XML Document Management (XDM) Specification

Approved Version 2.1 – 03 Apr 2012

Open Mobile Alliance

OMA-TS-XDM_Core-V2_1-20120403-A

Contents

1. SCOPE 7
2. REFERENCES 8
   2.1 NORMATIVE REFERENCES 8
   2.2 INFORMATIVE REFERENCES 11
3. TERMINOLOGY AND CONVENTIONS 13
   3.1 CONVENTIONS 13
   3.2 DEFINITIONS 13
   3.3 ABBREVIATIONS 15
4. INTRODUCTION 17
   4.1 VERSION 1.1 17
   4.2 VERSION 2.0 17
   4.3 VERSION 2.1 17
5. COMMON PROCEDURES 19
   5.1 SECURITY PROCEDURES 19
      5.1.1 Authentication 19
      5.1.2 Principal Identity Assertion 19
      5.1.3 Principal Identity Sharing 21
      5.1.4 Integrity and Confidentiality Protection 21
      5.1.5 Authorization 21
   5.2 COMMON EXTENSIONS 22
      5.2.1 URI Lists defined in List XDMS 22
      5.2.2 Authorization Rules 22
      5.2.3 Detailed Conflict Reports 27
   5.3 COMMON APPLICATION USAGES 28
      5.3.1 XCAP Server Capabilities 28
      5.3.2 XML Documents Directory 31
   5.4 COMMON CONTENT TYPES 34
      5.4.1 Search Document 34
      5.4.2 XDCP Document 35
   5.5 GLOBAL DOCUMENTS 39
   5.6 ACCESS PERMISSIONS DOCUMENT 39
      5.6.1 Structure 39
      5.6.2 Application Unique ID 41
      5.6.3 Default Namespace 41
      5.6.4 XML Schema 41
      5.6.5 MIME Type 42
      5.6.6 Validation Constraints 42
      5.6.7 Data Semantics 42
      5.6.8 Naming Conventions 44
      5.6.9 Global Documents 45
      5.6.10 Resource Interdependencies 45
      5.6.11 Authorization Policies 45
      5.6.12 Subscription to changes 45
      5.6.13 Search Capabilities 45
      5.6.14 XDM Preferences Document 45
      5.6.15 History Information Documents 45
      5.6.16 Forwarding 45
      5.6.17 Restore 45
      5.6.18 Document Reference 45
      5.6.19 Differential Read and Write 45
   5.7 HISTORY INFORMATION 46
      5.7.1 Modification History Information Document 46

5.7.2 Request History Information Document 50
5.8 XDM PREFERENCES DOCUMENT 54
5.8.1 Structure 54
5.8.2 Application Unique ID 56
5.8.3 Default Namespace 56
5.8.4 XML Schema 56
5.8.5 MIME Type 56
5.8.6 Validation Constraints 56
5.8.7 Data Semantics 56
5.8.8 Naming Conventions 59
5.8.9 Global Documents 59
5.8.10 Resource Interdependencies 59
5.8.11 Authorization Policies 59
5.8.12 Subscription to changes 59
5.8.13 Search Capabilities 59
5.8.14 XDM Preferences Document 59
5.8.15 History Information Documents 59
5.8.16 Forwarding 59
5.8.17 Restore 59
5.8.18 Document Reference 59
5.8.19 Differential Read and Write 60
6. DESCRIPTION OF PROCEDURES AT XDM FUNCTIONAL ENTITIES 61
6.1 PROCEDURES AT THE XDMC AND THE XDM AGENT 61
6.1.1 Document Management 61
6.1.2 Subscribing to Changes in the XDM Resources 68
6.1.3 Searching for Data in XML Documents 72
6.1.4 Retrieval of History Information 73
6.1.5 Management of Access Permissions 73
6.1.6 Management of XDM Preferences 73
6.1.7 XCAP Server Capabilities Retrieval 74
6.1.8 XCAP Directory Information Retrieval 74
6.2 PROCEDURES AT THE XDM SERVER 74
6.2.1 Document Management 74
6.2.2 Subscriptions to Changes in XDM Resources 75
6.2.3 Searching for Data in XML Documents 76
6.2.4 Handling of Access Permissions Documents 77
6.2.5 Enforcing XDM Access Permissions 78
6.2.6 Handling of XDCP Operations 79
6.2.7 Handling of History Information 87
6.2.8 XCAP Server Capabilities Retrieval 88
6.2.9 Directory Information Retrieval 88
6.3 PROCEDURES AT THE AGGREGATION PROXY 88
6.3.1 HTTP Request Handling 89
6.3.2 Compression 91
6.4 PROCEDURES AT THE SEARCH PROXY 91
6.4.1 Search Request Forwarding 91
6.4.2 Search Response Aggregation 92
6.5 PROCEDURES AT THE CROSS-NETWORK PROXY 92
6.5.1 Outbound Requests 93
6.5.2 Inbound Requests 93
6.6 PROCEDURES AT THE SUBSCRIPTION PROXY 94
6.6.1 Handling of the XDCP Subscribe Command 94
6.6.2 Handling of the SIP SUBSCRIBE Request 94
6.6.3 Establishing the Mapping between XDCP Subscribe and SIP Back-end SUBSCRIBE 95
6.6.4 Handling of the Back-end SIP NOTIFY for SIP Subscriptions 96
6.6.5 Handling of the Back-end SIP NOTIFY Request for XDCP Subscriptions 96

APPENDIX A. CHANGE HISTORY (INFORMATIVE) 98
A.1 APPROVED VERSION 2.1 HISTORY 98
APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) 99
B.1 XDMC 99
B.2 XDM AGENT 101
B.3 XDMS 102
B.4 AGGREGATION PROXY 103
B.5 SEARCH PROXY 104
B.6 CROSS-NETWORK PROXY 105
B.7 SUBSCRIPTION PROXY 106
APPENDIX C. EXAMPLES (INFORMATIVE) 107
C.1 SAMPLE XCAP OPERATION 107
C.2 SAMPLE XCAP MESSAGE FLOW 109
C.3 SAMPLE XCAP DIRECTORY RETRIEVAL OPERATION OF ALL USER XDM DOCUMENTS 111
C.4 SAMPLE XCAP DIRECTORY RETRIEVAL OPERATION OF SPECIFIC USER DOCUMENTS 113
C.5 SAMPLE SUBSCRIBING TO CHANGES IN XDM DOCUMENTS 114
C.5.1 Direct Subscription 114
C.5.2 Subscription Using Subscription Proxy 117
C.5.2.1 Initial Subscription 117
C.5.2.2 Subsequent Notifications 125
C.6 SAMPLE SEARCH OPERATION 130
C.7 EXAMPLES OF ACCESS PERMISSIONS DOCUMENTS 134
C.7.1 Administrator Controlled Access Permission Document 134
C.7.2 Administrator Controlled User Directory 134
C.7.3 Granting only a View of the Access Permissions Document to all users 135
C.7.4 Blocking a single User to retrieve the Access Permissions Document 136
C.7.5 Access Permissions Document with a filter 137
C.7.6 Administrator and Primary Principal Controlled User Directory 138
C.8 EXAMPLES OF XDCP OPERATIONS 139
C.8.1 Differential Read - No Filter 139
C.8.2 Differential Read - With Filter 141
C.8.3 "Reactive Authorization" via Differential Read and Write with Filter 142
C.8.4 Set Document Reference 147
C.8.5 Retrieve Document Reference 148
C.8.6 Remove Document Reference 148
C.8.7 Forwarding XDM Resources 150
C.9 EXAMPLES OF HISTORY INFORMATION DOCUMENTS 153
C.9.1 Request History Information Document Example 153
C.9.2 Modification History Information Document Examples 153
C.10 XDM PREFERENCES DOCUMENT EXAMPLES 153
APPENDIX D. XDMC PROVISIONING (NORMATIVE) 156
D.1 PROVISIONED XDMC PARAMETERS 156
D.2 APPLICATION CHARACTERISTICS 157
D.3 MANAGEMENT OBJECTS 158
APPENDIX E. OMA SPECIFIC URI-PARAMETERS (NORMATIVE) 159
E.1 AUID URI-PARAMETER 159
APPENDIX F. OMA XDCP OPERATIONS (NORMATIVE) 160
APPENDIX G. "REACTIVE AUTHORIZATION OF XDM REQUESTS USING REQUEST HISTORY INFORMATION DOCUMENTS" (INFORMATIVE) 161
APPENDIX H. " ACCESS PERMISSIONS CHANGE NOTIFICATIONS " (INFORMATIVE) 163
APPENDIX I. "FILTER ABNF" (NORMATIVE) 165

Figures

Figure C.1 - Sample XCAP operation 107
Figure C.2 - XDMC manipulating an XDM Document 109
Figure C.3 - Sample XCAP Directory retrieval operation 111
Figure C.4 - Sample XCAP Directory retrieval operation from a particular XDMS 113
Figure C.5 - XDM Client subscribes to changes in XDM Documents. 115
Figure C.6 - XDMC subscribes to changes in XDM Documents. 118
Figure C.7 - Notification of changes in XDM Documents. 126
Figure C.8 - Sample Search operation 130
Figure C.9 - Example Request History Information Document 139
Figure C.10 - XDCP Differential Read Request operation - no filter 140
Figure C.11 - XDCP Differential Read Request - No Filter 140
Figure C.12 - XDCP Differential Read Response - No Filter 141
Figure C.13 - XDCP Differential Read Request operation - with Filter 141
Figure C.14 - XDCP Differential Read Request - with Filter 142
Figure C.15 - XDCP Differential Read Response - with Filter 142
Figure C.16 - Starting State of Request History Information Document on the XDMS 143
Figure C.17 - "Reactive Authorization" with XDCP Differential Read and Write Request 143
Figure C.18 - XDCP Differential Read Request - Filter & Null E-Tag 144
Figure C.19 - XDCP Differential Read Response - Filter & Null E-Tag 145
Figure C.20 - XDMC Locally Stored Request History Information Document 145
Figure C.21 - XDCP Differential Write Request - Filter 146
Figure C.22 - XDCP Differential Write Response - Filter 146
Figure C.23 - Final State of Request History Information Document on the XDMS 147
Figure C.24 - XDM Forward Example Flows 150

Tables

Table 1: Example usages of URI Templates 157

1. Scope

This document specifies common protocols, data access conventions, common data application usages and functional entities that are needed to provide XDM services to other Enablers. Such Enablers can utilize this specification to support any required application-specific usages.

2. References

2.1 Normative References

OMA

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

[CPM_RD] "Converged IP Messaging Requirements", Open Mobile Alliance™, OMA-RD-CPM-V1_0, URL: http://www.openmobilealliance.org/

[Dict] "Dictionary for OMA Specifications", Version 2.7, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_7, URL: http://www.openmobilealliance.org/

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

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

[IM_TS] "Instant Messaging using SIMPLE", Version 1.0, Open Mobile Alliance™, OMA-TS-SIMPLE_IM-V1_0, URL: http://www.openmobilealliance.org/

[PoC_CP] "Push to talk Over Cellular (PoC) – Control Plane Specification", Version 2.1, Open Mobile Alliance™, OMA-TS-PoC-ControlPlane-V2_1, URL: http://www.openmobilealliance.org/

[Push_ERELD-V2_2] "Enabler Release Definition for Push", Version 2.2, Open Mobile Alliance™, OMA-ERELD-Push-V2_2, URL: http://www.openmobilealliance.org/

[SCRRULES] "SCR Rules and Procedures ",Version 1.0, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures-V1_0, URL: http://www.openmobilealliance.org/

[W3C-XQUERY] W3C Recommendation "XQuery 1.0: An XML Query Language", Scott Boag et al, January 23 2007, World Wide Web Consortium (W3C), URL: http://www.w3.org/TR/xquery/

[W3C-XQUERY_FullText] W3C Candidate Recommendation "XQuery and XPath Full Text 1.0", Sihem Amer-Yahia et al, 28 January 2010, World Wide Web Consortium (W3C), URL: http://www.w3.org/TR/xpath-full-text-10/

Note: Work in progress

[XDM_AC] "XDM Application Characteristics file of XDM V2.0", Version 1.0, Open Mobile Alliance™, OMA-SUP-AC_ap0007_xdm-v1_0, URL: http://www.openmobilealliance.org/

[XDM_AD] "XML Document Management Architecture", Version 2.1, Open Mobile Alliance™, OMA-AD-XDM-V2_1, URL: http://www.openmobilealliance.org/

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

[XDM_ERELD-V1_1] "Enabler Release Document for XDM", Version 1.1, Open Mobile Alliance™, OMA-ERELD-XDM-V1_1, URL: http://www.openmobilealliance.org/

[XDM_ERELD-V2_0] "Enabler Release Document for XDM", Version 2.0, Open Mobile Alliance™, OMA-ERELD-XDM-V2_0,

URL: http://www.openmobilealliance.org/

[XDM_ERELD-V2_1] "Enabler Release Document for XDM", Version 2.1, Open Mobile Alliance™, OMA-ERELD-XDM-V2_1, URL: http://www.openmobilealliance.org/

[XDM_Group] "Group XDM Specification", Version 1.1, Open Mobile Alliance™, OMA-TS-XDM_Group-V1_1, URL: http://www.openmobilealliance.org/

[XDM_List] "List XDM Specification", Version 2.1, Open Mobile Alliance™, OMA-TS-XDM_List-V2_1, URL: http://www.openmobilealliance.org/

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

[XDM_RD] "XML Document Management Requirements", Version 2.1, Open Mobile Alliance™, OMA-RD-XDM-V2_1, URL: http://www.openmobilealliance.org/

[XDM_UPPD] "UPP Directory XDM Specification", Version 1.0, Open Mobile Alliance™, OMA-TS-XDM_UPP_Directory-V1_0, URL: http://www.openmobilealliance.org/

[XSD_ap] "XML Schema Definition : XDM Access Permissions", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_access_permissions-V1_0, URL: http://www.openmobilealliance.org/

[XSD_commPol] "XML Schema Definition: "XDM – Common Policy", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_commonPolicy-V1_0, URL: http://www.openmobilealliance.org/

[XSD_ext] "XML Schema Definition: XDM Extensions", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_extensions-V1_0, URL: http://www.openmobilealliance.org/

[XSD_ext_2_1] "XML Schema Definition: "XDM 2.1 – Extensions", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_2_1_extensions-V1_0, URL: http://www.openmobilealliance.org/

[XSD_modHist] "XML Schema Definition: XDM – Modification History", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_modification_history-V1_0, URL: http://www.openmobilealliance.org/

[XSD_search] "XML Schema Definition: "XDM – Search", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_search-V1_0, URL: http://www.openmobilealliance.org/

[XSD_xcapDir] "XML Schema Definition: "XDM – XCAP Directory", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_xcapDirectory-V1_0, URL: http://www.openmobilealliance.org/

[XSD_xcapErr] "XML Schema Definition: "XDM – XCAP Error", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_xcapError-V1_0, URL: http://www.openmobilealliance.org/

[XSD_xdcp] "XML Schema Definition : XDM – XDCP Commands", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_xdcp-V1_0, URL: http://www.openmobilealliance.org/

[XSD_reqHist] "XML Schema Definition: "XDM – Request History", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_request_history-V1_0, URL: http://www.openmobilealliance.org/

IETF

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

[RFC2046] IETF RFC 2046 "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", N. Freed, N.

Borenstein, November 1996,
URL: http://www.ietf.org/rfc/rfc2046.txt

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

[RFC2234] IETF RFC 2234 "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. November 1997,
URL: http://www.ietf.org/rfc/rfc2234.txt

[RFC2246] IETF RFC 2246 "The TLS Protocol", T.Dierks at al, January 1999,
URL: http://www.ietf.org/rfc/rfc2246.txt

[RFC2616] IETF RFC 2616 "Hypertext Transfer Protocol – HTTP/1.1", R. Fielding, June 1999,
URL: http://www.ietf.org/rfc/rfc2616.txt

[RFC2617] IETF RFC 2617 "HTTP Authentication: Basic and Digest Access Authentication", Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and L. Stewart, June 1999,
URL: http://www.ietf.org/rfc/rfc2617.txt

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

[RFC3040] IETF RFC 3040 "Internet Web Replication and Caching Taxonomy", I. Cooper, I. Melve, G. Tomlinson, January 2001,
URL: http://www.ietf.org/rfc/rfc3040.txt

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

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

[RFC3428] IETF RFC 3428 "Session Initiation Protocol (SIP) Extension for Instant Messaging", B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, D. Gurle, December 2002,
URL: http://www.ietf.org/rfc/rfc3428.txt

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

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

[RFC3986] IETF RFC 3986 "Uniform Resource Identifier (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, January 2005,
URL: http://www.ietf.org/rfc/rfc3986.txt

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

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

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

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

[RFC4745] IETF RFC 4745 "Common Policy: A Document Format for Expressing Privacy Preferences", H. Schulzrinne, J. Morris, H. Tschofenig, J. Cuellar, J. Polk, J. Rosenberg, February 2007,
URL: http://www.ietf.org/rfc/rfc4745.txt

[RFC4825] IETF RFC 4825 "The Extensible Markup Language (XML) Configuration Access protocol (XCAP)", J. Rosenberg, May 2007,
URL: http://www.ietf.org/rfc/rfc4825.txt

[RFC4826] IETF RFC 4826 "Extensible Markup Language (XML) Formats for Representing Resource Lists", J.

Rosenberg, May 2007,
URL: http://www.ietf.org/rfc/rfc4826.txt

[RFC4975] IETF RFC 4975 "The Message Session Relay Protocol (MSRP)", B. Campbell, R. Mahy, C. Jennings, September 2007,
URL: http://www.ietf.org/rfc/rfc4975.txt

[RFC5261] IETF RFC 5261 "An Extensible Markup Language (XML) Patch Operations Framework Utilizing XML Path Language (XPath) Selectors", J. Urpalainen, August 2008,
URL: http://www.ietf.org/rfc/rfc5261.txt

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

[RFC5874] IETF RFC 5874 "An Extensible Markup Language (XML) Document Format for Indicating a Change in XML Configuration Access Protocol (XCAP) Resources", J. Rosenberg, J. Urpalainen, May 2010,
URL: http://tools.ietf.org/rfc/rfc5874.txt

[RFC5875] IETF RFC 5875 "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Diff Event Package", J. Urpalainen, D. Willis, May 2010,
URL: http://tools.ietf.org/rfc/rfc5875.txt

3GPP/3GPP2

[3GPP2-S.S0086] 3GPP2 S.S0068-B "IMS Security Framework",
URL: http://3gpp2.org/Public_html/specs/index.cfm

[3GPP2-X.P0027-002] 3GPP2 X.P0027-002 "Presence Security",
URL: http://3gpp2.org/Public_html/specs/index.cfm

Note: Work in progress, awaiting IETF drafts

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

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

[3GPP-TS_23.003] 3GPP TS 23.003 "Numbering, addressing and identification",
URL: http://www.3gpp.org/ftp/Specs/archive/23_series/23.003/

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

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

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

[3GPP-TS_33.141] 3GPP TS 33.141 "Presence service; Security",
URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.141/

[3GPP-TS_33.210] 3GPP TS 33 210 "Network Domain Security; IP network layer security",
URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.210/

[3GPP-TS_33.222] 3GPP TS 33.222 "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)",
URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.222/

[3GPP-TR_33.978] 3GPP TR 33.978 "Security aspects of early IP Multimedia Subsystem (Release 6)",
URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.978/

2.2 Informative References

OMA

[IM_XDM] "IM XDM Specification", Draft Version 1.0, Open Mobile Alliance™, OMA-TS-IM_XDM-V1_0, URL: http://www.openmobilealliance.org/

[XDM_Profile] "Profile XDM Specification", Version 1.1, Open Mobile Alliance™, OMA-TS-XDM_Profile-V1_1, URL: http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “SHALL”, “SHALL 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

Access Permissions	Use definition from [XDM_RD]
Access Permissions Document	Use definition from [XDM_AD].
Active User Preferences Profile	Use definition from [XDM_RD].
Alias Principal	Use definition from [XDM_RD].
Application Server	A functional entity that implements the service logic for SIP sessions (e.g. PoC Server or IM Server).
Application Unique ID	A unique identifier within the namespace of Application Unique IDs that differentiates XDM Resources accessed by one application from XDM Resources accessed by another. (Source: [RFC4825]).
Application Usage	Detailed information on the interaction of an application with an XCAP Server. (Source: [RFC4825]).
Document Selector	A sequence of path segments, with each segment being separated by a “/”, that identify the XDM Document within an XCAP Root that is being selected. (Source: [RFC4825]).
Document Reference	Use definition from [XDM_AD].
Document URI	The HTTP URI containing the XCAP Root and Document Selector, resulting in the selection of a specific XDM Document. (Source: [RFC4825]).
Enabler	Use definition from [Dict].
Entity Tag	A unique identifier allowing comparison of two or more entities from the same requested resource. (Source: [RFC2616]).
Forward XDCP Request	An XDCP Request to perform an XDM Forward operation.
Forwarding Notification List Document	Use definition from [XDM_List].
Global Document	An XDM Document placed under the Global Tree that applies to all users of that Application Usage.
Global Tree	A URI that represents the parent for all Global Documents for a particular Application Usage within a particular XCAP Root. (Source: [RFC4825]).
Group	Use definition from [XDM_RD].
Group Document	Use definition from [XDM_RD].
Group Identity	Use definition from [XDM_RD].
Group Usage List	Use definition from [XDM_RD].
History Information	Use definition from [XDM_AD].
HTTP URI	An HTTP Request-URI as defined by [RFC2616].
Modification History Information	A subset of History Information containing information about XDM operations that has modified an XDM Document (i.e., successful create, modify and delete operations).
Modification History Information Document	An XDM Document containing Modification History Information.

Node Selector	A sequence of path segments, with each segment being separated by a "/", that identify the XML node (element or attribute) being selected within an XDM Document. (Source: [RFC4825]).
Node Selector Separator	A single path segment equal to two tilde characters "~~" that is used to separate the Document Selector from the Node Selector within an HTTP URI. (Source: [RFC4825]).
Node URI	The HTTP URI containing the XCAP Root, Document Selector, Node Selector Separator and Node Selector, resulting in the selection of a specific XML node. (Source: [RFC4825]).
Primary Principal	Use definition from [XDM_RD].
Principal	Use definition from [Dict].
Principal Identity Assertion	An assertion of the identity of the Principal served by the XDMC or the XDM Agent such as described in section 5.1.2 *"Principal Identity Assertion"*.
Public User Identity	Use definition from [3GPP-TS_23.003] section 3.14 "*Public User Identity"*.
Push Enabler	Use definition from [XDM_AD].
Push OTA Message	Any of the over-the-air messages of OMA "Push Over The Air" [Push_ERELD-V2_2].
Quality of Experience	Use definition from [XDM_RD].
Reference Point	Use definition from [Dict].
Remote Network	Use definition from [XDM_AD].
Request History Information	A subset of History Information containing information about requests to perform XDM operations related to an XDM Document.
Request History Information Document	An XDM Document containing Request History Information.
Request-URI	A part of the start line of a request using the SIP protocol as defined by [RFC3261].
Restore XDCP Request	An XDCP request to perform an XDM Restore operation.
Reverse Proxy	A web server system that is capable of serving web pages sourced from other web servers (AS), making these pages look like they originated at the Reverse Proxy. (Source: [3GPP-TS_33.222])
Search Document	An XML document include as part of a Search Request or as part of the response to a Search Request.
Search Request	A request to perform a search operation towards one or more XDM Resources.
Service Provider	Use definition from [Dict].
SIP NOTIFY	The SIP method NOTIFY as defined by [RFC3265].
SIP SUBSCRIBE	The SIP method SUBSCRIBE as defined by [RFC3265].
SIP URI	A communication resource as defined by [RFC3261].
Subscription Proxy	Use definition from [XDM_AD].
Supporting XDM Document	An XDM Document that contains the supporting metadata necessary to manage the User Directory and its content (e.g. Access Permissions Document, Modification History Information Document, Request History Information Document or XDM Preferences Document).
Tel URI	A globally unique identifier used to describe a resource identified by a telephone number as defined by [RFC3966].
Trusted Network	Use definition from [XDM_AD].
URI	A Uniform Resource Identifier as defined by [RFC3986].
URI List	Use definition from [XDM_RD].
URI List Document	Use definition from [XDM_RD].
User	Use definition from [Dict].
User Address	A User Address identifies a User. The User Address can be used by one User to request communication with other Users. If the SIP/IP Core is 3GPP IMS or 3GPP2 MMD realization, the User Address is a Public User Identity.

User Directory	A directory that represents all User Documents that belongs to a particular XCAP User Identity of a particular Users Tree.
User Directory Document Selector	A sequence of path segments, with each segment being separated by a "/", that identify a particular User Document within a particular User Directory(i.e., a User Directory Document Selector can be derived by removing the Users Tree and the XCAP User Identifier parts of a Document URI).
User Directory Folder Selector	A sequence of path segments, with each segment being separated by a "/", that identify a particular folder within a particular User Directory (i.e., a User Directory Folder Selector can be derived by removing the XDM Document name part of a User Directory Document Selector).
User Document	An XDM Document that belongs to a particular User identified by an XCAP User Identifier.
User Preferences Profile	Use definition from [XDM_RD].
User Profile	Use definition from [XDM_RD].
User Profile Document	Use definition from [XDM_RD].
Users Tree	A URI that represents the parent for all User Documents for a particular Application Usage within a particular XCAP Root.
XCAP Client	An HTTP client that understands how to follow the naming and validation constraints defined in [RFC4825]. (Source: [RFC4825]).
XCAP Root	A context that includes all of the XDM Documents across all Application Usages and users that are managed by a server. (Source: [RFC4825]) In this specification, the XCAP Root means all XDM Documents in all XDMSs accessible via the Aggregation Proxy.
XCAP Root URI	An HTTP URI that represents the XCAP Root. Although a valid URI, the XCAP Root URI does not correspond to an actual resource. (Source:[RFC4825]).
XCAP Server	An HTTP server that understands how to follow the naming and validation constraints defined in [RFC4825]. (Source: [RFC4825]).
XCAP URI	An HTTP URI that represents an XDM Resource.
XCAP User Identifier	The XUI is a string, valid as a path element in an HTTP URI, that is associated with each user served by the XCAP Server. (Source: [RFC4825]).
XDCP Document	An XML document included as part of an XDCP Request or as part of an XDCP Response.
XDCP Response	A response to an XDCP Request.
XDCP Request	A request to perform an XDCP operation.
XDM Agent	Use definition from [XDM_AD].
XDM Capabilities Document	An XDM Document as described by the Application Usage in section 5.3.1 "*XCAP Server Capabilities*".
XDM Directory Document	An XDM Document containing meta data about the XDM Documents in a User Directory.
XDM Document	Use definition from [XDM_RD].
XDM Document Part	Use definition from [XDM_RD].
XDM Preferences	A set of preferences that a User can set per Application Usage as described in section 5.8.
XDM Preferences Document	An XDM Document containing XDM Preferences.
XDM Resource	Use definition from [XDM_RD].
XDMC	Use definition from [XDM_AD].
XDMS	Use definition from [XDM_AD].

3.3 Abbreviations

ABNF	Augmented Backus-Naur Form
APD	Access Permissions Document

AS	Application Server
AUID	Application Unique ID
E-Tag	Entity Tag
GAA	Generic Authentication Architecture
HTTP	Hyper Text Transfer Protocol
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extension
MMD	MultiMedia Domain
OMA	Open Mobile Alliance
OMNA	OMA Naming Authority
OTA	Over-The-Air
SCR	Static Conformance Requirement
SIP	Session Initiation Protocol
TLS	Transport Layer Security
UE	User Equipment
URI	Uniform Resource Identifier
XCAP	XML Configuration Access Protocol
XDCP	XDM Command Protocol
XDM	XML Document Management
XDMC	XDM Client
XDMS	XDM Server
XML	Extensible Markup Language
XUI	XCAP User Identifier

4. Introduction

Various OMA Enablers such as, Presence, Push to talk Over Cellular (PoC), Instant Messaging (IM), etc. need support for access to and manipulation of certain information that are needed by these Enablers. Such information is expressed as XDM Documents and stored in various document repositories in the network where such XDM Documents can be located, accessed and manipulated (created, changed, deleted) by authorised Principals.

This specification defines the common protocol for access and manipulation of such XDM Documents by authorized Principals. This specification reuses the IETF XML Configuration Access Protocol (XCAP).

Common, reusable as well as Enabler-specific XDM Document formats and associated XCAP Application Usages are described in separate specifications that make use of the XCAP protocol specified in this specification for their document management.

4.1 Version 1.1

The version 1.1 specifies the core functionality of the XDM Enabler:

- a convention for describing elements and attributes of an XDM Document as an HTTP resource, i.e., accessible via an HTTP URI;
- a technique for using HTTP GET, PUT and DELETE methods for various XDM Resource manipulation operations (e.g., retrieving/adding/deleting elements/attributes, etc.);
- the concept and structure of an Application Usage by which XDM Documents can be described; and
- a default authorization policy for accessing and manipulating XDM Resources.

4.2 Version 2.0

The version 2.0 includes the functionality of version 1.1 and in addition specifies:

- how changes to XDM Documents can be conveyed to XDMCs by reusing an IETF-defined SIP event package;
- basic requirements for handling XDM Documents in remote domains; and
- a way to facilitate finding certain information by limited search capabilities,

4.3 Version 2.1

The version 2.1 includes the functionality of version 2.0 and in addition specifies:

- an HTTP POST based protocol, XDCP, enabling additional XDM Resource operations:
  - XDM Document Reference
  - XDM Forward
  - XDM Differential read and write
  - Subscription to XDM Document Changes
  - Restore;
- Support for Alias Principal;
- Notification of changes in XDM Documents using Push Enabler;
- Access Permissions to XDM Resources;
- XDM Document history and restore; and

- User preference profiles.

All “Shared xyz XDMS”s defined by the XDM Enabler are renamed to “xyz XDMS”.

NOTE: XDM V2.1 inherits the Shared List XDMS, Shared Group XDMS, Shared Profile XDMS and Shared Policy XDMS from XDM V2.0, but renames them as the List XDMS, Group XDMS, Profile XDMS and Policy XDMS.

5. Common Procedures

5.1 Security Procedures

5.1.1 Authentication

The XDM-3 and XDM-5 Reference Points between the XDMC and the Aggregation Proxy (see [XDM_AD]) SHALL provide mutual authentication.

For a 3GPP IMS or 3GPP2 MMD realisation, the XDM-3 and XDM-5 Reference Points correspond to the Ut Reference Point. In this case the authentication between the XDMC and the Aggregation Proxy SHALL be performed according to [3GPP-TS_33.141] / [3GPP2-X.P0027-002].

If the Generic Authentication Architecture (GAA) as defined in [3GPP-TS_33.222] is not used, the XDMC and the Aggregation Proxy SHALL support the HTTP Digest mechanism for client authentication and MAY support early IMS authentication according to [3GPP-TR_33.978] section 6.3. If the Aggregation Proxy determines to apply early IMS authentication, the X-3GPP-Intended-Identity header is missing from the XCAP request and the request is to the "users" tree, then the Aggregation Proxy MAY extract the identity of the Principal from the XUI of the Request-URI for authentication.

The HTTP Digest authentication by this specification SHALL conform to [RFC2617] with the following clarifications:

1) The HTTP "401 Unauthorized" error response SHALL be used;

2) The "rspauth" parameter MAY be used to provide mutual authentication;

3) The "username" parameter SHALL have the value of the XUI (i.e. the SIP URI or Tel URI) identifying the Principal (the Public User Identity).

NOTE: The "username" can be a part of the Device Provisioning parameters (see Appendix D). When using such provisioned "username" the XDMC must use it exactly as provisioned.

The XDMC and the Aggregation Proxy SHALL support HTTP over Transport Layer Security (TLS) as specified in [RFC2818] for server authentication over the XDM-3 and XDM-5 Reference Points.

An HTTP "403 Forbidden" error response SHALL be sent to the XDMC after one or more failed responses to a challenge. The exact count of challenges is decided by local policy.

5.1.2 Principal Identity Assertion

The XDMC or XDM Agent SHALL provide the identity of the Principal.

This identity SHALL be asserted and safely shared within the XDMS infrastructure.

5.1.2.1 XDMC

For all requests from an XDMC, the Aggregation Proxy SHALL provide Principal Identity Assertion of the identity of the requesting Principal that has been successfully authenticated as described below:

When the 3GPP/3GPP2 GAA is not present the Aggregation Proxy:

1) SHALL insert the X-3GPP-Asserted-Identity header, as described in [3GPP-TS_24.109] Appendix G, to the HTTP requests after a successful HTTP Digest Authentication.

2) SHALL populate the X-3GPP-Asserted-Identity header with the SIP URI in quotation marks ("") provided by the "username" field in the HTTP Digest Authorization header.

3) SHOULD add the Tel URI as a value of the X-3GPP-Asserted-Identity header, if the Principal has a Tel URI associated with the SIP URI (the term 'associated' means that the Tel URI can be translated to the SIP URI and vice versa, for interchangeable usage in the SIP/IP Core).

When realized with 3GPP IMS or 3GPP2 MMD networks and the GAA is present or in case of an early IMS deployment as defined in [3GPP-TR_33.978], the procedures described [3GPP-TS_24.109] SHALL be followed with the following clarifications:

1) The XDMC MAY insert the X-3GPP-Intended-Identity header as defined in [3GPP-TS_24.109] to the HTTP requests to deliver the Principal's preferred identity for Principal Identity Assertion.

2) The Aggregation Proxy

   a) SHALL act as an Authentication Proxy defined in [3GPP TS 24.109].

   b) SHALL check whether an identity of the Principal has been inserted in the X-3GPP-Intended-Identity header of HTTP request;

      i. If the X-3GPP-Intended-Identity header is included, the Aggregation Proxy SHALL check if the value in the header is allowed to be used by the authenticated identity.

         - If it is allowed, the Aggregation Proxy SHALL insert a X-3GPP-Asserted-Identity header and populate it with the value of the X-3GPP-Intended-Identity header. The Aggregation Proxy MAY remove the X-3GPP-Intended-Identity header.

         - If not, the Aggregation Proxy SHALL send an HTTP “401 Unauthorized” error response.

      ii. If the X-3GPP-Intended-Identity header is not included, the Aggregation Proxy SHALL insert the authenticated identity in the X-3GPP-Asserted-Identity header of the HTTP request.

   c) SHOULD add the Tel URI as a value of the X-3GPP-Asserted-Identity header, if the Principal has a Tel URI associated with the SIP URI (the term 'associated' means that the Tel URI can be translated to the SIP URI and vice versa, for interchangeable usage in the SIP/IP Core).

NOTE 1: How the Aggregation Proxy determines the associated URIs is outside the scope of this specification.

NOTE 2: How the Aggregation Proxy checks if the value of the X-3GPP-Intented-Identity is allowed to be used is outside the scope of this specification.

