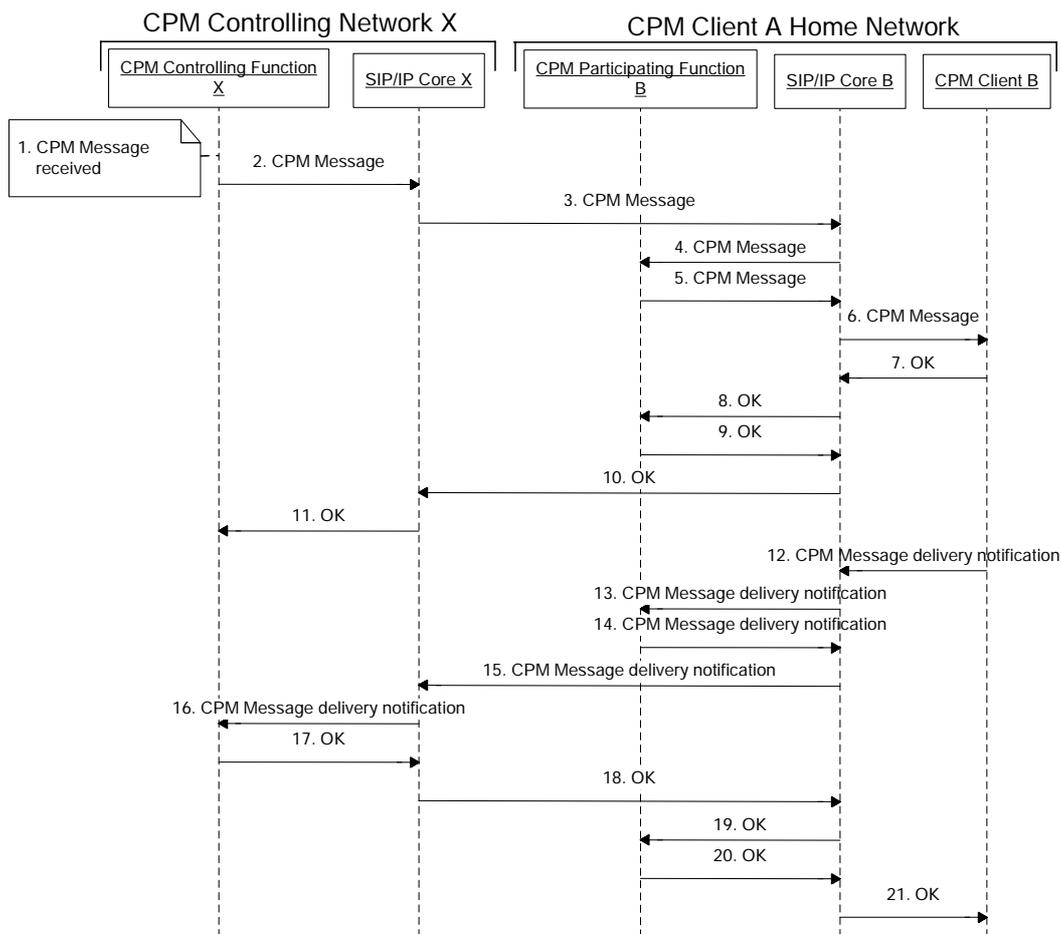
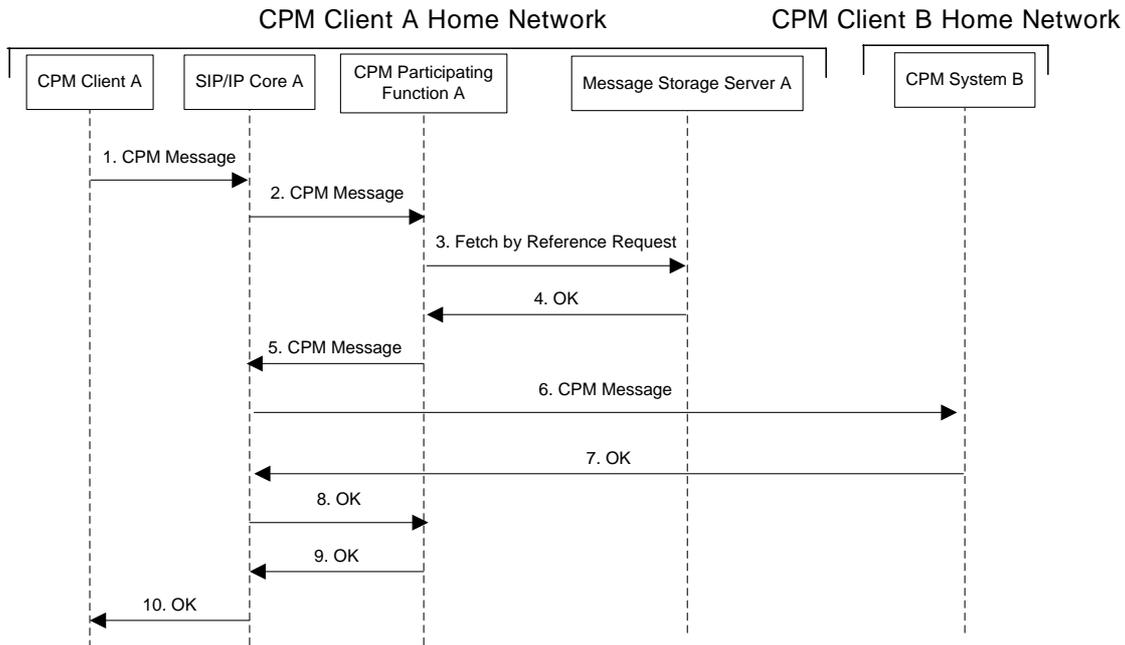


Figure 28: Pager Mode CPM Standalone Message to a CPM Pre-defined Group (terminating).

1. CPM Controlling Function X receives a CPM Standalone Message from CPM Client A as described in section **Error! Reference source not found.**
2. CPM Controlling Function X sends the CPM Standalone Message to SIP/IP Core X.
3. SIP/IP Core X routes the CPM Standalone Message to SIP/IP Core B.
4. SIP/IP Core B routes the request to CPM Participating Function B based on the CPM Feature Tag and CPM Client B's CPM Address.
5. CPM Participating Function B executes the necessary terminating service control and if successful, sends the CPM Standalone Message via SIP/IP Core B towards CPM Client B.
6. SIP/IP Core B routes the CPM Standalone Message to CPM Client B.
- 7-11. An OK response is forwarded to the CPM Controlling Function X.
  - NOTE 1: To send the CPM Standalone Message to all other members of the CPM Group, steps 2-11 are repeated as often as there are other members.
  - NOTE 2: When a recipient receives this CPM Standalone Message on multiple devices, handling is similar to how described in flow C.3.1.1.2
12. CPM Client B sends a CPM Standalone Message delivery notification towards the sender.
13. SIP/IP Core B routes the CPM Standalone Message delivery notification to CPM Participating Function B triggered on the CPM Feature Tag and CPM Client B's CPM Address.
14. CPM Participating Function B performs originating CPM services and then sends the CPM Standalone Message delivery notification to SIP/IP Core B.
15. SIP/IP Core B routes the CPM Standalone Message delivery notification to SIP/IP Core X.
16. SIP/IP Core X routes the CPM Standalone Message delivery notification to CPM Controlling Function X.
- 17-21. An OK response is forwarded to the CPM Client B.
  - NOTE 3: Other members of the CPM Group may also send a CPM Standalone Message delivery notification to the CPM Controlling Function. For these other members the steps executed are the same as steps 12-21.
  - NOTE 4: CPM Controlling Function X may collect the received CPM Standalone Message delivery notifications into a new, aggregated CPM Standalone Message delivery notification.

### **B.3.1.3 Forwarding/Including Data Stored in Message Storage**

This section describes the case when a CPM Client forwards/includes the stored data in Message Storage Server without downloading them to the local device.

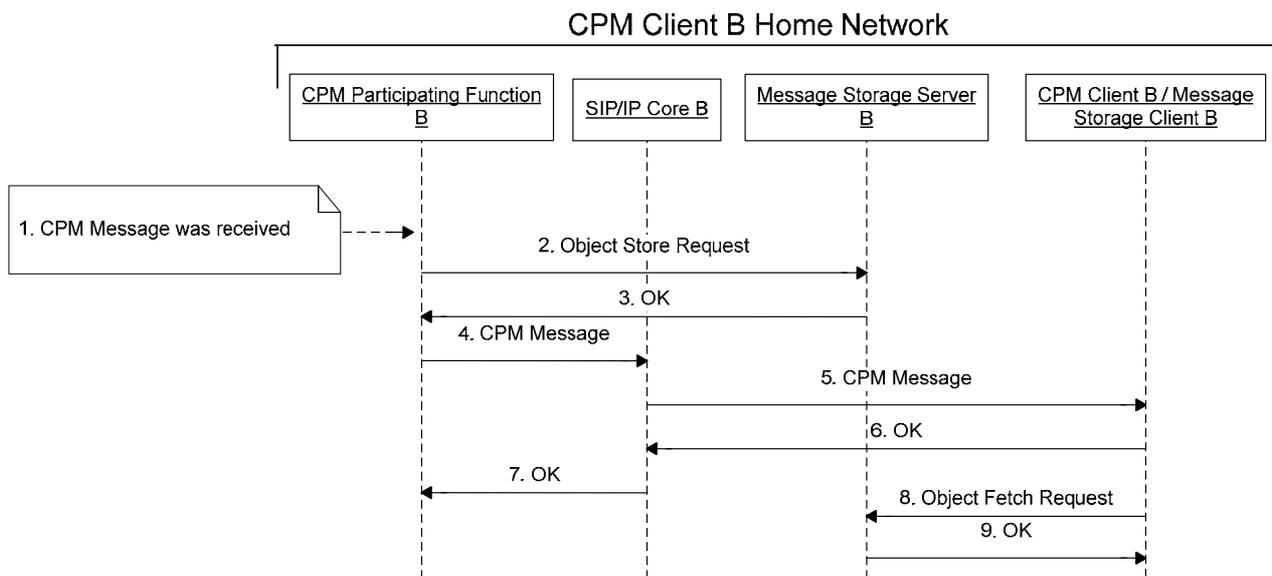


**Figure 29: Forwarding/including data stored in Message Storage Server**

1. CPM Client A sends a CPM Standalone Message towards a recipient. The CPM Standalone Message contains references to the stored data in Message Storage Server. CPM Client A can obtain these references by interacting with the Message Storage Client.
2. SIP/IP Core A routes the CPM Standalone Message to CPM Participating Function A.
3. CPM Participating Function A sends a fetch by reference request to Message Storage Server A to fetch the stored data as indicated by the reference.
4. Message Storage Server A returns the referenced data to CPM Participating Function A with an OK response. If the referenced data or part of it doesn't exist, the Message Storage Server may send a failure response to the CPM Participating Function A.
5. CPM Participating Function A replaces the reference with the fetched data, performs the originating CPM services and then sends the CPM Standalone Message to SIP/IP Core A.
6. SIP/IP Core A routes the CPM Standalone Message to the CPM System B.
7. CPM System B returns an OK response after delivering the CPM Standalone Message.
- 8-10. The OK response is routed to CPM Client A along the same Signalling Path.

### B.3.1.4 Indirect Media Delivery

This section describes the case when Media in a CPM Standalone Message is delivered indirectly.



**Figure 30: Indirect media delivery.**

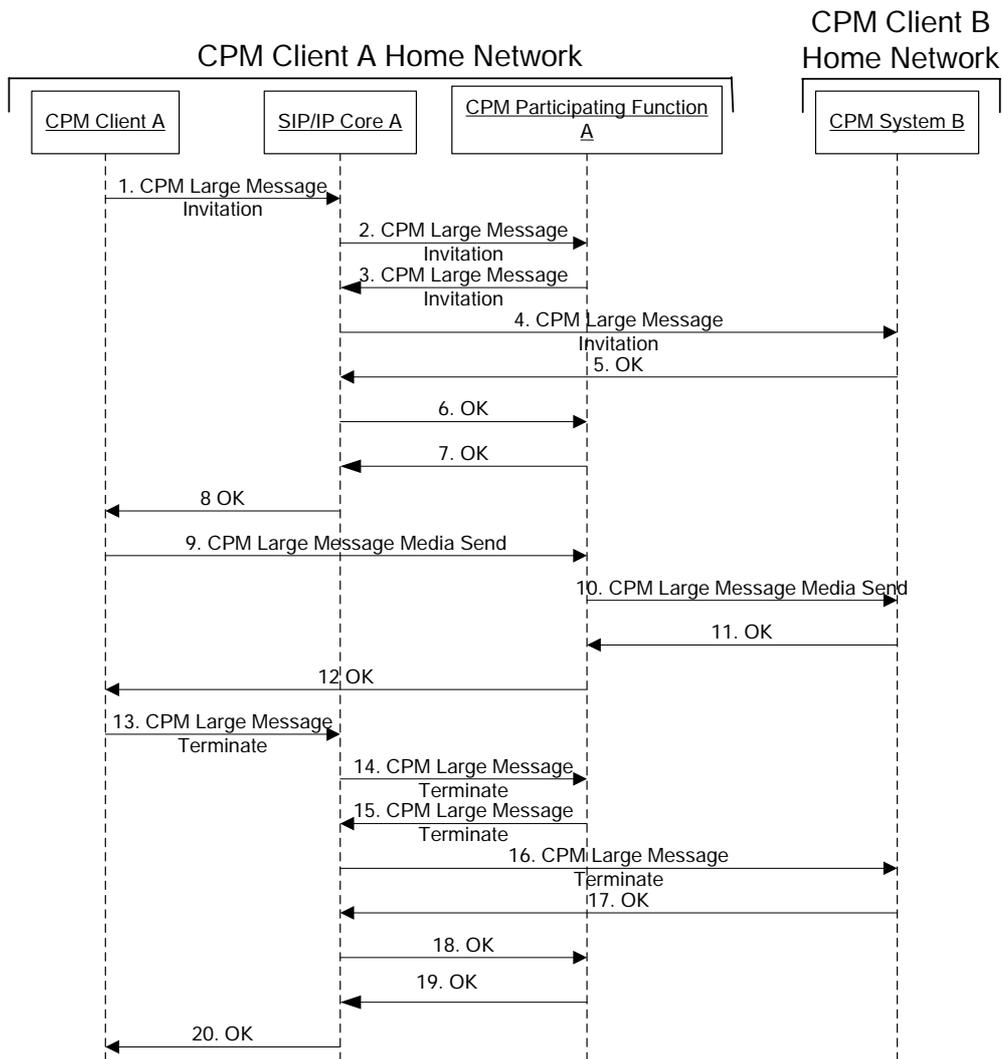
1. A CPM Standalone Message with one or more Media Objects is received by CPM Participating Function B.
2. CPM Participating Function B determines to send selected Media Objects indirectly according to user preferences and service provider policies and sends an object store request to store the CPM Standalone Message including all the Media Objects in Message Storage Server B.
3. Message Storage Server B returns OK response with the location information where the CPM Standalone Message including the Media Objects is stored.
4. CPM Participating Function B replaces the Media Objects selected in step 2 in the CPM Standalone Message with references indicating the respective locations of those Media Objects in Message Storage Server B, performs the terminating CPM services and then sends the CPM Standalone Message to SIP/IP Core B.
5. SIP/IP Core B routes the CPM Standalone Message to CPM Client B.
6. CPM Client B returns an OK response to SIP/IP Core B.
7. The OK response is routed back to the originating party along the same Signalling Path as the incoming CPM Standalone Message.
8. If requested by the CPM User or CPM Client, Message Storage Client B sends an object fetch request to Message Storage Server B.
9. Message Storage Server B returns OK response with the Media to Message Storage Client B.

## B.3.2 Large Message Mode CPM Standalone Message Flows

### B.3.2.1 Sending a Large Message Mode CPM Standalone Message

This section describes the case where a CPM User sends a Large Message Mode CPM Standalone Message to another CPM System.

Figure 26 shows the originating side of the flow.



**Figure 31: Sending a CPM Standalone Message in Large Message Mode to a CPM User (originating side)**

1. CPM Client A sends a CPM Large Message Invitation request towards a recipient.
2. SIP/IP Core A routes the CPM Large Message Invitation request to CPM Participating Function A triggered on the CPM Feature Tag and CPM Client A’s CPM Address.
3. CPM Participating Function A sends the CPM Large Message Invitation request towards CPM System B.
4. SIP/IP Core A routes the CPM Large Message Invitation request to CPM System B. If the CPM System B decides that the CPM Large Message is to be deferred, interworked or to be delivered to the Message Storage Server, then CPM System B accepts the message on behalf of CPM User B.
5. CPM System B sends an OK for the CPM Large Message Invitation request towards SIP/IP Core A.
6. SIP/IP Core A sends the OK response to CPM Participating Function A.
7. CPM Participating Function A sends the OK response for the CPM Large Message Invitation request towards CPM Client A.
8. SIP/IP Core A forwards the OK to CPM Client A.

9. CPM Client A sends the media to CPM Participating Function A.

NOTE: Content screening can be done after step 9. If this results in a decision not to send the message CPM Participating Function A will not send the MSRP data to CPM Client B and will send a BYE request to end the session.

10 CPM Participating Function A sends the media to CPM System B.

11. CPM System B sends an OK response for the received media.

12. CPM Participating Function A sends the OK response for the received media.

13. CPM Client A closes the message transmission by sending a CPM Large Message Termination request.

14. SIP/IP Core A routes the CPM Large Message Termination request to CPM Participating Function A.

15. CPM Participating Function A closes the message transmission by sending a CPM Large Message Termination request.

16. SIP/IP Core A routes the CPM Large Message Termination request to CPM System B.

17-18. CPM System B sends the OK response, which is forwarded by the SIP/IP Core to CPM Participating Function A.

19-20. CPM Participating Function A sends the OK response, which is forwarded by the SIP/IP Core A to CPM Client A.

Figure 27 shows the terminating side of the flow. Multiple devices are able to receive the Large Message Mode CPM Standalone Message.

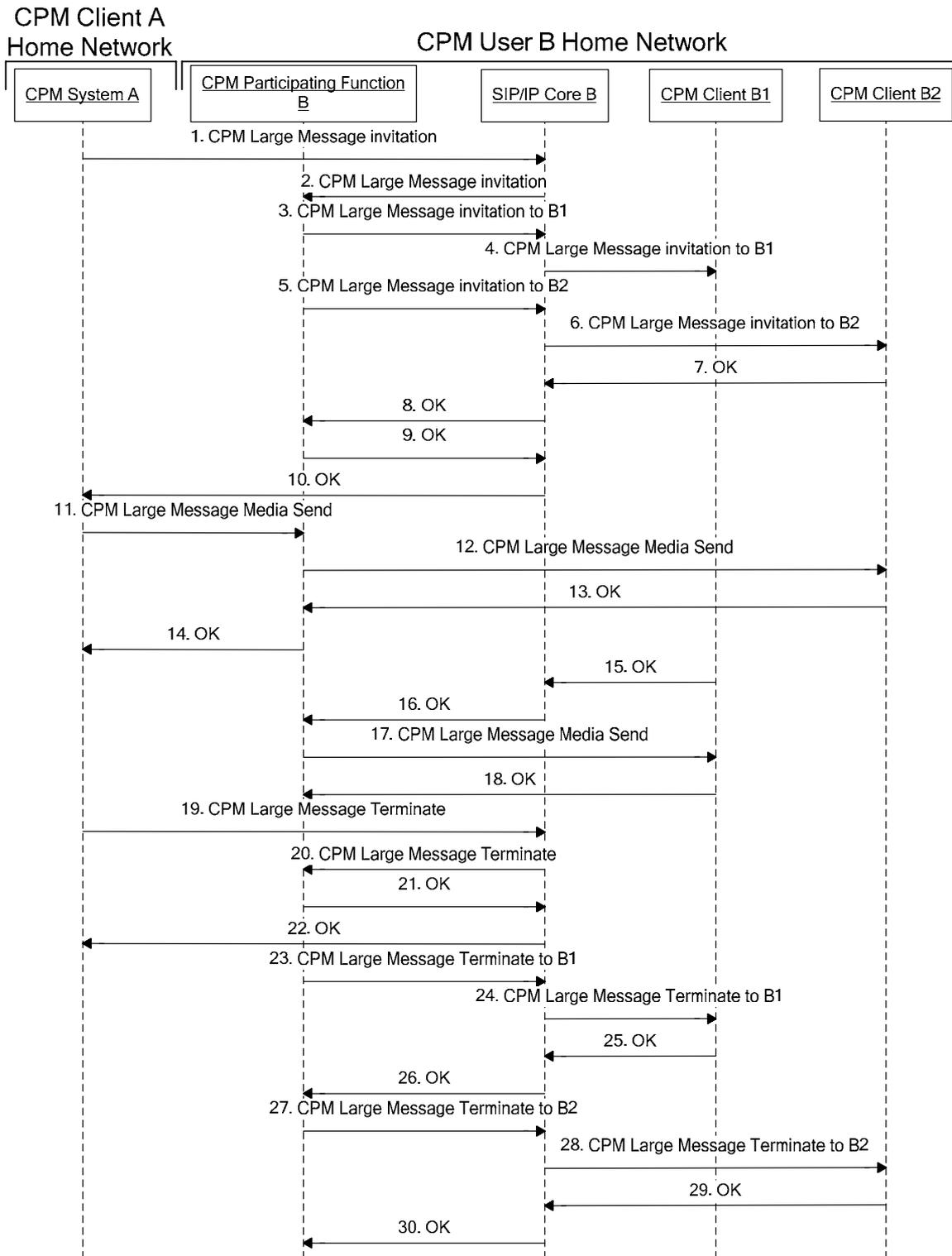


Figure 32: Sending a CPM Standalone Message in Large Message Mode to a CPM User (terminating side)

1. CPM System A sends a CPM Large Message Invitation request towards a recipient.

2. SIP/IP Core B routes the CPM Large Message Invitation request to CPM Participating Function B triggered on the CPM Feature Tag and CPM User B's CPM Address. If the CPM Participating Function B decides that the CPM Large Message is to be deferred, interworked or to be delivered to the Message Storage Server, then CPM System B accepts the message on behalf of CPM User B.

3-4. CPM Participating Function B determines that more than one device is available to receive the Large Message Mode CPM Standalone Message. It sends the CPM Large Message Invitation request towards CPM Client B1.

NOTE 1: There might be a CPM Client B3 for which the CPM Participating Function B decides not to send the CPM Large Message Invitation request.

5-6. CPM Participating Function B sends the CPM Large Message Invitation request towards CPM Client B2.

NOTE 2: CPM Participating Function B must send a CPM Large Message invitation to each client to avoid the SIP/IP core accepting only one CPM Large Message invitation from all clients.

7-8 CPM Client B2 sends an OK response for the CPM Large Message Invitation request towards CPM Participating Function B.

9-10 Since an OK response has now been received for this request; CPM Participating Function B sends the OK response for the CPM Large Message Invitation request towards CPM System A.

11. CPM System A sends the media to CPM Participating Function B.

NOTE 3: Content screening can be done after step 11. If this results in a decision not to send the message CPM Participating Function B will not send the MSRP data to CPM Client B and will send a BYE request to end the session. Other terminating services are also executed at this time.

12. CPM Participating Function B sends the media to CPM Client B2.

13. CPM Client B2 sends an OK response for the received media.

14. CPM Participating Function B sends an OK response for the received media to CPM Participating Function A.

15-16. CPM Client B1 accepts the Large Message Invitation and sends an OK response for the received invitation.

17. CPM Participating Function B sends the media to CPM Client B1.

18. CPM Client B1 sends an OK response for the received media.

19. CPM System A closes the message transmission by sending a CPM Large Message Termination request.

NOTE 4: Step 19 can occur any time after step 13.

20. SIP/IP core B routes the CPM Large Message Termination request to CPM Participating Function B.

21-22. CPM Participating Function B sends an OK response for session termination towards CPM Participating Function A.

23-24. CPM Participating Function B closes the message transmission by sending a CPM Large Message Termination request to CPM Client B1.

25-26 CPM Client B1 responds with an OK response, which is forwarded by SIP/IP Core B to CPM Participating Function B.

27-28. CPM Participating Function B closes the message transmission by sending a CPM Large Message Termination request to CPM Client B2.

29-30. CPM Client B2 responds with an OK response, which is forwarded by SIP/IP Core B to CPM Participating Function B.

### B.3.2.2 Extending to a Large Message Mode CPM Standalone Message after Fetching Media

This section describes the case where a CPM Participating Function has to extend a Pager Mode CPM Standalone Message to a Large Message Mode CPM Standalone Message after fetching Media Objects, CPM Messages, CPM File Transfer Histories, CPM Session Histories or CPM Conversation History stored in the Message Storage Server. The Pager Mode CPM Standalone Message is extended to a Large Message Mode CPM Standalone Message when the resulting CPM Standalone Message exceeds the Pager Mode CPM Standalone Message size limitations.