The SIP/IP Core SHALL provide Principal Identity Assertion. When realized with 3GPP IMS or 3GPP2 MMD networks, the XDMC MAY use P-Preferred-Identity SIP header to deliver the Principal's preferred identity for Principal Identity Assertion and the Privacy SIP header to set its privacy preference, and the SIP/IP Core SHALL use P-Asserted-Identity SIP header to carry the asserted identity of the Principal within trusted networks, as described in [3GPP-TS_24.229]/[3GPP2-X.S0013-004].

5.1.2.2 XDM Agent

Within Trusted Networks the XDM Agent SHALL provide the Principal Identity Assertion when it generates an HTTP request to XDMS on behalf of a Principal. In this case, the XDM Agent SHALL use the X-3GPP-Asserted-Identity HTTP header (see [3GPP-TS_24.109] Appendix G) to carry the identity of the Principal for whom it generates the HTTP request.

If the Principal has both a Tel URI and its associated SIP URI (the term 'associated' means that the Tel URI can be translated to the SIP URI and vice versa, for interchangeable usage in the SIP/IP Core), then the XDM Agent SHOULD use both URIs as values of the X-3GPP-Asserted-Identity header.

NOTE: How the XDM Agent determines the associated URIs is outside the scope of this specification.

5.1.3 Principal Identity Sharing

The identity of the Principal which has been authenticated SHALL be shared, through Principal Identity Assertion, on the following Reference Points (see [XDM_AD]) within Trusted Networks:

1) the XDM-4 Reference Point between the Aggregation Proxy and the XDMSs;
2) the XDM-6 Reference Point between the Aggregation Proxy and the Search Proxy;
3) the XDM-7 Reference Point between the Search Proxy and the XDMSs;
4) the XDM-8 Reference Point between the Aggregation Proxy and the Cross-Network Proxy;
5) the XDM-9 Reference Point between the Search Proxy and the Cross-Network Proxy;
6) the XDM-13 Reference Point between the XDM Agent and the Search Proxy;
7) the XDM-14 Reference Point between the XDM Agent and the XDMSs;
8) the NNI-1 Reference Point between the Cross-Network Proxy and the Remote Network in case of a trusted Remote Network.

Further details of the security mechanisms for the above listed Reference Points are out of scope of this specification.

For a 3GPP/3GPP2 realization, the above listed Reference Points SHALL use the security mechanisms as defined in the corresponding 3GPP/3GPP2 specifications.

5.1.4 Integrity and Confidentiality Protection

The integrity and confidentiality protection for XCAP/HTTP traffics SHALL be provided on the following Reference Points (see [XDM_AD]):

1) the XDM-3 Reference Point between the XDMC and the Aggregation Proxy;
2) the XDM-5 Reference Point between the XDMC and the Aggregation Proxy;
3) the NNI-1 Reference Point between the Cross-Network Proxy and the Cross-Network Proxy of Remote Network.

The TLS SHALL be supported as specified in [RFC2246] with the following clarifications:
TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher suite SHALL be supported; other cipher suites defined in [RFC2246] MAY be supported.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD networks, the XDMC and the Aggregation Proxy SHALL support the TLS version and profile as specified in clause 5.3 of [3GPP-TS_33.222].

The XDM-3 and XDM-5 Reference Points SHALL protect HTTP requests by enabling TLS transport security mechanism. The TLS resumption procedure SHALL be used as specified in [RFC2818].

The NNI-1 Reference Point SHALL protect HTTP traffic by enabling the TLS transport security mechanism or other inter-network domain security mechanism. When using TLS, the TLS resumption procedure SHALL be used as specified in [RFC2818]. When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the protection of the HTTP traffic between trusted domains MAY be implemented using Network Domain Security as defined in [3GPP-TS_33.210] and [3GPP2-S.S0086] respectively.

The integrity and confidentiality protection for SIP traffic SHALL be provided per the underlying SIP/IP Core.

5.1.5 Authorization

For the authorization of HTTP requests, the XDMS SHALL check that the identity provided through Principal Identity Assertion has been granted access rights to perform the requested operations: The XDMS SHALL use the information in the X-3GPP-Asserted-Identity HTTP header (see [3GPP-TS_24.109] Appendix G) provided by the Aggregation Proxy, the

XDM Agent or the Remote Network to determine the identity of the Principal. For backwards compatibility reason, the XDMS SHALL also be able to obtain the identity of the requesting Principal from the X-XCAP-Asserted-Identity (see [XDM_Core-V2_0] Appendix E) or the X-3GPP-Intended-Identity HTTP (see [3GPP-TS_24.109] Appendix G) headers.

For the authorization of SIP requests, when the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the XDMS SHALL use the identity information in P-Asserted-Identity SIP header as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] to authorize the identity provided by the requesting XDMC or XDM Agent.

For XDM Resources in the Users Tree, Application Usages MAY define their own authorization policies. Such authorization policy MAY make use of the Access Permissions Document described in sections 5.6 and 6.2.4. In the absence of an Application Usage specific authorization policy, the default SHALL be as follows:

1) The Primary Principal, Admin Principal and associated Alias Principals SHALL have permission to perform all operations defined in sections 6.1.1.*"Document Management"* and 6.1.2 *"Subscribing to changes in the XDM Resources*"; and

2) Principals other than the Primary Principal, Admin Principal and associated Alias Principals SHALL NOT have permissions to perform operations defined in sections 6.1.1*"Document Management"* and 6.1.2 *"Subscribing to changes in the XDM Resources"*.

NOTE: Local policy may allow trusted applications to be granted some or all of the permissions defined in sections 6.1.1*"Document Management"* and 6.1.2, *"Subscribing to changes in the XDM Resources"*.

For XDM Resources in the Global Tree, Application Usages defining the use of Global Documents SHALL specify the authorization policies associated with their use.

An HTTP "403 Forbidden" error response SHALL be sent to the XDMC or the XDM Agent if the HTTP request by the XDMC or the XDM Agent fails to get authorized by the XDMS per the authorization policy defined by the target Application Usage.

5.2 Common Extensions

5.2.1 URI Lists defined in List XDMS

Various Application Usages may wish to refer to URI Lists stored in the List XDMS (see [XDM_List]). The <external> element provides the means to make such references, in a similar manner across different Application Usages.

The attribute "anchor" of the <external> element SHALL contain a Node URI pointing to a <list> element within a "resource-lists" XDM Document in the List XDMS.

The value of the attribute "anchor" SHALL be percent-encoded as defined by the procedures in [RFC4825] section 6 before it is inserted into an XDM Document.

NOTE: There is an <external-list> condition element defined in section 5.2.2. It points to e.g. URI Lists in the List XDMS [XDM_List], against which the authorization rules are specified according to [RFC4745].

Application Usages that utilize the <external> element SHALL resolve a Node URI only to URIs within the specific <list> element that is pointed to.

In order to avoid circular referencing when resolving a URI List, an <external> element that has already been resolved SHALL be ignored.

5.2.2 Authorization Rules

Authorization rules (also called authorization policies) are based on the common policy framework described in [RFC4745], and extended by OMA defined common extensions in order to meet some additional requirements of OMA applications. These include the need to:

- reference identities in external URI Lists, which is an explicit non-goal of [RFC4745];
- enable the user to define a default rule that applies in the absence of any other matching rule;
- allow rules to be matched based on hierarchical precedence assigned to the different types of allowed conditions, prior to combining permissions;
- constrain, for predictability in UE design and end user expectation, the conditions in a rule to no more than a single expression or set of expressions.

NOTE 1: Individual Enablers may also define extensions to [RFC4745] to meet application-specific needs. Such extensions must not change or cause to change the semantics of the common extensions defined in section 5.2.2.1 or the evaluation algorithm for combining permissions defined in section 5.2.2.4.

NOTE 2: An authorization policy using the extensions defined in this section must declare the "urn:ietf:params:xml:ns:common-policy", "urn:oma:xml:xdm:common-policy", "urn:oma:xml:xdm:extensions" and "urn:oma:xml:xdm:2.1extensions" namespace names in the XML schema.

5.2.2.1 Structure

The <conditions> element within a rule in an authorization policy:

1) MAY include the <identity> condition element as defined in [RFC4745];
2) MAY include the <external-list> condition element as described in section 5.2.2.2 and [XSD_commPol];
3) MAY include the <anonymous-request> condition element as described in section 5.2.2.2 and [XSD_commPol];
4) MAY include the <other-identity> condition element as described in section 5.2.2.2 and [XSD_commPol];
5) MAY include the <media-list> condition element as described in section 5.2.2.2 and [XSD_ext];
6) MAY include the <service-list> condition element as described in section 5.2.2.2 and [XSD_ext];
7) MAY include the <validity> condition element as defined in [RFC4745];
8) MAY include the <invited-identities> condition element as described in section 5.2.2.2 and [XSD_ext_2_1];
9) MAY include the <activities> condition element as described in section 5.2.2.2 and [RFC4480];
10) MAY include the <qoe-list> condition element as described in section 5.2.2.2 and [XSD_ext_2_1];
11) MAY include the <country-region-list> condition element as described in section 5.2.2.2 and [XSD_ext_2_1];
12) MAY include the <location-list> condition element as described in section 5.2.2.2 and [XSD_ext_2_1];
13) MAY include the <upp-list> condition element as described in section 5.2.2.2 and [XSD_ext_2_1];
14) MAY include the <expired> element as described in section 5.2.2.2 and [XSD_ext_2_1]; and
15) MAY include the <deferred-messages> element as described in section 5.2.2.2 and [XSD_ext_2_1].

NOTE: According to [RFC4745], a rule is applicable to a request only if all <conditions> child elements of the rule evaluate to TRUE. Therefore, if a rule contains a <conditions> child element from a namespace that the Application Server does not understand or support, then that rule is not applicable.

The <conditions> element of a rule SHALL contain no more than one of the <identity>, <external-list>, <anonymous-request> or <other-identity> elements, but it MAY contain other elements (e.g. a <media-list> element and a <service-list> element).

The <external-list> element MAY include the <entry> element. If present, the <entry> element SHALL include the "anc" attribute, whose value SHALL be percent-encoded as defined by [RFC4825] section 6 before it is inserted into an XDM Document.

The <media-list> element SHALL include one of:

1) an <all-media-except> element as described in section 5.2.2.2; or

2) a list of one or more media elements selected from the list of possible media elements below.

List of possible media elements:

1) The <message-session> media element indicating session based messaging as defined in [RFC4975];

2) The <pager-mode-message> media element indicating pager mode message requests as defined in [RFC3428];

3) The <file-transfer> media element indicating file transfer as defined in [IM_TS];

4) The <audio> media element indicating a streaming media type as defined in [RFC3840];

5) The <video> media element indicating a streaming media type as defined in [RFC3840];

6) The <poc-speech> media element indicating a PoC speech media type as defined in [PoC_CP];

7) The <group-advertisement> media element indicating a group advertisement as defined in [XDM_Group] section "*Extended Group Advertisements*";

8) The <media-streaming-control> media element indicating a PoC media streaming control media type as defined in [PoC_CP];

9) The <text> media element indicating a discrete media type as defined in [RFC2046];

10) The <image> media element indicating a discrete media type as defined in [RFC2046];

11) The <binary-data> media element indicating a discrete media type of arbitrary binary data (e.g., as defined in [RFC2046]);

12) The <standalone-message> media element indicating a standalone message as defined in [CPM_RD]; and

13) Any elements from any other namespaces defining a media element.

The <all-media-except> element MAY include a list of one or more media elements selected from the list of possible media elements above.

The <audio>, <video> and <message-session> elements:

1) MAY include the <full-duplex> element as described in section 5.2.2.2 indicating that media can be exchanged in both directions simultaneously; or

2) MAY include the <half-duplex> element as described in section 5.2.2.2 indicating that media can be exchanged in only one direction at a time; or

3) MAY include any element from any other namespace for the purpose of extensibility; and

4) MAY include attributes from any other namespaces for the purpose of extensibility.

The <pager-mode-message>, <file-transfer>, <poc-speech>, <group-advertisement>, <text>, <image>, <binary-data> and <standalone-message> elements:

1) MAY include any element from any other namespace for the purpose of extensibility; and

2) MAY include attributes from any other namespaces for the purpose of extensibility.

The <service-list> element SHALL include one of:

1) a list of one or more <service> elements or any elements from any other namespaces for the purpose of extensibility; or

2) an <all-services-except> element.

The <all-services-except> element MAY include a list of one or more <service> elements or any elements from any other namespaces for the purpose of extensibility.

The <service> element:

1) MAY include an attribute "enabler" including a string defining an Enabler ;

2) MAY include attributes from any other namespaces for the purpose of extensibility to have other service identification definitions than using the "enabler" attribute; and

3) MAY include any other elements from any other namespaces for the purpose of extensibility to have other service identification definitions than using attributes.

The <invited-identities> element SHALL include the same elements and sub-elements as defined for the <identity> element in [RFC4745].

The <activities> element SHALL include a list of one or more activity-describing elements as defined in [RFC4480].

The <qoe-list> element SHALL include a list of one or more <qoe> elements indicating a Quality of Experience profile as defined in [XDM_Group].

The <country-region-list> element SHALL include a list of one or more <civicAddress> elements as defined in [RFC4119] but restricted to include only a subset of the subelements defined in [RFC4119].

The <civicAddress> element:

1) MAY include a <country> element as specified in [RFC4119];

2) MAY include a <A1> element as specified in [RFC4119];

3) MAY include a <A2> element as specified in [RFC4119].

The <location-list> element SHALL include a list of one or more <location-info> elements as defined in [RFC4119].

The <upp-list> element SHALL include one of:

1) a list of one or more <upp> elements as described in section 5.2.2.2; or

2) an <all-upp-except> element as described in section 5.2.2.2.

The <all-upp-except> element MAY include a list of zero or more <upp> elements.

The <upp> element SHALL include an attribute "upp-id" including a User Preferences Profile identifier as described in section "*UPP Directory*" in [XDM_UPPD].

5.2.2.2 Data Semantics

If present in any rule, the <external-list> element SHALL match those identities of the Principals that are contained in a URI List [XDM_List] or contained in an XDM Document Part as described by the Application Usage defining an authorization policy according to this section. The "anc" attribute of the <entry> elements of an <external-list> element SHALL contain a Node URI or a Node URI without the XCAP Root URI part pointing to an XDM Document Part containing an element with a list of identities in the form of URIs. A URI in this list MAY be an identity of a Principal (e.g. a Tel URI) or a reference to another list of identities (e.g. an element containing a Node URI to a URI List). The "anc" attribute of the <entry> elements of an <external-list> element SHALL therefore be resolved in such way that already resolved URIs are ignored to avoid circular referencing (i.e. a resolving process that never ends).

If present in any rule, the <anonymous-request> element SHALL match those incoming requests that have been identified as anonymous.

NOTE 1: In certain cases, the <identity> condition can also match anonymous requests. For example, the <many/> child element of the <identity> condition matches any authenticated identity, either anonymous or not. However, any rules matching the <anonymous-request> condition would have precedence as described in section 5.2.2.4 "*Combining Permissions*".

When the SIP/IP Core corresponds to 3GPP IMS or 3GPP2 MMD, an AS SHALL use the procedures as defined in section 5.7.1.4 in [3GPP-TS_24.229]/[3GPP2-X.S0013-004] to identify the source of the anonymous request.

NOTE 2: If the authorization policy document includes a rule having an <anonymous-request> condition element, an XDMC should not specify another rule containing an <identity> condition element with a <many/> child element and the same <actions> and/or <transformations> element(s) as the rule with the <anonymous-request> condition element.

If present in any rule, the <other-identity> element, which is empty, SHALL match all identities that are not referenced in any rule. It allows for specifying a default policy.

If present in any rule, the <media-list> element SHALL match incoming requests associated with particular media types. A <media-list> element with a list of media elements SHALL be used to specify allowed media types. A <media-list> element with an <all-media-except> element SHALL be used to specify that all media types are allowed apart from those listed as child elements. The <media-list> condition SHALL be considered TRUE if any of its child media elements evaluate to TRUE, i.e., the results of the individual child elements are combined using a logical OR. The <media-list> condition SHALL also be considered TRUE if all of the child media elements to an <all-media-except> element evaluate to FALSE.

If neither a <full-duplex> nor <half-duplex> duplex specific sub element is included, it means that the access rule is applicable to both cases (i.e. half-duplex and full-duplex).

If a child element of a media element is not known or not supported, the child element SHALL be ignored and evaluated as FALSE.

NOTE 3: How the AS determines the media type of the incoming request (i.e. in order to evaluate if a match exists for a rule containing the <media-list> condition) must be specified by the individual Enabler.

If present in any rule, the <service-list> element SHALL match incoming requests associated with a particular service. A <service-list> element with a list of <service> element SHALL be used to specify allowed services. A <service-list> element with an <all-services-except> element SHALL be used to specify that all services are allowed apart from those listed as child elements. A <service-list> element with an <all-service-except> element without any child element SHALL be used to specify that all services are allowed. The <service-list> element SHALL be evaluated to TRUE if one of its child <service> elements evaluates to TRUE. The <service-list> element SHALL also be evaluated to TRUE if all of the child <service> elements to an <all-services-except> element evaluate to FALSE. The <service-list> element SHALL be evaluated to TRUE if it contains an <all-services-except> element without any child elements.

The <service> element SHALL be used to define a service.
The attribute "enabler" SHALL specify the Enabler defining the service. The "enabler" attribute SHALL be used only for OMA defined Enablers. The Enabler SHALL use the OMNA registered Enabler XML schema domain token as the value of the "enabler" attribute (e.g. "poc" for the Push to talk over Cellular Enabler and "im" for the IM SIMPLE Enabler).

NOTE 4: Usage of the <service> element outside OMA can be done by extending the <service> element.

The individual Enabler specifies how an Application Server can use the information in an incoming request to recognize a request for a service. A <service> element SHALL be evaluated to TRUE if the incoming request to the Application Server contains the information defined and to FALSE if not.

If present in any rule, the <invited-identities> element SHALL match identities of Users being invited to communications sessions.

If present in any rule, the <activities> element SHALL match invited identities with particular activities.

If present in any rule, the <qoe-list> element SHALL match incoming requests associated with particular Quality of Experience profiles.

If present in any rule, the <country-region-list> element SHALL match invited identities with particular countries or regions of their home network.

If present in any rule, the <location-list> element SHALL match either inviting or invited identities with particular geographical locations.

If present in any rule, the value of "upp-id" attribute of <upp> element SHALL match any of the User Preference Profiles as described in section "*UPP Directory*" in [XDM_UPPD] related to the Primary Principal of the authorization policy document where the element is included.

If present in any rule, the <all-upp-except> element, when empty, SHALL match all of the User Preferences Profiles as described in section "*UPP Directory*" in [XDM_UPPD].

If present in any rule, the <expired> element SHALL match incoming communication requests that have expired.

If present in any rule, the <deferred-messages> element SHALL match deferred messages.

5.2.2.3 XML Schema

The authorization policy document SHALL conform to the XML schema described in [RFC4745] section 13 with the extensions described in [XSD_commPol], in [XSD_ext], [XSD_ext_2_1] and in an Application Usage defined extensions to [RFC4745].

5.2.2.4 Combining Permissions

When evaluating any authorization policy document based on [RFC4745] together with the extensions described in section 5.2.2.1 against a URI value, the algorithm for obtaining the different rules that are applicable SHALL be as follows:

1) Those rules matching the URI value against the <anonymous-request> element SHALL take precedence over those rules based on matching it against an <identity> element. That is, if there are applicable rules based on <anonymous-request> matches, only those will be used for the evaluation of the combined permission.

2) Those rules matching the URI value against the <identity> element SHALL take precedence over those rules based on matching it against an <external-list> or an <other-identity> element. That is, if there are applicable rules based on <identity> matches, only those will be used for the evaluation of the combined permission.

3) Those rules containing an <other-identity> element SHALL be used for the evaluation of the combined permission only if there are no other matching rules.

NOTE**:** The above algorithm for obtaining all the applicable rules differs from that described in [RFC4745].

After the applicable rules have been derived based on the above algorithm, the evaluation of the combined permission SHALL be based on [RFC4745] section 10.2.

5.2.3 Detailed Conflict Reports

Detailed conflict reports provide the means to indicate the possible cause of a validation error. They are based on the definition specified in [RFC4825], and extended by OMA defined common extensions in order to handle violations of constraints defined by local policy appropriately.

The XDMC and the XDM Agent SHALL support the types of <error-element> defined in [RFC4825] and this section. Other types of <error-element> elements MAY be ignored by the XDMC or the XDM Agent. It is thus RECOMMENDED that the XDMS does not use other types of <error-element> elements than those defined in [RFC4825] and this section.

5.2.3.1 Structure

The <extension> element defined in the xcap-error namespace in [RFC4825] MAY include the <local-constraint-failure> error element.

The <local-constraint-failure> SHALL be used when a constraint is violated that is defined by the local policy.

The <local-constraint-failure> element:

1) MAY include the "phrase" attribute;
2) MAY include one or more <alt-value> elements with the mandatory "field" attribute, providing one or more alternate values for the element or attribute indicated by the "field" attribute;
3) MAY include one or more <description> elements with an optional "lang" attribute, providing one or more descriptions documenting the local constraint failure, possibly in different languages.

The <local-constraint-failure> SHALL NOT be used when a constraint is violated that is defined by the Application Usage. The <constraint-failure>, as defined in [RFC4825], SHALL be used for this, unless specified otherwise by the Application Usage.

When the <local-constraint-failure> contains one or more <alt-value> elements, the XDMC or the XDM Agent MAY repeat the XCAP request in which the indicated field SHOULD be assigned one of the proposed values.

5.2.3.2 XML Schema

The <local-constraint-failure> element SHALL conform to the XML schema described in [XSD_xcapErr].

5.3 Common Application Usages

5.3.1 XCAP Server Capabilities

The XCAP Server Capabilities Application Usage allows an XDMC or XDM Agent to retrieve an XDM Capabilities Document to determine what extensions, Application Usages, namespaces, XDM Documents and XDM requests are supported by the Service Provider before making an XDM request targeting e.g. a particular Application Usage.

The XDM Capabilities Document SHALL conform to what is specified in the section.

5.3.1.1 Structure

The structure of an XDM Capabilities Document SHALL conform to the structure of XDM Document as defined in [RFC4825] with extensions given in this section.

The <xcap-caps> element MAY include a new child element <xdm-caps> as described in [XSD_ext_2_1] and in section 5.3.1.7.

The <xdm-caps> element:

1) MAY include a sequence of <au> elements as described in [XSD_ext_2_1] and in section 5.3.1.7;
2) MAY include a <subscribe> element as described in [XSD_ext_2_1] and in section 5.3.1.7; and
3) MAY include other elements from other namespaces for the purposes of extensibility.

The <au> element:

1) SHALL include an "auid" attribute as described in [XSD_ext_2_1] and in section 5.3.1.7;
2) MAY include a <requests> element as described in [XSD_ext_2_1] and in section 5.3.1.7;
3) MAY include a <docs> element as described in [XSD_ext_2_1].and in section 5.3.1.7; and

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

The <requests> element:

1) MAY include a <diff-read> element as described in [XSD_ext_2_1] and in section 5.3.1.7;
2) MAY include a <diff-write> element as described in [XSD_ext_2_1] and in section 5.3.1.7;
3) MAY include a <reference> element as described in [XSD_ext_2_1] and in section 5.3.1.7;
4) MAY include a <forward> element as described in [XSD_ext_2_1] and in section 5.3.1.7;
5) MAY include a <restore> element as described in [XSD_ext_2_1] and in section 5.3.1.7;
6) MAY include a <search> element as described in [XSD_ext_2_1] and in section 5.3.1.7;
7) MAY include a <subscribe> element as described in [XSD_ext_2_1] and in section 5.3.1.7; and
8) MAY include other elements from other namespaces for the purposes of extensibility.

The <docs> element:

1) MAY include an <apd> element as described in [XSD_ext_2_1] and in section 5.3.1.7;
2) MAY include a <mod-hist> element as described in [XSD_ext_2_1] and in section 5.3.1.7;
3) MAY include a <prefs> element as described in [XSD_ext_2_1] and in section 5.3.1.7;
4) MAY include a <request-hist> element as described in [XSD_ext_2_1] and in section 5.3.1.7; and
5) MAY include other elements from other namespaces for the purposes of extensibility.

The <subscribe> element:

1) MAY include a <sip> element as described in [XSD_ext_2_1] and in section 5.3.1.7; and
2) MAY include an <xdcp> element as described in [XSD_ext_2_1] and in section 5.3.1.7.

5.3.1.2 Application Unique ID

The default namespace SHALL be as defined in [RFC4825] section 12.1 "*Application Unique ID (AUID)*".

5.3.1.3 XML Schema

The XDM Capabilities Document SHALL conform to the XML Schema define in [RFC4825] section 12.2 "*XML Schema*" and to the XML Schema defined in [XSD_ext_2_1].

5.3.1.4 Default Namespace

The default namespace SHALL be as defined in [RFC4825] section 12.3 "*Default Document Namespace*".

5.3.1.5 MIME Type

The MIME type of the XDM Capabilities Document SHALL be as defined in [RFC4825] section 12.4 "*MIME Type*".

5.3.1.6 Validation Constraints

There are no additional validation constraints associated with this application usage.

5.3.1.7 Data Semantics

The data semantics for the elements defined in [RFC4825] SHALL be as defined in [RFC4825] section 12 "*XCAP Server Capabilities*".

The <xdm-caps> element defines which XDM capabilities the Server Provider supports. The <au> element defines the capabilities that a particular Application Usage supports. The "auid" attribute SHALL contain the AUID specified for the Application Usage.

The <requests> element defines which optional XDM requests the Application Usage supports.

The <forward> element defines that the Application Usage supports XDM Resource forwarding as described in section 6.2.6.2.

The <restore> element defines that the Application Usage supports XDM restore as described in section 6.2.6.5.

The <reference> element defines that the Application Usage supports Document Reference as described in section 6.2.6.1.

The <search> element defines that the Application Usage supports search as described in section 6.2.3.

The <subscribe> element defines that the Application Usage supports subscribe to changes as described in section 6.2.2.

The <diff-read> element defines that the Application Usage supports Differential Read as described in section 6.2.6.3.

The <diff-write> element defines that the Application Usage supports Differential Write as described in section 6.2.6.4.

The <docs> element defines which optional XDM Documents the Application Usage supports.

The <apd> element defines that the Application Usage supports the Access Permissions Document as described in section 5.6.

The <prefs> element defines that the Application Usage supports the XDM Preferences Document as described in section 5.8.

The <request-hist> element defines that the Application Usage supports the Request History Information Document as described in section 5.7.2.

The <mod-hist> element defines that the Application Usage supports the Modification History Document as described in section 5.7.1.

The <subscribe> element defines subscribe for changes capabilities of Server Provider's Subscription Proxy.

The <sip> element defines that the Subscription Proxy supports subscribe to changes using a SIP SUBSCRIBE request as described in section 6.6.2.

The <xdcp> element defines that the Subscription Proxy supports subscribe to changes using an XDCP Request as described in section 6.6.1.

5.3.1.8 Naming Conventions

The XDM Capabilities Document shall have the name as defined in [RFC4825] section 12.7 "*Naming Conventions*".

5.3.1.9 Global Document

This Application Usage defines one Global Document and no XDM Documents in the Users Tree.

5.3.1.10 Resource Interdependencies

As defined in [RFC4825] section 12.8 "*Resource Interdependencies*"

5.3.1.11 Authorization Policies

All authenticated Principals SHALL have access to the XDM Capabilities Document.

5.3.1.12 Subscription to changes

Not applicable.

5.3.1.13 Search Capabilities

Not applicable.

5.3.1.14 XDM Preferences Document

Not applicable.

5.3.1.15 History Information Documents

Not applicable.

5.3.1.16 Forwarding

Not applicable.

5.3.1.17 Restore

Not applicable.

5.3.1.18 Document Reference

Not applicable.

5.3.1.19 Differential Read and Write

Not applicable.

5.3.2 XML Documents Directory

The XML Documents Directory Application Usage allows an XDMC and an XDM Agent (corresponding to a given XUI) to fetch:

1) the list of all XCAP managed XDM Documents corresponding to that XUI across all Application Usages, or
2) the list of all XDM Documents for a given AUID corresponding to that XUI stored for that AUID.

5.3.2.1 Structure

The structure of an XDM Directory Document SHALL be as follows:

It is a well-formed and valid XML document encoded in UTF-8 that begins with the root element <xcap-directory>.

The <xcap-directory> element SHALL include one or more <folder> elements.

The <folder> element:

1) SHALL include an “auid” attribute, whose value corresponds to an AUID and for which there are XDM Documents in the User Directory corresponding to a given XUI; and
2) SHALL include either one or more <entry> elements or an <error-code> element.

The <entry> element:

1) SHALL include a “uri” attribute;
2) SHALL include an “etag” attribute;
3) MAY include a “last-modified”attribute;

4) MAY include a “size” attribute;

5) MAY include a “reference” attribute; and

6) MAY include a “reference-display-name” attribute.

The character escaping SHALL be applied in HTTP URI representation according to [RFC4825] section 6.3.

5.3.2.2 Application Unique ID

The AUID SHALL be “org.openmobilealliance.xcap-directory”.

5.3.2.3 MIME Type

The MIME type for the XDM Directory Document SHALL be “application/vnd.oma.xcap-directory+xml”.

5.3.2.4 Default Namespace

The default namespace SHALL be “urn:oma:xml:xdm:xcap-directory”.

5.3.2.5 XML Schema

The XDM directory XDM Document SHALL conform to the XML schema described in [XSD_xcapDir] and in [XSD_xcapDirExt].

5.3.2.6 Additional Constraints

None.

5.3.2.7 Data Semantics

The “uri” attribute SHALL contain the Document URI or Document Selector for an XDM Document corresponding to the “auid” attribute value in the parent <folder> element and for the given XUI. The character escaping SHALL be applied in HTTP URI representation according to [RFC4825] section 6.3.

The “etag” attribute SHALL contain either the server computed E-Tag value of the current instance of the XDM Document identified by the “uri” attribute value or if the value of the “uri” attribute identifies an XDM Document Reference, the E-Tag value of the referenced XDM Document. (This allows the XDMC or an XDM Agent to determine whether the locally cached copy of an XDM Document is up-to-date).

The “last-modified” attribute SHALL contain either the date and time the XDM Document identified by the “uri” attribute was last modified or if the value of the “uri” attribute identifies an XDM Document Reference, the date and time the referenced XDM Document was last modified. (This allows an XDMC or an XDM Agent to determine if whether an XDM Document has changed recently or not.)

The “size” attribute SHALL contain either the size, in octets, of the XDM Document as identified above or if the value of the “uri” attribute identifies an XDM Document Reference, the size, in octets, of the referenced XDM Document. (This can help an XDMC or an XDM Agent determine if it wants to upload the entire XDM Document or an XDM Document Part, as appropriate based on any resource limitation such as bandwidth.)

The “reference” attribute SHALL be a reference to an XDM Document. The value SHALL correspond to the <reference> element provided by the XDCP Request with the <set-doc-ref > child element as described in sections 6.1.1.3.1 and 6.2.6.1.

The “reference-display-name” SHALL be present only when a “reference” attribute is present. The value of “reference-display-name” attribute SHALL correspond to the <display-name> element provided by the XDCP Request with the <set-doc-ref> child element as described in sections 6.1.1.3.1 and 6.2.6.1.

The <error-code> child element of a <folder> element SHALL contain the Status-Code and Reason-Phrase retrieved from the Status-Line of the received HTTP response message returned by an XDMS (see [RFC2616]).

5.3.2.8 Naming Conventions

There SHALL be only one XDM Directory Document per XUI in each XDMS. The name of the XDM Directory Document SHALL be “directory.xml”.

5.3.2.9 Data Interdependencies

For every XDM Document created/deleted/modified in the “users” tree for a particular XUI and Application Usage, the XDMS SHALL add/delete/update the appropriate <entry> child element in the appropriate <folder> element of the XDM Directory Document corresponding to that XUI.

NOTE 1: This does not imply that the server must actually store this XDM Directory Document. All that is required is that the XDMS is able to serve an up-to-date version of such an XDM Document when requested.

The XDMS SHOULD NOT generate an E-Tag value for the the XDM Directory Document.

NOTE 2: This implies that conditional operations are not supported against the XDM Directory Document. An XDMC or an XDM Agent should always refresh any cached copy.

5.3.2.10 Authorization Policies

The XDMS SHALL be the only entity allowed to create and modify the XDM Directory Document. Thus, the Authorized Principals SHALL only be allowed to retrieve this XDM Document.

The authorization policies for retrieving an XDM Directory Document SHALL conform to those described in section 5.1.5.

If an Access Permissions Document exists for a User Directory, Authenticated Principals that have permission to read the User Directory and its content SHALL be authorized to retrieve the XDM Directory Document information related to the User Directory. Authenticated Principals that have permission to read a single XDM Document or XDM Document Part SHALL only be authorized to retrieve information about that XDM Document.

If an Access Permissions Document does not exist only the Primary Principal and an associated Alias Principal SHALL be authorized to retrieve the XDM Directory Document.

5.3.2.11 Global Document

This Application Usage defines no Global Documents.

5.3.2.12 Subscription to changes

Not applicable.

5.3.2.13 Search Capabilities

Not applicable.

5.3.2.14 XDM Preferences Document

Not applicable.

5.3.2.15 History Information Documents

Not applicable.

5.3.2.16 Forwarding

Not applicable.

5.3.2.17 Restore

Not applicable.

5.3.2.18 Document Reference

Not applicable.

5.3.2.19 Differential Read and Write

Not applicable.

5.4 Common Content Types

5.4.1 Search Document

A Search Document included in a Search Request or a Search Response SHALL comply with the specification defined in this section.

5.4.1.1 MIME Type

The MIME type for the Search Document SHALL be "application/vnd.oma.search+xml".

5.4.1.2 XML Schema

The Search Document SHALL conform to the XML schema described in [XSD_search].

5.4.1.3 Structure

The Search Document SHALL conform to the XML schema described in section 5.4.1.2 "*XML Schema*", with the clarifications given in this section.

The <search> element:

1) SHALL include the "id" attribute with the value unique among the Search Requests generated by the same XDMC or the same XDM Agent. The Search Request generated by the XDMC or the XDM Agent SHALL include a <request> element. The non-error response generated by the XDMS SHALL include a <response> element. The value of the "id" attribute in case of response SHALL be the same as in the request for which the response was generated;

2) MAY include the "max-results" attribute with the positive integer value indicating the maximum number of results requested by the XDMC or the XDM Agent; and

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

The <query> element SHALL include an XQuery expression as described in [W3C-XQUERY] and [W3C-XQUERY_FullText]. It is RECOMMENDED to include the XQuery expression into the CDATA section.

The XQuery expression SHALL include one input function – collection. The collection of the data to be searched is created:

1) as a set of all XDM Documents stored in the Users Tree of an appropriate Application Usage excluding Supporting XDM Documents;

2) as a set of all XDM Documents in the particular User's home directory of an appropriate Application Usage excluding Supporting XDM Documents;

3) as a particular XDM Document in the particular User's home directory of an appropriate Application Usage; or,

4) as a set of all XDM Documents stored in the Global Tree of an appropriate Application Usage.