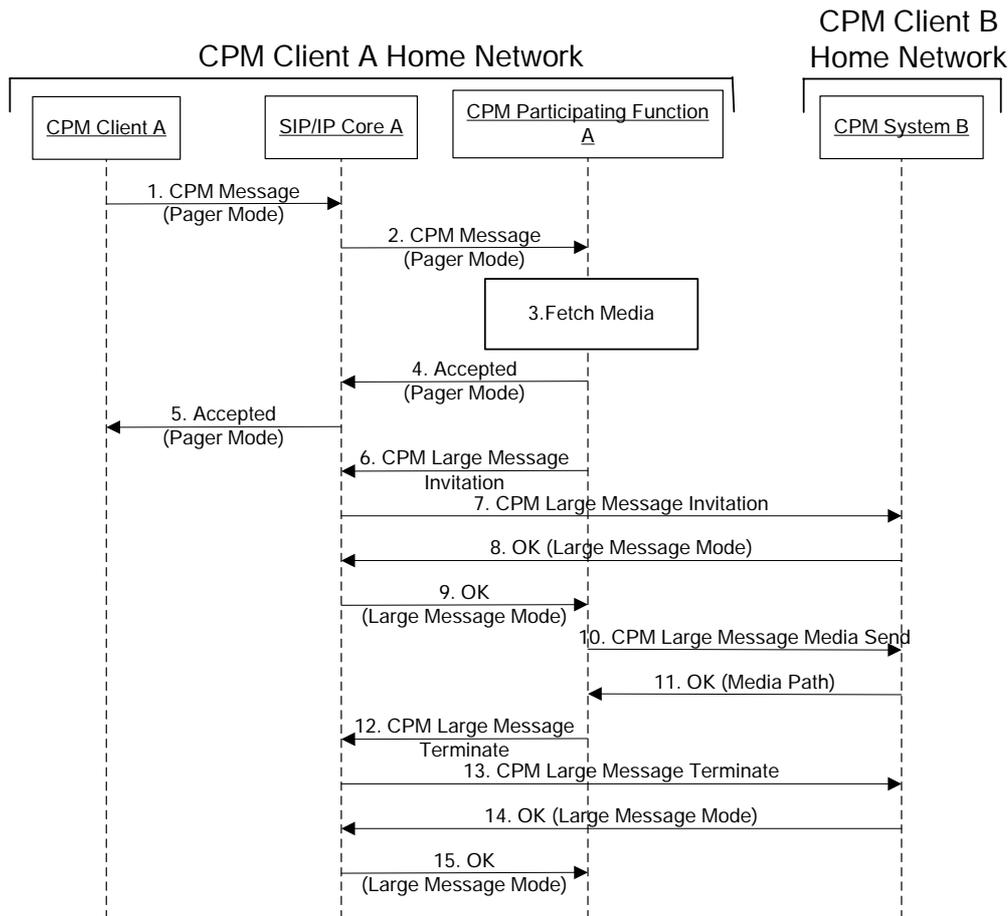


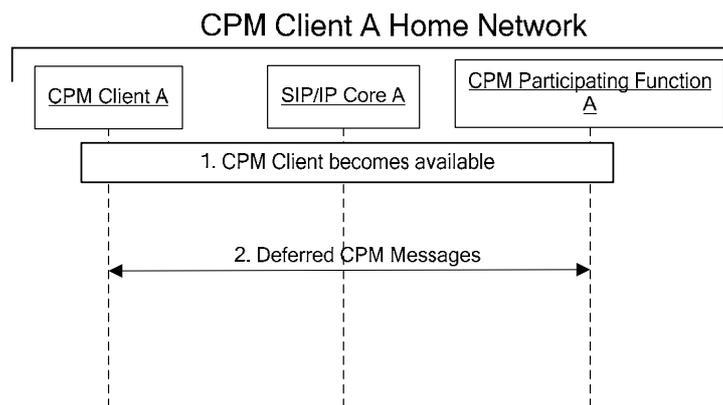
Figure 33: Extending to a Large Message Mode CPM Standalone Message in CPM Participating Function

1. CPM Client A sends a Pager Mode CPM Standalone Message towards a recipient.
2. SIP/IP Core A routes the Pager Mode CPM Standalone Message to CPM Participating Function A triggered on the CPM Feature Tag and CPM Client A's CPM Address.
3. CPM Participating Function A fetches the Media Objects, CPM Messages, CPM File Transfer Histories, CPM Session Histories or CPM Conversation History from the Message Storage Server as described in sections C.4. After that CPM Participating Function A checks the new size of the CPM Standalone Message and detects that the maximum Pager Mode CPM Standalone Message size has been exceeded.
4. CPM Participating Function A sends an Accepted response for the Pager Mode CPM Standalone Message towards CPM Client A.
5. SIP/IP Core A forwards the Accepted response to CPM Client A.

6. CPM Participating Function A sends a CPM Large Message Invitation request towards CPM System B.
7. SIP/IP Core A routes the CPM Large Message Invitation request to CPM System B.
8. CPM System B sends an OK response for the CPM Large Message Invitation request towards CPM Participating Function A
9. SIP/IP Core A forwards the OK response to CPM Participating Function A.
10. CPM Participating Function A sends the Media to CPM System B.
11. CPM System B sends an OK response for the received Media.
12. CPM Participating Function A closes the message transmission by sending a CPM Large Message Termination request.
13. SIP/IP Core A routes the CPM Large Message Termination request to CPM System B.
- 14-15. CPM System B sends an OK response, which is forwarded by SIP/IP Core A to CPM Participating Function A.

### B.3.3 Deferred CPM Message Related Flows

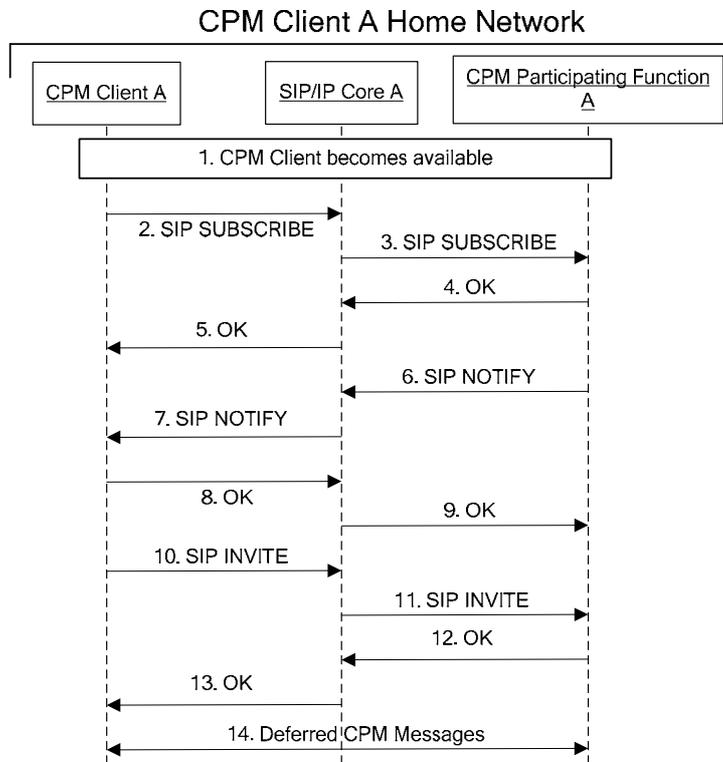
#### B.3.3.1 Deferred CPM Message Pushing



**Figure 34: Deferred CPM Message Pushing**

1. There are Deferred CPM Messages in the CPM Participating Function A awaiting delivery and the receiving CPM User has not set (anymore) a preference to not disturb. CPM Client A becomes available.
2. CPM Participating Function A delivers all those messages for which CPM Client A matches the delivery criteria.

### B.3.3.2 Deferred CPM Message Pulling



**Figure 35: Deferred CPM Message pulling**

1. A CPM Client becomes available.

NOTE: A preceding out-of-band notification may have caused the client to become available, i.e., to register.

2-3. CPM Client A subscribes to the Deferred CPM Message information (via SIP/IP Core A).

4-5. CPM Participating Function A sends an OK response (via SIP/IP Core A).

6-7. CPM Participating Function A sends a notification with the Deferred CPM Message information (via SIP/IP Core A).

8-9. CPM Client A acknowledges reception of the notification (via SIP/IP Core A).

10. CPM Client A sends an invitation to SIP/IP Core A. CPM Client A includes the actions to be performed for each of the selected messages from the list received in the notification.

11. SIP/IP Core A routes the invitation to Participating Function A.

12. CPM Participating Function A sends an OK response to SIP/IP Core A.

13. SIP/IP Core A routes the OK response to CPM Client A.

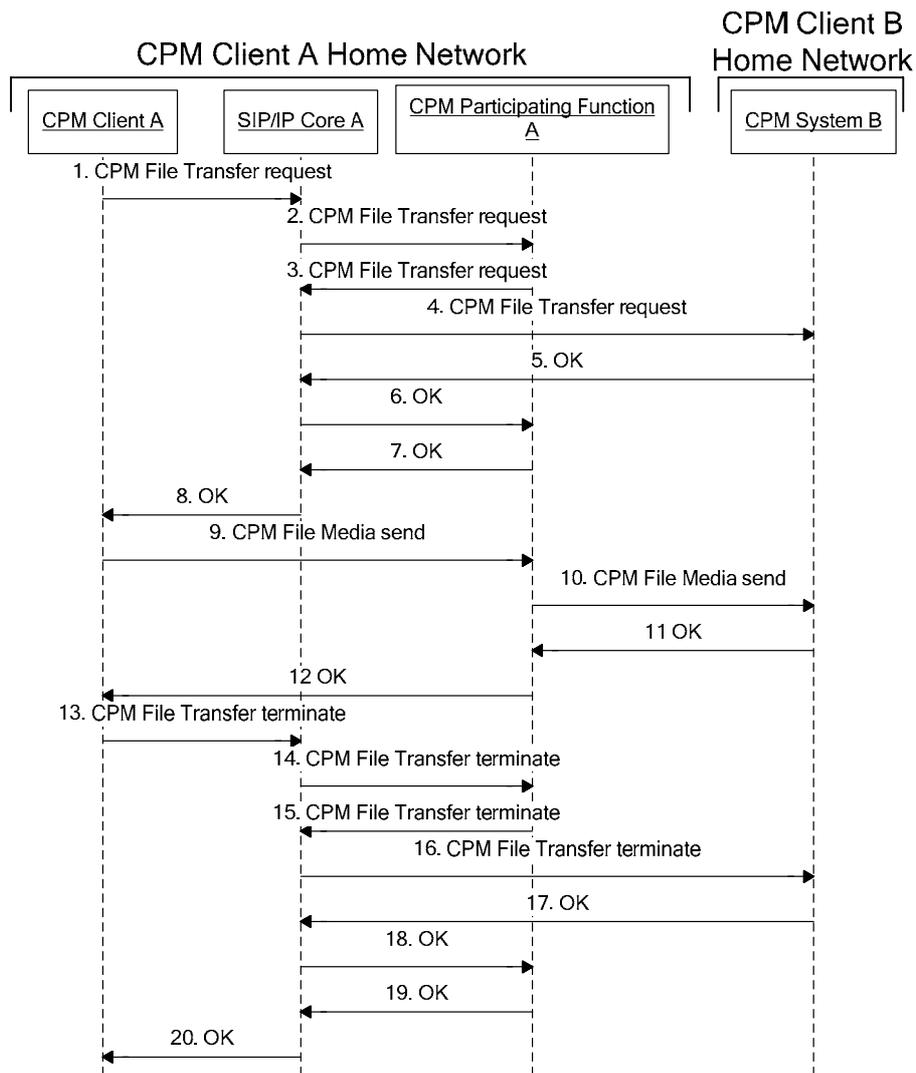
14. If some messages are to be retrieved, CPM Participating Function A delivers those messages to CPM Client A.

## B.4 CPM File Transfer Flows

### B.4.1 Transferring a File

This section describes the case where a CPM User sends one or more files to another CPM User. Each new CPM File Transfer request starts a new SIP session.

Figure 31 shows the originating side of the flow.



**Figure 36: Sending a CPM File Transfer to a CPM User (originating side)**

1. CPM Client A sends a CPM File Transfer request towards a recipient.
2. SIP/IP Core A routes the CPM File Transfer request to CPM Participating Function A triggered on the CPM Feature Tag and CPM Client A’s CPM Address.
3. CPM Participating Function A sends the CPM File Transfer request towards CPM System B.