As such, the parameter of the collection function SHALL be either:

1) the string of "[AUID]/users/";
2) the string of "[AUID]/users/[XUI]";
3) the string of "[AUID]/users/[XUI]/[User Directory Document Selector]"; or,
4) the string of "[AUID]/global/".

For example:

collection("org.openmobilealliance.user-profile/users/") represents all User Profile Documents stored in the Users Tree on Profile XDMS to which the Search Request is targeted;

collection("org.openmobilealliance.groups/users/sip:joebloggs@example.com") represents all Group Documents stored in the home directory of "sip:joebloggs@example.com" on Group XDMS to which the Search Request is targeted;

collection ("resource-lists/users/sip:joebloggs@example.com/index") represents the URI List Document with the name "index" stored in the home directory of "sip:joebloggs@example.com" on List XDMS to which the Search Request is targeted.

The <request> element MAY include any other element or attribute from any other namespace for the purpose of extensibility.

The <response> element MAY include any other element or attribute from any other namespace for the purpose of extensibility.

In addition, each Application Usage that supports the Search feature SHALL define one or more basic XQuery expressions that are supported by the Application Usage selected from [W3C-XQUERY] and [W3C-XQUERY_FullText]. Such basic XQuery expressions allows the Application Usage to restrict the data that can be searched and also restrict the results provided to the XDMC or the XDM Agent.

5.4.2 XDCP Document

An XDCP Document SHALL be included both in an XDCP Request and in an XDCP Response. When included within an XDCP Request the XDCP Document SHALL contain the requested operation and data needed to perform that operation. When included within an XDCP Response the XDCP Document SHALL contain the result of the requested operation.

5.4.2.1 MIME Type

The MIME type for the XDCP Document SHALL be "application/vnd.oma.xdcp+xml".

5.4.2.2 XML Schema

The XDCP Document SHALL conform to the XML schema described in [XSD_xdcp], the "urn:ietf:params:xml:ns:patch-ops-error" XML schema described in [RFC5261] and the "urn:ietf:params:xml:ns:xcap-error" described in [RFC4825].

5.4.2.3 Structure

The XDCP Document SHALL conform to the XML schema described in section 5.4.2.2 "*XML Schema*", with the clarifications given in this section.

The XDCP Document SHALL include an <xdcp-document> root element.

The <xdcp-document> element SHALL include either a <request> element or a <response> element.

The <request> element SHALL include one of the following elements:

1) a <diff-read> element;
2) a <diff-write> element;
3) a <forward> element;

4) a <forward-accept> element;

5) a <forward-delivery-report> element;

6) a <forward-reject> element;

7) a <forward-remote> element;

8) a <remove-doc-ref> element;

9) a <restore> element;

10) a <retrieve-doc-ref> element;

11) a <set-doc-ref> element;

12) a <subscribe> element; or,

13) any element from other namespaces for the purpose of extensibility.

The <diff-read> element:

1) MAY include an <etag> element; and,

2) MAY include a <filter-set> element as described in [RFC4661].

The <diff-write> element:

1) MAY include a <filter-set> element as described in [RFC4661].

The <forward> element:

1) SHALL include a < recipients-list> element that:

    a) SHALL include a <list> element as described in [RFC4826].

2) MAY include a <note> element;

3) MAY include an <expiration-time> element;

4) MAY include a <filter-set> element as described in [RFC4661];

5) MAY include a <display-name> element;

6) MAY include a <delivery-report> element; and,

7) MAY include a <request-id> element.

The <forward-accept> element:

1) SHALL include a <document-uri> element;

2) MAY include either:

    a) a <do-not-store> element; or

    b) a <store> element with a “udds” attribute..

The <forward-delivery-report> element:

1) SHALL include a <request-id> element;

2) SHALL include a <recipient-uri> element; and,

3) SHALL include a <status> element.

The <forward-reject> element:

1) SHALL include a <document-uri> element.

The <forward-remote> element:

1) SHALL include a <document-uri> element;
2) MAY include a <note> element;
3) MAY include a <size> element;
4) SHALL include a <sender-identity> element;
5) MAY include a <display-name> element;
6) MAY include an <expiration-time> element;
7) SHALL include an <content-type> element;
8) SHALL include a < recipients-list> element that:
    a) SHALL include a <list> element as described in [RFC4826].
9) MAY include a <delivery-report> element; and,
10) MAY include a <request-id> element.

The <restore> element:

1) SHALL include an <back-to-etag> element; and,
2) MAY include a <current-etag> element.

The <set-doc-ref> element:

1) SHALL included a <reference> element; and,
2) MAY include a <display-name> element.

The <subscribe> element:

1) SHALL include a <subscription> element that:
    a) SHALL include a <target-documents> element;
    b) SHALL include a <duration> element; and,
    c) MAY include other elements from other namespaces for the purpose of extensibility.
2) SHALL include a <notification> element that either:
    a) SHALL include either a <push> element; or,
    b) SHALL include any element from other namespaces for the purpose of extensibility.

The <target-documents> element:

1) MAY include a <resource-uri> element that:
    a) SHALL include on or more <uri> elements; and,
    b) MAY include a <filter-set> element as described in [RFC4661].
2) MAY include a <resource-list> element that:

a) SHALL includes one or more <list> elements as described in [RFC4826]; and,

b) MAY include a <filter-set> element as described in [RFC4661].

3) MAY include a <list-uri> element that:

a) SHALL included one or more <uri> elements; and,

b) MAY include a <filter-set> element as described in [RFC4661].

The <push> element:

1) SHALL include a <push-address> element;

2) SHALL include a <wap-application-id> element;

3) SHALL include a <user-interaction-level> element;

4) MAY include a <preferred-notification-type> element;

5) MAY include a <gzip> element; and,

6) MAY include any elements from other namespaces for the purpose of extensibility.

The <response> element:

1) SHALL include one of the following elements:

a) a <done> element;

b) a <done-new-etag> element;

c) a <forward-result> element that:

i. MAY include a <not-found-recipients-list> element that:

- MAY include a <list> element as described in [RFC4826].

ii. MAY include a <not-authorized-recipients-list> element that:

- MAY include a <list> element as described in [RFC4826].

d) a <remote-forward-result> element that:

i. MAY include a <not-found-recipients-list> element that:

- MAY include one or more <list> elements as described in [RFC4826].

e) a <retrieve-doc-ref-result> element that:

i. SHALL include a <reference> element; and,

ii. MAY include a <display-name> element.

f) a <new-etag-value> element;

g) an <application-usage-defined-conflict> element that:

i. SHALL include a “phrase” attribute; and,

ii. MAY include any elements from other namespaces for the purpose of extensibility.

h) a <constraint-failure> element;

i) a <doc-ref-active> element;

j) an <etag-missing> element;

k) a <filter-set-not-allowed> element;

l) an <invalid-recipient> element;

m) an <invalid-request> element;

n) a <no-diff-document> element;

o) a <not-found> element;

p) a <not-latest-etag-value> element;

q) a <not-supported-request> element;

r) a <restore-version-not-found> element;

s) an <unknown-etag-value> element;

t) a <diff-write-conflict> element that

i. MAY include a <patch-ops-error> element as described in [RFC5261];

ii. MAY include an <xcap-error> element as described in [RFC4825]; or,

iii. MAY include any elements from other namespaces for the purpose of extensibility.

u) an <xcdp-document-structure-not-ok> element;

v) an <other-conflict> element that includes one or more elements from other namespaces; or,

w) any elements from other namespaces for the purpose of extensibility.

2) MAY include a <display-text> element.

5.5 Global Documents

[RFC4825] specifies a Global Tree which is used to place XDM Documents applicable to a particular Application Usage but which are not specific to any particular user. An example of this is the XDM Capabilities Document (see section 5.3.1 *"XCAP Server Capabilities"*) describing the Application Usages supported by an XDMS.

If Global Documents are used, each Application Usage SHALL describe how each Global Document is constructed and whether there is any associated authorization policy that controls the access to the Global Document.

Such a definition of a Global Document does not imply that the XDMS must actually store this Global Document. But, this implies that the XDMS SHALL always be prepared to process the requests against this Global Document and the contents of this XDM Document at any point in time SHALL always accurately represent the state of all XDM Documents (with appropriate restrictions, if defined) in the Users Tree of the corresponding Application Usage.

5.6 Access Permissions Document

Access Permissions Document defined in any Application Usage SHALL comply with the specification defined in this section.

5.6.1 Structure

The structure of an Access Permissions Document SHALL be as follows:

The Access Permissions Document SHALL include one root element <ap-rules>.

The <ap-rules> element:

1) MAY include one <access-permissions-document-rule> element as described in section 5.6.7;

2) MAY include one <directory-rule> element as described in section 5.6.7; and

3) MAY include one or more <document-rule> elements as described in section 5.6.7.

The <access-permissions-document-rule> element SHALL include one <rule-set> element conforming to [RFC4745].

The <rule-set> element SHALL consist of one or more <rule> elements conforming to [RFC4745].

The <rule> element SHALL contain one <conditions> and one <actions> element but NOT a <transformations> element.

The <conditions> element:

1) SHALL include either one <identity> element as described in [RFC4745] or one <external-list> element as described in section 5.2.2.1.

The <actions> element:

1) SHALL include either one <allow-any-operation> element as described in section 5.6.7 or one <allow-retrieve-own-data> element as described in section 5.6.7.

The <directory-rule> element SHALL include one <rule-set> element conforming to [RFC4745].

The <rule-set> element SHALL consist of one or more <rule> elements conforming to [RFC4745].

The <rule> element SHALL contain one <conditions> and one <action> element but NOT a <transformations> element.

The <conditions> element in the <rule> element:

1) SHALL include either one <identity> element as described in [RFC4745] or one <external-list> element as defined in section 5.2.2.1.

The <actions> element in the <rule> element SHALL include either one <allow-any-operation> element as described in section 5.6.7 or one or more of the following elements:

1) the <allow-retrieve> element as described in this section 5.6.7;

2) the <allow-modify> element as described in section 5.6.7;

3) the <allow-create> element as described in section 5.6.7; and

4) the <allow-delete> element as described in section 5.6.7.

The <document-rule> element:

1) SHALL include an attribute named “path” specifying the User Directory Document Selector of the XDM Document for which the rule applies to; and

2) SHALL include a <rule-set> element conforming to [RFC4745].

The <rule-set> element SHALL contain one or more <rule> elements conforming to [RFC4745].

The <conditions> child element of the <rule> element SHALL include one of the following child elements:

1) the <identity> element as described in [RFC4745] and in section 5.2.2.2.

2) the <external-list> element as defined in section 5.2.2.1; and

3) the <other-identity> element as defined in section 5.2.2.1.

The <actions> child element of the <rule> element SHALL include either one <allow-any-operation> element as described in section 5.6.7, one <allow-any-operation-own-data> element as described in section 5.6.7 or one or more of the following elements:

1) the <allow-retrieve> element as described in section 5.6.7;
2) the <allow-modify> element as described in section 5.6.7;
3) the <allow-create> element as described in section 5.6.7;
4) the <allow-delete> element as described in section 5.6.7;
5) the <allow-forwarding> element as described in section 5.6.7; and
6) the <allow-restore> element as described in section 5.6.7.

The <allow-forwarding> element MAY include the <recipients-list> element as defined in section 5.6.7.

The <recipients-list> element SHALL either include:

1) the <black-list> element as defined in section 5.6.7 or the <white-list> element as defined in section 5.6.7.

The <black-list> and <white-list> elements:

1) MAY include one or more <one> element(s) as described in [RFC4745]; and
2) MAY include one <external-list> element as described in 5.2.2.2.

The <transformations> element MAY be included. If included it SHALL include one of the following elements:

1) the <filter-set> element as described in [RFC 4661] , Appendix I "*Filter ABNF*" and section 5.6.7; and
2) elements from a namespace defined by a particular Application Usage.

The <filter-set> element SHALL have a structure as defined by [RFC 4661] with the following clarifications.

1) the <filter-set> element SHALL have a structure that include one or more <filter> element and MAY include a <ns-bindings> element but SHALL NOT include a "package" attribute; and
2) the <filter> element SHALL include one <what> element and a "id" attribute but SHALL NOT include a <trigger> element, a "uri" attribute, a "domain" attribute, an "enabled" attribute or a "remove" attribute.

The <what> element MAY include one or more <include> elements and MAY include one or more <exclude> elements. If an <exclude> element is present then, if any <include> elements are present they SHALL include a "type" attribute with the value set to "namespace".

5.6.2 Application Unique ID

An Application Usage MAY require the use of an Access Permissions Document in order to control access to XDM Resources. The Access Permissions Document SHALL share the Application Unique ID with the XDM Documents it controls access to.

5.6.3 Default Namespace

The default namespace SHALL be "urn:oma:xml:xdm:ap".

5.6.4 XML Schema

The Access Permissions Document SHALL conform to the XML schema defined in [XSD_ap], [RFC4661] and the XML schemas described in section 5.2.2.3.

5.6.5 MIME Type

The MIME Type for the Access Permissions Document is “application/vnd.oma.xdm-apd+xml”.

5.6.6 Validation Constraints

The Access Permissions Document SHALL conform to the XML schema described in section 5.6.4 “*XML Schema*” and the data semantics described in section 5.6.7 "*Data Semantics*" with the following clarifications:

1) The <document-rule> element attribute “path” SHALL contain the path to an existing XDM Document in the directory. Only one <document-rule> element per XDM Document in the directory SHALL be allowed. If this constraints is violated, an HTTP “409” Conflict” response SHALL be returned with the error condition identified by the <constraint-failure> element. If include, the “phase” attribute of this element SHOULD be set to “Only one document rule per document is allowed”; and

2) A <filter-set> element SHALL conform to the constraints defined in section 5.6.7 and to any constraint defined in the documentation for a particular Application Usage. If any of these constraints is violated, an HTTP “409” Conflict” response SHALL be returned the error condition identified in the <constraint-failure> element. If include, the “phase” attribute of this element SHOULD be set to “This type of filter is not allowed”.

5.6.7 Data Semantics

The <access-permissions-document-rule> element SHALL contain the Access Permissions to grant access to the Access Permissions Document itself or its History Information if such exists. An <access-permissions-document-rule> element SHALL only grant permissions to perform an XDM operation.

If the <access-permissions-document-rule> element is not included in the Access Permissions Document, the User Directory’s Primary Principal or associated Alias Principal SHALL be the Admin Principal of the Access Permissions Document or its History Information if such exists.

If the <access-permissions-document-rule> element is included in the Access Permissions Document, the <access-permissions-document-rule> element:

1) SHALL contain one <rule> element that grants at least one Principal access to all operations. The “id” attribute of this rule SHALL have the value “ap-admin”. The only allowed <actions> child element included in the <rule> element SHALL be the <allow-any-operations> element;

2) MAY contain one <rule> element that allows any Principal to retrieve an Access Permissions Document with only the requesting Principal’s Access Permissions and to block particular Principals to retrieve this information. The “id” attribute of this <rule> element SHALL have the value “ap-own-many”. The only allowed <actions> child element SHALL be the <allow-retrieve-own-data> element. The only allowed <conditions> child elements SHALL be the <identity> element including a <many> element. To block a particular Principal from retrieving its Access Permissions the identity of the Principal SHALL be included in an <except> child element to the <many> element; and

3) SHALL NOT contain a <rule> element with “id” attribute beginning with the sub-string “ap-” unless it is a rule with OMA defined data semantics.

The <directory-rule> element SHALL contain the Access Permissions rules to grant access to the Application Usage User Directory. This element SHALL enumerate Principals allowed to create and delete XDM Documents in the User Directory and allowed to perform management operations on existing XDM Documents. The element SHALL NOT define permissions for the Access Permissions Document.
This element SHALL be checked on every access to the User Directory.
A <directory-rule> element SHALL only grant permissions to perform an XDM operation.

If the <directory-rule> element is not included in the Access Permissions Document, the default Access Permissions as described by the Application Usage SHALL be applied, i.e. this is the same Access Permissions as described in section 5.1.3 for a User Directory without an Access Permissions Document.

If the <directory-rule> element is included in the Access Permissions Document and the Primary Principal is to be given more Access Permissions then the default retrieve permission, a <rule> element with the "id" attribute value "ap-xui" SHALL be included in the <rule-set> element. The <conditions> element SHALL include an <identity> element with only one <one> element with the identity of the Primary Principal included.

If the <directory-rule> element is included in the Access Permissions Document, the <directory-rule> element SHALL NOT contain a <rule> element with "id" attribute beginning with the sub-string "ap-" unless it is a rule with OMA defined data semantics.

The <document-rule> element SHALL contain the Access Permissions for a particular XDM Document in the Application Usage's User Directory. This element SHALL be checked when the <directory-rule> element does not grant a Principal access to the particular XDM Document for a particular XDM operation. The "path" attribute SHALL include the User Directory Document Selector of the XDM Document.
The <document-rule> element SHALL only grant permissions to perform an XDM operation.

If a < document-rule > element is included in the Access Permissions Document, the < document-rule > element SHALL NOT contain a <rule> element with "id" attribute beginning with the sub-string "ap-" unless it is a rule with OMA defined data semantics.

Each Application Usage SHALL define how the <external-list> element is used and what XDM Document Parts this element is allowed to reference.
One example of such definition can be found in [XDM_Group] section "*Group*".

The <allow-any-operation> element SHALL be a child element to the <actions> element. This element SHALL grant access to any operation when the <conditions> element of a <rule> element is evaluated to true.

The <allow-retrieve> element SHALL be a child element to the <actions> element. This element SHALL grant access to the "retrieve", "search" and "subscribe" operations when the <conditions> element of a <rule> element is evaluated to true.

The <allow-create> element SHALL be a child element to the <actions> element. This element SHALL grant access to the "create a document" operation and all "document reference" operations when the <conditions> element of a <rule> element is evaluated to true.

The <allow-modify> element SHALL be a child element to the <actions> element. This element SHALL grant access to modify an XDM Resource in the directory when the <conditions> element of a <rule> element is evaluated to true. The element SHALL also granted access to the "retrieve", "search' and "subscribe" operations.

The <allow-delete> element SHALL be a child element to the <actions> element. This element SHALL grant access to delete an XDM Document in the directory when the <conditions> element of a <rule> element is evaluated to true.

The <allow-forwarding> element SHALL be a child element of the <actions> element. This element SHALL grant access to the "forwarding of a document" operation for an XDM Resource in the directory when the <conditions> element of a <rule> element is evaluated to true. The element SHALL also grant access to the "retrieve", "search" and "subscribe" operations.

The <recipients-list> element SHALL be a child element of the <allow-forwarding> element. This element SHALL include either the <black-list> or the <white-list> element as child element. The <black-list> element SHALL be used to list a set of Principals who are restricted to receive the forwarded XDM Resource. The <white-list> element SHALL be used to list a set of Principals who are allowed to receive the forwarded XDM Resource.

The <allow-restore> element SHALL be a child element of the <actions> element. This element SHALL grant access to the "restore" operation of an XDM Document when the <conditions> element of a <rule> element is evaluated to true.

The <allow-any-operation-own-data> element SHALL be a child element of the <actions> element. This element SHALL grant access to XML nodes containing information related to the requesting Principal when the <conditions> element of a <rule> element is evaluated to true. Which such XML nodes are and how to determine which element or attribute that contain the identity of the requesting Principal is defined per Application Usage. An example of the use of this element is to allow a Principal to delete or modify an XML element pointed out using the Principals own identity in a list of identities. This element is used to allow authorized Principals to administer data related to themselves in other Principals' XDM Documents. This element for example can be used together with the <conditions> child elements <identity> <many> and <except> to control that all Principals apart from the ones in the <except> element are allowed to modify data related to themselves.

The <allow-retrieve-own-data> element SHALL be a child element of the <actions> element. This element SHALL grant retrieve access to XDM Documents that only contain information related to the requesting Principal when the <conditions> element of a <rule> element is evaluated to true. Application Usage SHALL define rules to determine what constitutes information related to the requesting Principal and what elements and attributes contain that information. This element is used to allow authorized Principals to retrieve data related to themselves in other Principals' XDM Documents. This element, for example, can be used together with the <conditions> child elements <identity> <many> and <except> to control that all Principals apart from the ones in the <except> element are allowed to retrieve data related to themselves.
When this element is used in a <rule> element inside an <access-permissions-document-rule> element, the requesting Principal, as result of a retrieve request, SHALL obtain an Access Permissions Document that contains only the rules that specify the requesting Principal's Access Permissions to the Application Usage's XDM Resources. Information related to other Principals SHALL NOT be included.

The <filter-set> element SHALL be a child element of the <transformations> element. This element SHALL define which parts of an XDM Document the Principals identified by the <conditions> child element have permission to access.

The <filter-set> element has the following use in the context of different XDM operations.

1) Retrieve operation: The <filter-set> element defines which parts of an XDM Document a requesting Principal is allowed to retrieve. The filter SHALL be applied to the XDM Document before the XDM Document is returned to the requesting Principal.

2) Subscribe for changes operations: The <filter-set> element defines which parts of an XDM Document a requesting Principal is allowed to get changes about. The requesting Principal SHALL receive notifications only about changes to the parts of the XDM Document that are defined by the <filter-set> element.

3) Modification operations: The <filter-set> element defines which parts of an XDM Document a requesting Principal is allowed to modify. The filter SHALL be applied to the XDM Document modification information received in the modification request before the XDM Document is allowed to be modified. If the modification information contains more XDM Document Parts than the filter defines, the modification request SHALL be denied else it SHALL be accepted and the XDM Document SHALL be modified as requested.

4) Forwarding operations: The <filter-set> element defines which parts of an XDM Document the requesting Principal is allowed to forward to another Principal. The filter SHALL be applied to the XDM Document before it is forwarded to the receiving Principal. The <exclude> child elements of the <what> element is used to block particular content of an XDM Document to be forwarded. The <include> elements are used to allow only particular content of the XDM Document to be forwarded.

5) Restore operations: The <filter-set> element SHALL be ignored when checking access permissions for the restore operation.

6) Differential Read operations: The <filter-set> element defines which parts of an XDM Document a requesting Principal is allowed to retrieve. The <filter-set> element SHALL be applied to the XDM Document before the differential information is returned to the requesting Principal.

7) Differential Write operations: The <filter-set> element defines which parts of an XDM Document a requesting Principal is allowed to modify. The filter SHALL be applied to the XDM Document modification information received in the modification request before the XDM Document is allowed to be modified. If both the received Differential Write operation and the <transformations> element of the <rule> element that granted the requesting Principal access to the operations includes a <filter-set> element, the request SHALL be rejected.

8) Search operations: The <filter-set> element defines which XDM Document Parts a requesting Principal is allowed to search and which XDM Document Parts the search result is allowed to contain.

5.6.8 Naming Conventions

There is only one Access Permissions Document per Application Usage and XUI in each XDMS. The Access Permissions Document SHALL be addressed using the User Directory Document Selector "/oma_ap/access-permissions", i.e. the Document Selector to the Access Permissions Document SHALL be "[auid]/users/[xui]/oma_ap/access-permissions" and the XDM Document name SHALL be "access-permissions".

5.6.9 Global Documents

No Global Documents are defined.

5.6.10 Resource Interdependencies

None.

5.6.11 Authorization Policies

The <access-permissions-document-rule> element in the Access Permissions Document SHALL be checked before executing any XDM operation towards an Access Permissions Document.

If a Modification History Information Document containing information about the Access Permissions Document exists, Access Permissions to this Modification History Information Document SHALL be controlled by the <access-permissions-document> rule element in the Access Permissions Document and SHALL be checked by the XDMS before executing any XDM operation towards the Modification History Information Document.

5.6.12 Subscription to changes

Not applicable.

5.6.13 Search Capabilities

Not applicable.

5.6.14 XDM Preferences Document

An Access Permissions Document MAY support XDM Preferences Document for handling the preferences related only to History Information if History Information is supported.

5.6.15 History Information Documents

An Access Permissions Document MAY support Modification History Information Document as described in section 5.7.1 "*Modification History Information Document*".

An Access Permissions Document MAY support a Request History Information Document as described in section 5.7.2 "*Request History Information Document*".

5.6.16 Forwarding

Not applicable.

5.6.17 Restore

Restore of an Access Permissions Document as described in section 6.2.6.5 "*XDM Restore*" MAY be supported.

5.6.18 Document Reference

Document References of an Access Permissions Document as described in section 6.2.6.1 "*Document Reference*" MAY be supported.

5.6.19 Differential Read and Write

An Access Permissions Document MAY support Differential Write as described in section 6.2.6.4 "*Differential Write"*. A Differential Write request including a <filter-set> element is not supported.

An Access Permissions Document MAY support Differential Read as described in section 6.2.6.3 *"Differential Read"*.

5.7 History Information

The History Information function allows an XDMC or an XDM Agent to retrieve information about operation done towards an XDM Document. The information is divided into two parts, Modification History Information containing successful operations that modify the XDM Document (i.e. successful create, modify and delete operations) and Request History Information containing a subset of all other operations performed on the XDM Document (e.g. unauthorized retrieve and subscribe for changes operations).

The Modification History Information is available in the form of an XDM Document as defined in section 5.7.1"*Modification History Information Document*". The Request History Information is available in the form of an XDM Document as defined in section 5.7.2 "*Request History Information Document*".

5.7.1 Modification History Information Document

5.7.1.1 Structure

The structure of a Modification History Information Document SHALL be as follows:

The Modification History Information Document SHALL have one root element <history-information>.

The <history-information> element MAY include one or more <patch> element as described in section 5.7.1.7;

The <patch> element:

1) SHALL include an attribute "id" as described in section 5.7.1.7;
2) SHALL include an attribute "operation-requester" as described in section 5.7.1.7;
3) SHALL include an attribute "timestamp" as described in section 5.7.1.7;
4) SHALL include an attribute "new-etag" as defined in [RFC5874];
5) SHALL include an attribute "previous-etag" as defined in [RFC5874];
6) SHALL include an attribute "type" to capture the type of XDM Operation performed;
7) MAY include <modified-information> as described in section 5.7.1.7;
8) MAY include one of the following elements:
    a) an <add> element as defined in [RFC5874]; or
    b) a <remove> element as defined in [RFC5874] and described in section 5.7.1.7; or
    c) a <replace> element as defined in [RFC5874] and described in section 5.7.1.7; or
    d) a <restore> element as described in section 5.7.1.7; or
    e) a <diff-write> element as described in section 5.7.1.7.

The <remove> and <replace> elements SHALL include an attribute "modified-ref" as described in section 5.7.1.7.

The <modified-information> element:

1) SHALL include <change-logs> element as described in section 5.7.1.7; and
2) SHALL include an attribute "id".

The <change-logs> element SHALL include one of the following elements:

1) a <previous-element> element as described in section 5.7.1.7;

2) a <previous-attribute> element as described in section 5.7.1.7.

5.7.1.2 Application Unique ID

An Application Usage MAY requires the use of a Modification History Information Document. The Modification History Information Document SHALL have the same Application Unique ID as the XDM Document whose Modification History Information it contains.

5.7.1.3 Default Namespace

The default namespace SHALL be "urn:oma:xml:xdm:hist-mod".

5.7.1.4 XML Schema

The Modification History Information Document SHALL conform to the XML schema defined in [XSD_modHist].

5.7.1.5 MIME Type

The MIME Type for the Document is "application/vnd.oma.xdm-hi+xml"

5.7.1.6 Validation Constraints

The Modification History Information Document SHALL conform to the XML schema described in section 5.7.1.4 *"XML Schema"*.

5.7.1.7 Data Semantics

The <history-information> element contains the history information of an XDM Document with the same file name and folder name.

The <patch> element SHALL be the child of the <history-information> element which records each history of the operation towards the XDM Document. The operation requester SHALL be indicated by "operation-requester" attribute. The "timestamp" attribute SHALL be used to store the timestamp of when the operation towards the XDM Document has been processed. The "id" attribute in the <patch> element SHALL be unique among all <patch> elements in this Modification History Information Document and recorded in ascendant order.

NOTE: the "id" attribute can be used to address a particular <patch> element as the XDMS may remove the older ones if the storage reaches the limit based on the Service Provider's policy.

The <add> element as defined in [RFC5874] SHALL be the child of the <patch> element which records history information of adding an element or attribute located in corresponding XDM Document indicated by "sel" attribute as defined in [IETF-RFC5874]. The value of the "sel" attribute SHALL be the same as Request-URI in XCAP PUT request. The "new-etag" attribute of the <patch> element SHALL be set to the E-Tag of the corresponding XDM Document after performing XCAP PUT operation. The "previous-etag" attribute of the <patch> element SHALL be set to the E-Tag of the corresponding XDM Document before performing XCAP PUT operation.

The <replace> element as defined in [RFC5874] SHALL be the child of the <patch> element which records history information of replacing an element or attribute located in corresponding XDM Document indicated by "sel" attribute as defined in [RFC5874]. The value of the "sel" attribute SHALL be the same as Request-URI in XCAP PUT request. The "new-etag" attribute of the <patch> element SHALL be set to the E-Tag of the corresponding XDM Document after performing XCAP PUT operation. The "previous-etag" attribute of the <patch> element SHALL be set to the E-Tag of the corresponding XDM Document before performing XCAP PUT operation. The "modified-ref" attribute of the <replace> element SHALL be set to the same value of the "id" attribute of <modified-information> element which records the previous element or attribute before operating the XCAP PUT operation.

The <remove> element as defined in [RFC5874] SHALL be the child of the <patch> element which records history information of replacing an element or attribute located in corresponding XDM Document indicated by "sel" attribute as defined in [RFC5874]. The value of the "sel" attribute SHALL be the same as Request-URI in XCAP DELETE request. The

“new-etag” attribute of the <patch> element SHALL be set to the E-Tag of the corresponding XDM Document after performing XCAP DELETE operation. The “previous-etag” attribute of the <patch> element SHALL be set to the E-Tag of the corresponding XDM Document before performing XCAP DELETE operation. The “modified-ref” attribute of the <remove> element SHALL be set to the same value of the “id” attribute of <modified-information> element which records the previous element or attribute before operating the XCAP DELETE operation.

The <diff-write> element SHALL be the child of the <patch> element which records history information of differential writing located in corresponding XDM Document indicated by “sel” attribute as defined in [RFC5874]. The <diff-write> element SHALL be of type "documentType" [RFC5874] reflecting the actual changes made to the target XDM Document The value of the “sel” attribute SHALL be the same as Request-URI in Differential Write XDCP Request. The “new-etag” attribute of the <patch> element SHALL be set to the E-Tag of the corresponding XDM Document after performing Differential Write XDCP Request. The “previous-etag” attribute of the <patch> element SHALL be set to the E-Tag of the corresponding XDM Document before performing Differential Write XDCP Request.

The <restore> element SHALL be the child of the <patch> element which records history information of restoring located in corresponding XDM Document indicated by “sel” attribute as defined in [RFC5874]. The value of the “sel” attribute SHALL be the same as Request-URI in XDCP restore request. The “new-etag” attribute of the <patch> element SHALL be set to the E-Tag of the corresponding XDM Document after performing XDCP restore operation. The “previous-etag” attribute of the <patch> element SHALL be set to the E-Tag of the corresponding XDM Document before performing XDCP restore operation.

The <modified-information> element SHALL be the child of the <entry> element which is used to record the previous element or attribute before performing the operation towards to the XDM Document. The value of “id” attribute SHALL be unique in the whole Modification History Information Document

The <previous-element> SHALL be the child of the <change-logs> element which is used to record the previous element before performing the operation towards to the XDM Document.

The <previous-attribute> SHALL be the child of the <change-logs> element which is used to record the previous attribute before performing the operation towards to the XDM Document.

5.7.1.8 Naming Conventions

There is one Modification History Information Document per XDM Document having Modification History Information.The Modification History Information Document SHALL be addressed using the User Directory Document Selector “/oma_hist/[XDM-Document-name] i.e. the Document Selector to the Modification History Information Document SHALL be “[auid]/users/[xui]/oma_hist/index” if name of the XDM Document having Modification History Information is “index”.

5.7.1.9 Global Documents

No Global Documents are defined.

5.7.1.10 Resource interdependencies

None.

5.7.1.11 Authorization Policies

The XDMS is the only entity that is allowed to create or modify elements and attributes in the Modification History Information Document.

An authenticated Principal SHALL be able to retrieve or delete an XDM Document Part (including all) of the Modification History Information Document. However, an authenticated Principal SHALL not be able to delete the entire Modification History Information Document. Before allowing access or edit to the Modification History Information Document, the XDMS SHALL check the <document-rule> element in the Access Permissions Document.
If the operation is not allowed, the XDMS SHALL reject the operation with an HTTP “403 Forbidden” response, if it is an XCAP operation, or with a SIP “403 Forbidden” response, if it is a SIP operation.

The use of a <filter-set> element in an Access Permissions Document is not applicable to a Modification History Information Document.

5.7.1.12 Subscription to changes

Not applicable.

5.7.1.13 Search Capabilities

An Application Usage MAY support search in Modification History Information Document. If the search feature is supported, it SHALL be possible to search according to the following rules:

The Application Usage SHALL support a collection "[AUID]/users/[XUI]/oma_hist/" and a collection "[AUID]/users/[XUI]/oma_hist/[XDM-Document-document-name]". An example of a collection is as follows:

"org.openmobilealliance.user-profile/users/user1@example.com/oma_hist", or;

"org.openmobilealliance.user-profile/users/user1@example.com/oma_hist/user-profile".

The basic XQuery expression supported by this Application Usage SHALL be as follows:

```
xquery version "1.0";
declare default element namespace "urn:oma:xml:xdm:hist-mod";

for $g in collection({Data_Source})/history-information/patch/
where [{Condition}]
return <patch>{$g/@operatation-requester} {$g/@timestamp} {$g/patch} {$g/modified-
information}</patch>
```

where:

{*Data_Source*} represents the collection that SHALL be searched. In case that the value:

- "[AUID]/users/[XUI]/oma_hist/" is used, the search SHALL be executed over all Modification History Documents stored in the home directory of the User identified by XUI, in Application Usage identified by AUID.
- "[AUID]/users/[XUI]/oma_hist/[XDM-Document-document-name]" is used, the search SHALL be executed over the Modification History Documents identified by [XDM-Document-document-name]" stored in the home directory of the User identified by XUI, in Application Usage identified by AUID.

{*Condition*} SHALL represent a combination of logical and/or comparison expressions defined by an XDMC or an XDM Agent which makes use of one or more of the attributes/elements "operation-requestor", "timestamp", <add>, <remove> <replace>, <restore> or <diff-write> . All other XML elements/attributes referred to as part of {*Condition*} SHALL be forbidden.

An example of a condition is as follows:

$g /add and ($g/@operation-requester= "user1@example.com") and ($g/@timestamp > "2009-12-30T00:00:00" and $g/@timestamp < "2010-01-02T00:00:00")

NOTE: The comparison expression can be used in condition. In the example above, XDMS will return <patch> elements which record all operations requested by "user1@example.com" in the time period from "2009-12-30T00:00:00" to "2010-01-02T00:00:00"

All Search Requests that do not comply with the basic XQuery expression as defined in this section SHALL be responded to with an HTTP "409 Conflict" error response.

5.7.1.14 XDM Preferences Document

Not applicable.

5.7.1.15 History Information Documents

Not applicable.

5.7.1.16 Forwarding

Not applicable.

5.7.1.17 Restore

Not applicable.

5.7.1.18 Document Reference

Not applicable.

5.7.1.19 Differential Read and Write

Not applicable.

5.7.2 Request History Information Document

5.7.2.1 Structure

The structure of a Request History Information Document SHALL be as follows:

The Request History Information Document SHALL have one root element <request-history >.

The <request-history> element:

1) SHALL include zero or more <document> elements.

The <document> element:

1) SHALL include an “udds” attribute; and
2) SHALL include zero or more <requestor> elements.

The <requestor> element:

1) SHALL include an “id” attribute;
2) MAY include a <last-requests> element; and
3) MAY include a <request-log> element;
4) MAY include other elements from other namespaces for the purposes of extensibility; and
5) MAY include attributes from any other namespace for the purpose of extensibility.

The <last-requests> element

1) SHALL include zero or more <request> elements.

The <request-log> element:

1) SHALL include zero or more <request> elements.

The <request> element:

1) SHALL include a “type” attribute;

2) SHALL include a “result” attribute;

3) SHALL include a “timestamp” attribute;

4) MAY include a <details> element;

5) MAY include a “counter” attribute;

6) MAY include a <node-selector> element;

7) MAY include other elements from other namespaces for the purposes of extensibility; and

8) MAY include attributes from any other namespace for the purpose of extensibility.

The <details> element:

1) MAY include a <document-reference> element;

2) MAY include a <forward> element;

3) MAY include a <get> element;

4) MAY include a <modify> element;

5) MAY include a <restore> element;

6) MAY Include a <subscribe> element; and

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

5.7.2.2 Application Unique ID

An Application Usage MAY require the use of a Request History Information Document. The Request History Information Document SHALL have the same Application Unique ID as the XDM Document whose Request History Information it contains.

5.7.2.3 Default Namespace

The default namespace SHALL be “urn:oma:xml:xdm:request-history”.

5.7.2.4 XML Schema

The Request History Information Document SHALL conform to the XML schema defined in [XSD_reqHist].

5.7.2.5 MIME Type

The MIME Type for the Request History Information Document is “application/vnd.oma.xdm-rhi+xml”.

5.7.2.6 Validation Constraints

Not applicable.

5.7.2.7 Data Semantics

The <document> element SHALL contain information about XDM Request targeting a particular XDM Document. The “udds” attribute SHALL contain the XDM Document’s User Directory Document Selector.

The <requestor> element SHALL contain information about a particular Principal’s XDM Requests targeting the XDM Document. The “id” attribute SHALL contain the identity of the requesting Principal as received in the XDM request.

The <last-requests> SHALL contain the last XDM request for each type of XDM requests performed by the requesting Principal.

The <request-log> element SHALL contain a list of XDM requests performed by the requesting Principal. Local policy SHALL define the size of the list.

The <request> element SHALL contain information about the performed XDM request:

1) the type of the request in the "type" attribute. Two values are possible "modify" or "retrieve";
2) the result of the request in the "result" attribute. Two values are possible "authorized" and "unauthorized";
3) the time the XDM request was performed in the "timestamp" attribute;
4) the Node Selector of XDM Document Part in the <node-selector> child element if the XDM request was made targeting a particular XDM Document Part;
5) the number of repeated identical requests of the same type in the "counter" attribute; and
6) details about the request in the child elements of the <details> element.

The "result" attribute SHALL have the value "authorized" if the requesting Principal was allowed to perform the operation as described in section 5.1.5.

The "result" attribute SHALL have the value "unauthorized" if the requesting Principal was not allowed to perform the operation as described in section 5.1.5. Requests that failed for other reasons SHALL NOT be recorded in the Request History Information Document.

The "type" attribute and the child elements of the <details> element SHALL be combined as the following:

1) if the request is a "set-doc-ref" or a "remove-doc-ref" XDCP Request as described in section 6.2.6.1, the "type" attribute value SHALL be "modify" and the <details> child element SHALL be the <document-reference> element;
2) if the request is a "retrieve-doc-ref" XDCP Request as described in section 6.2.6.1, the "type" attribute value SHALL be "retrieve" and the <details> child element SHALL be the <document-reference> element;
3) if the request is an "restore" XDCP Request as described in section 6.2.6.5 , the "type" attribute value SHALL be modify" and the <details> child element SHALL be the <restore> element;
4) if the request is a "diff-read" XDCP Request as described in section 6.2.6.3, the "type" attribute value SHALL be retrieve" and the <details> child element SHALL be the <get> element;
5) if the request is a "diff-write" XDCP Request as described in section 6.2.6.4, the "type" attribute value SHALL be modify" and the <details> child element SHALL be the <modify> element;
6) if the request is a "forward" XDCP Request as described in section 6.2.6.2, the "type" attribute value SHALL be "retrieve" and the <details> child element SHALL be the <forward> element;
7) if the request is an XCAP operation as described in sections 6.2.1.1 and 6.2.1.3, the "type" attribute value SHALL be "modify" and the <details> child element SHALL be the <modify> element;
8) if the request is an XCAP operation as described in section 6.2.1.2, the "type" attribute value SHALL be retrieve" and the <details> child element SHALL be the <get> element; and
9) if the request is a "subscribe for changes" XDM operation as described in section 6.2.2, the "type" attribute value SHALL be" retrieve" and the <details> child element SHALL be the <subscribe> element.

5.7.2.8 Naming conventions

There is one Request History Information Document per XUI in the Users Tree of an Application Usage. The Request History Information Document SHALL be addressed using the User Directory Document Selector "/oma_requests/history i.e. the Document Selector to the Request History Information Document SHALL be "[auid]/users/[xui]/oma_requests/history".

5.7.2.9 Global Documents

Not applicable.

5.7.2.10 Resource interdependencies

The Request History Information Document SHALL contain information about XDM Requests targeting XDM Resources residing in the same User Directory as the Request History Information Document.

5.7.2.11 Authorization Policies

The XDMS is the only entity that is allowed to create or modify elements and attributes in the Request History Information Document.

An authenticated Principal SHALL be able only to retrieve the whole Request History Information Document or an XDM Document Part of it and to delete the whole Request History Information Document or an XDM Document Part of it. Before the XDMS accept such request the XDMS SHALL do authorization as described in section 5.1.5 making use of the Access Permissions Document associated the Application Usage if such exists. If such Access Permissions Document does not exist, the XDMS SHALL apply the default authorization policy as described in section 5.1.5. The use of a <filter-set> element in an Access Permissions Document is not applicable to a Request History Information Document.

5.7.2.12 Subscription to changes

A Request History Information Document SHALL support subscription to changes.

5.7.2.13 Search Capabilities

An Application Usage MAY support search in Request History Information Document. If the search feature is supported, it SHALL be possible to search according to the following rules:

The Application Usage SHALL support a collection "[AUID]/users/[XUI]/ oma_requests/history". An example of a collection is as follows:

"org.openmobilealliance.user-profile/users/user1@example.com/oma_requests/history"

The basic XQuery expression supported by this Application Usage SHALL be as follows:

```
xquery version "1.0";
declare default element namespace " urn:oma:xml:xdm:request-history ";

for $g in collection({Data_Source})/request-history/
where [{Condition}]
return <document>{$g/@udds} {$g/requestor/@id}</ document >
```

where:

{*Data_Source*} represents the collection that SHALL be searched. In case that the value:

- "[AUID]/users/[XUI]/oma_requests/history" is used, the search SHALL be executed over the Request History Information Document stored in the home directory of the User identified by XUI, in Application Usage identified by AUID.

{*Condition*} SHALL represent a combination of logical and/or comparison expressions defined by an XDMC or an XDM Agent which makes use of one or more of the attributes "id", "type", "result" or "timestamp". All other XML elements/attributes referred to as part of {*Condition*} SHALL be forbidden.

Examples of conditions are as follows:

$g /document/requestor/request-log/request/@type=”retrieve” and $g /document/requestor/request-log/request/@ result=”unauthorized”

$g /document/requestor/request-log/request/@ result=”unauthorized” and ($g /document/requestor/request-log/request/@ timestamp > “2009-12-30T00:00:00” and $g /document/requestor/request-log/request/@ timestamp <“2010-01-02T00:00:00”

NOTE: The comparison expression can be used in condition. In the example above, XDMS will return <document> elements which record unauthorized operations in the time period from “2009-12-30T00:00:00” to “2010-01-02T00:00:00”

All Search Requests that do not comply with the basic XQuery expression as defined in this section SHALL be responded to with an HTTP “409 Conflict” error response.

5.7.2.14 XDM Preferences Document

Not applicable.

5.7.2.15 History Information Documents

Not applicable.

5.7.2.16 Forwarding

Not applicable.

5.7.2.17 Restore

Not applicable.

5.7.2.18 Document Reference

Not applicable.

5.7.2.19 Differential Read and Write

A Request History Information Document SHALL support Differential Read as described in section 6.2.6.3 “*Differential Read”*.

The Request History Information Document MAY support Differential Write, as described in section 6.2.6.4 “*Differential Write”*. See also section 5.7.2.11 "Authorization Policies" for associated details.

5.8 XDM Preferences Document

5.8.1 Structure

The structure of the XDM Preferences Document SHALL be as follows:

The XDM Preferences SHALL include the root element <preferences>. The <preferences> element:

1) MAY include <history-prefs> element as described in section 5.8.7;
2) MAY include <forward-prefs> element as described in section 5.8.7;
3) MAY include any other elements from any other namespaces for the purposes of extensibility; and
4) MAY include any other attributes from any other namespaces for the purposes of extensibility.

The <history-prefs> element:

1) SHALL include <history-info> element as described in section 5.8.7; and
2) MAY include one or more <filter> elements as described in section 5.8.7.

The <history-info> element:

1) SHALL include an attribute “state” as described in section 5.8.7; and
2) MAY include <except> element as described in section 5.8.7.

The <except> element MAY include one or more <document> element.

The <document> element SHALL include an attribute “path” which carries a User Directory Document Selector, a User Directory Folder Selector or a “/” character as its string value.

The <filter> element:

1) SHALL include <conditions> element as described in section 5.8.7;
2) SHALL include <actions> element as described in section 5.8.7;
3) SHALL include an attribute “state” as described in section 5.8.7; and
4) SHALL include an attribute “id” as described in section 5.8.7.

The <conditions> element SHALL include any of the following child elements:

1) <operation-type> element as described in section 5.8.7;
2) < operation-result> element as described in section 5.8.7;
3) <identity> element as described in section 5.8.7;
4) <external-list> element as described in section 5.2.2.2;
5) <other-identity> element as described in section 5.2.2.2;
6) <validity> element as described in section 5.8.7; and
7) <document> element as described in section 5.8.7.

The <operation-type> element SHALL include any of the following child elements:

1) <modify> element as described in section 5.8.7; and
2) <retrieve> element as described in section 5.8.7.

The <operation-result> element SHALL include any of the following child elements:

1) <authorized> element as described in section 5.8.7; and
2) <un-authorized> element as described in section 5.8.7.

The <actions> element SHALL include any of the following child elements:

1) <store-changelog> element as described in section 5.8.7;
2) <store-request-log> element as described in section 5.8.7.

The <forward-prefs> element:

1) MAY include one or more <rule> element as described in section 5.8.7

The <conditions> child element of <rule> element:

1) MAY include <identity> element as described in section 5.8.7;
2) MAY include <content-type> element as described in section 5.8.7; and
3) MAY include <max-size> element as described in section 5.8.7.

The <actions> child element of <rule> element:

4) MAY include <accept> element as described in section 5.8.7;
5) MAY include <reject> element as described in section 5.8.7; and
6) MAY include <confirm> element as described in section 5.8.7.

5.8.2 Application Unique ID

An Application Usage MAY require the use of XDM Preferences Document in order to control information to be stored in Modification History Information Document (see section 5.7.1) and Request History Information Document (see section 5.7.2) and also to handle the Forward XDCP Requests received from other users. The XDM Preferences Document SHALL share the Application Unique ID with the XDM Documents to which Forward XDCP Request is targeted and also the XDM Document to which it controls the storage of History Information.

5.8.3 Default Namespace

The default namespace SHALL be “urn:oma:xml:xdm:xdm-prefs”.

5.8.4 XML Schema

The XDM Preferences Document SHALL conform to the XML schema defined in [XSD_XDM2_XP].

5.8.5 MIME Type

The MIME Type for the XDM Preferences Document is “application/vnd.oma.xdm-prefs+xml”.

5.8.6 Validation Constraints

The XDM Preferences Document SHALL conform to the XML schema described in section 5.8.4 “XML Schema”.

5.8.7 Data Semantics

5.8.7.1 History-Prefs Elements

The <history-info> element SHALL be used to enable or disable the recording of History Information of all or few XDM Documents residing under the same AUID as XDM Preferences Document by setting the “state” attribute of the <history-info> element with the following values:

- “on” - instructs the XDMS to record the History Information for particular XDM Documents.
- “offf” - instructs the XDMS not to record the History Information for particular XDM Documents.

When value of “state” attribute of the <history-info> element is set to “on”, the XDMS SHALL store the following information into the Modification History Information Document irrespective of the preferences set in the XDM Preferences Document:

- Identity of the Principal who performed the XDM Operation
- Type of the XDM Operation
- E-Tag value of the XDM Document before performing the XDM Operation
- E-Tag value of the XDM Document after performing the XDM Operation.

The XDMS SHALL store the above listed information in the Modification History Information Document when no <filter> element is present and the “state” attribute of the <history-info> element is set to “on”.

If the state attribute of <history-info> element is set to “off”, the XDMS SHALL NOT record History Information for any XDM Documents other than the ones listed under the <except> element.

The <except> element SHALL be used to instruct XDMS to exclude some of the XDM Documents when enabling or disabling History Information as instructed by the <history-info> element. I.e. when “state” attribute of <history-info> element is set to “on” and some XDM Documents are listed under <except> element the XDMS has to enable the History Information for all other XDM Documents than listed under <except> element.

The <filter> element specifies the conditions upon which the History Information to be stored and also what information need to be stored in the Modification History Information Document or in the Request History Information Document. When no <filter> element is present and “state” attribute of the <history-info> element is set to “on”,the XDMS SHALL store last XDM request for each type of XDM requests performed by the requesting Principal in the Request History Information Document. The “id” attribute of the <filter> element SHALL be unique within the XDM Preferences Document.

The <conditions> element SHALL be the child element of <filter> element which includes the conditions set by the User for storing the History Information into the History Information Document and conforms to [RFC4745].

The <operation-type> SHALL be the child element of the <conditions> element. This element specifies that History Information has to be stored when a particular XDM operation is applied to the corresponding XDM Document. Details of the History Information to be stored are based on <actions>element.

The <operation-type> element element with <modify> child element instructs the XDMS to store the <request> element with the "type" attribute value "modify" as described in section 5.7.2.7.

The <operation-type> element with <retrieve> child element instructs the XDMS to store the <request> element with the "type" attribute value "retrieve" as described in section 5.7.2.7.

The <operation-result> specifies that History Information has to be stored depending on whether the Principal performing the XDM Operation is authorized or not.

The <operation-result> element with <authorized> child element instructs the XDMS to store the details of the XDM Operation if the Principal performing the XDM Operation is authorized.

The <operation-result> element with <un-authorized> child element instructs the XDMS to store the details of the XDM Operation if the Principal performing the XDM Operation is not authorized.The <validity> element SHALL be used to indicate the time period for recording the XDM History Information and conforms to [RFC4745]. The XDMS SHALL record the History Information for the time period specified by this element. Even if recording of History Information is enabled the XDMS SHALL NOT record the History Information if the validity period specified by this element is expired.

The <identity> element SHALL be the child element of the <conditions> element and conforms to [RFC4745]. The <identity> condition is used to indicate that the XDMS SHALL record the History Information if the XDM Operation is performed by the principals identified by this element.

The <document> element SHALL be the child element of the <conditions> element which can be used to indicate the path of the XDM Document to which the preferences mentioned as part of the parent <filter> element has to be applied while storing the History Information. The “path” attribute carries a User Directory Document Selector, a User Directory Folder Selector or a “/” character as its string value. If the preferences have to be applied to a particular XDM Document then the then the ”path” attribute carries a User Directory Document Selector as its string value. If the preferences have to be applied to all the XDM Documents in a particular folder then the ”path” attribute carries the User Directory Folder Selector as its string value. If the preferences have to be applied to all XDM Documents in the User Directory not located under a particular folder then the “path” attribute carries the “/” character.

The <actions> element SHALL be the child element of <filter> element and conforms to [RFC4745]. This element is used to instruct the XDMS about what History Information SHALL be stored.

The <store-change-log> element SHALL be the child element of the <actions> element. This element is used to control storage of change details in to the Modification History Document. The value is of a Boolean type:

- "false" - instructs the XDMS not to store change details. This is the default value taken in the absence of the element.
- "true" - instructs the XDMS to store change details.

The <store-request-log> element SHALL be the child element of the <actions> element. This element is used to control the storage of request details in to the Request History Document. The value is of Boolean type:

- "false" - instructs the XDMS not to store request details. This is the default value taken in the absence of the element.
- "true - instructs the XDMS to store the request details.

5.8.7.2 Forward-Prefs Elements

The <rule> element SHALL be the child element of <forward-prefs> element which includes the conditions and actions set by the User for handling the Forward XDCP Request received from other users. It SHALL conform to the structure of <rule> element as described in [RFC4745] with the following clarifications:

1) The <rule> element SHALL NOT include <transformations> element.
2) The <identity> element SHALL be the child element of <conditions> element which is used to instruct the XDMS on how to handle the Forward XDCP Request when it is received from the users matching its value. It SHALL conform to the structure of the <identity> element as described in [RFC4745].
3) The <content-type> element SHALL be the child element of the <conditions> element which is used to instruct the XDMS on how to handle the Forward XDCP Request when the MIME type of the received content matches with the value of this element.
4) The <max-size> element SHALL be the child element of the <conditions> element which is used to instruct the XDMS on how to handle the Forward XDCP Request if the size of the received content exceeds the size specified by the value of this element. The value specified is in bytes.
5) The <actions> element SHALL be the child element of the <forward-prefs> element which specifies the action to be taken by the XDMS on receiving the Forward XDCP Request from other user.
6) The <actions> element MAY include any of the following child elements:
    a) <accept> - instructs the XDMS to behave as if it had received aForward Accept XDCP Request from the receiving Principal automatically.
    b) <confirm> - instructs the XDMS to notify the user about the received Forward XDCP Request and await user's disposition.
    c) <reject> - instructs the XDMS to behave as if it had received a Forward Reject XDCP Request from the receiving Principal automatically .

Handling of XDCP ForwardAccept and ForwardReject XDCP Requests is specified in section 6.2.6.2.7 *"Handling of XDCP ForwardAccept and ForwardReject Requests"* and handling of received Forward XDCP Request based on recipient's preferences is specified in section 6.2.6.2.3 *"Handling of Received Forward XDCP Request Based on Recipient Preferences"*.

5.8.8 Naming Conventions

There is only one XDM Preferences Document per Application Usage and XUI in each XDMS. The XDM Preferences Document SHALL be addressed using the User Directory Document Selector "/oma_xdm_pref/preferences", i.e. the Document Selector to the XDM Preferences Document SHALL be "[auid]/users/[xui]/oma_xdm_pref/preferences" and the document name SHALL be "preferences".

5.8.9 Global Documents

No Global Documents are defined.

5.8.10 Resource Interdependencies

None.

5.8.11 Authorization Policies

Application Usages supporting XDM Preferences Document SHALL apply the authorization policies as described in section 5.1.5.
If the operation is not allowed the XDMS SHALL reject the operation with a HTTP "403 Forbidden" response if it is an XCAP operation or with a SIP "403 Forbidden" response if it is a SIP operation.

5.8.12 Subscription to changes

Not applicable.

5.8.13 Search Capabilities

Not applicable.

5.8.14 XDM Preferences Document

An XDM Preferences Document MAY support the XDM Preferences Document. The XDM Preferences, if any, are stored in the XDM Preferences Document itself.

5.8.15 History Information Documents

An XDM Preferences Document MAY support Modification History Document as described in section 5.7.1 "*Modification History Information Document*".

An XDM Preferences Document MAY support a Request History Information Document as described in section 5.7.2 "*Request History Information Document*".

5.8.16 Forwarding

Not applicable.

5.8.17 Restore

Restore of an XDM Preferences Document as described in section 6.2.6.5 "*XDM Restore*" MAY be supported.

5.8.18 Document Reference

Document Reference of an XDM Preferences Document as described in section 6.2.6.1 "*Document Reference*" MAY be supported.

5.8.19 Differential Read and Write

Not applicable.

6. Description of Procedures at XDM Functional Entities

6.1 Procedures at the XDMC and the XDM Agent

An XDMC and an XDM Agent are entities that accesses XDM Resources in an XDMS. An XDM Resource is identified via an HTTP URI following the conventions for constructing URIs in [RFC4825].

The XDMC SHALL support the following security functions:

1) authentication function described in section 5.1.1 *"Authentication"*;

2) Principal Identity Assertion function described in section 5.1.2 *"Principal Identity Assertion"*;

3) HTTP traffic protection function described in section 5.1.4 *"Integrity and Confidentiality Protection"*.

The XDMC SHALL, when generating HTTP requests, include the User-Agent HTTP header as defined in [RFC2616] with the value set to "XDM-client/OMA2.1" to indicate that the XDMC is compliant with this specification.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the XDMC MAY be implemented in a UE as defined in [3GPP-TS_23.228] and [3GPP2-X.S0013-002] respectively.

The XDM Agent SHALL support the following security function:

1) Principal Identity Assertion function described in section 5.1.2 "Principal Identity Assertion".

The XDM Agent SHALL, when generating HTTP requests, include the User-Agent HTTP header as defined in [RFC2616] with the value set to "XDM-client/OMA2.1" to indicate that the XDM Agent is compliant with this specification.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the XDM Agent MAY be implemented in an AS as defined in [3GPP-TS_23.228] and [3GPP2-X.S0013-002] respectively.

6.1.1 Document Management

The XDMC and the XDM Agent SHALL support document management as described in this subsection.

6.1.1.1 XDM URI Construction

An XCAP URI represents an XDM Resource (i.e an XML document, an element within an XML document or an attribute of an element with an XML document) stored in an XDMS. The rules for constructing such XCAP URIs SHALL follow the rules described in [RFC4825] section 6 with the clarifications given in this section.

NOTE 1: An XCAP URI would be of the form [XCAP Root URI]/[AUID]/users/[XUI]/… (See Appendix C for examples.)

The XCAP Root URI SHALL include host address of the Aggregation Proxy in the XDMC's home domain. The XDMC that resides in an UE SHALL use the XCAP Root URI provisioned to the XDMC as described in Appendix D "XDMC Provisioning".

The XDM Agent SHALL use the XCAP Root URI as preconfigured and it SHALL have the possibility to address the XDMS directly without going through the Aggregation Proxy; in this case, the XDM Agent SHALL be preconfigured per AUID with the host address of the XDMS, in addition to the XCAP Root URI.

The XDMC SHALL compare whether the XCAP Root URI of any XDM Resource to be accessed is the same as the XCAP Root URI that has been provisioned or preconfigured. If the validation fails, the XDMC SHALL replace the XCAP Root URI with the provisioned/preconfigured XCAP Root URI.

NOTE 2: The XDMC may become aware of XDM Resources having XCAP Root URI that differs from the one that is provisioned/preconfigured, e.g., via links.

The path segment corresponding to the XUI SHALL either be a User Address that is a SIP URI of form sip: user@domain or a Tel URI, e.g., tel:+1720-555-1212, identifying the Primary Principal of the XDM Document.

NOTE 3: If the User has multiple User Addresses available, each single User Address constitutes an independent and unrelated XUI unless two XUIs identify a Primary Principal and an associated Alias Principal. For example, if a user has two User Addresses of sip:user_public1@example.com and sip:user_public2@example.com, the XUIs of sip:user_public1@example.com and sip:user_public2@example.com represent two different XUIs unless the two XUIs identify a Primary Principal and an associated Alias Principal.

If the user has both a Tel URI and its associated SIP URI then the XDMC SHALL use the SIP URI in preference to the Tel URI as an XUI. Here the term 'associated' means that the Tel URI can be translated to the SIP URI and vice versa, for interchangeable usage in the SIP / IP Core e.g. the two XUIs identify a Primary Principal and an associated Alias Principal. Both the translation and the interchangeable usage are out of the scope of this specification. The XDMC MAY use the Application Usage "Alias Principals List" as defined in [XDM_List] section "*Alias Principals List*" to retrieve information about which Alias Principals that are associated with a Primary Principal.

If the Node Selector Separator is used in the XCAP URI, then:

1) The Node Selector Separator SHALL convey the meaning as defined in [RFC4825].

2) The Node Selector Separator SHALL appear only once, as a URI separator (i.e. in the form of "/~~/").

3) The Node Selector Separator SHOULD NOT be percent-encoded according to the procedures defined in [RFC 3986].

NOTE 4: Using double tilde or the percent-encoded format as part of a name is still allowed. For example, "/first~~last/", "/first~~/" and "/~~last/" are valid expressions.

6.1.1.2 XDM Operations using XCAP

An XDMC or an XDM Agent manipulates an XDM Resource by invoking certain HTTP operations (defined in sub-sections below) on the XDM Resource identified in the Request-URI of the HTTP header.

The XDMC and the XDM Agent SHALL construct the Request-URI based on its knowledge of the Application Usage governing that XDM Document.

The XDMC and the XDM Agent MAY implement the conditional operations of [RFC4825] section 7.11.

The XDMC MAY support HTTP compression using content encoding. If the XDMC utilizes HTTP compression, it SHALL set the Accept-Encoding header as defined in [RFC2616].

6.1.1.2.1 Create or Replace a Document

Creating or replacing an XDM Document SHALL follow the procedures described in [RFC4825] section 7.1.

6.1.1.2.2 Delete a Document

Deleting an XDM Document SHALL follow the procedures described in [RFC4825] section 7.2.

6.1.1.2.3 Retrieve a Document

Retrieving an XDM Document SHALL follow the procedures described in [RFC4825] section 7.3.

6.1.1.2.4 Create or Replace an Element

Creating or replacing an XDM Document Part being an XML element SHALL follow the procedures described in [RFC4825] section 7.4.

6.1.1.2.5 Delete an Element

Deleting an XDM Document Part being an XML element SHALL follow the procedures described in [RFC4825] section 7.5.

6.1.1.2.6 Retrieve an Element

Retrieving an XDM Document Part being an XML element SHALL follow the procedures described in [RFC4825] section 7.6.

NOTE: When an XDM Document Part is received as a result of a retrieve operation, the XDM Document Part does not always contain all needed namespace bindings. XDMCs and XDM Agents that do not already have knowledge about the namespace bindings must fetch these by doing a separate namespace binding request as defined in section 6.1.1.2.10.

6.1.1.2.7 Create or Replace an Attribute

Creating or replacing an XDM Document Part being an XML attribute SHALL follow the procedures described in [RFC4825] section 7.7.

6.1.1.2.8 Delete an Attribute

Deleting an XDM Document Part being an XML attribute SHALL follow the procedures described in [RFC4825] section 7.8.

6.1.1.2.9 Retrieve an Attribute

Retrieving an XDM Document Part being an XML attribute SHALL follow the procedures described in [RFC4825] section 7.9.

6.1.1.2.10 Fetch Namespace Bindings

Fetching namespace bindings of an XDM Document Part SHALL follow the procedures described in [RFC4825] section 7.10.

6.1.1.3 XDM Operations using XDCP

An XDMC or XDM Agent MAY support XDM operations as described in this section.

When performing an XDCP operation, the XDMC or XDM Agent SHALL issue an XDCP Request by using a HTTP POST request containing an XDCP Document as defined in section 5.4.2 "*XDCP Document*".

The HTTP Request-URI for the XDCP Request SHALL be set with any of the following values accordingly:

1) "http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]/users/[XUI]/[User Directory Document Selector]", if the XDCP Request is targeting the XDM Document identified by the XCAP URI "http://[XCAP Root URI]/[AUID]users/[XUI]/[User Directory Document Selector]".

2) "http://[XCAP Root URI]/org.openmobilealliance.xdcp.sp", if the XDCP Request is targeting the Subscription Proxy.

3) "http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]", if the XDCP Request is targeting the XDMS serving the specific Application Usage.

6.1.1.3.1 Document Reference Operations

When setting or modifying Document Reference information, the XDMC or XDM Agent SHALL make an XDCP Request containing an XDCP Document as described in section 5.4.2 with the following clarifications:

1) SHALL include <set-doc-ref> element as child element of the <request> element;

2) The <reference> element SHALL contain the Document Selector pointing to the XDM Document to be shared; and

3) The <display-name> element MAY contain a descriptive text of the reference used e.g. “Alice’s document”.

When removing all Document Reference information, the XDMC or XDM Agent SHALL make an XDCP Request containing an XDCP Document as described in section 5.4.2 with the following clarifications:

1) SHALL include a <remove-doc-ref> element as child element of the <request> element.

When retrieving Document Reference information, the XDMC or XDM Agent SHALL make an XDCP Request containing an XDCP Document as described in section 5.4.2 with the following clarifications:

1) SHALL include a <retrieve-doc-ref> element as child element of the <request> element.

As part of a Document Reference operation, the XDMC or XDM Agent SHALL set the HTTP Request-URI of the XDCP Request to the URI of the XDM Document containing the reference.

6.1.1.3.2 XDM Resource Forwarding Operations

When forwarding an XDM Resource, the XDMC or XDM Agent SHALL make an XDCP Request containing an XDCP Document as described in section 5.4.2 with the following clarifications:

1) SHALL include a <forward> element as child element of the <request> element;

2) SHALL include the <list> element as child element of <recipients-list> which carries the list of recipients to whom the XDM Resource has to be forwarded. The <list> element SHALL conform to the structure of <list> element specified in [RFC4826] with the following clarifications:

    a) The “name” attribute of the <list> element is not needed and is ignored by the XDMS even if present; and

    b) The <entry> element SHALL contain the "uri" attribute set to a valid User Address, e.g. a SIP URI (as defined in [RFC3261]) or a Tel URI (as defined in [RFC3966]).

3) MAY include <note> element which carries the information set by the Principal forwarding the XDM Resource wants to send to the recipients;

4) MAY include <expiration-time> element which can be used to indicate the time period of keeping the Forward XDCP Request active. The maximum and default time periods for keeping a Forward Request active SHALL be determined by the Service Providers local policy;

5) MAY include a <filter-set> element containing information about filters to be applied to the XDM Resource before forwarding it to the recipients. The <filter-set> element SHALL have a structure as defined by [RFC 4661] with the following clarifications:

    a) The <filter-set> element SHALL have a structure that include one or more <filter> element and MAY include a <ns-bindings> element but SHALL NOT include a “package” attribute;

    b) The <filter> element SHALL include one <what> element and a “id” attribute but SHALL NOT include a <trigger> element, a “uri” attribute, a “domain” attribute, an “enabled” attribute or a “remove” attribute;

    c) The <what> element MAY include one or more <include> elements and MAY include one or more <exclude> elements. If an <exclude> element is present then, if any <include> elements are present they SHALL include a “type” attribute with the value set to “namespace”; and

    d) The <include> and <exclude> element including a “type” attribute with the value “xpath” SHALL conform to what is described in Appendix I “*Filter ABNF*”.

    and;

6) MAY include <display-name> element which carries the name suggested by the Principal forwarding the XDM Resource to send to the recipients;

7) MAY include <delivery-report> element to indicate whether the Principal forwarding the XDM Resource wants to receive delivery report of the Forward XDCP Request from each of the recipients. The <delivery-report> SHALL contain one of the following values:

   - "true" - if the Principal forwarding the XDM Resource wants to receive the delivery report from each of the recipients.
   - "false" - if the Principal forwarding the XDM Resource does not want to receive the delivery report from each of the recipients. This is the default value taken in the absence of <delivery-report> element.

8) SHALL include <request-id> element which carries a unique identifier for a particular Forward XDCP Request if <delivery-report> element value is set to "true".

6.1.1.3.3 Handling of XDM Resource Forwarding Notifications

When child element of <actions> element of the <forward-prefs> element of the XDM Preferences Document defined in section 5.8 is the <confirm> element, the XDMS sets the <status> element of the <request> element in the Forwarding Notification List Document to "pending" before notifying the XDMC or XDM Agent about the received Forward XDCP Request. Upon receiving that notification, the XDMC or XDM Agent SHALL obtain receiving Principal's disposition for the Forward XDCP Request and act as follows:

If the disposition is "accept and store forwarded XDM Document in a default location", the XDMC or XDM Agent SHALL make an XDCP Request containing an XDCP Document as described in section 5.4.2 with the following clarifications:

1) The HTTP Request-URI for the XDCP Request SHALL be set to: "http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]" where the AUID corresponds to the Application Usage of the forwarded document;

2) SHALL include a < forward-accept > element as child element of the <request> element; and

3) SHALL include the <document-uri> element set to the value of "document-uri" attribute of the <request> element contained in the Forwarding Notification List Document defined in [XDM_List] section 5.3, and as specified in section 6.2.6.2.4 *"Notifying the Recipients about the Status of the Received Forward XDCP Request"*.

NOTE: The default location is determined by local policy or by the Application Usage of the forwarded XDM Document.

If the disposition is "accept without storing forwarded XDM Document in the XDMS", the XDMC or the XDM Agent SHALL make an XDCP Request containing an XDCP Document as described in section 5.4.2 and SHALL perform steps 1-3 of the "accept and store forwarded XDM Document in a default location" scenario, along with the following clarifications:

1) SHALL include a <do-not-store> element.

If the disposition is "accept and store forwarded XDM Document in a location specified by the XDMC or XDM Agent", the XDMC or the XDM Agent SHALL make an XDCP Request containing an XDCP Document as described in section 5.4.2 and SHALL perform steps 1-3 of the "accept and store forwarded XDM Document in a default location" scenario, along with the following clarifications:

1) SHALL include a <store> element with an "udds" attribute containing the User Directory Document Selector to the location in the XDMS where the XDMC or XDM Agent requires the document to be stored.

If the disposition is "reject", the XDMC or XDM Agent SHALL make an XDCP Request containing an XDCP Document as described in section 5.4.2 with the following clarifications:

1) The HTTP Request-URI for the XDCP Request SHALL be set to: "http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]" where the AUID corresponds to the Application Usage of the forwarded document;

2) SHALL include a < forward-reject > element as child element of the <request> element; and

3) SHALL include the <document-uri> element set to the value of "document-uri" attribute of the <request> element contained in the Forwarding Notification List Document defined in [XDM_List] section 5.3, and as specified in section 6.2.6.2.4 *"Notifying the Recipients about the Status of the Received Forward XDCP Request"*.