4. SIP/IP Core A routes the CPM File Transfer request to CPM System B. CPM System B presents the CPM File Transfer request to CPM User B and collects input from CPM User B who may accept one or more files or reject the CPM File Transfer. In this case, CPM User B accepts one or more files and sends an OK to CPM System B.

5. CPM System B sends an OK for the CPM File Transfer request towards SIP/IP Core A.

6. SIP/IP Core A sends the OK response to CPM Participating Function A.

7. CPM Participating Function A sends the OK response for the CPM File Transfer request towards CPM Client A.

8. SIP/IP Core A forwards the OK to CPM Client A.

9. CPM Client A sends the file media to CPM Participating Function A.

10. CPM Participating Function A sends the file media to CPM System B.

11. CPM System B sends an OK response for the received media.

12. CPM Participating Function A sends the OK response for the received media.

13. CPM Client A closes the transmission by sending a CPM File Transfer terminate request.

14. SIP/IP Core A routes the CPM File Transfer terminate request to CPM Participating Function A.

15. CPM Participating Function A closes the transmission by sending a CPM File Transfer terminate request.

16. SIP/IP Core A routes the CPM File Transfer terminate request to CPM System B.

17-18. CPM System B sends an OK response, which is forwarded by the SIP/IP Core to CPM Participating Function A.

19-20. CPM Participating Function A sends an OK response, which is forwarded by the SIP/IP Core A to CPM Client A.

Figure 31 shows the originating side of the flow. Multiple devices are able to receive the CPM File Transfer request but only one device can receive the actual file.

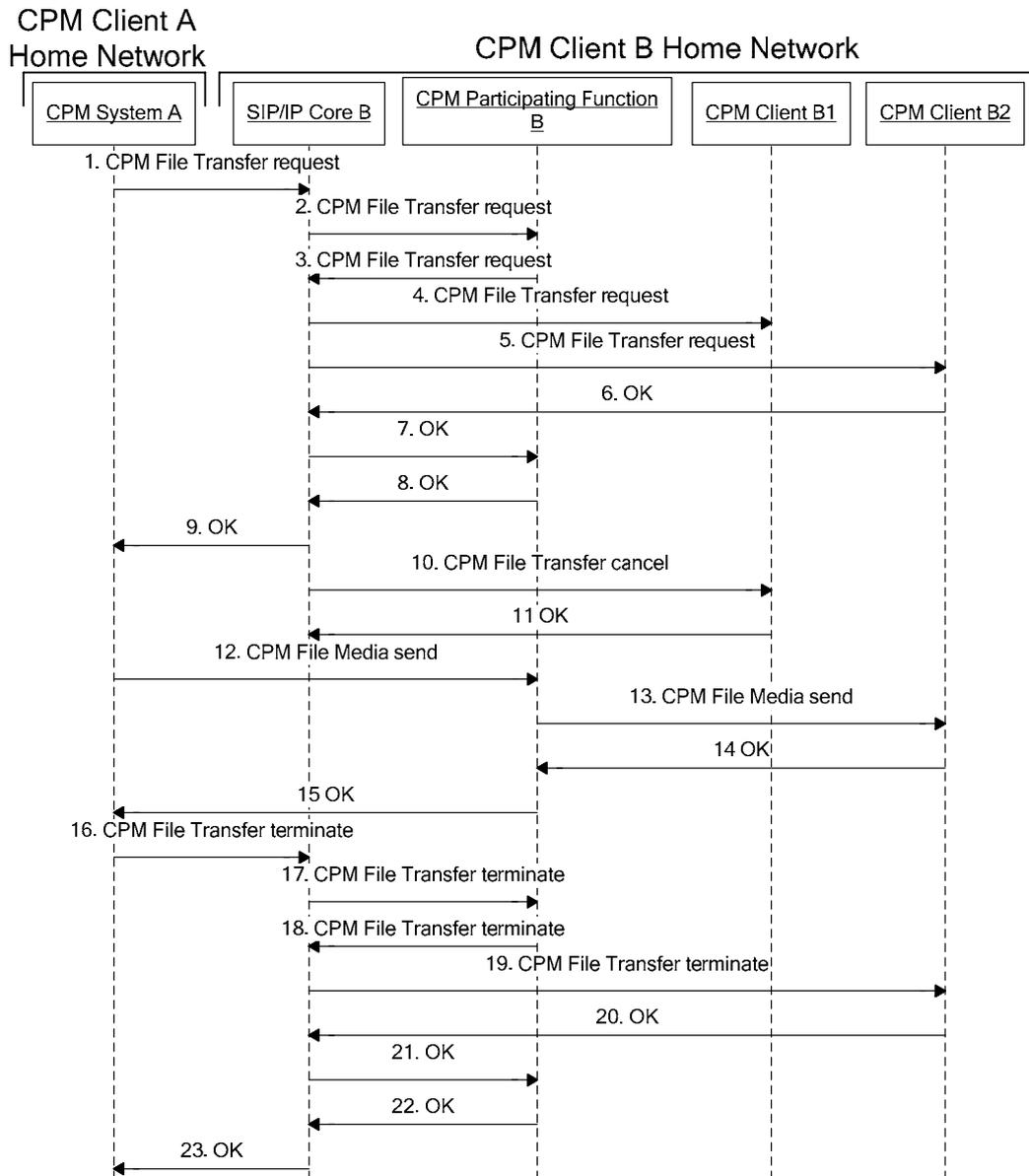


Figure 37: Sending a File Transfer to a CPM User (terminating side)

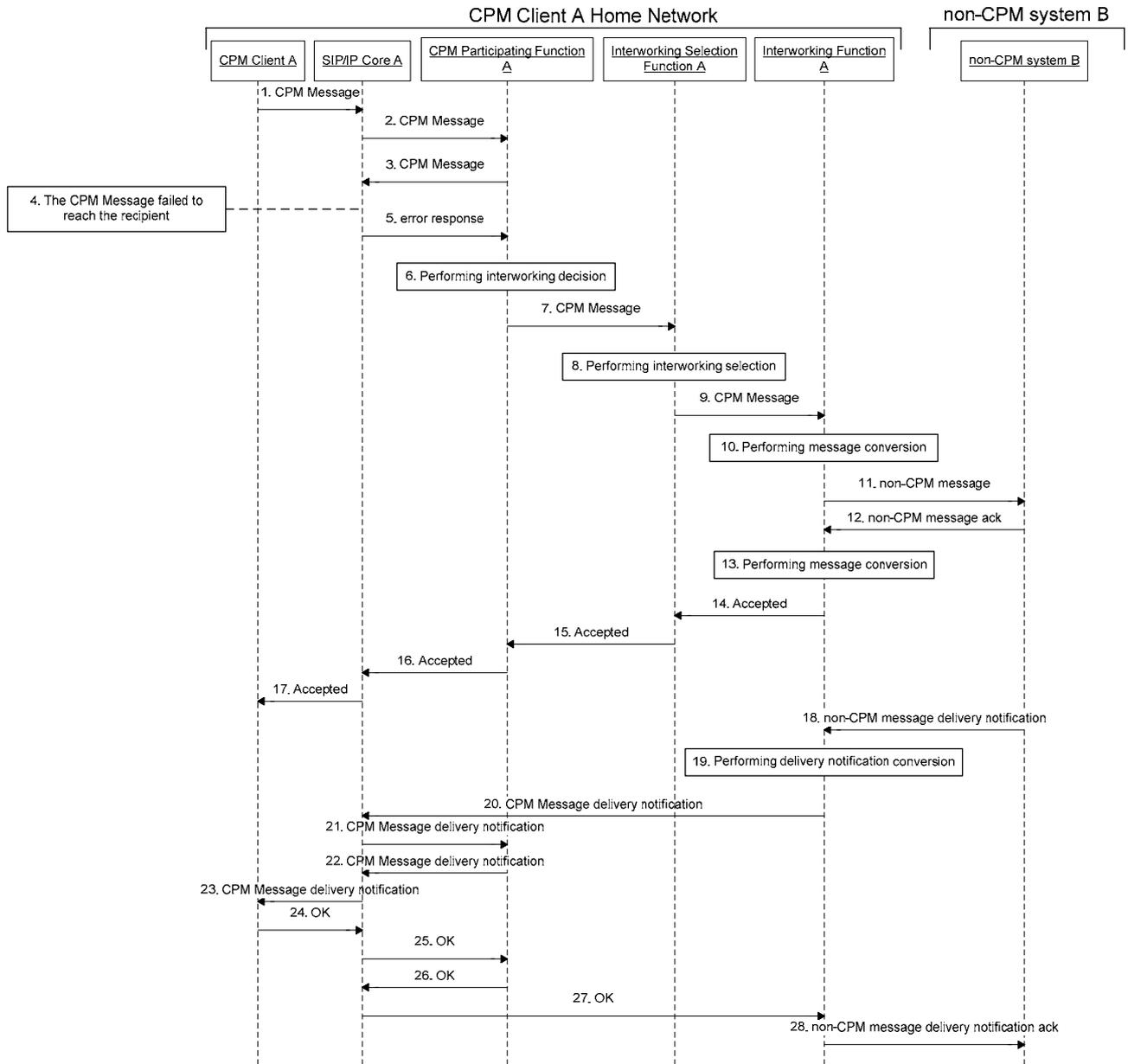
1. CPM System A sends a CPM File Transfer request towards a recipient.
2. SIP/IP Core B routes the CPM File Transfer request to CPM Participating Function B triggered on the CPM Feature Tag and CPM User B’s CPM Address.
3. CPM Participating Function B sends the CPM File Transfer request towards SIP/IP Core B. CPM Participating Function B executes the necessary terminating service control and determines to which CPM Clients of CPM User B to deliver/not to deliver the CPM File Transfer request as described in section **Error! Reference source not found.** In this flow, CPM Participating Function B determines to route the CPM Session Invitation to both CPM Clients B1 and B2 and sends the CPM File Transfer request to SIP/IP Core B.
- 4-5. SIP/IP Core B sends the CPM File Transfer request to CPM Client B1 and CPM Client B2.

6. CPM Client B2 sends an OK response for the CPM File Transfer request towards CPM Participating Function B indicating which file(s) are acceptable as indicated by CPM User B.
- 7-9. The OK response is sent towards CPM System A.
10. Since CPM Client B2 has already accepted the CPM File Transfer, SIP/IP Core B cancels the CPM File Transfer to CPM Client B1 using the CPM File Transfer cancel request.
11. CPM Client B1 confirms reception of the CPM File Transfer cancel request.
12. CPM System A sends the media to CPM Participating Function B.
13. CPM Participating Function B sends the media to CPM Client B2.
- 14-15. CPM Client B2 acknowledges reception of the media with an OK response.
16. CPM System A closes the transmission by sending a CPM File Transfer terminate request.
17. SIP/IP Core B routes the CPM File Transfer terminate request to CPM Participating Function B.
- 18-19. CPM Participating Function B sends the CPM File Transfer terminate request to CPM Client B2.
- 20-21. CPM Client B2 responds with an OK response which is forwarded by SIP/IP Core B to CPM Participating Function B
- 22-23. CPM Participating Function B sends an OK response for CPM File Transfer termination towards CPM Participating Function A.

## **B.5 Outbound Interworking**

### **B.5.1 Pager Mode CPM Standalone Message Interworking in the Originating Network**

This section describes the case where a sending CPM User sends a Pager Mode CPM Standalone Message to a recipient, and it is decided that to reach the recipient, interworking is required.



**Figure 38: Interworking at the originating network**

1. CPM Client A sends a CPM Standalone Message towards a recipient. CPM Client A may include a disposition notification request for a delivery notification and/or read report in the CPM Standalone Message. In this flow CPM Client A requests a delivery notification.
2. SIP/IP Core A routes the CPM Standalone Message to CPM Participating Function A triggered on the CPM Feature Tag and CPM Client A’s CPM Address.
3. CPM Participating Function A performs originating CPM services and send the CPM Standalone Message to SIP/IP Core A.
4. The CPM Standalone Message fails to reach the recipient for one of the reasons described in section 5.3.1.1.2, section 5.3.1.1.5 and section 5.3.1.1.6.

5. The CPM Participating Function A receives an error response to the CPM Standalone Message.

NOTE: In case the received address is neither a SIP URI nor a TEL URI, steps 3 to 5 will not be executed.

6. CPM Participating Function A decides whether interworking is required.

7. CPM Participating Function A sends the request to Interworking Selection Function A.

8. Interworking Selection Function A selects the most appropriate Interworking Function for handling the CPM Standalone Message as defined in section 5.3.1.2.