6.1.1.3.4 Subscription to Changes in XDM Resources

When subscribing to changes in XDM Resources, the XDMC SHALL make an XDCP Request containing an XDCP Document as described in section 5.4.2 with the following clarifications:

1) SHALL include a <subscribe> element as child element of the <request> element;

2) The <target-documents> element of the <subscription> element SHALL contain a <resource-uri> element containing XCAP URIs of XDM Resources of interest, which MAY include a <filter-set> element per [RFC4661] and/or, a <resource-lists> element as specified in [RFC4826] containing a list of XDM Resources of interest, which MAY include a <filter-set> element per [RFC4661], and/or a <list-uri> element containing URIs of already existing resource lists containing XDM Resources of interest. Note that the <uri> attribute SHALL be set to the Document Selector that selects the XDM Document to which the filter applies.

   The <filter-set> element SHALL have a structure as defined by [RFC 4661] with the following clarifications:
   a) The <filter-set> element SHALL have a structure that include one or more <filter> element, and MAY include a <ns-bindings> element but SHALL NOT include a "package" attribute;

   b) The <filter> element SHALL include one <what> element and a "id" attribute, but SHALL NOT include a <trigger> element, a "uri" attribute, a "domain" attribute, an "enabled" attribute, nor a "remove" attribute;

   c) The <what> element MAY include one or more <include> elements, and MAY include one or more <exclude> elements. If an <exclude> element is present then, if any <include> elements are present, they SHALL include a "type" attribute with the value set to "namespace"; and

   d) The <include> and <exclude> element including a "type" attribute with the value "xpath" SHALL conform to what is described in Appendix I "*Filter ABNF*".

NOTE: The mechanism used by the XDMC to retrieve the Document Selector, Node Selector or part of the Document Selector used to identify the collection as described in [RFC5875] of the XDM Resources to be watched is out of scope of the present specification.

3) The <duration> element of the <subscription> element SHALL contain the duration of the subscription in seconds;

4) The <gzip> element of the <subscription> element SHOUD be included if the XDMC supports the GZIP algorithm [RFC1952] processing of push document change notifications;

5) The <notification> element of the <subscription> element SHALL contain a <push> element;

6) The <push-address> element of the <push> element SHALL contain the Tel URI to which the notification of changes will be sent;

7) The <wap-application-id> element of the <push> element SHALL contain the OMNA registered application identifier of the application acting as an XDMC;

8) The <user-interaction-level> element of the <push> element SHALL contain the level of user interaction required for processing notifications. If the value is set to "none", which is defined to be the default value of this element, the notifications will be processed in the background without user interaction. If the value is set to "low", "medium" or "high" the user will be prompted with corresponding degrees of urgency (i.e., low, medium, or high). The values of this element correspond to the values of the "action" attribute associated with the <indication> element of the "Service Indication" type of the Push OTA Message as specified in [Push_ERELD-V2_2]; and,

9) The <preferred-notification-type> element of the <push> element SHALL specify the preferred contents of the Push OTA Message notifications.

a) If this element is set to "push", the XDMC expects notifications containing XDM Document changes in the format of "XCAP-Diff" MIME type [RFC5874] as part of the Push OTA Message payload.

b) If this element is set to "pull", the XDMC expects notifications containing as part of the Push OTA Message payload the XCAP URI of another document that containins document changes in XCAP-Diff format.

c) If this element is set to "none", the XDMC expects notifications containing as part of the Push OTA Message payload the XCAP URI of the changed XDM Document.

6.1.1.3.5 Differential Read

The XDMC or XDM Agent MAY issue a Differential Read XDCP Request to retrieve the difference between the current (i.e. latest version in XDMS) and the version of the XDM Document identified by the supplied E-Tag.

When performing the Differential Read, the XDMC or XDM Agent SHALL create and send an XDCP Request as follows:

1) SHALL include the XDCP Document with the following clarifications:

   a) SHALL include a <diff-read> element as child element of the <request> element;

   b) SHALL either

      i. include the <etag> element containing the E-Tag value of the XDM Document that the XDMC or XDM Agent currently possesses, if the XDMC or XDM Agent seeks the difference between the version of the E-Tag and the current version on the XDMS; or,

      ii. omit the <etag> element, if the XDMC or XDM Agent seeks to read the entire XDM Document according to an included <filter-set>;

   c) MAY include a <filter-set> element with the structure as described in [RFC 4661] to perform Differential Read on a subset of the target XDM Document with the following clarifications:

      i. The <filter-set> element SHALL have a structure that include one or more <filter> elements, and MAY include a <ns-bindings> element, but SHALL NOT include a "package" attribute;

      ii. The <filter> element SHALL include one <what> element and an "id" attribute, but SHALL NOT include a <trigger> element, a "uri" attribute, a "domain" attribute, an "enabled" attribute, nor a "remove" attribute;

      iii. The <what> element MAY include one or more <include> elements and MAY include one or more <exclude> elements. If an <exclude> element is present then, if any <include> element is also present then the <include> element SHALL include a "type" attribute with the value set to "namespace";and

      iv. The <include> and <exclude> element including a "type" attribute with the value "xpath" SHALL conform to what is described in Appendix I "*Filter ABNF*".

Once an XDMC or XDM Agent uses a filter-set to read an XDM Document, the XDMC or XDM Agent SHALL use the same filter-set until the XDMC or XDM Agent refreshes the entire XDM Document or a new subset of the XDM Document.

6.1.1.3.6 Differential Write

The XDMC or XDM Agent MAY use the Differential Write XDCP Request to specify changes that need to be applied to the XDM Document identified in the Request URI of the XDCP Request.

When performing the Differential Write, the XDMC or XDM Agent SHALL create and send an XDCP Request as follows:

1) SHALL include the XDCP Document as the first part of a multipart MIME body with the following clarifications:

   a) SHALL include a <diff-write> element as child element of the <request> element;

b) MAY include a <filter-set> element with the structure as described in [RFC 4661] to perform Differential Write on a subset of the target XDM Document with the following clarification:

i. The <filter-set> element SHALL have a structure that includes one or more <filter> elements, and MAY include a <ns-bindings> element, but SHALL NOT include a "package" attribute;

ii. The <filter> element SHALL include one <what> element and an "id" attribute but SHALL NOT include a <trigger> element, a "uri" attribute, a "domain" attribute, an "enabled" attribute, nor a "remove" attribute;

iii. The <what> element MAY include one or more <include> elements and MAY include one or more <exclude> elements. If an <exclude> element is present then, if any <include> element is also present then the <include> element SHALL include a "type" attribute with the value set to "namespace"; and

iv. The <include> and <exclude> element including a "type" attribute with the value "xpath" SHALL conform to what is described in Appendix I "*Filter ABNF*".

2) SHALL include an XCAP Diff document as described in [RFC5874] as the second part of a multipart MIME body with the following clarification:

a) SHALL include a "previous-etag" attribute in a single XCAP Diff document <document> element.

Once an XDMC or XDM Agent uses a filter-set, the XDMC or XDM Agent SHALL use the same <filter-set> element until the XDMC or XDM Agent entirely refreshes the entire XDM Document. Furthermore the XDMC or XDM Agent SHALL perform a Differential Read XDCP Request with a filter before performing any Differential Write XDCP Request with the same filter.

6.1.1.3.7 XDM Restore

When restoring an XDM Document, the XDMC or XDM Agent SHALL make an XDCP request containing an XDCP Document as described in section 5.4.2 with the following clarifications:

1) SHALL include a <restore> element as child element of the <request> element;

2) SHALL include the <back-to-etag> element containing the E-Tag value of XDM Document which is requested to restore to;

3) MAY include the <current-etag> element containing the value of the E-Tag of the XDM Document the XDMC or XDM Agent currently possesses.

6.1.2 Subscribing to Changes in the XDM Resources

An XDMC and an XDM Agent MAY support subscription to changes in XDM Resources as described in this subsection. Two mechanisms for subscription to changes are supported: SIP and XDCP/Push. While SIP subscriptions and notifications apply to both the XDMC and the XDM Agent, XDCP/Push based subscriptions and notifications apply only to the XDMC.

6.1.2.1 Initial SIP Subscription

If the XDMC or the XDM Agent subscribes to changes in XDM Resources using SIP, then it SHALL be done by sending a SIP SUBSCRIBE request according to [RFC3265] and [RFC5875] with the clarifications given in the section 6.1.2.1.1 for an XDM Agent and section 6.1.2.1.2 for an XDMC.

The responses to the SIP SUBSCRIBE request SHALL be handled in accordance with [RFC3265], [RFC5875], and the procedures of the SIP/IP Core.

6.1.2.1.1 XDM Agent

An XDM Agent SHALL subscribe for the notification of changes in XDM Resources either via the Subscription Proxy, when XDM Resources from different AUIDs or different users are to be subscribed for, or directly to the XDMS, when XDM Resources from a single AUID and user are to be subscribed for.

When the XDM Agent subscribes for the notification of changes in the XDM Resources from multiple AUIDs or users, the XDM Agent:

1) SHALL set the Request-URI to the preconfigured SIP URI identifying the Subscription Proxy;

2) SHALL include in the body of the SIP SUBSCRIBE request a list of XCAP URIs pointing to all XDM Resources it is subscribing to. The format of the list is specified in [RFC5875], and MAY include a <filter-set> element contained in the MIME type "application/simple-filter+xml" per [RFC4660] and [RFC4661]. The <uri> attribute of a <filter> element in the <filter-set> element SHALL be set to the Document Selector that selects the XDM Document to which the filter applies;

   The <filter-set> element SHALL have a structure as defined by [RFC 4661] with the following clarifications:

   a) The <filter-set> element SHALL have a structure that include one or more <filter> element and MAY include a <ns-bindings> element but SHALL NOT include a “package” attribute;

   b) The <filter> element SHALL include one <what> element and an “id” attribute but SHALL NOT include a <trigger> element, a “uri” attribute, a “domain” attribute, an “enabled” attribute or a “remove” attribute;

   c) The <what> element MAY include one or more <include> elements and MAY include one or more <exclude> elements. If an <exclude> element is present then, if any <include> elements are present they SHALL include a “type” attribute with the value set to “namespace”; and

   d) The <include> and <exclude> element including a “type” attribute with the value “xpath” SHALL conform to what is described in Appendix I “*Filter ABNF*”;

NOTE: The mechanism used by the XDM Agent to retrieve the Document Selector, Node Selector or part of the Document Selector used to identify the collection as described in [RFC5875] of the XDM Resources to be watched is out of scope of the present specification.

3) SHALL include Accept header fields with the following values:

   a) “application/xcap-diff+xml” to indicate the support for partial XML updates as described in [RFC5875];

   b) “multipart/related”; and

   c) “application/rlmi+xml” to indicate the support for the event notification extension for resource list as described in [RFC4662].

4) MAY indicate that it supports that the body of a SIP NOTIFY request is compressed by the GZIP algorithm [RFC1952] by including an Accept-Encoding header field with the value “gzip” in the SIP SUBSCRIBE request; and

NOTE: If this is a subscription to document change notifications applicable for a large XCAP element node, then subscribe to the entire XDM Document and use a filter to select the particular XCAP element node of interest. This technique prevents XCAP element components from being sent in notifications associated with an initial or refreshing type subscription request, and is advantageous for purposes of saving network resources when the XDM Agent already has a locally cached copy that is up-to-date (i.e., does not need to be conveyed) or the XDM Agent can use XDCP Differential Read operations to catch-up via document changes.

5) SHALL send the SIP SUBSCRIBE request towards the SIP/IP Core according to the procedures of the SIP/IP Core.

When the XDM Agent subscribes for the notification of changes in the XDM Resource(s) from a single AUID and user or a single service instance, the XDM Agent:

1) SHALL set the Request-URI to either the XUI part of the XCAP URI pointing to the XDM Resource(s) or to the SIP or Tel URI identifying the service instance (e.g. Group URI), and set the "auid" parameter defined in Appendix E.1 "*AUID URI Parameter*" to the value of AUID of the requested XDM Resource(s);

2) SHALL include all relevant XDM Resources in the body of the SIP SUBSCRIBE request as described in [RFC5875];

3) SHALL include an Accept header field with the value "application/xcap-diff+xml" to indicate support for partial XML updates described in [RFC5875]; and

NOTE: If this is a subscription to document change notifications applicable for a large XCAP element node, then subscribe to the entire XDM Document and use a filter to select the particular XCAP element node of interest. This technique prevents XCAP element components from being sent in notifications associated with an initial or refreshing type subscription request, and is advantageous for purposes of saving network resources when the XDM Agent already has a locally cached copy that is up-to-date (i.e., does not need to be conveyed) or the XDM Agent can use XDCP Differential Read operations to catch-up via document changes.

4) SHALL send the SIP SUBSCRIBE request towards the SIP/IP Core according to the procedures of the SIP/IP Core.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD an XDM Agent SHALL use 3GPP IMS or 3GPP2 MMD requirements respectively, mechanisms and procedures as defined in section 5.7.3 [3GPP-TS_24.229] /[3GPP2-X.S0013-004] with the clarifications given in the respective sub clauses.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD, the XDM Agent SHALL apply the mechanisms of the "Application Server acting as originating User Agent" as defined in [3GPP-TS_24.229]/[3GPP2-X.S0013-004] section 5.7.3 and the XDM Agent SHALL set its Public User Identity or the Public User Identity of the Principal on which the XDM Agent is acting on behalf of in the P-Asserted-Identity header.

6.1.2.1.2 XDMC

The XDMC:

1) SHALL set the Request-URI to the SIP URI of the Subscription Proxy if that SIP URI was provisioned to the XDMC as described in Appendix D "*XDMC Provisioning*"; or

2) SHALL set the Request-URI to the XUI part of the XCAP URI pointing to the XDM Resource(s) to be subscribed for and set the "auid" parameter defined in Appendix E.1 "*AUID URI Parameter*" to the value of AUID of the requested XDM Resouce(s), if the SIP URI of the Subscription Proxy was not provisioned to the XDMC;

3) SHALL include in the body of the SIP SUBSCRIBE request a list of XCAP URIs pointing to all XDM Resources it is subscribing to. The format of the list is specified in [RFC5875], and MAY include a <filter-set> element contained in the MIME type "application/simple-filter+xml" per [RFC4660] and [RFC4661]; Note that the <uri> attribute SHALL be set to the Document Selector that selects the XDM Document to which the filter applies. In case that the SIP URI of the Subscription Proxy was not provisioned to the XDMC, the XDM Resources subscribed in a single subscription SHALL be from a single AUID and single user;

Furthermore: The <filter-set> element SHALL have a structure as defined by [RFC 4661] with the following clarifications:

a) The <filter-set> element SHALL have a structure that include one or more <filter> element and MAY include a <ns-bindings> element but SHALL NOT include a "package" attribute;

b) The <filter> element SHALL include one <what> element and an "id" attribute but SHALL NOT include a <trigger> element, a "uri" attribute, a "domain" attribute, an "enabled" attribute or a "remove" attribute;

c) The <what> element MAY include one or more <include> elements and MAY include one or more <exclude> elements. If an <exclude> element is present then, if any <include> elements are present they SHALL include a "type" attribute with the value set to "namespace"; and

d) The <include> and <exclude> element including a "type" attribute with the value "xpath" SHALL conform to what is described in Appendix I "*Filter ABNF*"..

NOTE: The mechanism used by the XDMC to retrieve the Document Selector, Node Selector or part of the Document Selector used to identify the collection as described in [RFC5875] of the XDM Resources to be watched is out of scope of the present specification.

4) SHALL include an Accept header field with the value "application/xcap-diff+xml" to indicate the support for partial XML updates as described in [RFC5875RFC5874];

5) SHALL include Accept header fields with the following values if the SIP URI of the Subscription Proxy was provisioned to the XDMC as described in Appendix D "*XDMC Provisioning"*:

a) "multipart/related" and

b) "application/rlmi+xml" to indicate support for the event notification extension for resource lists described in [RFC4662].

6) MAY indicate that it supports that the body of a SIP NOTIFY request is compressed by the GZIP algorithm [RFC1952] by including an Accept-Encoding header field with the value "gzip" in the SIP SUBSCRIBE request; and,

NOTE: If this is a subscription to document change notifications applicable for a large XCAP element node, then subscribe to the entire XDM Document and use a filter to select the particular XCAP element node of interest. This technique prevents XCAP element components from being sent in notifications associated with an initial or refreshing type subscription request, and is advantageous for purposes of saving network resources when the XDMC already has a locally cached copy that is up-to-date (i.e., does not need to be conveyed) or the XDMC can use XDCP Differential Read operations to catch-up via document changes.

7) SHALL send the SIP SUBSCRIBE request towards the SIP/IP Core according to the procedures of the SIP/IP Core.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD, a UE acting as the XDMC SHALL use 3GPP IMS or 3GPP2 MMD requirements respectively, mechanisms and procedures as defined in section 5.1 in [3GPP-TS_24.229] / [3GPP2-X.S0013-004].

6.1.2.2 SIP NOTIFY Processing

Upon receiving an incoming SIP NOTIFY request that is part of the same dialog as the previously sent SIP SUBSCRIBE request the XDMC and the XDM Agent:

1) SHALL handle the request according to [RFC3265], [RFC5875], [RFC4662] and the procedures of the SIP/IP Core; and

2) SHOULD update the stored XDM Document based on the information in the SIP NOTIFY request. The structure of the information in the body of SIP NOTIFY request is defined [RFC5875], [RFC4662] and section 6.6.

An XDMC or an XDM Agent indicating support for GZIP compression SHALL, when receiving a SIP NOTIFY request with the Content-Encoding header field with the value "gzip", decompress the received body as defined by [RFC1952] before performing processing of the notification.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD, the XDMC and the XDM Agent SHALL use 3GPP IMS or 3GPP2 MMD requirements respectively, mechanisms and procedures as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] with the clarifications given in this sub-clause.

6.1.2.3 Initial XDCP Subscription

If the XDMC subscribes to changes in XDM Resources using XDCP, then it SHALL be carried out by sending an XDCP Request as specified in section 6.1.1.3 with the following clarifications:

1) The XDMC SHALL set the HTTP Request-URI to: "http://[XCAP Root URI]/org.openmobilealliance.xdcp.sp"; and

2) The XDMC SHALL provide as the payload of the request the XDCP Document as specified in section 6.1.1.3.4.

Due to scarce spectrum and potential Push Over-The-Air Message size limitations, the XDMC SHOULD retrieve XDM Resources prior to creating an XDCP Subscription to those resources. The XDCP Subscription <preferred-notification-type> element being set to "pull" or "none" will serve to limit the size of the Push OTA Message.

Furthermore, if a subscription to document change notifications for XCAP element nodes is desired, the XDMC SHOULD subscribe to the entire XDM Document, and then include a filter in the XDCP Subscribe Request to select the particular XCAP element node of interest.

NOTE: This technique prevents XCAP element components from being sent in notifications associated with an initial or refreshing type subscription request, which is advantageous if the XDMC already has a locally cached copy that is up-to-date (i.e., does not need to be conveyed) or the XDMC can use XDCP Differential Read operations to catch-up via document changes.

6.1.2.4 PUSH OTA Message Processing

Upon receiving an incoming Push OTA Message, the XDMC:

1) SHALL determine the XDM Resource to which the Push OTA Message applies; and

2) SHALL process the received Push OTA Message based on the <preferred-notification-type> set in the XDCP Subscribe request and update the locally stored XDM Document, if necessary.

NOTE: The structure of the information in the body request is defined in [RFC5874] and in section 6.6.5.

If an XDMC indicates support for GZIP compression in the <gzip> element of the <subscription> element of the XDCP Documents of the initial XDCP Subscription Request, the XDMC SHALL, when receiving a Push OTA Message with a Content-Encoding header field of value "gzip", decompress the received body as defined by the GZIP algorithm [RFC1952] before performing processing of the Push OTA Message.

6.1.3 Searching for Data in XML Documents

An XDMC and an XDM Agent MAY support searching for data in XDM Documents using Limited XQuery over HTTP as described in this subsection.

When performing a search operation, the XDMC and the XDM Agent SHALL generate the Search Request by using HTTP POST request containing a Search Document as defined in section 5.4.1 "*Search Document*".

The HTTP Request-URI for the Search Request SHALL be constructed as http://[XCAP Root URI]/org.openmobilealliance.search. For routing purposes, the HTTP Request-URI of the Search Request SHALL include the mandatory query parameter of "target" and whose value is equal to the parameter of the collection input function of the XQuery request in the Search Document that identifies the XDM Document(s) to be searched as described in section 5.4.1 *"Search Document"*. When the search target is the set of all XDM Documents stored in the Users Tree or in the Global Tree of an appropriate Application Usage, the domain to be searched needs to be specified. For the identification of target search domain, the HTTP Request-URI of the Search Request MAY include the optional query parameter of "domain" and whose value includes 'home' to request home domain search, 'all' to request to expand the search to all possible remote domains, or target domain names to request the particular domain to be searched. Multiple values are separated using the percent encoded whitespace (i.e., "%20") as specified in [RFC3986]. The default interpretation in the absence of "domain" query parameter SHALL be home domain search.

When using the "target" and "domain" query parameter, the HTTP Request-URI for the Search Request SHALL be constructed as http://[XCAP Root URI]/org.openmobilealliance.search?target=[value of collection input function]&domain=[home, all, or target domains].

Example: http://xcap.example.com/org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/&domain=all

Example: http://xcap.example.com/org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/&domain=home%20example.com%20example2.com

The Search Request SHALL include the XML body of the content type "application/vnd.oma.search +xml" as defined in section 5.4.1.1 *"MIME Type"*.

The XQuery expression in the Search Request SHALL conform to the constraints as defined by the target Application Usage to be searched for.

The XDMC and the XDM Agent MAY limit the number of Search results using the optional "max-results" attribute of the <search> element in the Search Document.

6.1.4 Retrieval of History Information

An XDMC or XDM Agent MAY support retrieval of History Information XDM Documents as described in this section.

6.1.4.1 Modification History Information

An XDMC or an XDM Agent MAY support the retrieval of Modification History Information Documents as described in section 5.7.1. If supported the XDMC or the XDM Agent SHALL use an XCAP URI of the form [XCAPRootURI]/[AUID]/users/[XUI]/oma_hist/[UserDirectory Document Selector] as described in section 5.7.1.8 to address the Modification History Document related to an XDM Document addressed using an XCAP URI of the form [XCAPRootURI]/[AUID]/users/[XUI]/[User Directory Document Selector] and SHALL use the procedures described in section 6.1.1.

6.1.4.2 Request History Information

An XDMC or an XDM Agent MAY support the retrieval of Request History Information Documents as described in section 5.7.2. If supported the XDMC or the XDM Agent SHALL use an XCAP URI of the form [XCAPRootURI]/[AUID]/users/[XUI]/oma_requests/history as described in section 5.7.2.8 to address the Request History Information Document in the User Directory and SHALL use the procedures described in sections 6.1.1 and 6.1.2.

6.1.4.2.1 Reactive authorization using Request History Information

An XDMC or an XDM Agent MAY subscribe to the Request History Information Document and use notifications of unauthorized XDM Requests targeting an XDM Resource in the User Directory to implement reactive authorization. An example of an implementation is shown in Appendix G "*Reactive Authorization of XDM Requests using Request History Information*"

6.1.5 Management of Access Permissions

An XDMC or an XDM Agent MAY support management of Access Permissions Documents as described in section 5.6. If supported, the XDMC or the XDM Agent SHALL use an XCAP URI of the form [XCAPRootURI]/[AUID]/users/[XUI]/oma_ap/access-permissions as described in section 5.6.8 to address the Access Permissions Document in the User Directory and SHALL use the procedures described in sections 6.1.1 and 6.1.2.

6.1.6 Management of XDM Preferences

An XDMC or an XDM Agent MAY support mangement of XDM Preferences Documents as described in section 5.8. If supported, the XDMC or the XDM Agent SHALL use an XCAP URI of the form [XCAPRootURI]/[AUID]/users/[XUI]/oma_xdm_pref/preferences as described in section 5.8.8 to address the XDM Preferences Document in the User Directory and SHALL use the procedures described in sections 6.1.1 and 6.1.2.

6.1.7 XCAP Server Capabilities Retrieval

The XDMC MAY support the retrieval of an XDM Capabilities Document described in section 5.3.1 using retrieve operations as described in sections 6.1.1.2.3 and 6.1.1.2.6.

6.1.8 XCAP Directory Information Retrieval

The XDMC or XDM Agent MAY support the retrieval of an XDM Directory Document as described in section 5.3.2 using procedures defined in section 6.1.1.2.3.

The XDMC or XDM Agent MAY support the retrieval of the XDM Document Part of an XDM Directory Document that corresponds to a particular Application Usage as described in section 5.3.2 by retrieving a <folder> element with the particular "auid" attribute set to the AUID value of the Application Usage using procedures defined in section 6.1.1.2.6.

6.2 Procedures at the XDM Server

An XDMS is a HTTP origin server that manipulates XDM Resources according to the conventions described in [RFC4825], and processes Search Requests.

An XDMS SHALL authorize the requests as described in section 5.1.5 *"Authorization"*.

An XDMS receiving an HTTP POST request containing an HTTP Request-URI of the form http://[XCAP Root URI]/org.openmobilealliance.search SHALL process the request as described in section 6.2.3 "*Searching for Data in XML Documents*". An XDMS receiving an HTTP POST request containing an HTTP Request-URI of the form http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]/users/[XUI]/[User Directory Document Selector] or http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]" SHALL process the request as described in section 6.2.6 "*Handling of XDCP Operations*". The XDMS SHALL reject any other HTTP POST requests with an HTTP "405 Method not allowed" response.

An XDMS receiving an XCAP Request SHALL process the request as described in section 6.2.1 "*Document Management*".

When generating HTTP responses, the XDMS MAY include a Server HTTP header as defined in [RFC2616] with the value set to "XDM-serv/OMA2.1" to indicate that the XDMS is compliant with this specification.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the XDMS SHALL be implemented in an AS as defined in [3GPP-TS_23.228] and [3GPP2-X.S0013-002] respectively.

6.2.1 Document Management

The XDMS SHALL support document management as described in this subsection.

An XDMS SHALL conform to [RFC4825] section 8.5 for the management of E-Tags.

An XDMS SHALL implement the conditional operations of [RFC4825] section 7.11.

If the XDMS implements parallel processing of requests, it SHALL ensure the integrity of the resulting XDM Document.

An XDMS SHALL check if the XUI of an XCAP Request is an identity of an Alias Principal associated with a Primary Principal and execute the XCAP Request as if the XDMS received an XCAP Request with an XUI identifying the Primary Principal.

6.2.1.1 PUT Handling

HTTP PUT requests targeted at an XDM Resource SHALL be processed as described in [RFC4825] section 8.2.

Additional validation constraints might be applied which may result in a HTTP "409 Conflict" error response. An HTTP "409 Conflict" error response SHALL include an XML document in the HTTP body that conforms to that defined in [RFC4825] section 11 and the extensions defined in this specification section 5.2.3 *"Detailed Conflict Reports*".
For additional details of the handling of those, see [RFC4825] section 8.2.5 and this specification section 5.2.3 *"Detailed*

Conflict Reports". Other specifications MAY define the value of the "phrase" attribute, which contains text for rendering to a human user, that is optionally present in an error element identifying an error condition.

6.2.1.2 GET Handling

HTTP GET requests targeted at an XDM Resource SHALL be processed as described in [RFC4825] section 8.3.

6.2.1.3 DELETE Handling

HTTP DELETE requests targeted at an XDM Resource SHALL be processed as described in [RFC4825] section 8.4.

6.2.2 Subscriptions to Changes in XDM Resources

The XDMS MAY support subscription to changes in XDM Resources as described in this subsection. If subscription to changes is not supported the XDMS SHALL return appropriate error response upon reception of a SIP SUBSCRIBE request for the "xcap-diff" event defined in [RFC5875].

6.2.2.1 Initial Subscription

Upon receiving a SIP SUBSCRIBE request for the "xcap-diff" event defined in [RFC5875] the XDMS:

1) SHALL perform necessary checks on the XCAP resources listed in the body of the SUBSCRIBE request. In case that any resource is not recognized as resource from appropriate Application Usage, the XDMS SHALL return the SIP "404 Not found" error response;

2) SHALL perform the necessary authorization checks on the originator. When the SIP/IP Core corresponds to 3GPP IMS or 3GPP2 MMD the XDMS SHALL use the "P-Asserted-Identity" as defined in [3GPP-TS_24.229]/[3GPP2-X.S0013-004] to ensure that this particular XDMC is authorized to track the XDM Document changes. If the authorization check fails, the XDMS SHALL return the SIP "403 Forbidden" error response;

    a) For XDM Documents in the Users tree, by default the Primary Principal and an associated Alias Principal of the XDM Document SHALL be authorized to subscribe to the "xcap-diff" event package;

    b) For XDM Documents in the Global Tree, other principals (e.g. XDMCs and XDM Agents) MAY be authorised to subscribe based on local policy or other Enabler-specific policy;

    c) Additional authorization policy MAY be defined for an Application Usage in the respective application-specific XDM Technical Specifications.

3) SHALL create a subscription to changes of XDM Documents listed in the body of the SIP SUBSCRIBE request as described in [IETF-XCAP_Diff_Event] with the following clarification:

    a) For the collection "/[AUID]/users/[XUI]/" the XDMS SHALL create a subscription to all XDM Documents in the User Directory excluding Supporting XDM Documents;

4) SHALL send a SIP "200 OK" in accordance with [RFC3265], [RFC5875], and the procedures of the SIP/IP Core;

5) SHALL generate and send an initial SIP NOTIFY request as specified in section 6.2.2.2 "*Generating a SIP NOTIFY request*".

When a change in the subscribed XDM Document occurs, the XDMS SHALL generate and send a SIP NOTIFY request as specified in section 6.2.2.2 "*Generating a SIP NOTIFY request*".

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD, the XDMS SHALL use 3GPP IMS or 3GPP2 MMD requirements respectively, mechanisms and procedures as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] with the clarifications given in this section.

6.2.2.2 Generating a SIP NOTIFY Request

If the "xcap-diff" event is supported the XDMS SHALL generate a SIP NOTIFY request as described in the [RFC3265] and [IETF-XCAP_Diff_Event] with the clarifications given in this section.

The XDMS

1) SHALL if the SIP Subscribe contained a body with a MIME Type apply the filter to the patches of the XDM Document and only keep patches that relates to the, by the filter defined ,XDM Document;

2) SHALL include an "application/xcap-diff+xml" body as defined in [RFC5874];

3) SHALL check if a <rule> element that granted access to the subscribe operation also included a <filter-set> element. If a <filter-set> element is included, the XDMS SHALL apply the filter and remove all information from the body of the SIP NOTIFY, that the filter prevents the requesting Principal to monitor before continuing the next step; and

4) SHALL send the SIP NOTIFY request towards the SIP/IP Core according to the procedures of the SIP/IP Core.

The responses to the SIP NOTIFY request SHALL be handled in accordance with [RFC3265], [RFC5875], and the procedures of the SIP/IP Core.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD, the XDMS SHALL use 3GPP IMS or 3GPP2 MMD requirements respectively, mechanisms and procedures as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] with the clarifications given in this section.

6.2.3 Searching for Data in XML Documents

The XDMS MAY support searching for data in XDM Documents using Limited XQuery over HTTP as described in this section.

The Search Request SHALL contain a Search Document as defined in section 5.4.1 "*Search Document*".

Upon receiving the Search Request, the XDMS:

1) SHALL generate an HTTP 400 "Bad Request" response and not continue with the following steps if the Search Request does not contain a body;

2) SHALL generate an HTTP 415 "Unsupported Media Type" and not continue with the following steps if the MIME type indicated in the Search Request is not "application/vnd.oma.search+xml";

3) SHALL generate an HTTP 400 "Bad Request" and not continue with the following steps if the Search Document included in the Search Request does not conform to the structure defined in section 5.4.1 "*Search Document*";

4) SHALL get the AUID from the "collection" input function of the XQuery and based on this AUID validate the XQuery expression included in the body of the Search Request against the XQuery restrictions as defined by the corresponding target Application Usage of the XDMS;

5) SHALL generate an HTTP "409 Conflict" error response containing a <constraint-failure> error condition element as defined in [RFC4825] and not continue with the step 6 if validation in the previous step fails. If the basic XQuery expressions as defined by the corresponding Application Usage do not allow:

    a) the Search operation as requested in the included XQuery expression, the "phrase" attribute, if it is included, SHOULD be set to "Search request not allowed";

    b) the types of Search Result as requested in the included XQuery expression, the "phrase" attribute, if it is included, SHOULD be set to "Search result types not allowed".

6) SHALL, if the "collection" input of the XQuery contains an XUI and if the XUI is an identity of an Alias Principal associated with a Primary Principal, act as if it received a Search Request with an XUI identifying the Primary Principal;

7) SHALL, for each XDM Document stored in the Users Tree or Global Tree of the corresponding Application Usage included in the "collection" input function of the XQuery request:

   a) execute the query; and

   b) verify the search result against corresponding Access Permissions Document if such exist and remove all XDM Document Parts that the requesting Principal is not authorized to retrieve as result of a Search Request as described in sections 5.1.5 and 5.6.

8) SHALL generate a response aggregating the results of the previous step, with the following precisions:

   a) In case that "max-results" attribute is included in the Search Request, the XDMS SHALL include in the response only the number of results of the Search up to and including the value specified in the "max-results" attribute;

   b) The XDMS MAY restrict the number of results of the Search based on local policy.

Each Application Usage that supports the Search feature SHALL define one or more basic XQuery expressions that are supported by the Application Usage. Such basic XQuery expressions allows the Application Usage to restrict the data that can be searched and also restrict the results provided to an XDMC or an XDM Agent.

6.2.4 Handling of Access Permissions Documents

6.2.4.1 Creating the Access Permissions Document

If the Application Usage defines an Access Permissions Document, the XDMS SHALL create a single Access Permissions Document as described in section 5.6 for the Primary Principal when a User Directory is created in the Users Tree. The Document URI for the Access Permissions Document SHALL be "[XCAP Root URI]/[AUID]/users/[XUI]/oma_ap/access-permissions".
By default the Access Permissions Document SHALL grant the Primary Principal access to all operations towards the User Directory including the created Access Permissions Document.

6.2.4.2 Deleting the Access Permissions Document

The XDMS SHALL delete the Access Permissions Document when the corresponding Primary Principal's User Directory is deleted.

6.2.4.3 Updating the Access Permissions Document

The XDMS SHALL follow the procedures defined in sections 6.2.1 and 6.2.2 except that a PUT handling (see section 6.2.1.1) with the purpose to create an Access Permissions Document or DELETE handling (see section 6.2.1.3) with the purpose to delete an Access Permissions Document SHALL NOT be allowed. If an XDMC performs a PUT or DELETE operation of this kind, the XDMS SHALL reject the request with a HTTP "403 Forbidden" response.

The XDMS SHALL control access to the Access Permissions Document by applying the rules specified in the <access-permissions-document-rule> element inside the Access Permissions Document itself. If access is not granted the XDMS SHALL reject the request with an HTTP "403 Forbidden" response if the request is an XCAP request and with SIP "403 Forbidden" response if it is a SIP request.

The XDMS SHALL verify that the Access Permissions Document conforms to what is specified in section 5.6.