9. Interworking Selection Function A routes the CPM Standalone Message to the selected Interworking Function A.

10. Interworking Function A converts the CPM Standalone Message into a non-CPM message routable in the Non-CPM System B.

11. Non-CPM system B receives the message.

12. Non-CPM system B acknowledges reception.

NOTE 1: Step 13 could be skipped and steps 14-17 could be initiated before step 12 (e.g. per service provider policies, to prevent the risk of timeout).

13. Interworking Function A converts the acknowledgement of the non-CPM message into an Accepted response to the CPM Message.

14. Interworking Function A sends the Accepted response to Interworking Selection Function A.

15. Interworking Selection Function A forwards the Accepted response to CPM Participating Function A.

16-17. The Accepted response is routed to CPM Client A along the Signalling Path that the CPM Standalone Message experienced before

NOTE 2: The procedure can end in step 17. Steps 18 to 28 occur only if the CPM User requested a delivery notification in the CPM Standalone Message sent in step 1.

18. Non-CPM system B generates a delivery notification towards the sender.

19. Interworking Function A converts the non-CPM message delivery notification into a CPM Standalone Message delivery notification.

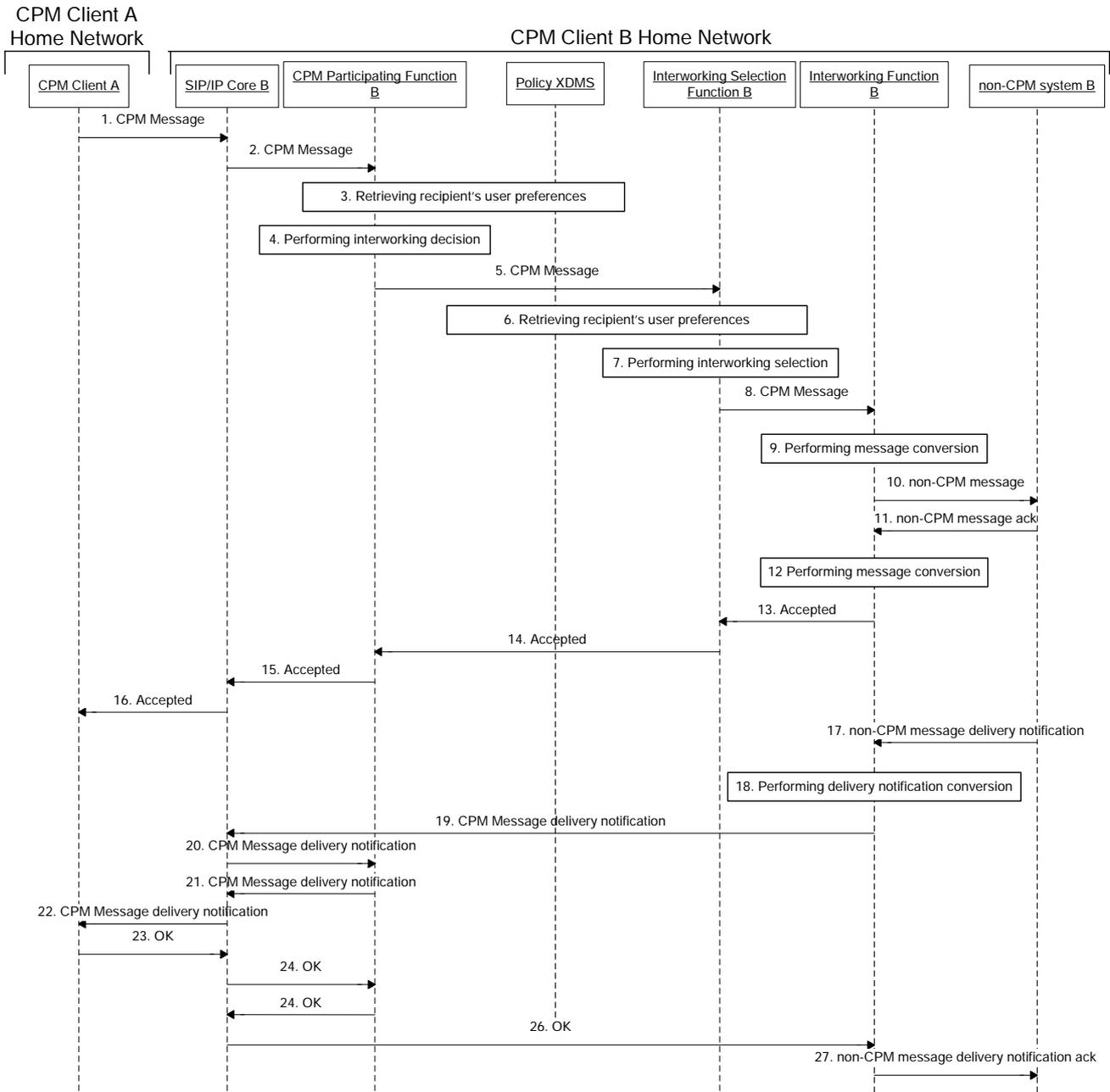
20-23. The Interworking Function A routes the CPM Standalone Message delivery notification towards CPM Client A.

24-27. CPM Client A sends an OK response towards non-CPM system B along the same Signalling Path.

28. Interworking Function A routes the OK response towards the sender, if required by the particular non-CPM system.

## **B.5.2 Pager Mode CPM Standalone Message Interworking in the Terminating Network**

This section describes an interworking message flow in a case that a recipient has subscription to CPM service and other Non-CPM Communication services but he is not available or not willing to receive a CPM Standalone Message.



**Figure 39: Interworking based on recipient’s user preferences at the terminating network**

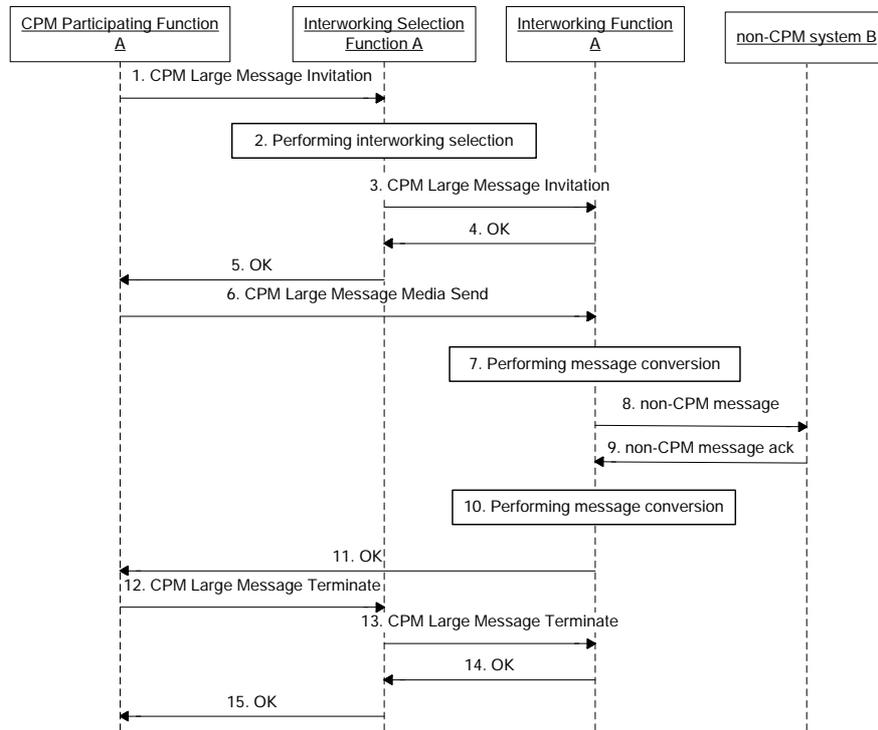
1. In order for a CPM Standalone Message to reach to CPM Client B, CPM System A passes the CPM Standalone Message to SIP/IP Core B.
2. SIP/IP Core B routes the CPM Standalone Message to CPM Participating Function B serving CPM Client B.
3. CPM Participating Function B retrieves the recipient’s user preferences from Policy XDMS.
4. CPM Participating Function B makes a decision about whether interworking is needed or not.
5. If the decision is taken to interwork, CPM Participating Function B routes the CPM Standalone Message to Interworking Selection Function B.

6. Interworking Selection Function B retrieves the recipient's user preferences for interworking selection from Policy XDMS.
7. Interworking Selection Function B selects the most appropriate Interworking Function for handling the CPM Standalone Message as defined in section 5.3.1.2.
8. Interworking Selection Function B routes the CPM Standalone Message to the selected Interworking Function B.
9. Interworking Function B converts the CPM Standalone Message into a non-CPM message routable in non-CPM System B.
10. Interworking Function B routes the non-CPM message to non-CPM system B.
11. The acknowledgement of the non-CPM message is returned from non-CPM system B to Interworking Function B.  
NOTE 1: Step 12 could be skipped and steps 13-16 could be initiated before step 11, (e.g. per service provider policies, to prevent the risk of timeout).
12. Interworking Function B converts the acknowledgement of the non-CPM message into an OK response to the CPM Message.
13. Interworking Function B sends the Accepted response to Interworking Selection Function A.
14. Interworking Selection Function A forwards the Accepted response to CPM Participating Function A.
- 15-16. The OK response is routed to the CPM Client A Home Network along the Signalling Path that the CPM Standalone Message experienced before.
17. If supported by the non-CPM system B, it generates a delivery notification towards the sender.  
NOTE 2: Client B may opt to disallow a delivery notification, even though asked to by Client A.
18. Interworking Function B converts the non-CPM message delivery notification into a CPM Standalone Message delivery notification.
- 19-22. Interworking Function B routes the CPM Standalone Message delivery notification towards CPM System A, via CPM Participating Function B and SIP/IP Core B.
- 23-26. CPM System A sends an OK response towards non-CPM system B along the same Signalling Path, via SIP/IP Core B and CPM Participating Function B.
27. Interworking Function B routes the ack towards non-CPM System B.  
NOTE 3: Non-CPM System B may or may not require an ack.  
NOTE 4: CPM Client A served by the CPM System A might have requested a read report. If so, and if supported by the non-CPM system, the recipient client in non-CPM System B might provide one. The read report will be handled in a similar way to the delivery notification (steps 21 to 31).

### B.5.3 Large Message Mode CPM Standalone Message Interworking

This section describes the case where a Large Message Mode CPM Standalone Message needs to be delivered to a recipient, and it is decided that to reach the recipient, interworking is required.

NOTE 1: The flow only focuses on the interactions between the CPM Participating Function, the Interworking Selection Function and the Interworking Function. Upstream interactions are purposely not described.



**Figure 40: Large Message Mode CPM Standalone Message Interworking**

1. The CPM Standalone Message is to be sent as a Large Message Mode CPM Standalone Message (the size of the message is larger than 1300 bytes), so CPM Participating Function A sends a CPM Large Message Invitation request to Interworking Selection Function A.
2. Interworking Selection Function A extracts the message details (size, content types, etc) from the CPM Large Message Invitation request and selects the most appropriate Interworking Function for handling the CPM Standalone Message.
3. Interworking Selection Function A sends the CPM Large Message Invitation request to Interworking Function A.
4. Interworking Function A returns an OK response.
5. The OK response is forwarded to CPM Participating Function A.
6. CPM Participating Function A sends the Large Message Mode CPM Standalone Message (as the media of the established session) directly toward Interworking Function A.
7. Interworking Function A converts the received Large Message Mode CPM Standalone Message into a non-CPM message routable in non-CPM System B.
8. Non-CPM system B receives the message.
9. Non-CPM system B acknowledges reception of the message.
 

NOTE 2: This acknowledgement means that Non-CPM system B has accepted the message for delivery, e.g. at a non-CPM system service centre. The acknowledgement does not necessarily mean that the message has been delivered to non-CPM Principal B.

NOTE 3: Step 10 could be skipped and step 11 could be initiated before step 9, (e.g. per service provider policies, to prevent the risk of timeout).

10. Interworking Function A converts the acknowledgement of the non-CPM message into a RESPONSE for the received media.

11. Interworking Function A sends this RESPONSE to CPM Participating Function A.

12-15. CPM Participating Function A closes the message transmission.

## B.5.4 CPM File Transfer Interworking

### B.5.4.1 File Transfer supported by Non-CPM Communication Service

This section describes the case where a CPM File Transfer needs to be set up to a recipient, and it is decided that to reach the recipient, interworking is required. In this case the Non-CPM Communication Service supports file transfers as well.

NOTE 1: The flow only focuses on the interactions between the CPM Participating Function, the Interworking Selection Function and the Interworking Function. Upstream interactions are purposely not described.