6.2.4.4 Updating the Access Permissions List Document

When updating or deleting the Access Permissions Document as described in sections 6.2.4.2 and 6.2.4.3 the XDMS SHALL update the Access Permissions List Documents [XDM_List] associated with Principals whose Access Permissions were changed. The XDMS SHALL act on behalf of the Primary Principal of the Access Permissions Document as an XDM Agent as described in section 6.1 and use appropriate update or delete XDM operations to update the Access Permissions List Documents with <entry> elements referencing the Access Permissions Documents that correspond to the Application Usage

and the Principals for which Access Permissions have been changed. The XDMS SHALL update an Access Permissions List Document with a new or modified <entry> element only if the associated Principal has Access Permissions to retrieve the referenced Access Permissions Document. The XDMS SHALL set the value of the "etag" attribute to an E-Tag value generated using an Access Permissions Document containing the same content as if retrieved by the Principal. If a Principal's Access Permissions to retrieve an Access Permissions Document is removed, the XDMS SHALL update the Principal's Access Permissions List Document by removing any <entry> element containing a "reference" attribute with a reference to the Access Permissions Document.

6.2.5 Enforcing XDM Access Permissions

The XDMS SHALL apply access control to all XDM operations described in sections 6.2.1. *"Document Management"*, 6.2.2 *"Subscribing to changes in the XDM Resources*" and 6.2.6 "*Handling of XDCP Operations*" targeting the Primary Principal's User Directory and its contents.

The XDMS SHALL apply access control as it is defined in the Application Usage pertaining to the XDM Resource being accessed.

6.2.5.1 Default Access Permissions

If the Application Usage does not specify the use of the Access Permissions Document, the XDMS SHALL ensure that only the Primary Principal associated with the XUI, an associated Alias Principal and Trusted Application Server are granted access to the User Directory and its contents.

6.2.5.2 Using the Access Permissions Document for Access Permission Control

The Primary Principal associated with the XUI and an associated Alias Principal SHALL always be permitted to retrieve any XDM Document and subscribe to receive notifications of changes of any XDM Documents in the User Directory.

If the Application Usage specifies the use of an Access Permissions Document as defined in section 5.6, the XDMS SHALL apply the rules contained in this Access Permissions Document to control access to XDM Resources in the User Directory. Application Usage MAY define rules to prohibit certain combinations of Access Permissions in order to ensure the validity of XDM Documents after applying Access Permissions.

If a Principal requesting an XDM Document has access rights only to certain XDM Document Parts as defined by child elements to the <transformations> element (e.g. the <filter-set> element) in the Access Permissions Document, the XDMS SHALL generate a subset of the XDM Document and include all XDM Document Parts required by the XML Schema in order to ensure the validity of the generated subset. Mandatory XDM Document Parts for which the Principal does not have retrieve rights SHALL be supplied without any values (i.e. blank), or with specially defined values indicating access restriction.

If an Application Usage mandates the use of the Access Permissions Document, then the XML Schema for the XDM Documents specified by the Application Usage SHOULD define a mandatory attribute with a unique value for every XML element used in an XML sequence defined in the same XML Schema in order to ensure consistency of sequence element addressing.

The value of the mandatory unique attribute is assigned by the XDMC or the XDM Agent whenever a new element is added to a sequence. The XDMS SHALL verify that value of the attribute is unique within the sequence. If the value of the attribute is not unique the update operation SHALL fail and the XDMS SHALL generate a HTTP 409 "Conflict" response as described in [RFC4825] section 11 "*Detailed Conflict Reports*" with a <uniqueness-failure> element with a suggested unique value in the <alt-value> element.

Alternatively, if an Application Usage mandates the use of the Access Permissions Document and it does not define a mandatory attribute with a unique value for every XML element used in an XML sequence, it MAY require that all elements in an XML sequence defined in the XML Schema defined by the Application Usage SHALL have the same Access Permissions.

6.2.6 Handling of XDCP Operations

The XDMS MAY support one or more XDCP operations as described in this section. The Application Usage defines which XDCP operations it requires.

Upon receiving an XDCP Request, the XDMS

1) SHALL generate an HTTP 415 "Unsupported Media Type" response and not continue with the following steps, if the XDCP Request does not contain an XDCP Document or if the MIME type indicated in the XDCP Request is not "application/vnd.oma.xdcp+xml";

2) SHALL check if the XUI of the XDCP Request is an identity of an Alias Principal associated with a Primary Principal, and continue to handle the XDCP Request as if it had received an XDCP Request with an XUI identifying the Primary Principal;

3) SHALL generate an HTTP 404 "Not Found" response and do not continue with the following steps, if the XDM Document addressed in the Request URI of XDCP Request does not exist;

4) SHALL generate an HTTP 409 "Conflict" response with an XDCP document containing a <response> element with an <xdcp-document-structure-not-ok> child element and not continue with the following steps, if the XDCP Document included in the body of the XDCP Request does not conform to the structure defined in section 5.4.2 "*XDCP Document*";

5) SHALL generate an HTTP 409 "Conflict" response with an XDCP Document containing a <response> element with a <not–supported-request> child element and not continue with the following steps, if the XDMS does not support the requested XDCP operation;

6) SHALL generate an HTTP 409 "Conflict" response with an XDCP Document containing a <response> element with an <application-usage-defined-conflict> child element with "phrase" attribute as defined by the Application Usage and not continue with the following steps, if the execution of the XDCP Request can not be fulfilled due to a reason described in the technical specification of the Application Usage;

7) SHALL generate an HTTP 409 "Conflict" response with an XDCP Document containing a <response> element with an <other-conflict> child element and not continue with the following step, if the XDMS applies a service provider policy resulting in a decision not to fulfil the XDCP Request; and

8) SHALL continue to handle the requested XDCP operations as described per XDCP Request in the subsections to section 6.2.6.

6.2.6.1 Document Reference

Upon receiving an XDCP Request, the XDMS SHALL check the child element of the <request> element of the XDCP Document included in the XDCP Request body.

If the child element is the <set-doc-ref> element and the XDMS supports Document Reference Operations, the XDMS:

1) SHALL generate an HTTP 409 "Conflict" response with an XDCP Document containing a <response> element with a <not-found> child element and not continue with the following steps, if the referenced XDM Document in the <reference> element does not exist or is not retrievable by the requesting Principal;

2) SHALL create the XDM Document reference and maintain it until the reference is deleted or replaced;

3) SHALL delete any XDM Document associated with the Primary Principal having the same name as the XDM Document addressed in the XDCP Request;

4) SHALL maintain the display name, if received in the request, until it is deleted or replaced;

5) SHALL generate an HTTP 201 "Created" response to the requestor with an XDCP Document containing a <response> element with a <done> element if the XDM Document addressed in the XDCP Request did not exist before the XDCP Request and therefore is created; and

6) SHALL generate an HTTP 200 "OK" response to the requestor with an XDCP Document containing a <response> element with a <done> element if the XDM Document addressed in the XDCP Request did exist before the XDCP Request.

If the child element is the <remove-doc-ref> element and the XDMS supports Document Reference operations, the XDMS:

1) SHALL remove the XDM Document reference and any information related to it; and

2) SHALL generate an HTTP 200 "OK" to the requestor with XDCP Document containing a <response> element with a <done> child element.

If the child element is the <retrieve-doc-ref> element and the XDMS supports Document Reference operations, the XDMS:

1) SHALL generate an HTTP 200 "OK" to the requestor with XDCP Document containing a <response> element with a <retrieve-doc-ref-result> child element that includes the XDM Document reference if such exists.

If the XDMS receives any other XDM requests targeting an XDM Document than Document Reference XDCP Requests described in this section and the XDM Document is referencing another XDM Document, the XDMS:

1) SHALL generate an HTTP 409 "Conflict" response with an XDCP Document containing a <response> element with a <doc-ref-active> child element and not continue with the following steps, if the XDM request is a Forward XDCP Request or a Restore XDCP Request;

2) SHALL generate an HTTP 403 "Forbidden" or a SIP 403 "Forbidden" response and not continue the following steps, if the requesting Principal is not permitted as described in section 5.1.5 to request execution of the XDM request.

3) SHALL act as an XDM Agent of behalf of the Primary Principal of the addressed XDM Document and propagate the XDM request to the XDMS handling the referenced XDM Document; and

4) SHALL propagate the response, when received, back to the requestor; and

5) SHALL NOT update a Modification History Information Document if such exists.

6.2.6.2 XDM Resource Forwarding Operations

Upon receiving an XDCP request, the XDMS SHALL check the child element of the <request> element of the XDCP Document included in the XDCP request body.

If the child element is the <forward> element and the XDMS supports XDM Resource forwarding operations, the XDMS:

1) SHALL generate an HTTP 403 "Forbidden" response and not continue with the following steps, if the requesting Principal is not allowed to forward the XDM Resource as described in section 5.1.5;

2) SHALL extract the list of recipients from the XDCP Request body;

3) SHALL generate an HTTP 403 "Forbidden" response and not continue with the following steps if none of the recipients extracted from the XDCP Request are authorized to receive the XDM Resource based on the <recipients-list> element based on the Access Permissions Document of the requestor as described in section 5.6.7;

4) SHALL generate an HTTP 404 "Not Found" response and not continue with the following steps if none of the recipients extracted from the XDCP Request are found;

5) SHALL check whether <delivery-report> element is present in the XDCP Document. If it is present and if its value is set to "true", XDMS SHALL add a new element <delivery-notification> to the Forwarding Notification List Document of the requestor as described in the Forwarding Notification List Document (see [List_XDMS]);

NOTE: Recipients not authorized to receive the XDM Resource or not found are excluded from the Forwarding Notification List Document

6) SHALL fetch the XDM Resource to be forwarded specified in the Request-URI of the XDCP request and follow the procedures as described in the section 6.2.6.2.1 "*Creation of the XDM Document to be forwarded*";

7) SHALL check the preferences set by all the recipients residing in the same domain as the XDMS for handling the received Forward XDCP Request as described in the section 6.2.6.2.3 "*Handling of received Forward XDCP Request based on recipients preferences*";

8) SHALL generate the Forward XDCP Request if there are any recipients not residing in its domain as described in the section 6.2.6.2.2 "*Generation of a Forward XDCP Request to remote recipients*"; and

9) SHALL follow the procedure as described in the section 6.2.6.2.4 *"Notifying the recipients about the status of the received Forward XDCP Request"* for notifying about the status of the received Forward XDCP Request to all of the recipients residing in the same domain as the XDMS; and

10) SHALL generate an HTTP 200 "OK" response to the requestor with an XDCP Document containing a <response> element with a <forward-result> child element with the following clarifications:

    a) if any recipient could not be found, the <forward-result> element SHALL contain a <not-found-recipients-list> child element containing the list of the recipients that could not be found; and

    b) if any recipient was not authorized, the <forward-result> element SHALL contain a <not-authorized-recipients-list> child element containing the list of the recipients that were not authorized.

If the child element is the <forward-remote> element and the XDMS supports XDM Resource forwarding operations, the XDMS:

1) SHALL extract the list of recipients from the XDCP Request body;

2) SHALL check the recipient's preferences for handling of the received Forward XDCP Request content as described in the section 6.2.6.2.3 "*Handling of Received Forward XDCP Request based on recipients preferences*";

3) SHALL follow the procedure as described in the section 6.2.6.2.4 *"Notifying the recipients about the status of received Forward XDCP Request"* for notifying the recipients about the status of the received Forward XDCP Request; and

4) SHALL generate an HTTP 200 "OK" response to the requestor with an XDCP Document containing a <response> element with a <remote-forward-result> child element including any recipients that can not be found.

6.2.6.2.1 Creation of the XDM Document to be Forwarded

Upon receiving a Forward XDCP Request, the XDMS:

1) SHALL check if the <rule> element that granted access to the forwarding operation also included a <filter-set> element. If a <filter-set> element is included, the XDMS SHALL apply the filter on the XDM Document to be forwarded before continuing the next step; and

2) SHALL check for the presence of <filter> element within the XDCP Document included in the XDCP Request body. If no <filter> element is present, then the XDMS SHALL store the XDM Document to be forwarded in a temporary storage. If a <filter> element is present, then the XDMS SHALL apply the filters to the XDM Document to be forwarded and store the resulting XDM Document in a temporary storage. XDM Documents stored within temporary storage SHALL be accessible to the recipients.

The XDMS SHALL delete the XDM Document stored in the temporary storage if all the recipients have either fetched it or rejected it, orafter a time period specified by local policy, has elapsed.

6.2.6.2.2 Generation of a Forward XDCP Request to Remote Recipients

The XDMS SHALL create the Forward XDCP Request as explained in this section to forward the XDM Resource to the Users residing in the remote domains. When creating this Forward XDCP Request, the XDMS SHALL act as an XDM Agent as described in section 6.1 "Procedures at the XDMC and the XDM Agent".

The recipients of the forwarded XDM Resource could reside in several remote domains. To each of the remote domain the XDMS SHALL make an XDCP Request containing the XDCP Document with the following clarifications:

1) SHALL include the <forward-remote> element as child element of the <request> element;

2) SHALL include the <document-uri> element and set its value to the URI of the document to be forwarded residing in the temporary storage as described in the section 6.2.6.2.1 "*Creation of an XDM Document to be forwarded*";

3) SHALL include the <note> element if the <note> element is present in the received Forward XDCP Request and set its value with the content of the <note> element present in the received Forward XDCP Request;

4) MAY include the <size> element which carries the size of the XDM Resource in bytes;

5) SHALL include the <sender-identity> element whose value is set to the Identity of the Principal forwarding the XDM Resource;

6) MAY include the <display-name> element if the <display-name> element is present in the received Forward XDCP Request and set its value with the content of the <display-name> element present in the received Forward XDCP Request;

7) SHALL include the <expiration-time> element which carries the expiration time of the XDM Resource being forwarded;

8) SHALL include the <content-type> element which carries the MIME type of the XDM Resource being forwarded; and

9) SHALL include the <list> element as child element of <recipients-list> element which carries the list of recipients residing in the remote domain to whom the XDM Resource has to be forwarded. The <list> element SHALL conform to the structure of <list> element specified in [RFC4826] with the following clarifications:

    a) The "name" attribute of the <list> element is not included; and

    b) The <entry> element SHALL contain the "uri" attribute set to a valid User Address, e.g. a SIP URI (as defined in [RFC3261]) or a Tel URI (as defined in [RFC3966]).

10) SHALL include <delivery-report> element if the <delivery-report> element is present in the received Forward XDCP Request and set its value equal to the content of the <delivery-element> element present in the received Forward XDCP Request; and

11) SHALL include the <request-id> element if the <request-id> element is present in the received Forward XDCP Request and set its value equal to the content of the <request-id> element present in the received Forward XDCP Request.

NOTE: The procedure of XDMS for determining and grouping the recipients belonging to same domain is outside the scope of this specification. If grouping of recipients is not possible the implementation may choose to make individual XDCP Requests to each of the recipients.

6.2.6.2.3 Handling of Received Forward XDCP Request Based on Recipients Preferences

The XDMS SHALL apply the preferences related to handling of XDM Resource forwarding operation, if any, as described in the section 5.8 "*XDM Preferences Document*" as follows:

The XDMS SHALL behave as if it had received a ForwardAccept XDCP Request and follow the procedure described in section 6.2.6.2.7 *"Handling of XDCP ForwardAccept and ForwardReject Requests"*, if the child element of the <actions> element of the <forward-prefs> element in the XDM Preferences Document (see section 5.8) of the recipient is the <accept> element; or,

The XDMS SHALL behave as if it had received a ForwardReject XDCP Request and follow the procedure described in section 6.2.6.2.7 *"Handling of XDCP ForwardAccept and ForwardReject Requests"* if the child element of the <actions>

element of the <forward-prefs> element in the XDM Preferences Document (see section 5.8) of the recipient is the <reject> element; or,

The XDMS, if the forwarded XDM Resource is not delivered to the recipient within the time period specified to keep the forwarded XDM Resource alive, SHALL:

1) Update the Forwarding Notification List [XDM_List] of the Principal that initiated the XDM Forward operation and set the value of the "status" attribute of the <entry> element corresponding to all the Principals that have neither accepted nor rejected the forwarded XDM Resource to "expired" if the forwarded XDM Resource resides in the local XDMS; or

2) Generate the ForwardDeliveryReport XCDP Request as described in section 6.2.6.2.6 *"Generation of XDM Forward Delivery Report Request"* on behalf of all the Principals that have neither accepted nor rejected the forwarded XDM Resource and set the <status> element to "expired" if the forwarded XDM Resource resides in a remote XDMS.

6.2.6.2.4 Notifying the Recipients about the Status of the Received Forward XDCP Request

The XDMS SHALL update the Forwarding Notification List Document of each of the recipients extracted from the XDCP Request body as explained in this section.

The XDMS SHALL update the Forwarding Notification List Document by adding a <request> element as child element of <request-notification-list> element whose "auid" attribute value matches with its AUID with the following clarifications:

1) MAY include the <timestamp> element with the value set to the time when XDCP Request was received;

2) SHALL include the <note> element if it is present in the received XDCP Request and set its value same as the value of the <note> element present in the received XDCP Request;

3) SHALL include the <status> element with values based on the receiving Principal's XDM Preferences as described in section 5.8;

4) If the value of the <request> element received in the XDCP Request is "forward-remote", the XDMS:

    a) SHALL include the "document-uri" attribute and set it to a URI of a local temporary storage specifying the Document Reference. The Document Reference SHALL point to the XDM Document indicated by the value of the <document-uri> element received in the XDCP Request. This Document Reference allows the XDMC or XDM Agent to fetch the forwarded XDM Document before determining what to do with it and allowing the XDMS to monitor if the temporary document is fetched or not; and

    b) SHALL include the following elements with the same value as received in the XDCP Request:

        i. <sender-identity>

        ii. <size>

        iii. <content-type>

5) If the value of the <request> element received in the XDCP Request is "Forward", the XDMS:

    a) SHALL include the "document-uri" attribute whose value is set to the URI of the forwarded XDM Resource residing in the temporary storage;

    b) SHALL include <sender-identity> element whose value is set to the identity of the Principal forwarding the XDM Resource;

    c) SHALL include <size> element whose value is set to the size of the forwarded XDM Resource; and

    d) SHALL include <content-type> element whose value is set to the MIME type of the forwarded XDM Resource.

6) SHALL include the <display-name> as child element of <sender-identity> if it is present in the received XDCP Request and set its value same as the value of the <display-name> element present in the received XDCP Request.

6.2.6.2.5 Generation of XDM Forward Delivery Report Request

The XDMS SHALL create a Forward Delivery Report XDCP Request as explained in this section to send the delivery status of the received Forward XDCP Request. When creating this ForwardDeliveryReport XDCP Request, the XDMS SHALL act as an XDM Agent as described in section 6.1 "Procedures at the XDMC and the XDM Agent". The XDMS SHALL include the XDCP Document in the ForwardDeliveryReport XDCP Request with the following clarifications:

1) SHALL include the < forward-delivery-report> child element of the <request> element;

2) SHALL include the <request-id> element and set it's value with the value of the <request-id> element received in the Forward XDCP Request;

3) SHALL include the <recipient-uri> containing the value of the valid User Address, e.g. a SIP URI (as defined in [RFC3261]) or a Tel URI (as defined in [RFC3966]) on whose behalf the ForwardDeliveryReport XDCP Request is initiated;

4) SHALL include the <status> element with any of the following values:

   a) "delivered" - If the value of the <actions> child element of the <forward-prefs> element in the XDM Preferences Document (see section 5.8) of the recipient is set to "accept".

   b) "rejected" - If the value of the <actions> child element of the <forward-prefs> element in the XDM Preferences Document (see section 5.8) of the recipient is set to "reject"

   c) "expired" - If the forwarded XDM Resource is not delivered to the recipient and the time to keep the forwarded XDM Resource alive is expired,

5) SHALL send the XDCP Request as described in section 6.1.1.3 to the Application Usage from where the Forward XDCP Request was received using a HTTP Request URI of the form http [XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]/users/[XUI].

6.2.6.2.6 Handling of XDM Forward Delivery Report Request

Upon receiving an XDCP request, the XDMS SHALL check the child element of the <request> element of the XDCP Document included in the XDCP request body.

If the child element is the <forward-delivery-report> element, the XDMS SHALL act as an XDM Agent and update the Forwarding Notification List Document [XDM_List] of the Principal identified by the XUI received in the XDCP Request with the following clarifications:

1) SHALL check whether any <delivery-notification> element whose "request-id" attribute is equal to the value of the <request-id> element in the XDCP Document is present in the Forwarding Notification List Document [XDM_List]. If not, the XDMS SHALL generate an HTTP 409 "Conflict" with an XDCP Document containing a <response> element with an <invalid-request> child element and not continue with the following steps;

2) SHALL check whether the matching <delivery-notification> element in the Forwarding Notification List Document [XDM_List] contains a <entry> child element whose "uri" attribute value is equal to the value of <recipient-uri> element of the XDCP Document is present. If not, the XDMS SHALL generate an HTTP 409 "Conflict" with an XDCP Document containing a <response> element with a <invalid-recipient> child element and not continue with the following steps;

3) SHALL update the "status" attribute of the matching <entry> element in the Forwarding Notification List Document [XDM_List] with the value of the <status> element of the XDCP Document; and,

4) SHALL generate an HTTP 200 "OK" response with an XDCP Document containing a <response> element with a <done> element.

6.2.6.2.7 Handling of XDCP ForwardAccept and ForwardReject Requests

Upon receiving an XDCP request, the XDMS SHALL check the child element of the <request> element of the XDCP Document in the XDCP request body.

If the child element is the <forward-accept> element the XDMS:

1) SHALL if the request contained a <do-not-store> element proceed with step 5 below;

2) SHALL if not already fetched, fetch the XDM Resource pointed to by the <document-uri> contained in the request;

3) SHALL if the request contained a <store> element with an "udds" attribute, store the fetched document in the User Directory of the Principal that made the ForwardAccept XDCP Request in the location provided in the attribute and then proceed with step 5 below;

4) SHALL store the fetched document in the User Directory of the Principal that made the ForwardAccept XDCP Request in a default location determined by the XDMS.

5) SHALL update the Forwarding Notification List [XDM_List] of the Principal that initiated the XDM Forward operation and set the value of the "status" attribute of the <entry> element corresponding to the Principal that made the ForwardAccept request to "delivered" if the forwarded XDM Resource resides in the local XDMS; or

6) SHALL generate the ForwardDeliveryReport XDCP Request as described in section 6.2.6.2.5 *"Generation of XDM Forward Delivery Report Request"* and set the <status> element to "delivered" if the forwarded XDM Resource resides in a remote XDMS.

If the child element is the <forward-reject> element the XDMS:

1) SHALL update the Forwarding Notification List [XDM_List] of the Principal that initiated the XDM Forward operation and set the value of the "status" attribute of the <entry> element corresponding to the Principal that made the ForwardReject XDCP Request to "rejected" if the forwarded XDM Resource resides in the local XDMS; or

2) SHALL generate the ForwardDeliveryReport XDCP Request as described in section 6.2.6.2.5 *"Generation of XDM Forward Delivery Report Request"* and set the <status> element to "rejected" if the forwarded XDM Resource resides in a remote XDMS.

6.2.6.3 Differential Read

The XDMS MAY support Differential Read. Upon receiving an XDCP Request, the XDMS SHALL check the child element of the <request> element of the XDCP Document included in the XDCP Request body.

If the child element is the <diff-read> element and the XDMS supports Differential Read, the XDMS:

1) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with a <unknown-etag-value> child element, and do not continue with the following steps, if the E-Tag value contained in the <etag> element of the XDCP Document does not correspond to a known version of the XDM Document;

2) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with a <constraint-failure> child element, and do not continue with the following steps, if a <filter-set> element included in the XDCP Document is invalid;

3) SHALL generate an HTTP 304 "Not Modified" response with an empty body and do not continue with the following steps, if the E-Tag value supplied in the XDCP Document matches with the E-Tag value of the latest version of the XDM Document;

4) SHALL generate an HTTP 200 "OK" response with content-type header set to "multipart/mixed" ; and

5) SHALL include XDCP Document and XCAP Diff document (see [RFC5874]) in the body of the response with the following clarifications:

   a) The XDCP Document SHALL contain the <response> element with the child element <done>;

   b) The XCAP Diff document (see [RFC5874]) SHALL contain a single <document> element with the new and previous E-Tag values and the document change information between the two XDM Documents versions, based on the <filter> element, if one had been included in the request. The "previous-etag" attribute of the <document> element SHALL contain the E-Tag value received in the XDCP Document and the "new-etag" attribute SHALL contain the latest E-Tag value of the XDM Document; and,

   c) If the <etag> element of the XDCP Document is not present, the XCAP Diff document (see [RFC5874]) SHALL contain a <document> element with document change information that replaces the root element of the XDM Document.

6.2.6.4 Differential Write

The XDMS MAY support Differential Write. Upon receiving an XDCP Request, the XDMS SHALL check the child element of the <request> element of the XDCP Document included in the XDCP Request body.

If the child element is the <diff-write> element and the XDMS supports Differential Write, the XDMS:

1) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with <constraint-failure> child element, and do not continue with these steps, if the filter expression of the <filter-set> element included in the XDCP Request is invalid or not in accordance with the constraints of the Application Usage specification;

2) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with an <etag-missing> child element, and do not continue with the following steps, if there is no <previous-etag> element in the XCAP-Diff Document of the XDCP Request;

3) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with an <not-latest-etag-value> child element, and do not continue with the following steps, if the value of the <etag> element in the XDCP Document does not match the E-Tag of the current document stored in the XDMS;

4) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with <no-diff-document> child element , and do not continue with the following steps, if there is no content in the XCAP Diff document (see [RFC5874]);

5) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with an <diff-write-conflict> child element with an <patch-ops-error> child element as described in [RFC5261], and do not continue with the following steps, if the XCAP Diff document (see [RFC5874]) is invalid;

6) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with a <diff-write-conflict> child element with an <xcap-error> child element as described in [RFC4825], and do not continue with the following steps, if the XDM Document can not be modified due to a constraint described in [RFC4825] or in the technical specification for the Application Usage related to the XDM Document;

7) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with a <diff-write-conflict> child element, and do not continue with the following steps, if the XCAP Diff document (see [RFC5874]) included in the XDCP Request is inconsistent with the requested XDM Document;

8) SHALL apply the differential contained in the XCAP Diff document (see [RFC5874]) as described in section 6.2.6.4.1 "*Applying Filters*"; and

9) SHALL generate an HTTP 200 "OK" response including an XDCP Document containing a <response> element with a <done-new-etag> child element that includes the new E-Tag value.

6.2.6.4.1 Applying Filters

The set of XML elements of an XDM Document identified by a <filter-set> is defined as the “Target Node Set”. In addition to this definition, the set complement of the “Target Node Set” of the XDM Document is defined as the “Preserved Node Set”.

When a <filter-set> element is present in a Differential Read XDCP Request, and there is a <filter-set> element in the <transformation> element of the Access Permissions Document that is used to restrict the information returned to the XDMC or XDM Agent, then the <filter-set> of the <transformation> element is applied after the <filter-set> element of the XDCP Differential Read Request is applied to the XDM Document."

When a <filter-set> element is present in an Differential Write XDCP Request, the XDMS SHALL edit the “Target Node Set” of the XDM Document using XCAP Diff document (see [RFC5874]) received in the Differential Write XDCP Request body. That resulting set of XML elements is defined as the “Edited Node Set”. When processing this Differential Write XDCP Request, the XDMS SHALL replace the XDM Document with the “Edited Node Set”, preserving XML elements of the “Preserved Node Set”, except where the parent element of a preserved element is deleted in the “Edited Node Set”.

If the Access Permissions Document contains a <filter-set> element as child element of <transformations> element applicable for a particular Differential Write XDCP Request (i.e., implying the requesting XDMC or XDM Agent does not have complete access to the XDM document it seeks to modify), and there is a <filter-set> element in the Differential Write XDCP Request, the XDMS SHALL reject the Differential Write XDCP Request with an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with a <filter-set-not-allowed> child element.

6.2.6.5 XDM Restore

Upon receiving an XDCP request, the XDMS SHALL check the child element of the <request> element of the XDCP Document included in the XDCP request body.

If the child element is the <restore> element and the XDMS supports XDM restore XDCP operations, the XDMS:

1) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with a <not-latest-etag-value> child element and not continue with the following steps, if the E-Tag value contained in the <current-etag> element, if existing, does not corresponds to the latest version of the XDM Document;

2) SHALL generate an HTTP 409 "Conflict" response including an XDCP Document containing a <response> element with a <restore-version-not-found> child element and not continue with the following steps, if the <back-to-etag> element of the XDCP Request does not identify an XDM Document version restore candidate;

3) SHALL restore the XDM Document to the version identified by the <back-to-etag> element value;

4) SHALL remove all the change details stored in the Modification History Information Document newer than the E-Tag value specified by the <back-to-etag> element of the XDCP Document and add an XDM restore XDCP operation as last XDM operation applied to the XDM Document; and

5) SHALL generate an HTTP 200 "OK" response including an XDCP Document containing a <response> element with a <new-etag> child element containing the E-Tag value of the restored XDM Document.

6.2.7 Handling of History Information

6.2.7.1 Modification History Information handling

After execution of an XDM management operation targeting an XDM Document, the XDMS SHALL check if an XDM Preferences Document as described in section 5.8 exists in the User Directory. If such XDM Preferences Document exists, the XDMS SHALL apply the XDM Preferences related to History Information and update the Modification History Information Document as described in section 5.7.1 with information about successful XDM management operations that creates, modifies or deletes the XDM Document.

Upon reception of an XDM Request targeting a Modification History Information Document, the XDMS SHALL apply the authorization policies as described section 5.7.1.11 before executing the XDM Request.

6.2.7.2 Request History Information handling

After execution of an XDM management operation targeting the content of a User Directory, the XDMS SHALL check if an XDM Preferences Document as described in section 5.8 exists in the User Directory. If such XDM Document exists, the XDMS SHALL apply the history XDM Preferences and update the Request History Information Document described in section 5.7.2 with information about the XDM management operation.

The XDMS SHALL be able to apply a service provider policy and limit the number of <requestor> or <request> elements in a Request History Information Document.

When modifying a <request> element's attribute in a Request History Information Document, the XDMS SHALL NOT alter the E-Tag value related to the Request History Information Document.

Upon reception of an XDM Request targeting a Request History Information Document, the XDMS SHALL apply the authorization policies as described section 5.7.2.11 before executing the XDM Request.

Upon handling a subscription to changes operation as described in section 6.2.2 targeting a Request History Information Document and the XDMS modifies a <request> element's attribute, the XDMS SHALL NOT send a SIP NOTIFY with the modified attribute. The XDMS SHALL instead include this information in the next SIP NOTIFY generated due to other XDM Document Part modifications.

6.2.8 XCAP Server Capabilities Retrieval

The XDMS SHALL support the HTTP GET operation as described in section 6.2.1.2.for retrieval of an XDM Capabilities Document described in section 5.3.1.

6.2.9 Directory Information Retrieval

The XDMS SHALL support the XML Documents Directory Application Usage as described in section 5.3.2.

The XDMS SHALL maintain one XDM Directory Document in the Users Tree per XUI named "directory.xml" as described in section 5.3.2.8.

The XDMS SHALL maintain one <folder> element per supported Application Usage in the XDM Directory Document.

The XDMS SHALL authorize an authenticated Principal retrieving the XDM Directory Document as described in sections 5.1.5 and 5.3.2.10.

For an XCAP Get Request targeting at the XDM Directory Document belonging to a user, for example, with a Document Selector "org.openmobilealliance.xcap-directory /users/sip:joe@example.com/directory.xml", the XDMS SHOULD return an XDM Directory Document containing a <folder> element for each supported AUID providing a list of all authorized XDM Documents associated with the respective AUID for the user identified by sip:joe@example.com. If an XDMS is aware of the XCAP Root URI, the XDMS SHALL include the Document URI as the value of the "uri" attribute returned otherwise, it SHALL include the Document Selector.

For an XCAP GET request targeting at a specific AUID as specified by the Node Selector, for a user, for example URI http://[XCAP Root URI]/org.openmobilealliance.xcap-directory/users/sip:joe@example.com/directory.xml/~~/directory/folder[@auid="org.openmobilealliance.groups"], the XDMS serving the AUID SHOULD return to a <folder> element containing a list of all authorized XDM Documents associated with the AUID for the user. The list in this example would be a list of all XDM Documents for Group belonging to sip:joe@example.com. The content type SHALL be "application/xcap-el+xml".

6.3 Procedures at the Aggregation Proxy

The Aggregation Proxy is the single contact point for XDMC to access XDM Resources stored in any XDMS.

When acting as a contact point for an XDMC, the Aggregation Proxy:

1) SHALL act as an HTTP Proxy defined in [RFC2616] and be configured as an HTTP Reverse Proxy [RFC 3040];

2) SHALL, upon receiving an XCAP or HTTP request targeted to the Aggregation Proxy, authenticate the originating XDMC as specified in the section 5.1.1 *"Authentication"*;

3) SHALL, upon the successful authentication, assert the identity of the Principal served by the originating XDMC as described in the section 5.1.2 *"Principal Identity Assertion"*; and

4) SHALL forward the requests as described in the section 6.3.1 *"HTTP Request Handling"*.

The Aggregation Proxy is also a contact point for XDM entites within the Trusted Network (i.e. the XDM Agent, the Cross Network Proxy and the XDMSs) that have a need to make use of the Aggregation Proxy HTTP request handling.

When acting as a contact point for an XDM entity within the Trusted Network, the Aggregation Proxy:

1) SHALL act as an HTTP Proxy defined in [RFC2616] and be configured as an HTTP Reverse Proxy [RFC 3040];

2) SHALL forward the requests as described in the section 6.3.1 *"HTTP Request Handling"*.

6.3.1 HTTP Request Handling

6.3.1.1 General

Upon receiving an XCAP Request targeted to the Aggregation Proxy, the Aggregation Proxy:

1) SHALL check whether the domain of the XUI matches with the domain of the Aggregation Proxy;

2) SHALL perform one of the following:

   a) If the domain of the XUI matches with the domain of the Aggregation Proxy, forward the XCAP request to the corresponding XDMS based on the HTTP Request-URI; or

   b) If the domain of the XUI does not match the domain of the Aggregation Proxy, forward the XCAP request to the Cross-Network Proxy.

Upon receiving an HTTP POST request containing an HTTP Request-URI of the form http://[XCAP Root URI]/ org.openmobilealliance.search, the Aggregation Proxy SHALL forward the HTTP POST request to the Search Proxy.

Upon receiving an HTTP POST request containing an HTTP Request-URI of the form http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]/users/[XUI]/[User Directory Document Selector] the Aggregation Proxy SHALL perform one of the following:

1) If the domain of the XUI matches with the domain of the Aggregation Proxy, forward the XDCP Request to the corresponding XDMS based on the HTTP Request-URI; or

2) If the domain of the XUI does not match the domain of the Aggregation Proxy, forward the XDCP Request to the Cross-Network Proxy.