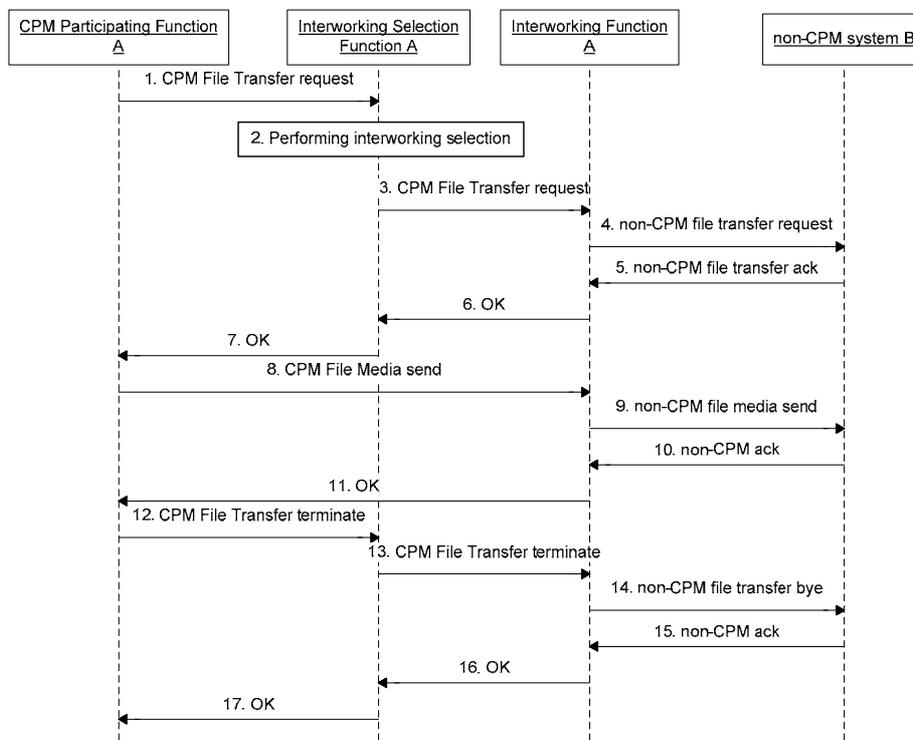


Figure 41: CPM File Transfer Interworking to non-CPM file transfer

1. CPM Participating Function A sends the CPM File Transfer request to Interworking Selection Function A.
2. Interworking Selection Function A extracts the file transfer details from the CPM File Transfer request and selects the most appropriate Interworking Function for handling the CPM File Transfer.
3. Interworking Selection Function A sends the CPM File Transfer request to Interworking Function A.
4. Interworking Function A sends a non-CPM file transfer request to non-CPM system B.
5. Non-CPM system B accepts the file transfer by sending a non-CPM file transfer ack.

NOTE 2: Steps 6-7 could be initiated before step 5 (e.g. per service provider policies).

6. Interworking Function A returns an OK response to Interworking Selection Function A.
7. Interworking Selection Function A forwards the OK response to CPM Participating Function A.
- 8-11. The file media is exchanged with non-CPM System B.
- 12-17. The CPM File Transfer is terminated both on the CPM side and the non-CPM side.

### B.5.4.2 File Transfer not supported by Non-CPM Communication Service

This section describes the case where a CPM File Transfer needs to be set up to a recipient, and it is decided that to reach the recipient, interworking is required. In this case the Non-CPM Communication Service does not support file transfers.

NOTE 1: The flow only focuses on the interactions between the CPM Participating Function, the Interworking Selection Function and the Interworking Function. Upstream interactions are purposely not described.

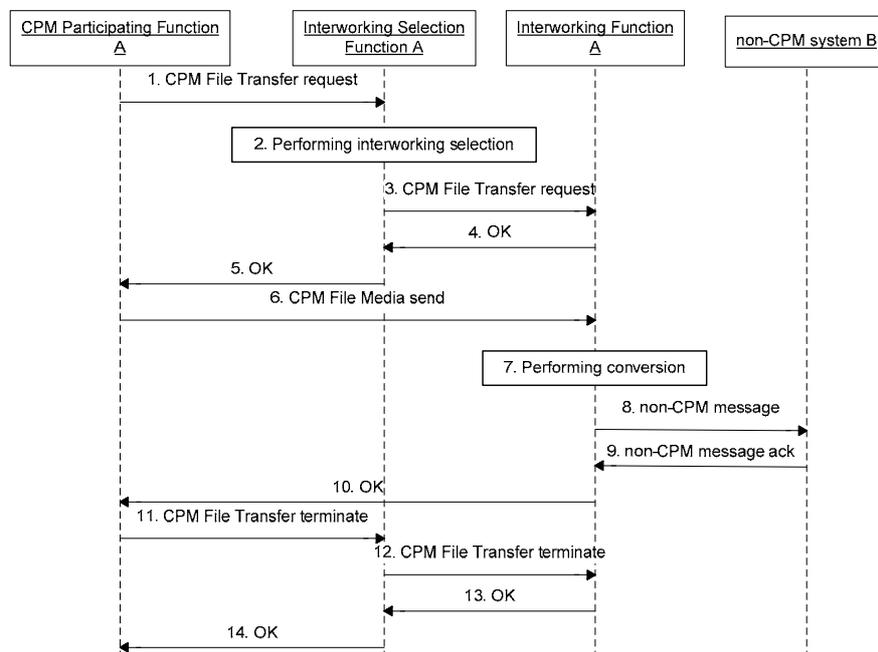


Figure 42: CPM File Transfer Interworking to non-CPM message

1. CPM Participating Function A sends the CPM File Transfer request to Interworking Selection Function A.
2. Interworking Selection Function A extracts the file transfer details from the CPM File Transfer request and selects the most appropriate Interworking Function for handling the CPM File Transfer.
3. Interworking Selection Function A sends the CPM File Transfer request to Interworking Function A.
4. Interworking Function A returns an OK response to Interworking Selection Function A.
5. Interworking Selection Function A forwards the OK response to CPM Participating Function A.
6. CPM Participating Function A sends the file media to Interworking Function A.
7. Interworking Function A converts the CPM File Transfer in a non-CPM message.
8. Interworking Function A sends the non-CPM message to non-CPM System B.

- 9. Non-CPM system B acknowledges reception of the message.
- 10. Interworking Function A sends the OK response for the received media to CPM Participating Function A.
- 11-14. The CPM File Transfer is terminated.

## B.5.5 CPM Session Interworking

### B.5.5.1 Interworking with a Single Interworking Function

This section describes the case where a CPM Session needs to be set up to a recipient, and it is decided that to reach the recipient, interworking is required. In this case the CPM Session can be interworked via a single Interworking Function.

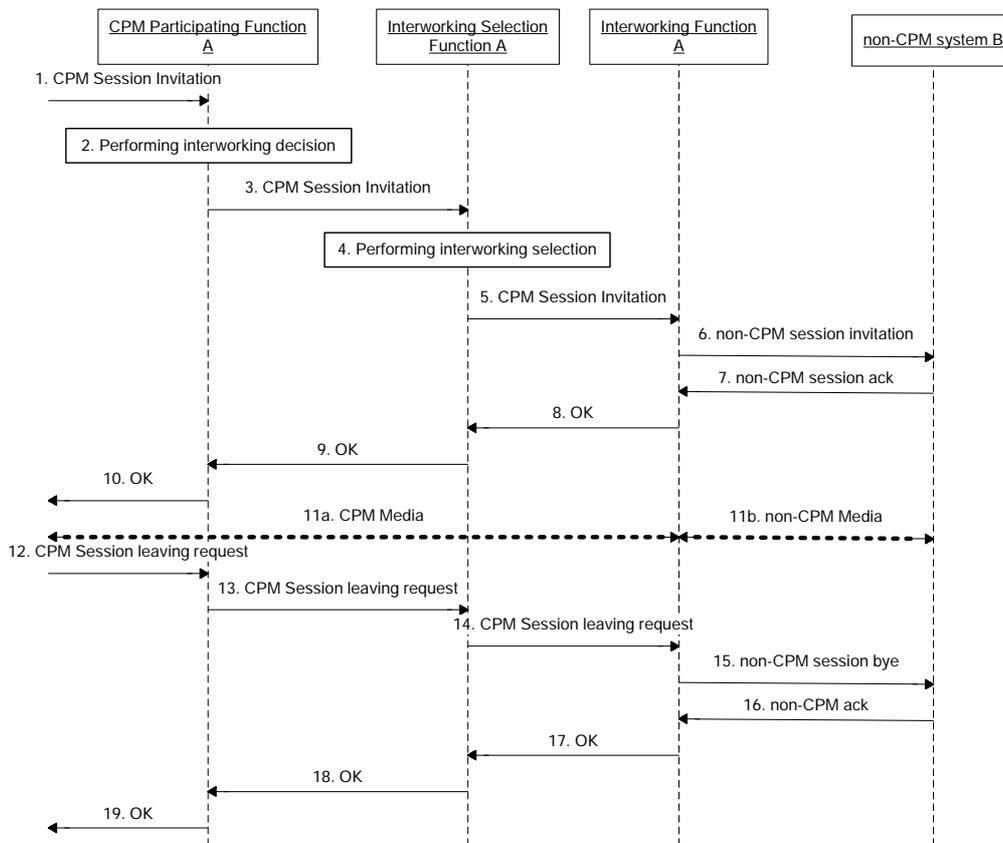


Figure 43: CPM Session Interworking with a single Interworking Function

- 1. A CPM Session Invitation arrives at CPM Participating Function A.
- 2. CPM Participating Function A performs originating CPM services and decides whether interworking is required.
- 3. CPM Participating Function A sends the CPM Session Invitation to Interworking Selection Function A.
- 4. Interworking Selection Function A extracts the session details from the session description of the CPM Session Invitation and selects the most appropriate Interworking Function for handling the CPM Session.
- 5. Interworking Selection Function A sends the CPM Session Invitation to Interworking Function A.
- 6. Interworking Function A sends a non-CPM session invitation to non-CPM system B.

7. Non-CPM system B accepts the session by sending a non-CPM session ack.

NOTE: Steps 8-10 could be initiated before step 7, (e.g. per service provider policies).

8. Interworking Function A returns an OK response to Interworking Selection Function A.

9. Interworking Selection Function A forwards the OK response to CPM Participating Function A.

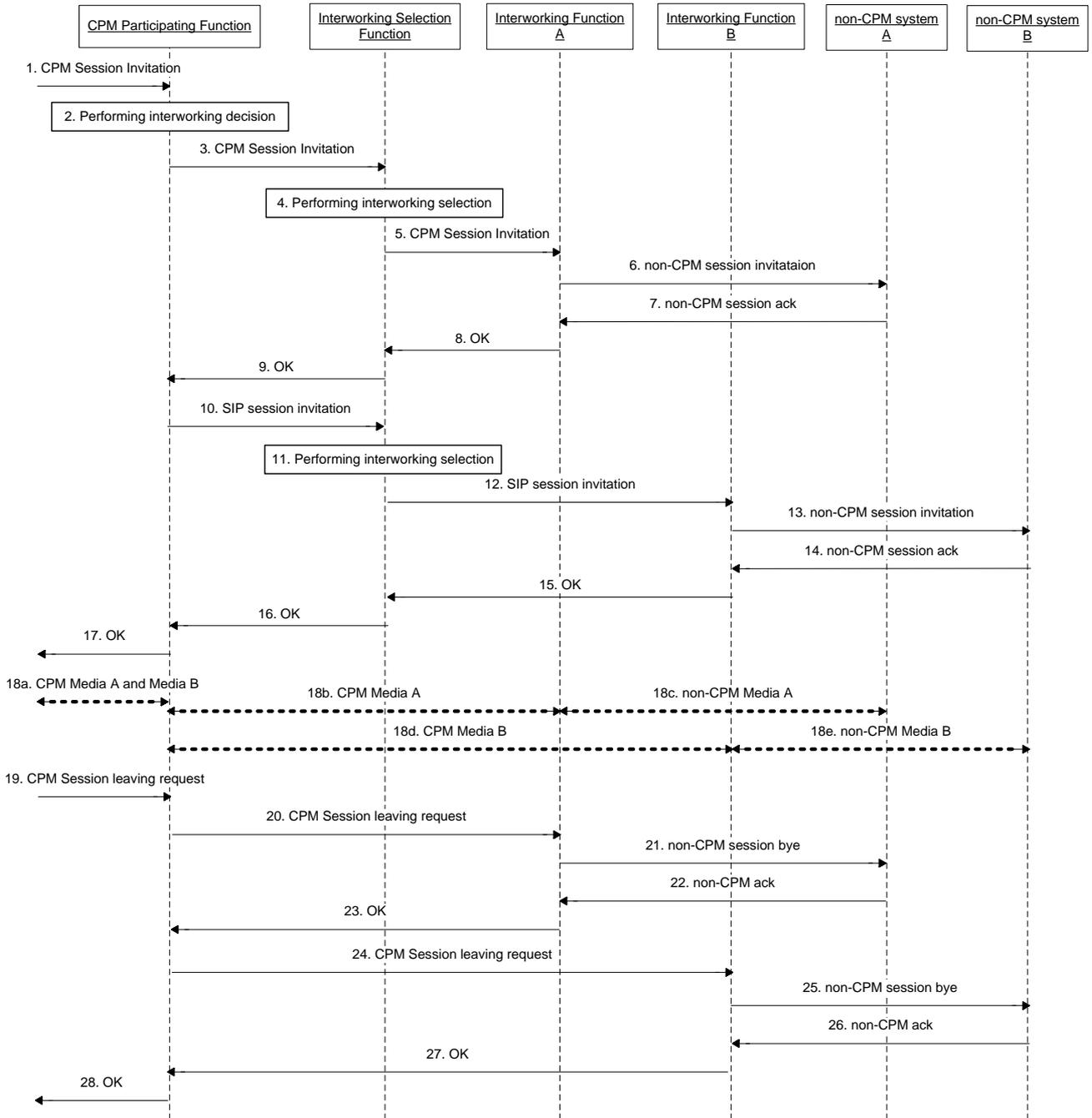
10. The OK response is routed further upstream by CPM Participating Function A.

11. Media is exchanged via the session that has been set up. The CPM Media that is sent on the CPM side is translated into non-CPM Media sent on the non-CPM side by Interworking Function A and vice-versa.

12-19. The CPM Session is terminated both on the CPM side and the non-CPM side.

### **B.5.5.2 Interworking with Multiple Interworking Functions**

This section describes the case where a CPM Session needs to be set up to a recipient, and it is decided that to reach the recipient, interworking is required. In this case the CPM Session needs to be interworked via multiple Interworking Functions.



**Figure 44: CPM Session Interworking with multiple Interworking Function**

1. A CPM Session Invitation arrives at the CPM Participating Function.
2. The CPM Participating Function performs originating CPM services and decides whether interworking is required.
3. The CPM Participating Function sends the CPM Session Invitation to the Interworking Selection Function.

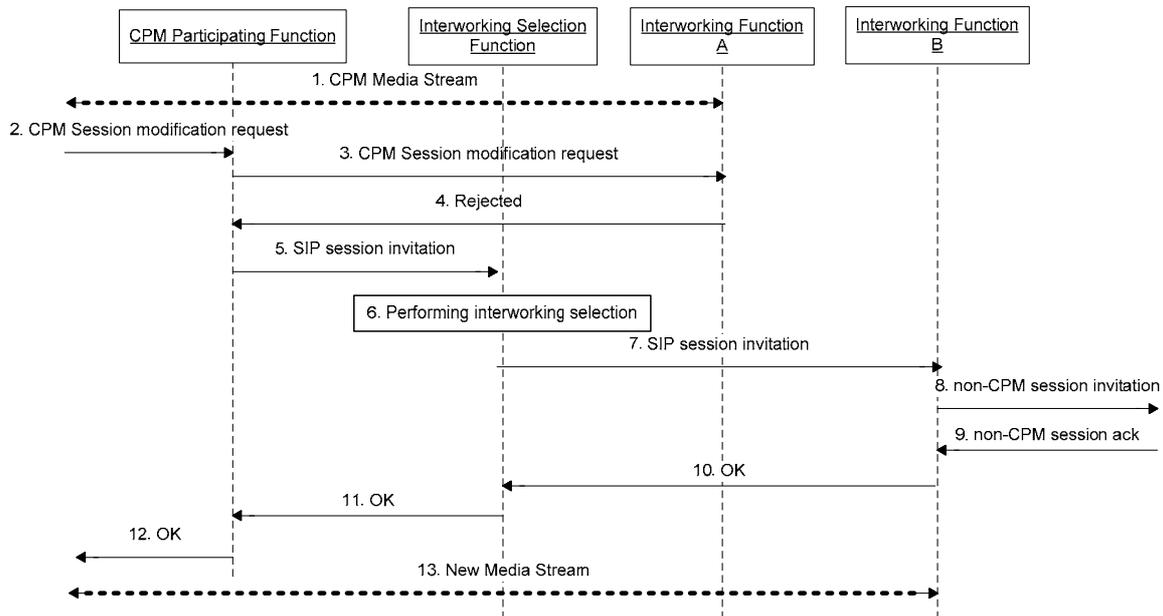
4. The Interworking Selection Function extracts the session details from the session description of the CPM Session Invitation and selects the most appropriate Interworking Function for handling the CPM Session.
5. The Interworking Selection Function sends the CPM Session Invitation to the Interworking Function A.
6. The Interworking Function A sends a non-CPM session invitation for the Media Stream A to the non-CPM system A.
7. The non-CPM system A accepts the session by sending a non-CPM session ack.  
NOTE: Steps 8-9 could be initiated before step 7, (e.g. per service provider policies).
8. The Interworking Function A returns to the Interworking Selection Function an OK response which contains an SDP answer indicating that the Media Stream A has been accepted.
9. The Interworking Selection Function A forwards the OK response to CPM Participating Function.
10. By examining the Ok response the CPM Participating Function becomes aware that the Media Stream B was declined because the Interworking Function A does not support it and sends a SIP session invitation containing offering the Media Stream B to the Interworking Selection Function according to service provider policy.
- 11-16. Steps 4-9 are repeated for the SIP session invitation with a difference that the Interworking Function interworks with a non-CPM system B this time.
17. The OK response to the CPM Session Invitaitaion is routed further upstream by the CPM Participating Function.
18. Media is exchanged via the session that has been set up. The CPM Media A/Media B that is sent on the CPM side is translated into non-CPM Media A/Media B sent on the non-CPM side by Interworking Function A/Interworking Function B and vice-versa.
- 19-28. The CPM Session is terminated both on the CPM side and the non-CPM side.

## **B.5.6 CPM Session Modification**

### **B.5.6.1 Adding a New Media Stream**

This section describes the case where a CPM User adds a new Media Stream during a CPM Session with a non-CPM Principal. The non-CPM Principal is subscribed to a Non-CPM Communication Service A that does not support the new Media Stream. However he is subscribed to another Non-CPM Communication Service B that supports the new Media Stream.

In the flows the two Non-CPM Communication Services A and B are taken care of by two different Interworking Functions, i.e. an Interworking Function A for the Non-CPM Communication Service A and an Interworking Function B for the Non-CPM Communication Service B.



**Figure 45: Adding a new Media Stream to the ongoing CPM Session**

1. There exists an ongoing SIP session conveying a Media Stream between the CPM Client and the Interworking Function A.
2. The CPM Participating Function receives a CPM Session modification request from a CPM Client.
3. The CPM Participating Function identifies the corresponding session and the Interworking Function in the Media path for the CPM Session modification request and forwards the request towards Interworking Function A, the other end point of the existing session.
4. Interworking Function A sends a Rejected response towards the CPM Participating Function because it does not support the newly offered Media Stream in the CPM Session modification request.
5. The CPM Participating Function sends a new SIP session invitation towards the Interworking Selection Function.
 

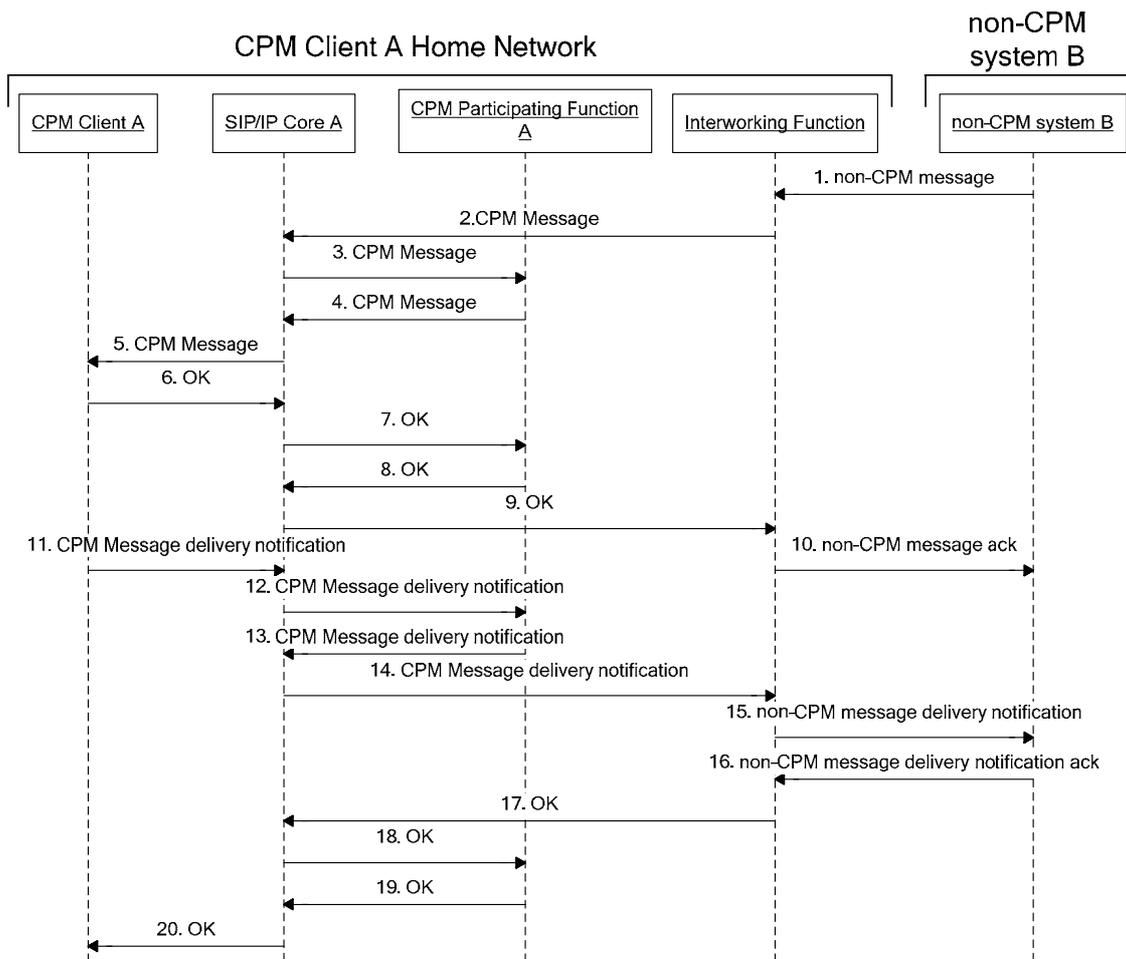
NOTE: If the rejection for the CPM Session modification request results from willingness of the recipient non-CPM Principal the CPM Participating Function forwards the received Rejected response to the originator.
6. The Interworking Selection Function selects the Interworking Function B for the received SIP session invitation
7. The Interworking Selection Function sends the SIP session invitation to Interworking Function B.
8. The Interworking Function B converts the received SIP session invitation into a non-CPM session invitation or non-CPM inviting message according to the Non-CPM Communication Service B handled by the Interworking Function B or service provider policy and sends the non-CPM session invitation or non-CPM inviting message towards the corresponding Non-CPM Communication Service B.
9. The Interworking Function B receives a non-CPM ack from the Non-CPM Communication Service B.
10. The Interworking Function B converts the non-CPM ack into an OK response to the SIP session invitation routable in the CPM network and sends it to the Interworking Selection Function.
11. The Interworking Selection Function forwards the OK response to the CPM Participating Function.

12. The CPM Participating Function routes towards the CPM Client the OK response to the CPM Session modification request.

13. A new media stream is created between the CPM Client and the Interworking Function B.

## B.6 Inbound Interworking

This section describes the case where the sending user sends a non-CPM message to a recipient that is a CPM User.



**Figure 46: Non-CPM message to CPM User**

1. Non-CPM system B sends a message that arrives at the Interworking Function. Non-CPM system B may request a delivery notification and/or read report. In this flow non-CPM system B requests a delivery notification.

2. The Interworking Function maps the message to a CPM Standalone Message and routes it towards CPM Client A.

3. SIP/IP Core A routes the CPM Standalone Message to CPM Participating Function A based on the CPM Address of the recipient CPM Client and CPM Feature Tag.

4. CPM Participating Function A executes the necessary terminating service control and if successful, sends the CPM Standalone Message via SIP/IP Core A towards CPM Client A.