Upon receiving an HTTP POST request containing an HTTP Request-URI of the form http://[XCAP Root URI]/org.openmobilealliance.xdcp.sp, the Aggregation Proxy SHALL forward the HTTP POST request to the Subscription Proxy.

Upon receiving the responses to the XCAP Request, the Aggregation Proxy SHALL aggregate and forward responses back to the XDMC or XDM Agent.

Upon receiving the responses to the HTTP POST request, the Aggregation Proxy SHALL forward the responses back to the XDMC or XDM Agent.

The Aggregation Proxy MAY, when generating HTTP responses to XDMC or XDM Agent (e.g., when challenging the XDMC for authentication), include the Server HTTP header [RFC2616] with the value set to "XDM-proxy/OMA2.1" to indicate that the Aggregation Proxy is compliant with this specification.

NOTE: It is out of scope of this specification how the Aggregation Proxy to handle the received Server HTTP headers included in the received HTTP responses, when aggregating and forwarding those HTTP responses to XDMC or XDM Agent.

The Aggregation Proxy SHALL protect the HTTP traffic between the XDMC and the Aggregation Proxy as specified in section 5.1.4 *"Integrity and Confidentiality Protection"*.

6.3.1.2 Error Cases

If the Aggregation Proxy receives an HTTP request targeted at an XDM Resource whose Application Usage is not recognized, the Aggregation Proxy or XDMS SHALL reject the request with an HTTP "404 Not Found" error response.

Upon receiving an HTTP request containing an HTTP Request-URI of the form http://[XCAP Root URI]/ org.openmobilealliance.search where HTTP method is different from POST, the Aggregation Proxy SHALL reject the request with an HTTP "405 Method not allowed" error response.

Upon receiving an HTTP request containing an HTTP Request-URI of the form http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]/users/[XUI]/[User Directory Document Selector] or http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID], where HTTP method is different from POST, the Aggregation Proxy SHALL reject the request with an HTTP "405 Method not allowed" error response.

Upon receiving an HTTP POST request not containing an HTTP Request-URI of either the form http://[XCAP Root URI]/ org.openmobilealliance.search or http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]/users/[XUI]/[User Directory Document Selector] or http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID] or http://[XCAP Root URI]/org.openmobilealliance.xdcp.sp, the Aggregation Proxy SHALL reject the request with an HTTP "405 Method not allowed" error response.

Upon receiving an HTTP POST request containing an HTTP Request-URI of the form http://[XCAP Root URI]/org.openmobilealliance.xdcp/[AUID]/users/[XUI]/[User Directory Document Selector] where the Application Usage described by the [AUID] part of the URI is not recognized, the Aggregation Proxy SHALL reject the request with an HTTP "404 Not Found" error response.

6.3.1.3 XCAP Server Capabilities Retrieval

Upon receiving an XCAP GET request for the XDM Capabilities Document as described in section 5.3.1, the Aggregation Proxy:

1) SHALL act as an HTTP Reverse Proxy;

2) SHALL obtain XCAP Server Capabilities from all XDMSs that serve the request originator. To perform this operation the Aggregation Proxy SHALL forward the XCAP request to all XDMSs that serve the request originator and if the target XDMSs respond with HTTP "200 OK" response, collect the <auid>, <extension>, <namespace> and <au> elements;

3) SHALL add information about the subscribe capabilities of the domain by updating the <subscribe> element with the subscribe methods supported; and

4) SHALL return the HTTP "200 OK" response with the "application/xcap-caps+xml" body including all received <auid>, <extension>, <namespace> and <xdm-caps> elements.

Upon receiving any other XCAP requests than XCAP GET for an XDM Capabilities Document, the Aggregation Proxy SHALL respond with an HTTP "405 Method Not Allowed" response.

6.3.1.4 XCAP Directory Retrieval

Upon receiving an XCAP GET request for the "org.openmobilealliance.xcap-directory" AUID (described in section 5.3.2), the Aggregation Proxy:

1) SHALL act as an HTTP Reverse Proxy;

2) SHALL obtain the requested XCAP Directory from the corresponding XDMSs that serve the Primary Principal as indicated in the XUI of the request. To perform this operation the Aggregation Proxy:

    a) SHALL forward the XCAP request either to all XDMSs that serve the Primary Principal as indicated in the XUI of the request if the request is targeted at the XDM Directory Document, or to the XDMS serving the specific AUID if the request is targeted at a specific AUID as specified by the Node Selector;

    b) SHALL if the target XDMSs responded with HTTP "200 OK" response, collect the <folder> elements;

    c) SHALL add the XCAP Root URI in front of the received "uri" attribute value if it contains only the Document Selector;

    d) SHALL aggregate all received <folder> element from the different XDMSs responding wih a HTTP "200 OK" into one XDM Directory Document; and

    e) SHALL when an XDMS response is received with an error message, insert one <folder> element per AUID, that the XDMS serves, containing an <error-code> child element with the error message included for every corresponding AUID.

3) SHALL return an HTTP "200 OK" response, either with the "application/vnd.oma.xcap-directory+xml" body as defined in section 5.3.2.2 that contains XDM Directory Document including all received <folder> elements if the request was targeted at the XDM Directory Document, or with the "application/xcap-el+xml" body that includes the received <folder> element for a specific AUID if the request was targeted at a specified Node Selector.

Upon receiving any other XCAP requests for an XDM Directory Document than XCAP GET, the Aggregation Proxy SHALL respond with an HTTP "405 Method Not Allowed" response.

6.3.2 Compression

The Aggregation Proxy MAY support compression using content encoding.

If the Aggregation Proxy supports compression it SHALL follow the procedures defined in [RFC2616].

6.4 Procedures at the Search Proxy

The Search Proxy performs request forwarding / response aggregation procedure for HTTP traffic carrying Search Requests / Search Responses as described in this section.

The Search Proxy SHALL share the Principal Authentication and its identity assertion provided by the Aggregation Proxy or the XDM Agent as described in section 5.1.3 *"Principal Identity Sharing"*.

The Search Proxy SHALL protect the HTTP traffics as described in section 5.1.4 *"Integrity and Confidentiality Protection"*.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the Search Proxy MAY be implemented in an AS as defined in [3GPP-TS_23.228] and [3GPP2-X.S0013-002] respectively.

6.4.1 Search Request Forwarding

Upon receiving the HTTP Search Request, the Search Proxy:

1) SHALL get the AUID from the "target" query parameter included in the HTTP URI;

2) SHALL get the target search domain information from the optional “domain” query parameter included in the HTTP URI:

   a) If the “domain” query parameter does not exist, the value of the “domain” parameter is “home” or the value of the “domain” parameter includes the same target domain as is the domain of the Search Proxy handling the Search Request, the Search Proxy SHALL forward the Search Request to a single or multiple instances of the appropriate XDMS based on the AUID;

   b) If the "domain” query parameter value is “all”, the Search Proxy SHALL forward the Search Request to a single or multiple instances of the appropriate XDMS based on the AUID and also SHALL forward the Search Request to the Cross-Network Proxy for each domain defined by the local policy of the Search Proxy if inter-domain search is requested and supported. When forwarding the Search Request to the Cross-Network Proxy, the Search Proxy SHALL set the value of the “domain” query parameter to the domain of the targeted Remote Network; and

   c) If the “domain” query parameter includes one or more target domain names, the Search Proxy SHALL forward the Search Request to the Cross-Network Proxy for each target domain different than the domain of the Search Proxy handling the Search Request if inter-domain search is requested and supported. When forwarding the Search Request to the Cross-Network Proxy, the Search Proxy SHALL set the value of the “domain” query parameter to the domain of the targeted Remote Network.

6.4.1.1 Error Cases

If the Search Proxy receives an HTTP Search Request where:

1) The HTTP method is different from POST, the Search Proxy SHALL reject the request with an HTTP “405 Method Not Allowed” error response.

2) The value in “target” query parameter is not recognized as known Application Usage, the Search Proxy SHALL reject the request with an HTTP “409 Conflict” error response with the <constraint-failure> error condition element defined in [RFC4825]. If included, the “phrase” attribute SHOULD be set to “Search not supported for indicated Application Usage”.

6.4.2 Search Response Aggregation

Upon receiving the responses for the Search Requests, the Search Proxy:

1) SHALL forward the response back to the originator in case that the corresponding Search Request was forwarded to a single instance of the XDMS.

2) SHALL aggregate and forward responses back to the originator in case that the corresponding Search Request was forwarded to multiple instances of the XDMS in the Search Proxy’s domain.

3) SHALL aggregate and forward responses back to the originator in case that the corresponding Search Request was forwarded to multiple XDMSs in different domains.

When the responses are aggregated, the Search Proxy SHALL ensure that the total amount of results do not exceed the value of “max-results” attribute in corresponding Search Request if included. The mechanism of the selection of the subset of results in case that total amount of aggregated results is higher than requested by an XDMC or an XDM Agent is out of scope of this specification.

NOTE: It is out of scope of this specification how the Search Proxy to handle the received Server HTTP headers included in the received HTTP responses, when aggregating and forwarding those HTTP responses towards an XDMC or an XDM Agent.

6.5 Procedures at the Cross-Network Proxy

The Cross-Network Proxy SHALL act as an HTTP Proxy defined in [RFC2616] with the following clarifications.

6.5.1 Outbound Requests

Upon receiving XCAP or XDCP requests from the Aggregation Proxy or Search Requests from the Search Proxy, the Cross-Network Proxy:

1) SHALL act as an HTTP Reverse Proxy (see [RFC3040]);

2) SHALL share the PrincipalI Identity Assertion with the Aggregation Proxy and the Search Proxy as described in section 5.1.3*"Principal Identity Sharing"*;

3) SHALL verify whether the target remote domain is a trusted domain. If not, the Cross-Network Proxy SHALL reject the request with an HTTP "404 Not Found" error response. Otherwise, continue with the rest of the steps;

4) SHALL forward the request to the Cross-Network Proxy of Remote Network;

   a) When forwarding an XCAP request, the Cross-Network Proxy SHALL set the XCAP Root URI of the forwarded XCAP request to the XCAP Root URI of the Remote Network based on the domain of the XUI.

   b) When forwarding a Search Request, the Cross-Network Proxy SHALL set the XCAP Root URI of the forwarded Search Request to the XCAP Root URI of the Remote Network based on the "domain" query parameter and SHALL remove the "domain" query parameter from the HTTP URI.

   c) When forwarding an XDCP Request, the Cross-Network Proxy SHALL set the XCAP Root URI of the forwarded XDCP Request to the XCAP Root URI of the Remote Network based on the domain of the XUI.

5) SHALL protect the HTTP traffic as described in section 5.1.4*"Integrity and Confidentiality Protection"*.

NOTE 1: How the Cross-Network Proxy determines the target remote domain is outside the scope of this specification.

NOTE 2: How the Cross-Network Proxy determines if the target remote domain is a trusted domain is outside the scope of this specification.

6.5.2 Inbound Requests

Upon receiving XCAP requests, XDCP Requests or Search Requests from Remote Networks, the Cross-Network Proxy:

1) SHALL act as an HTTP Reverse Proxy (see [RFC3040]);

2) SHALL share the Principal Identity Assertion with the originating network as described in section 5.1.3 *"Principal Identity Sharing"* if they are from trusted domains;

3) SHALL verify whether the request comes from a trusted remote domain. If not, the Cross-Network Proxy SHALL reject the request with an HTTP "403 Forbidden" error response. Otherwise, continue with the rest of the steps;

4) SHALL verify whether the Cross-Network Proxy is responsible for the target domain of the received XCAP requests, XDCP requests or Search Requests. If not, the Cross-Network Proxy SHALL reject the request with an HTTP "404 Not Found" error response. Otherwise, continue with the rest of the steps;

5) SHALL forward XCAP requests or XDCP Requests to the Aggregation Proxy and forward Search Requests to the Search Proxy;

6) SHALL protect the HTTP traffic as described in section 5.1.4 *"Integrity and Confidentiality Protection"*.

NOTE: How the Cross-Network Proxy determines if the remote domain is a trusted domain is outside the scope of this specification.

6.6 Procedures at the Subscription Proxy

The Subscription Proxy is the contact point for the XDMC and the XDM Agent to subscribe for notification of changes in XDM Resources stored in any XDMS.

6.6.1 Handling of the XDCP Subscribe Command

Upon receiving an XDCP request, the Subscription Proxy SHALL check the child element of the <request> element of the XDCP Document included in the XDCP request body.

If the child element is the <subscribe> element, the Subscription Proxy:

1) SHALL act as an XDMS and execute the steps 1, 2, 4, 5, 6 and 7 described in section 6.2.6;

2) SHALL extract XDM Resource URIs from the <resource-uri> element of the <target-documents> element if it is present in the request and generate back-end subscriptions as described in the section 6.6.2 and clarified below;

3) SHALL extract the resource list from the <resource-lists> element of the <target-documents> element if it is present in the request and generate back-end subscriptions as described in the section 6.6.2and clarified below;

4) SHALL save the extracted resource list and provide in the response the URI of the created resource list as specified in [RFC5367] (refer to section 6.6.3 "*Establishing the Mapping between XDCP Subscribe and SIP Back-end SUBSCRIBE*");

5) SHALL extract the URIs of the existing resource lists from the <list-uri> element of the <target-documents> element if it is present in the request and then retrieve those lists and generate back-end subscriptions as described in the section 6.6.2 with the following clarifications:

    a) SIP “Expires” header SHALL be set to the value of the <duration> element of the XDCP Document included in the body of the XDCP Request;

    b) SIP “P-Asserted-Identity” header SHALL be set to the value of the “X-XCAP-Asserted-Identity” header or “X-3GPP-Asserted-Identity” or “X-3GPP-Intended-Identity”from the HTTP POST request forwarded by the Aggregation Proxy.

    c) If the<preferred-notification-type> element is set to "push", the Subscription Proxy SHALL use the "diff-processing" Event header field parameter set to "aggregate" in the back-end subscription;

    d) If the <preferred-notification-type> element is set to "pull", the Subscription Proxy SHALL use the diff-processing" Event header field parameter set to "aggregate" in the back-end subscription;

    e) If the <preferred-notification-type> element is set to "none", the Subscription Proxy SHALL use the diff-processing" Event header field parameter set to "no-patching" in the back-end subscription.

6.6.2 Handling of the SIP SUBSCRIBE Request

The Subscription Proxy SHALL act as a Resource List Server as defined in [RFC4662] with the following clarifications.

Upon receiving the subscription request for the “xcap-diff” event package, the Subscription Proxy SHALL for each entry in the resource list included in the body of the SIP SUBSCRIBE request generate a back-end subscription to the appropriate XDMS or Subscription Proxy in Remote Network. The Subscription Proxy SHALL include XDM Resources from the same AUID and user in a single back-end subscription.

Each back-end subscription SHALL be realized by sending a SIP SUBSCRIBE request according to [RFC3265] and [IETF-XCAP_Diff_Event] with the following clarifications:

1) The Subscription Proxy SHALL check whether the resource list entry indicates a collection, an XDM Document from the Global Tree, or an XDM Document from the Users Tree.

a) For XDM Documents from the Global Tree, the Subscription Proxy SHALL set the Request-URI and the To SIP header to the SIP URI obtained from the P-Asserted-Identity SIP header identifying the request initiator and set the "auid" parameter defined in Appendix E.1 "*AUID URI Parameter*" to the value of the AUID of the XDM Document.

b) For XDM Documents from the Users Tree, the Subscription Proxy SHALL check the domain from the XUI in the Document Selector.

   i. When the domain is the same as the domain of the Subscription Proxy, the Subscription Proxy SHALL set the Request URI and the To SIP header to the XUI from the Document Selector and set the "auid" parameter defined in Appendix E.1 "*AUID URI Parameter*" to the value of AUID of the XDM Document.

   ii. When the domain is different from the domain of the Subscription Proxy, the Subscription Proxy SHALL set the Request URI to the preconfigured SIP URI identifying the Subscription Proxy in the Remote Network. The Subscription Proxy SHALL include all XDM Resources in the same remote domain in one back-end subscription, regardless of the AUID or XUI of the XDM Document. If no such SIP URI is preconfigured for the Subscription Proxy in Remote Network, the Subscription Proxy SHALL set the Request URI and the To SIP header to the XUI from the Document Selector and set the "auid" parameter defined in Appendix E.1 "*AUID URI Parameter*" to the value of AUID of the XDM Document.

c) For the collection "/" the Subscription Proxy SHALL generate back-end subscription to each preconfigured XDMS in the same domain for each AUID that supports the subscription for notification of changes in XDM Resources and replace the value "/" with the value "/[AUID]/users/[XUI]" where the SIP URI received in the P-Asserted-Identity header is used as an XUI value. The Subscription Proxy SHALL set the Request URI and the To SIP header for each AUID as the SIP URI from the P-Asserted-Identity SIP header identifying the request initiator and set the "auid" parameter defined in Appendix E.1 *"AUID URI Parameter"* to the value of the AUID as preconfigured.

d) For the collection "/[AUID]/" or "/[AUID]/users/" the Subscription Proxy SHALL generate back-end subscription to the appropriate XDMS and replace the value "/[AUID]/" or "/[AUID]/users/" with the value "/[AUID]/users/[XUI]/" where the SIP URI received in the P-Asserted-Identity header is used as an XUI value. The Subscription Proxy SHALL set the Request URI and the To SIP header for each AUID as the SIP URI from the P-Asserted-Identity SIP header identifying the request initiator and set the "auid" parameter defined in Appendix E.1 *"AUID URI Parameter*" to the value of the AUID from the collection.

2) If the SIP SUBSCRIBE contains a MIME-TYPE "application/simple-filter+xml" body with a <filter> element relalated to an XDM Document address in the backend SIP Subscribe, the Subscription SHALL include a MIME-TYPE "application/simple-filter+xml" body with relevant <filter> elements for the received SIP SUBSCRIBE.

NOTE: This is in contradiction with [IETF-XCAP_Diff_Event] but for some Application Usages, the User can have read privilege also to XDM Documents owned by other users and so using such collection can generate very long notifications.

6.6.3 Establishing the Mapping between XDCP Subscribe and SIP Back-end SUBSCRIBE

Upon receiving the XDCP "Subscribe" request and generating the back-end subscriptions, the Subscription Proxy SHALL wait for the initial back-end SIP NOTIFY, or SIP 4xx Client Failure response before sending the response to the XDMC.

If the SIP 4xx Client Failure response is received and the response code is in the range 400-415 inclusive, the Subscription Proxy SHALL forward the response to the XDMC as an HTTP response and terminate the processing of the XDCP Subscribe request.

If the SIP 4xx Client Failure response code is greater than 415, the Subscription Proxy SHALL send the HTTP 400 "Bad Request" response to the XDMC with an XDCP document containing a <response> element with an <other-conflict> child

element containing the details of the received SIP 4xx Client Failure response and terminate the processing of the XDCP Subscribe request.

If the SIP 5xx Server Failure response is received and the response code is in the range 500-504 inclusive, the Subscription Proxy SHALL forward the response to the XDMC as an HTTP response and terminate the processing of the XDCP Subscribe request.

If the SIP 5xx Server Failure response code is greater than 504, the Subscription Proxy SHALL send the HTTP 500 "Internal Server Error" response to the XDMC with an XDCP document containing a <response> element with an <other-conflict> child element containing the details of the received SIP 5xx Server Failure response and terminate the processing of the XDCP Subscribe request.

If the initial SIP NOTIFY is not received within the predefined period of time, the Subscription Proxy SHALL send the HTTP 504 "Gateway Timeout" response to the XDMC and terminate the processing of the XDCP Subscribe request.

Upon receiving the initial SIP NOTIFY, the Subscription Proxy SHALL store the data contained in the <push> element of the <notification> element contained in the XDCP Request and associate the stored data with the SIP dialogue used for the back-end subscription in order to be able to forward subsequent SIP NOTIFY messages to appropriate XDMC. Then, the Subscription Proxy SHALL send the HTTP 200 "OK" response to the XDMC. If the XDCP "Subscribe" request contained a resource list, the response SHALL contain the URI of the created resource list as described in section 6.6.1 "*Handling of the XDCP Subscribe Command*".

6.6.4 Handling of the Back-end SIP NOTIFY for SIP Subscriptions

Upon receiving the SIP NOTIFY from the back-end subscription, the Subscription Proxy SHALL generate notification according to procedures described in [RFC4662] with the following clarifications:

1) The "uri" attribute of the <list> element SHALL include the domain of the Subscription Proxy generating the notification.

2) The value of the "uri" attribute in the <resource> element SHALL include the XUI and "auid" parameter defined in Appendix E.1 "*AUID URI Parameter*" as was included in the Request URI of related SIP SUBSCRIBE request in case of XDM Documents from the Users Tree.

3) The value of the "uri" attribute in the <resource> element SHALL include the SIP URI obtained from the P-Asserted-Identity header of related SIP SUBSCRIBE request and "auid" parameter defined in Appendix E.1 "*AUID URI Parameter*" as was included in the Request URI of related SIP SUBSCRIBE request in case of XDM Documents from the Global Tree in the same domain.

4) The value of the "uri" attribute in the <resource> element SHALL include the domain from the SIP URI of the Subscription Proxy in Remote Network in case of XDM Documents from the Remote Network obtained using a Subscription Proxy in the Remote Network.

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

6.6.5 Handling of the Back-end SIP NOTIFY Request for XDCP Subscriptions

Upon receiving the SIP NOTIFY request from the back-end subscription created with an XDCP Subscribe command, the Subscription Proxy:

1) SHALL retrieve the subscription related data stored after the successful XDCP Subscribe request was processed based on the SIP dialog id and,

2) SHALL send a Push Message Request to the Push Proxy Gateway acting as a Push Initiator as described in [Push_ERELD-V2_2] with following clarifications:

a) If the <xcap-diff> element received in the body of the SIP NOTIFY request contains changes for multiple XDM Documents, each XDM Document change SHALL be sent in a separate Push Message Request;

b) The X-Wap-Application-Id header of the Push Message Request SHALL be set to the value <wap-application-id> element contained in the <push> element of the <notification> element;

c) The control entity carried in the body SHALL contain one <pap> element and the Content-Type set to "application/xml";

d) The <address> element of the <push-message> element of the <pap> element SHALL be set to the value of <push-address> element contained in the <push> element of the <notification> element supplied with the XDCP Subscribe request;

e) The content entity carried in the body of the Push Message Request SHALL be of the type "Service Indication", i.e., it SHALL contain one <si> element and the Content-Type set to "text/vnd.wap.si";

f) The "action" attribute of the <indication> element of the <si> element SHALL be set to the value of the <user-interaction-level> element contained in the <push> element of the <notification> element supplied with the XDCP Subscribe request;

g) The "href" attribute of the <indication> element of the <si> element SHALL contain the XCAP URI of the changed XDM Resource;

    i. For a Push Message Request of an XDCP Subscription that is associated with a new XDCP Subscription, the <quality-of-service> element of the <push-message> element SHALL contain an attribute "delivery-method" value of "confirmed" indicating the Subscription Proxy requests a successful delivery indication of delivery to the XDMC. For subsequent Push Message Request, the <quality-of-service> element of the <push-message> element MAY contain an attribute "delivery-method" value of "confirmed" indicating the Subscription Proxy requests a successful delivery indication of delivery to the XDMC.

h) Note: Confirmation after the first Push Message Request is per local service provider policy. The <item> element of the <info> element of the <si> element:

    i. SHALL contain XCAP-Diff based documents changes for one document according to the <xcap-diff> element contained in the body of the SIP NOTIFY request, if the <preferred-notification-type> element contained in the <push> element of the <notification> element is set to "push". The "class" attribute of the <item> element SHALL be set to "none";

    ii. SHALL contain an XCAP URI pointing to an XDM Document that contains XCAP-Diff based documents changes for one document according to the <xcap-diff> element contained in the body of the SIP NOTIFY request, if the <preferred-notification-type> element contained in the <push> element of the <notification> element is set to "pull". The creation of this indirect XDM Document of XCAP-Diff document changes is outside the scope. The "class" attribute of the <item> element SHALL be set to "document-uri";

    iii. SHALL be empty if the <preferred-notification-type> element contained in the <push> element of the <notification> element is set to "none". The "class" attribute of the <item> element SHALL be set to "none".

    iv. MAY according to local policy compress the body using the GZIP algorithm [RFC1952] and include an Accept-Encoding header field with the value "gzip", if the XDMC included a <gzip> element in the <subscription> element of the XDCP Document of the initial XDCP Request.

If the Subscription Proxy receives a Push Message Response from the Push Proxy Gateway with a <response-result> element that possesses a "response-result" attribute code-type value other than 100x, and the <quality-of-service> element of the <push-message> element SHALL contain an attribute "delivery-method" value of "confirmed", then the Subscription Proxy SHALL cancel the associated back-end subscription by sending a SIP SUBSCRIBE Request to the associated XDMS on the associated dialog with an expiration time set to zero.

Appendix A. Change History (Informative)

A.1 Approved Version 2.1 History

Reference	Date	Description
OMA-TS-XDM_Core-V2_1-20120403-A	03 Apr 2012	Status changed to Approved by TP: OMA-TP-2012-0136-INP_XDM_V2_1_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (normative)

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

The SCRs defined in the following tables include SCR for:

- Aggregation Proxy
- XDMS
- XDMC
- XDM Agent
- Search Proxy
- Cross-Network Proxy
- Subscription Proxy

The following tags are used in the Function column to identify the relationship of the requirements in this Enabler release [XDM_ERELD-V2_1] with the requirements of the previous Enabler release [XDM_ERELD-V2_0]:

- XDMv1.1 – Requirement that is the same in this Enabler release [XDM_ERELD-V2_1], as in the previous Enabler release [XDM_ERELD-V1_1].
- XDMv2.0 – Requirement that is the same in this Enabler release [XDM_ERELD-V2_1], as in the previous Enabler release [XDM_ERELD-V2_0].
- XDMv2.1exp – Requirement that exists in one of the previous Enabler releases (i.e., [XDM_ERELD-V1_1 and [XDM_ERELD-V2_0], but is expanded in this Enabler release [XDM_ERELD-V2_1].

B.1 XDMC

Item	Function	Reference	Requirement
XDM_Core-XOP-C-001-M	Support rules for constructing XDM URIs (XDMv1.1)	6.1.1.1	
XDM_Core-XOP-C-002-M	Including User-Agent HTTP header with the required value (XDMv2.0)	6.1	
XDM_Core-XOP-C-003-M	Support for XCAP Operations (XDMv1.1)	6.1.1.2	
XDM_Core-SUB-C-001-O	Initial Subscription using the SIP SUBSCRIBE message (XDMv2.0)	6.1.2.1	XDM_Core-SUB-C-002-O
XDM_Core-SUB-C-002-O	Processing Received SIP NOTIFY Request (XDMv2.0)	6.1.2.2	XDM_Core-SUB-C-001-O
XDM_Core-SEC-C-001-M	Support HTTP Digest authentication (XDMv1.1)	6.1, 5.1.1	
XDM_Core-SEC-C-002-M	Support HTTP over TLS using the required cipher suite (XDMv1.1)	6.1, 5.1.4	
XDM_Core-SEC-C-003-O	Support other cipher suites defined in RFC2246 (XDMv1.1)	6.1, 5.1.4	
XDM_Core-HCOM-C-001-O	Support HTTP Compression (XDMv1.1)	6.1.1.2	

Item	Function	Reference	Requirement
XDM_Core-SRC-C-001-O	Searching for XDM Documents in Users Tree (XDMv2.0)	6.1.3	XDM_Core-SRC-C-002-O
XDM_Core-SRC-C-002-O	Support Search Document (XDMv2.0)	5.4.1	XDM_Core-SRC-C-001-O
XDM_Core-SRC-C-003-O	Searching for XDM Documents in Global Tree (XDMv2.1)	6.1.3, 5.4.1.3	XDM_Core-SRC-C-002-O
XDM_Core-SRC-C-004-O	Search in Modification History Information (XDMv2.1)	6.1.3, 5.7.1.13	XDM_Core-SRC-C-001-O
XDM_Core-SRC-C-005-O	Search in Request History Information (XDMv2.1)	6.1.3, 5.7.2.13	XDM_Core-SRC-C-001-O
XDM_Core-SEC-C-005-O	Management of Access Permissions (XDMv2.1)	6.1.5	XDM_Core-XOP-C-003-M
XDM_Core-CAPS-C-001-O	Support Application Usage “xcap-caps” (XDMv1.1)	6.1.7, 5.3.1	
XDM_Core-CAPS-C-002-O	Support Application Usage “xcap-caps” with XDM2.1 extensions (XDM v2.1)	6.1.7, 5.3.1	XDM_Core-CAPS-C-001-O
XDM_Core-DIR-C-001-O	Support Application Usage “org.openmobilealliance.xcap-directory” (XDMv1.1)	6.1.8, 5.3.2	
XDM_Core-ERR-C-001-M	Support types of <error-element> as required (XDMv1.1)	5.2.3	
XDM_Core-REF-C-001-O	Document Reference operations (XDMv2.1)	6.1.1.3.1	
XDM_Core-FWD-C-001-O	XDM Resource Forwarding Operations (XDMv2.1)	6.1.1.3.2	
XDM_ Core-FWD-C-002-O	XDM Resource Forwarding Notifications (XDMv2.1)	6.1.1.3.3	
XDM_ Core-SUB-C-003-O	Subscription to Changes in XDM Resources using XDCP operations (XDMv2.1)	6.1.1.3.4, 6.1.2.3	
XDM_ Core-DIFF-C-001-O	Differential Read Operation without the use of a filter (XDMv2.1)	6.1.1.3.5	
XDM_ Core-DIFF-C-002-O	Differential Read Operation with the use of a filter (XDMv2.1)	6.1.1.3.5	XDM_ Core-DIFF-C-001-O
XDM_ Core-DIFF-C-003-O	Differential Write Operation without the use of a filter (XDMv2.1)	6.1.1.3.6	
XDM_ Core-DIFF-C-004-O	Differential Write Operation with the use of a filter (XDMv2.1)	6.1.1.3.6	XDM_ Core-DIFF-C-003-O
XDM_ Core-RES-C-001-O	XDM Restore Operation (XDMv2.1)	6.1.1.3.7	
XDM_ Core-MHI-C-001-O	Modification History Information (XDMv2.1)	6.1.4.1	
XDM_ Core-RHI-C-001-O	Request History Information (XDMv2.1)	6.1.4.2	
XDM_ Core-RHI-C-002-O	Reactive Authorization using Request History Information (XDMv2.1)	6.1.4.2.1	
XDM_ Core-PRF-C-001-O	Management of XDM Preferences (XDMv2.1)	6.1.6	

B.2 XDM Agent

Item	Function	Reference	Requirement
XDM_Core-XOP-A-001-M	Support rules for constructing HTTP URIs (XDMv1.1)	6.1.1.1	
XDM_Core-XOP-A-002-M	Including User-Agent HTTP header with the required value (XDMv2.0)	6.1	
XDM_Core-XOP-A-003-M	Support for XCAP Operations (XDMv1.1)	6.1.1.2	
XDM_Core-SUB-A-001-O	Initial Subscription using the SIP SUBSCRIBE message (XDMv2.0)	6.1.2.1	XDM_Core-SUB-A-002-O
XDM_Core-SUB-A-002-O	Processing Received SIP NOTIFY Request (XDMv2.0)	6.1.2.2	XDM_Core-SUB-A-001-O
XDM_ Core-SEC-A-003-M	Principal Identity Assertion (XDMv1.1)	6.1	
XDM_ Core-SEC-A-004-O	Management of Access Permissions (XDMv2.1)	6.1.5	
XDM_Core-SRC-A-001-O	Searching for XDM Documents in the Users Tree (XDMv2.0)	6.1.3	XDM_Core-SRC-A-002-O
XDM_Core-SRC-A-002-O	Support Search Document (XDMv2.0)	5.4.1	XDM_Core-SRC-A-001-O
XDM_Core-SRC-A-003-O	Searching for XDM Documents in Global Tree (XDMv2.1)	6.1.3, 5.4.1.3	XDM_Core-SRC-A-001-O
XDM_Core-SRC-A-004-O	Search in Modification History Information (XDMv2.1)	6.1.3, 5.7.1.13	XDM_Core-SRC-A-001-O
XDM_Core-SRC-A-005-O	Search in Request History Information (XDMv2.1)	6.1.3, 5.7.2.13	XDM_Core-SRC-A-001-O
XDM_Core-ERR-A-001-M	Support types of <error-element> as required (XDMv1.1)	5.2.3	
XDM_ Core-DIFF-A-001-O	Differential Read Operation without the use of a filter (XDMv2.1)	6.1.1.3.5	
XDM_ Core-DIFF-A-002-O	Differential Read Operation with the use of a filter (XDMv2.1)	6.1.1.3.5	XDM_ Core-DIFF-A-001-O
XDM_ Core-DIFF-A-003-O	Differential Write Operation without the use of a filter (XDMv2.1)	6.1.1.3.6	
XDM_ Core-DIFF-A-004-O	Differential Write Operation with the use of a filter (XDMv2.1)	6.1.1.3.6	XDM_ Core-DIFF-A-003-O
XDM_ Core-RES-A-001-O	Support XDM Restore Operation (XDMv2.1)	6.1.1.3.7	
XDM_ Core-MHI-A-001-O	Modification History Information (XDMv2.1)	6.1.4.1	
XDM_ Core-RHI-A-001-O	Request History Information (XDMv2.1)	6.1.4.2	
XDM_ Core-RHI-A-002-O	Reactive Authorization using Request History Information (XDMv2.1)	6.1.4.2.1	
XDM_ Core-PRF-A-001-O	Management of XDM Preferences (XDMv2.1)	6.1.6	
XDM_ Core-DIR-A-001-O	XCAP Directory Information Retrieval (XDMv1.1)	6.1.8	