5. SIP/IP Core A routes the CPM Standalone Message to CPM Client A.

6-9. CPM Client A sends an OK response for the CPM Standalone Message. The OK response is sent to CPM Participating Function A along the same Signalling Path, and then on towards the Interworking Function.

10. The Interworking Function routes the ack towards the sender, if required by the particular non-CPM system.

NOTE 1: The procedure can end in step 10. Steps 11 to 20 occur only if a delivery notification has been requested in the non-CPM Message sent in step 1, and the Interworking Function mapped the non-CPM message to a CPM Standalone Message with delivery notification request.

11. CPM Client A generates a delivery notification and sends it towards the sender.

12. SIP/IP Core A routes the CPM Standalone Message delivery notification to CPM Participating Function A triggered on the CPM Feature Tag and CPM Client A's CPM Address.

13. CPM Participating Function A performs originating CPM services and then sends the CPM Standalone Message delivery notification to SIP/IP Core A.

14. SIP/IP Core A routes the request to the Interworking Function.

15. Non-CPM system B receives the delivery notification.

16. Non-CPM system B may acknowledge reception.

17-20. The Interworking Function sends an OK response towards CPM Client A along the same Signalling Path.

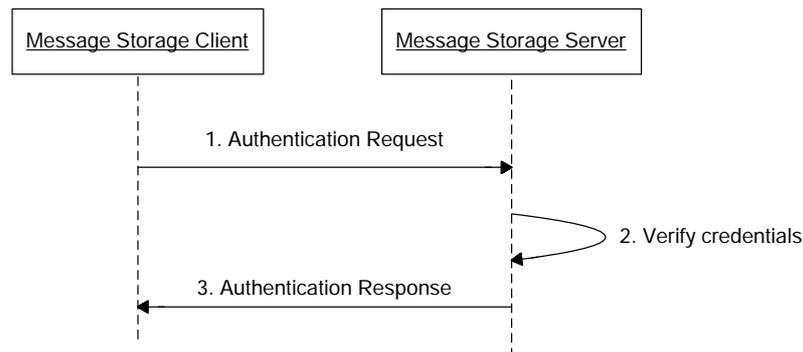
NOTE 2: Step 17 could be initiated before step 16 (e.g. per service provider policies or to prevent the risk of timeout).

## Appendix C. Message Storage Flows

(Informative)

### C.1 Authentication

This flow describes how a Message Storage Client authenticates to a Message Storage Server.



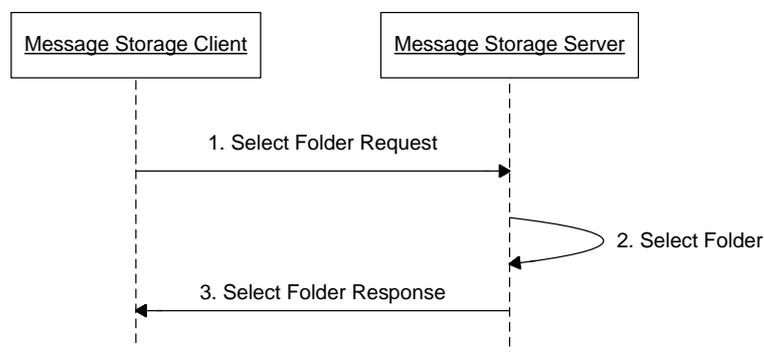
**Figure 47: Authentication to Message Storage Server**

1. Accessing a Message Storage Server requires authentication, therefore a Message Storage Client sends an authentication request to the Message Storage Server. The request includes the credentials to be used for authentication.
2. The Message Storage Server verifies the supplied authentication credentials.
3. The Message Storage Server sends back its response to the requestor. The response includes status information about the transaction.

NOTE: Mutual authentication can be performed using the same flow.

### C.2 Selecting the Active Folder

This flow describes how the Message Storage Client selects the active folder on the Message Storage Server.



**Figure 48: Selection the active folder on the Message Storage Server**

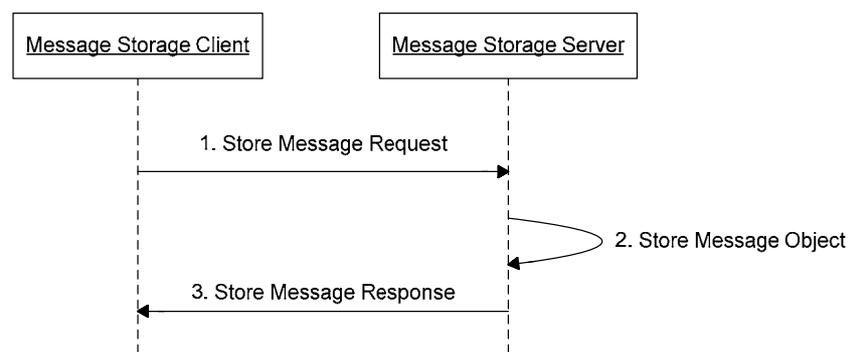
1. A Message Storage Client is required to select the active folder; therefore it sends a select folder request to the Message Storage Server. The request includes the name of the folder to be set as active.
2. The Message Storage Server verifies that the folder exists, and selects it as the active folder.
3. The Message Storage Server sends back its response to the requestor. The response includes status information about the transaction.

### C.3 Storing a CPM Standalone Message

This flow describes how the Message Storage Client stores a CPM Standalone Message in a particular folder of the CPM User on the Message Storage Server.

NOTE 1: Such a storage operation can be triggered by recording the CPM Standalone Message or by delivering it to the Message Storage Server.

NOTE 2: Storing other objects like CPM File Transfer Histories or CPM Session Histories works analogously

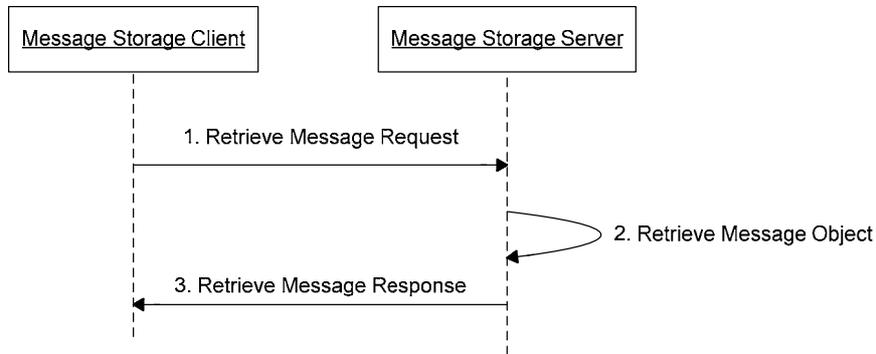


**Figure 49: Storing a CPM Standalone Message on the Message Storage Server**

1. To store a CPM Standalone Message, the Message Storage Client SHALL send an object store request to the Message Storage Server. The object store request includes the CPM Standalone Message and the name of the folder in which the CPM Message is to be stored.
2. The Message Storage Server SHALL verify that there is enough storage space, assign a unique identifier to the supplied CPM Standalone Message and store it into the indicated folder.
3. If successful, the Message Storage Server SHALL respond with an OK response to the requestor. The response includes the unique identifier that has been assigned for the newly stored item. Otherwise, if the object store operation is not successful, the Message Storage Server SHALL respond with an appropriate error response.

### C.4 Retrieving a Stored Object

This flow describes how the Message Storage Client retrieves a stored object from the Message Storage Server.



**Figure 50: Retrieving a stored object from the Message Storage Server**

1. To retrieve a stored object from the Message Storage Server, the Message Storage Client SHALL send an object fetch request to the Message Storage Server. The object fetch request SHALL include the identifier of the stored object and an indication of the required information to be retrieved.
2. Upon receiving an object fetch request, the Message Storage Server SHALL verify that a stored object with the supplied identifier exists in the active folder and retrieve the whole or part(s) of the indicated object from the storage.
3. If object fetch operation is successful, the Message Storage Server SHALL send back its response carrying the requested stored object or a part(s) of the stored object to the requestor. Otherwise, if the object fetch operation is not successful, it SHALL respond with an appropriate error.

## Appendix D. Handling of ISF Relevant Criteria (Informative)

This appendix describes a specific way for the Interworking Selection Function to select the most appropriate Interworking Function to send a CPM Standalone Message or a CPM Session Invitation to.

The interworking selection process could be steered by a configuration table. The rows of the configuration table represent different criteria to evaluate. The columns represent the Non-CPM Communication Services available to interwork to. The (x,y) entries in the table specify if and how the respective Non-CPM Communication Service satisfies the respective criterion.

- A boolean value for an (x,y) entry denotes if the respective Non-CPM Communication Service fulfils the respective criterion or not.
- A numeric value for an (x,y) gives a threshold value up onto which the respective Non-CPM Communication Service fulfils the respective criterion.

Table 1: Example routing table shows an example table as it might be used by a specific service provider. This service provider has five specific Non-CPM Communication Services to interwork to. His Interworking Selection Function chooses to restrict SMS interworking to messages that are no more than 160 characters. Similar restrictions are specified for the other Non-CPM Communication Services. For PS Voice, the limit is zero as it does not make sense to this service provider to interwork CPM Messages to PS Voice. Note that another service provider might have text-to-speech translation capabilities built into the IWF for PS Voice and, therefore, might specify a much higher limit than zero.

For video streams offered via CPM Session invitations, this service provider chooses to interwork them to Internet enabled television and nowhere else. Also, this service provider has a preference for MMS to interwork to, possibly because his MMSC has free capacity.

The two bottom rows are user specific, i.e., they might be taken from different places, e.g., a user profile or be computed by combining a user's service class (e.g. prepaid user) with service provider policies assigned to this service class. In this example, this particular target user cannot get incoming CPM traffic be interworked to Internet TV, possibly because it is not within his subscription plan. The target user's individual preferences are shown in the last row. Again, this information will not be directly stored in such a table but rather be taken from elsewhere – here from the user's preferences. This particular user dislikes SMS and does not want his (potentially existing) Internet TV set display CPM traffic.

Note that user specific criteria would be easily available in the target network when the target user is a CPM user (but CPM traffic is interworked to him for some reason). If no information is available about the target user, the service provider may choose to assign default values or the values would be simply empty.

	SMS	MMS	Email	PS Voice	Internet TV
<b>Max size (bytes)</b>	140	45000	10.000.000	0	1000
<b>Video stream ok</b>	No	No	No	No	Yes
<b>Audio stream ok</b>	No	No	No	Yes	Yes
<b>Video clip ok</b>	No	No	Yes	No	No
<b>Audio clip ok</b>	No	Yes	Yes	No	No
<b>Picture ok</b>	No	Yes	Yes	No	Yes
<b>Text ok</b>	Yes	Yes	Yes	No	Yes
<b>Bidirectional Voice ok</b>	No	No	No	Yes	No
<b>Preferred by service provider</b>	No	Yes	No	No	No
<b>Service provider allows for user</b>	Yes	Yes	No	Yes	No

<b>Fine with user</b>	No	Yes	Yes	Yes	No
-----------------------	----	-----	-----	-----	----

**Table 1: Example routing table**

For the individual entries, entries other than of boolean or integral value would be possible but harder to manage. Therefore, they are kept out of scope in this description.

One of the major benefits of such a scheme is its extensibility: at any time, the service provider can add new Non-CPM Communication Services, simply by adding a column to the table and assigning values to the different entries in this table (user specific entries can be given default values that then can be manipulated by individual users). Also, new evaluation criteria can be introduced by adding rows to the table. New evaluation criteria typically would be needed to either fine-tune the selection process or because a newly introduced Non-CPM Communication Service would need to be distinguished from other Non-CPM Communication Services by certain evaluation criteria not known yet.

In case, the evaluation process yields several Non-CPM Communication Services as being feasible to interwork to, the resulting tie needs to be broken. If no Non-CPM Communication Service is found feasible to interwork to, the CPM Standalone Message or CPM Session Invitation is rejected and handed back to the Participating Function or Conferencing Function.

One of the variations of this scheme would be to rank the criteria, i.e., some criteria would be more important than others. For example, a service provider might choose to overrule user preferences. A simple ranking scheme would be to just order the criteria from top to bottom (a more advanced ranking would be to assign weights to criteria). Such a ranking of criteria might be important to break ties when several non-CPM Communication Services would be eligible to interwork a specific CPM Standalone Message or CPM Session Invitation.

If a user has multiple devices and several Non-CPM Communication Services would be eligible to interwork a specific CPM Standalone Message or CPM Session Invitation, the choice of devices might be used as additional criterion to decide between the different Non-CPM Communication Services.