B.3 XDMS

Item	Function	Reference	Requirement
XDM_Core-XOP-S-001-M	Support for XCAP (XDMv1.1)	6.2, 6.2.1	XDM_Core-XOP-S-001-M
XDM_Core-XOP-S-002-M	Processing different HTTP requests (XDMv1.1)	6.2.1.1, 6.2.1.2, 6.2.1.3	
XDM_Core-SUB-S-001-O	Support Initial Subscription when SIP SUBSCRIBE message received (XDMv2.0)	6.2.2.1	XDM_Core-SUB-S-002-O
XDM_Core-SUB-S-002-O	Generating a SIP NOTIFY request (XDMv2.0)	6.2.2.2	XDM_Core-SUB-S-001-O
XDM_Core-SEC-S-001-M	Support identity of the Principal access authorization (XDMv1.1) using the default Access Permissions defined.	6.2.5.1, 5.1.5	
XDM_Core-SEC-S-002-O	Support Principal identity access authorization using an Access Permissions Document. (XDMv2.1)	6.2.5, 6.2.5.2	
XDM_Core-SEC-S-003-O	Management of an Access Permissions Document (XDMv2.1)	6.2.4.2, 6.2.4.3	
XDM_Core-SEC-S-004-O	Updating of an Access Permissions List Document (XDMv2.1)	6.2.4.4	
XDM_Core-ERR-S-001-M	Support Error Handling (XDMv1.1)	5.1.1, 5.1.5, 6.2, 6.2.1.1, 6.2.2, 6.3.1.2,	
XDM_Core-CAPS-S-001-M	Support Application Usage “xcap-caps” (XDMv1.1)	6.2.8, 5.3.1	
XDM_Core-CAPS-S-002-O	XDM Capabilities Document Retrieval with XDM 2.1 extensions (XDMv2.1)	6.2.8, 5.3.1	
XDM_Core-DIR-S-001-M	Support Application Usage “org.openmobilealliance.xcap-directory” (XDMv1.1)	6.2.9, 5.3.2	
XDM_Core-DIR-S-002-O	XCAP Directory Information Retrieval XDM 2.1 extensions (XDMv2.1)	6.2.9, 5.3.2	
XDM_Core-SRC-S-001-O	Support Search Document (XDMv2.0)	5.4.1	XDM_Core-SRC-S-002-O
XDM_Core-SRC-S-002-O	Searching for XDM Documents in Users Tree (XDMv2.0)	6.2.3	XDM_Core-SRC-S-001-O
XDM_Core-SRC-S-003-O	Search in Global Tree (XDMv2.1)	6.1.3	
XDM_Core-SRC-S-004-O	Search in Modification History Information (XDMv2.1)	5.7.1.13	XDM_Core-SRC-S-002-O
XDM_Core-SRC-S-005-O	Search in Request History Information (XDMv2.1)	5.7.2.13	XDM_Core-SRC-S-002-O
XDM_Core-XOP-S-003-O	Including Server HTTP header with the required value in HTTP response to XDMC/XDM Agent (XDMv2.0)	6.2	

Item	Function	Reference	Requirement
XDM_Core-ERR-S-002-O	Not using other types of <error-element> than what is recommended. (XDMv1.1)	5.2.3	
XDM_Core-XDCP-S-001-O	Handling of XDCP operations common procedures (XDMv2.1)	6.2.6	
XDM_Core-REF-S-001-O	Document Reference (XDMv2.1)	6.2.6.1	
XDM_Core-FWD-S-002-O	XDM Resource Forwarding operations (XDMv2.1)	6.2.6.2, 6.2.6.2.1	
XDM_Core-FWD-S-003-O	Forward XDCP Request to remote recipients (XDMv2.1)	6.2.6.2.2	
XDM_Core-FWD-S-004-O	Handling of received Forward XDCP Request based on XDM Preferences (XDMv2.1)	6.2.6.2.3	
XDM_Core-FWD-S-005-O	Notifying the recipients of a Forward XDCP Request (XDMv2.1)	6.2.6.2.4	
XDM_Core-FWD-S-006-O	Forward Delivery Report generation (XDMv2.1)	6.2.6.2.5	
XDM_Core-FWD-S-007-O	Forward Delivery Report reception (XDMv2.1)	6.2.6.2.6	
XDM_Core-FWD-S-008-O	Forward Accept and ForwardReject XDCP Requests (XDMv2.1)	6.2.6.2.7	
XDM_Core-DIFF-S-001-O	Differential Read without a filter (XDMv2.1)	6.2.6.3	
XDM_Core-DIFF-S-002-O	Differential Read with a filter (XDMv2.1)	6.2.6.3, 6.2.6.4.1	XDM_Core-DIFF-S-001-O
XDM_Core-DIFF-S-003-O	Differential Write without a filter (XDMv2.1)	6.2.6.4	
XDM_Core-DIFF-S-004-O	Differential Write with a filter (XDMv2.1)	6.2.6.4.1	XDM_Core-DIFF-S-003-O
XDM_Core-RES-S-001-O	XDM Restore (XDMv2.1)	6.2.6.5	
XDM_Core-MHI-S-001-O	Modification History Information (XDMv2.1)	6.2.7.1	
XDM_Core-MHI-S-001-O	Request History Information (XDMv2.1)	6.2.7.2	
XDM_Core-PRF-S-001-O	Management of XDM Preferences (XDMv2.1)	5.8	

B.4 Aggregation Proxy

Item	Function	Reference	Requirement
XDM_Core-XOP-S-004-M	Acting as an HTTP Proxy [RFC2616] and configuration as an HTTP Reverse Proxy [RFC3040] (XDMv1.1)	6.3	
XDM_Core-SEC-S-005-M	Support HTTP Digest authentication (XDMv1.1)	6.3, 5.1.1	
XDM_Core-SEC-S-006-M	Support HTTP over TLS using the required cipher suite (XDMv1.1)	6.3.1.1, 5.1.4	

Item	Function	Reference	Requirement
XDM_Core-SEC-S-007-O	Support other cipher suites defined in RFC2246 (XDMv1.1)	6.3.1.1, 5.1.4	
XDM_Core-SEC-S-008-M	Support Principal Identity Assertion (XDMv1.1)	6.3, 5.1.2	
XDM_Core-XOP-S-005-M	Support XCAP request forwarding (XDMv1.1)	6.3, 6.3.1	
XDM_Core-XOP-S-006-M	Sending XCAP response back (XDMv1.1)	6.3, 6.3.1.1	
XDM_Core-ERR-S-003-M	Handling error cases with appropriate HTTP error response (XDMv1.1mod)	6.3.1.2	
XDM_Core-HCOM-S-001-O	Support Compression (XDMv1.1)	6.3.2	
XDM_Core-SEC-S-009-O	Support for GAA (XDMv1.1)	6.3, 5.1.1	
XDM_Core-CAPS-S-003-M	XCAP Server Capabilities retrieval (Application Usage “xcap-caps”) (XDMv1.1)	6.3.1.3, 5.3.1	
XDM_Core-CAPS-S-004-M	XDM Capabilities Document aggregation of XDM 2.1 extensions (XDMv2.1)	6.3.1.3, 5.3.1	
XDM_Core-DIR-S-003-M	XCAP Directory retrieval (Application Usage “org.openmobilealliance.xcap-directory”) (XDMv1.1)	6.3.1.4, 5.3.2	
XDM_Core-DIR-S-004-M	XCAP Directory Information aggregation of XDM 2.1 extensions (XDMv2.1)	6.3.1.4, 5.3.2	
XDM_Core-ERR-S-004-M	Support Error Handling (XDMv1.1)	5.1.1, 6.3.1.3	
XDM_Core-SEC-S-010-M	Principal identity sharing (XDMv2.0)	5.1.3	
XDM_Core-EIMS-C-001-O	Support Early IMS authentication (XDMv1.1)	5.1.2	
XDM_Core-XOP-S-007-M	Forwarding XCAP request to Cross-Network Proxy (XDMv2.0)	6.3, 6.3.1.1	
XDM_Core-XDCP-S-002-M	Forwarding XDCP request to local XDMS (XDMv2.1)	6.3.1.1	XDM_Core-XDCP-S-002-M
XDM_Core-XDCP-S-003-M	Forwarding XDCP request to Cross-Network Proxy (XDMv2.1)	6.3.1.1	XDM_Core-XDCP-S-003-M
XDM_Core-XDCP-S-004-M	Sending XDCP response back to requestor (XDMv2.1)	6.3.1.1	XDM_Core-XDCP-S-004-M

B.5 Search Proxy

Item	Function	Reference	Requirement
XDM_Core-SRC-S-004-M	Forwarding Search Requests targeting the Users Tree (XDMv2.0)	6.4.1	XDM_Core-SEC-S-006-M
XDM_Core-SRC-S-005-M	Aggregating Search results from XDMSs and forwarding those back (XDMv2.0)	6.4.2	XDM_Core -SEC-S-006-M

Item	Function	Reference	Requirement
XDM_Core-SRC-S-006-M	Forwarding Search Requests targeting the Global Tree (XDMv2.1)	6.4.1	
XDM_Core-ERR-S-005-M	Handling error cases (XDMv2.0)	6.4.1.1	
XDM_Core-SEC-S-011-M	Integrity and Confidentiality Protection support (XDMv2.0)	6.4, 5.1.4	
XDM_Core-SEC-S-012-M	Sharing XDMC authentication and Principal Identity Assertion provided by the Aggregation Proxy (XDMv2.0)	6.4, 5.1.3	
XDM_Core-SEC-S-013-M	Integrity and confidentiality protection (XDMv2.0)	5.1.4	

B.6 Cross-Network Proxy

Item	Function	Reference	Requirement
XDM_Core-XOP-S-086-M	Acting as an HTTP Proxy [RFC2616] and configuration as an HTTP Reverse Proxy [RFC3040] (XDMv2.0)	6.5	
XDM_Core-XOP-S-009-M	Forwarding XCAP requests to trusted domains (XDMv2.0)	6.5.1	
XDM_Core-XOP-S-010-M	Forwarding XCAP responses back to trusted domains (XDMv2.0)	6.5.2	
XDM_Core-SEC-S-014-M	Protecting HTTP traffic (XDMv2.0)	6.5.1, 6.5.2, 5.1.4	
XDM_Core-XOP-S-011-M	Receiving XCAP requests/responses from trusted domains and forwarding them to the Aggregation Proxy (XDMv2.0)	6.5.1	
XDM_Core-XOP-S-012-M	Receiving XCAP requests/responses from Aggregation Proxy and forwarding them to the trusted domains (XDMv2.0)	6.5.2	
XDM_Core-SRC-S-009-M	Receiving Search requests/responses from trusted domains and forwarding them to the Search Proxy(XDMv2.0)	6.5.1	
XDM_Core-SRC-S-010-M	Receiving Search requests/responses from the Search Proxy and forwarding them to the trusted domains (XDMv2.0)	6.5.2	
XDM_Core-ERR-S-006-M	Reject a request from un-trusted Remote Network with an HTTP “403 Forbidden” error response (XDMv2.0)	6.5.2	
XDM_Core-ERR-S-007-M	Reject a request for a target domain that is not responsible for with an HTTP “404 Not Found” error response (XDMv2.0)	6.5.2	

Item	Function	Reference	Requirement
XDM_Core-XDCP-S-005-M	Receiving XDCP requests/responses from trusted domains and forwarding them to the Aggregation Proxy (XDMv2.1)	6.5.1	XDM_Core-XDCP-S-005-M
XDM_Core-XDCP-S-006-M	Receiving XDCP requests/responses from Aggregation Proxy and forwarding them to the trusted domains (XDMv2.1)	6.5.2	XDM_Core-XDCP-S-006-M

B.7 Subscription Proxy

Item	Function	Reference	Requirement
XDM_Core-SUB-S-003-M	Handling of subscriptions for changes in XDM Resources (XDMv2.0).	6.6	
XDM_Core-SUB-S-004-O	Handling of a Subscribe XDCP Request (XDMv2.1)	6.6.1, 6.6.3, 6.6.5	
XDM_Core-SUB-S-005-O	Handling of a Subscribe SIP request targeting Multiple Application Usages (XDMv2.0).	6.6.2, 6.6.3	
XDM_Core-SEC-S-016-M	Sharing the Principal Identity Assertion with the originating Aggregation Proxy (XDMv2.0)	6.6.1, 5.1.3	

Appendix C. Examples (informative)

C.1 Sample XCAP Operation

Figure C.1 describes how an XCAP operation is performed in 3GPP IMS or 3GPP2 MMD. The "resource-list" Application Usage (see [XDM_List]) i.e. the manipulation of a URI List is used in this specific example, but the same types of messages apply for other Application Usages (although the HTTP Request URI and the HTTP body content would, of course, be different). In this example is the XDMC in the same domain as the List XDMS. It is also assumed that the address of Aggregation Proxy is "xcap.example.com" and the XCAP Root URI is "xcap.example.com/".

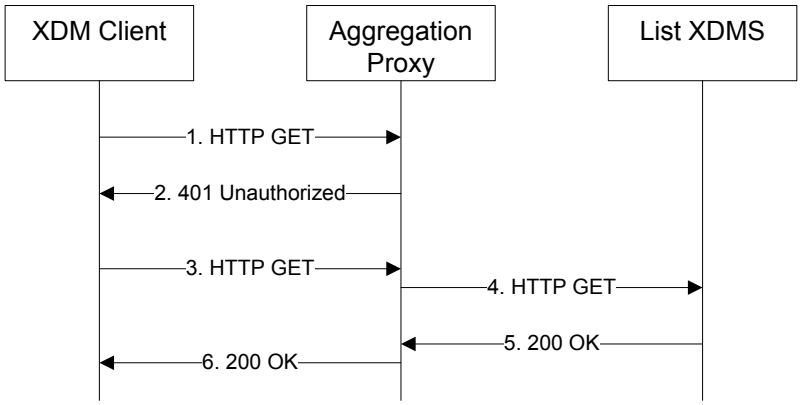

Figure C.1 - Sample XCAP operation

The details of the flows are as follows:

1) The user "sip:joebloggs@example.com" wants to obtain an XDM Document. For this purpose the XDMC sends an HTTP GET request to the Aggregation Proxy.

```
GET /resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.1
Date: Mon, 08 Jan 2007 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
```

2) Upon receiving an unauthorized HTTP GET the Aggregation Proxy chooses to authenticate the XDMC.

```
HTTP/1.1 401 Unauthorized
Server: XDM-proxy/OMA2.1
Date: Mon, 08 Jan 2007 10:50:35 GMT
WWW-Authenticate: Digest realm="xcap.example.com", nonce="47364c23432d2e131a5fb210812c", qop=auth-
   int
Content-Length: 0
```

3) The XDMC sends a HTTP GET request including the Authorization header.

```
GET /resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.1
Date: Mon, 08 Jan 2007 10:50:37 GMT
Authorization: Digest realm="xcap.example.com", nonce="47364c23432d2e131a5fb210812c",
   username="sip:joebloggs@example.com", qop=auth-int,
   uri="/resource-lists/users/sip:joebloggs@example.com/index",
   response="2c8ee200cec7f6e966c932a9242554e4", cnonce="dcd99agsfgfsa8b7102dd2f0e8b1", nc=00000001
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
```

```
Accept-Encoding: gzip
```

4) Based on the AUID the Aggregation Proxy forwards the request to appropriate XDMS.

```
GET /resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: shared-list-xdms.example.com
Via: HTTP/1.1 proxy.example.com (Apache/1.1)
User-Agent: XDM-client/OMA2.1
Date: Mon, 08 Jan 2007 10:50:37 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
```

NOTE: If the "X-3GPP-Intended-Identity" is not included in the message (3), the Aggregation Proxy will include the X-3GPP-Asserted-Identity header.

5) After the XDMS has performed the necessary authorisation checks on the request originator, the XDMS sends an HTTP "200 OK" response including the requested XDM Document in the body.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Date: Mon, 08 Jan 2007 10:50:39 GMT
Etag: "eti87"
Content-Type: application/resource-lists+xml; charset="utf-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list name="oma_buddylist">
     <external anchor="http://xcap.example.org/resource-lists/users/
       sip:joebloggs@example.com/index/~~/resource-lists/list%5B@name=%22list-a%22%5D">
     </external>
  </list>
  <list name="list-a">
    <display-name>My Friends</display-name>
    <entry uri="sip:hermione.blossom@example.com"/>
    <entry uri="tel:+43012349999"/>
  </list>
</resource-lists>
```

6) The Aggregation Proxy encodes (optionally) the content and routes the response back to the XDMC.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Via: HTTP/1.1 proxy.example.com (Apache/1.1)
Date: Mon, 08 Jan 2007 10:50:39 GMT
Authentication-Info: nextnonce="e966c32a924255e42c8ee20ce7f6"
Etag: "eti87"
Content-Encoding: gzip
Content-Type: application/resource-lists+xml; charset="utf-8"
Content-Length: (...)

(binary data)
```

C.2 Sample XCAP Message Flow

This example describes the message flows used to manipulate an XDM Document in an XDMS after authentication.

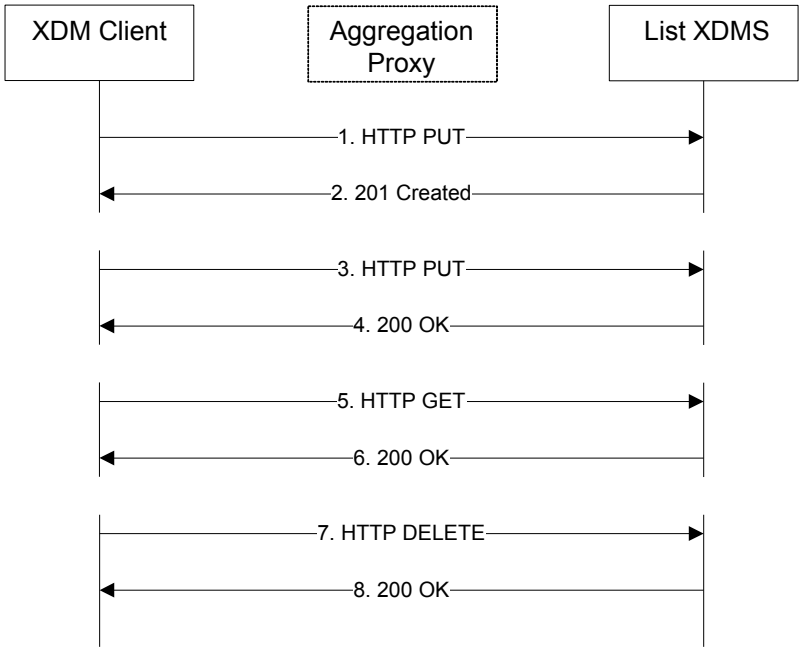

Figure C.2 - XDMC manipulating an XDM Document

NOTE 1: The request messages (1,3,5,7) are shown in one diagram for the convenience of the reader, but there is no implication that all of them have to be performed.

NOTE 2: The Aggregation Proxy is not shown in the flow diagram as its omission does not affect the content of the exchanged messages. The flow diagram also does not show the authentication headers and other HTTP headers not necessary to illustrate the XCAP functionality.

1) The XDMC sends an HTTP PUT request to create a new URI list XDM Document "index" for the user with a Public User Identity of "sip:joebloggs@example.com" in the List XDMS in the example.com domain.

```
PUT /resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: xcap.example.com
…
Content-Type: application/resource-lists+xml; charset="utf-8"
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list name="list-c">
    <display-name>My friends</display-name>
    <entry uri="sip:friend1@example.com">
      <display-name>Friend1</display-name>
    </entry>
  </list>
</resource-lists>
```

2) The List XDMS acknowledges the creation of the "index" XDM Document with a HTTP 201 Created message, assuming that the XDMC had the necessary authorisation to perform the operation, and the operation was successful.

```
HTTP/1.1 201 Created
Etag: "cdcdcdcd"
…
Content-Length: 0
```

3) The XDMC sends a HTTP PUT request to the just-created "index" XDM Document in "sip:joebloggs@example.com"'s home directory to add a new <entry> sub-element to the <list> element identified as "list-c".

```
PUT /resource-lists/users/sip:joebloggs@example.com/index /~~/resource-lists/list%5B@name=%22list-
   c%22%5D/entry%5B@uri=%22sip:friend2@example.com%22%5D HTTP/1.1
Host: xcap.example.com
…
Content-Type: application/xcap-el+xml; charset="utf-8"
Content-Length: (…)

<entry uri="sip:friend2@example.com">
  <display-name>Friend2</display-name>
</entry>
```

4) The List XDMS acknowledges the addition of new elements to the list with an HTTP "200 OK" reply.

```
HTTP/1.1 200 OK
Etag: "efefefef"

…
Content-Length: 0
```

5) The XDMC sends an HTTP GET request to retrieve "sip:joebloggs@example.com"'s "list-c" list from the List XDMS.

```
GET /resource-lists/users/sip:joebloggs@example.com/index/~~/resource-lists/list%5B@name=%22list-
   c%22%5D HTTP/1.1
Host xcap.example.com
```

6) The List XDMS returns the list to the XDMC in the body of an HTTP "200 OK" message.

```
HTTP/1.1 200 OK
…
Etag: "efefefef"
Content-Type: application/xcap-el+xml; charset="utf-8"
Content-Length: (…)

<list name="list-c">
 <display-name>My friends</display-name>
  <entry uri="sip:friend1@example.com">
   <display-name>Friend1</display-name>
  </entry>
  <entry uri="sip:friend2@example.com">
   <display-name>Friend2</display-name>
  </entry>
 </list>
```

7) The XDMC sends an HTTP DELETE request to delete an <entry> identified by the URI "sip:friend2@example.com" from sip:joebloggs@example.com's "list-c" list in the List XDMS.

```
DELETE /resource-lists/users/sip:joebloggs@example.com/index/~~/resource-
   lists/list%5B@name=%22list-c%22%5D/entry%5B@uri=%22sip:friend2@example.com%22%5D HTTP/1.1
Host: xcap.example.com
```

The List XDMS, after checking the privileges of the Principal, performs the deletion.

8) The List XDMS acknowledges the deletion of the "friend2" element from the list with an HTTP "200 OK".

```
HTTP/1.1 200 OK
Etag: "ghghgh"
…
Content-Length: 0
```

C.3 Sample XCAP Directory Retrieval Operation of all User XDM Documents

Figure C.3 describes how an XCAP operation is performed to retrieve all of a user's XDM Documents for all Application Usages. For simplicity, only three XDMSes are shown and the authentication steps are omitted. In this example is the XDMC in the same domain as List XDMS, Policy XDMS and Group XDMS. It is also assumed that the address of Aggregation Proxy is "xcap.example.com" and the XCAP Root URI is xcap.example.com/".

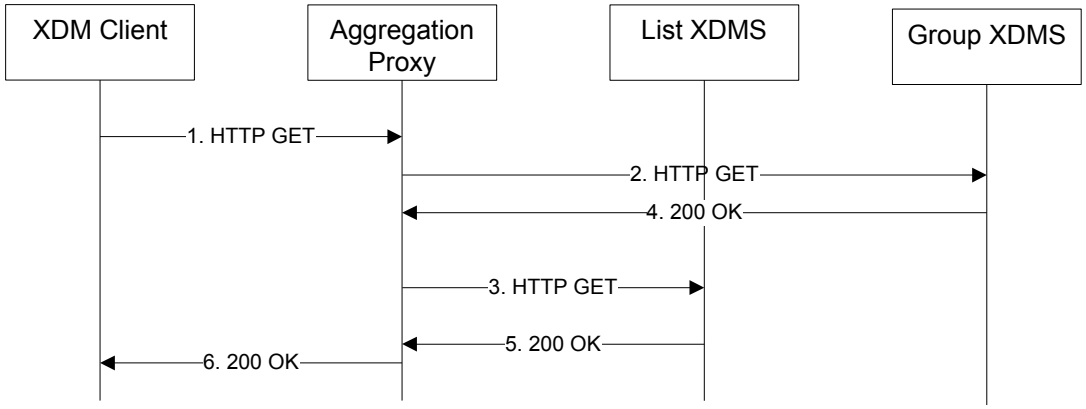

Figure C.3 - Sample XCAP Directory retrieval operation

The details of the flows are as follows:

1) The user "sip:joebloggs@example.com" wants to obtain a list of all his XDM Documents stored in all XDMSes. For this purpose the XDMC sends a HTTP GET request to the Aggregation Proxy.

```
GET /org.openmobilealliance.xcap-directory/users/sip:joebloggs@example.com/directory.xml HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.1
Date: Mon, 08 Jan 2007 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
```

2) The Aggregation proxy forwards the HTTP GET from step 1) to the Group XDMS.

3) The Aggregation proxy forwards the HTTP GET from step 1) to the Policy XDMS.

4) The Aggregation proxy forwards the HTTP GET from step 1) to the List XDMS.

5) The Group XDMS returns the XDM Directory Document containing a list of all the Group Usage List Documents belonging to sip:joebloggs@example.com in a HTTP "200 OK" response

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Date: Mon, 08 Jan 2007 10:50:39 GMT
Content-Type: application/vnd.oma.xcap-directory+xml; charset="utf-8"

Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-directory xmlns="urn:oma:xml:xdm:xcap-directory">
  <folder auid="org.openmobilealliance.groups">
    <entry
   uri="http://xcap.example.com/org.openmobilealliance.groups/users/sip:joebloggs@example.com/skiin
   g" etag="abc123"/>
    <entry
   uri="http://xcap.example.com/org.openmobilealliance.groups/users/sip:joebloggs@example.com/shopp
   ing" etag="def456"/>
  </folder>
</xcap-directory>
```

where the folder element identifies the supported AUID and each <entry> element lists an XDM Document containing one of sip:joebloggs@example.com's Groups called "skiing" and "shopping" in this example.

6) The Policy XDMS returns the XDM Directory Document containing the User Access Policy Document belonging to sip:joebloggs@example.com in a HTTP "200 OK" response

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Date: Mon, 08 Jan 2007 10:51:12 GMT
Content-Type: application/vnd.oma.xcap-directory+xml; charset="utf-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-directory xmlns="urn:oma:xml:xdm:xcap-directory">
  <folder auid="org.openmobilealliance.access-rules">
    <entry uri="http://xcap.example.com/org.openmobilealliance.access-
   rules/users/sip:joebloggs@example.com/access-rules" etag="hjk987"/>
   </folder>
</xcap-directory>
```

where the folder element identifies the supported AUID and <entry> element lists the sip:joebloggs@example.com's User Access Policy Document.

7) The List XDMS returns the XDM Directory Document containing the URI lists and the Group Usage List Documents belonging to sip:joebloggs@example.com in a HTTP "200 OK" response

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Date: Mon, 08 Jan 2007 10:51:44 GMT
Content-Type: application/vnd.oma.xcap-directory+xml; charset="utf-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-directory xmlns="urn:oma:xml:xdm:xcap-directory">
  <folder auid="resource-lists">
    <entry uri="http://xcap.example.com/resource-lists/users/sip:joebloggs@example.com/index"
   etag="pqr999"/>
  </folder>

<folder auid="org.openmobilealliance.group-usage-list">
    <entry uri="http://xcap.example.com/org.openmobilealliance.group-usage-
   list/users/sip:joebloggs@example.com/index" etag="stx111"/>
  </folder>

</xcap-directory>
```

where the folder element identifies the supported AUID and the <entry> element lists XDM Documents under that AUID.

8) The Aggregation Proxy returns the consolidated XDM Directory Document to the user in a HTTP "200 OK" response.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Date: Mon, 08 Jan 2007 10:55:39 GMT
Etag: "eti101"
Content-Type: application/vnd.oma.xcap-directory+xml; charset="utf-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-directory xmlns="urn:oma:xml:xdm:xcap-directory">
  <folder auid="resource-lists">
    <entry uri="http://xcap.example.com/resource-lists/users/sip:joebloggs@example.com/index"
   etag="pqr999"/>
  </folder>
<folder auid="org.openmobilealliance.group-usage-list">
    <entry uri="http://xcap.example.com/org.openmobilealliance.group-usage-
   list/users/sip:joebloggs@example.com/index" etag="stx111"/>
  </folder>
  <folder auid="groups">
    <entry
   uri="http://xcap.example.com/org.openmobilealliance.groups/users/sip:joebloggs@example.com/skiin
   g" etag="abc123"/>
    <entry
   uri="http://xcap.example.com/org.openmobilealliance.groups/users/sip:joebloggs@example.com/shopp
   ing" etag="def456"/>
  </folder>
<folder auid="org.openmobilealliance.access-rules">
    <entry uri="http://xcap.example.com/org.openmobilealliance.access-
   rules/users/sip:joebloggs@example.com/access-rules" etag="hjk987"/>
   </folder>

</xcap-directory>
```

C.4 Sample XCAP Directory Retrieval Operation of Specific User Documents

Figure C.4 describes how an XCAP operation is performed to retrieve all of a user's XDM Documents corresponding to a particular Application Usage. For simplicity, the authentication steps are omitted.

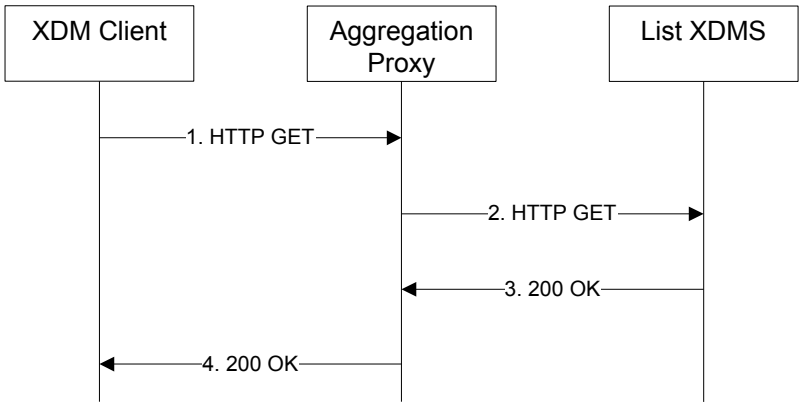

Figure C.4 - Sample XCAP Directory retrieval operation from a particular XDMS

The details of the flows are as follows:

1) The user "sip:joebloggs@example.com" wants to obtain a list of all his XDM Documents (URI lists) stored in the List XDMS. For this purpose the XDMC sends a HTTP GET request to the Aggregation Proxy.

```
GET /org.openmobilealliance.xcap-directory/users/sip:joebloggs@example.com/directory.xml/~~/xcap-
   directory/folder%5B@auid=%22resource-lists%22%5D HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.1
Date: Mon, 08 Jan 2007 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
```

2) The Aggregation proxy forwards the HTTP GET from step 1) to the List XDMS.

3) The List XDMS responds with a HTTP "200 OK" including the <folder> element containing the URI List Document belonging to sip:joebloggs@example.com

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Date: Mon, 08 Jan 2007 10:55:39 GMT
Content-Type: application/xcap-el+xml; charset="utf-8"
Content-Length: (...)

  <folder auid="resource-lists">
    <entry uri="/resource-lists/users/sip:joebloggs@example.com/index" etag="pqr999"/>
  </folder>
```

where the "uri" attribute contains the Document Selector as the XCAP Root URI is not known by the XDMS in this example.

4) The Aggregation proxy returns the consolidated XDM Directory Document to the user in a HTTP "200 OK" response including the addition of the XCAP Root URI to the "uri" attribute value.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Date: Mon, 08 Jan 2007 10:55:59 GMT
Content-Type: application/xcap-el+xml; charset="utf-8"
Content-Length: (...)

  <folder auid="resource-lists">
    <entry uri="http://xcap.example.com/resource-lists/users/sip:joebloggs@example.com/index"
   etag="pqr999"/>
  </folder>
```

C.5 Sample Subscribing to Changes in XDM Documents

This is an informative section to give illustrative examples on how the subscription and notification procedures happen when XDMC requests to subscribe to changes in the Group XDM Document. Note the procedure is identical no matter an XDMC is subscribing to an XML belonging to himself or others.

C.5.1 Direct Subscription

Figure C.5 is an example that demonstrates how an XDMC subscribes to changes in a Group XDM Document. As the subscription is targeted to a single AUID and a single user, the Subscription Proxy is not used in this example.

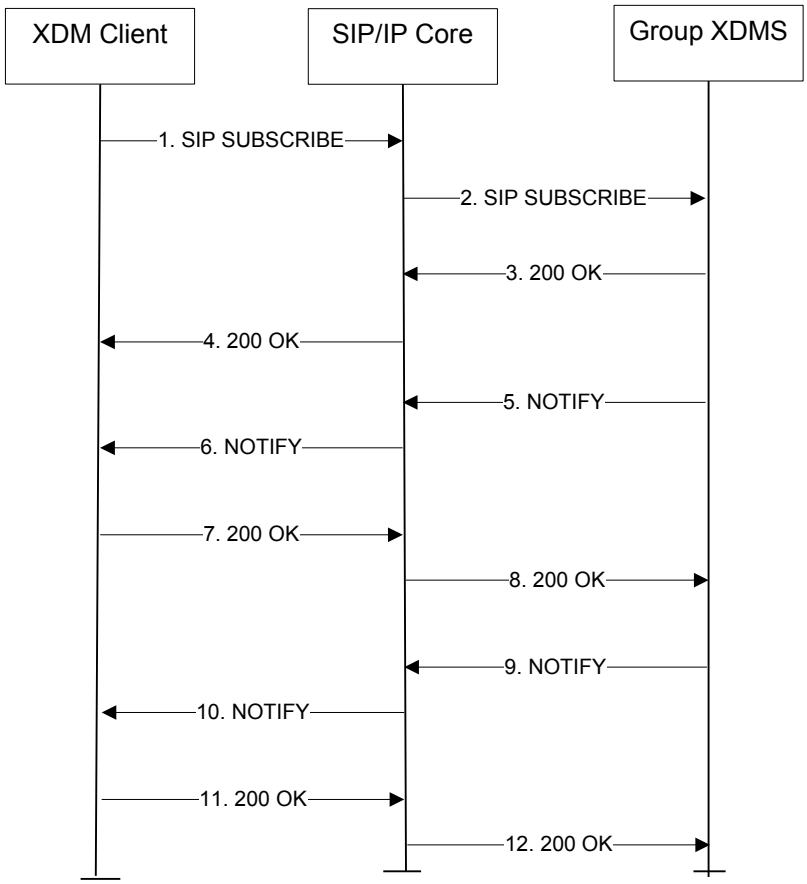

Figure C.5 - XDM Client subscribes to changes in XDM Documents.

1) XDMC (XUI=joe.bloggs@example.com) subscribes to his Group XDM Document named as 'joebloggs_friends', with the contact SIP URI ‘sip:joe.bloggs@example.com’, because he uses multiple devices and wants to keep them updated.

```
SUBSCRIBE sip:joe.bloggs@example.com;auid=org.openmobilealliance.groups SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 70
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>
Event: xcap-diff
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 85 SUBSCRIBE
P-Preferred-Identity: "Joe Bloggs" <sip:joe.bloggs@example.com>
Privacy: none
Expires: 600000
Accept: application/xcap-diff+xml
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Type: application/resource-lists+xml; charset="utf-8"
Content-Length: ..

<?xml version="1.0" encoding="UTF-8"?>
<resource-list xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list>
    <entry uri="org.openmobilealliance.groups/users/sip:joe.bloggs@example.com/joebloggs_friends"/>
  </list>
</resource-list>
```

2) The SIP/IP Core network forwards the SIP SUBSCRIBE request to the Group XDMS. When the SIP/IP Core network corresponds to 3GPP IMS or 3GPP2 MMD, the subscriber's preferred Public User Identity is inserted in P-Asserted-Identity header.

```
SUBSCRIBE sip:joe.bloggs@example.com;auid=org.openmobilealliance.groups SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1,
   SIP/2.0/UDP pcscf1.visited1.net:7531 branch=z9hG4bK240f34.1,
   SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr> <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
Route: <sip:sharedgroupxdms1.home1.netlr>
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>
Event: xcap-diff
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 85 SUBSCRIBE
P-Asserted-Identity: "Joe Bloggs" <sip:joe.bloggs@example.com>
Privacy: none
Expires: 600000
Accept: application/xcap-diff+xml
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Type: application/resource-lists+xml; charset="utf-8"
Content-Length: ..

<?xml version="1.0" encoding="UTF-8"?>
<resource-list xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list>
    <entry uri="org.openmobilealliance.groups/users/sip:joe.bloggs@example.com/joebloggs_friends"/>
  </list>
</resource-list>
```

3) Upon receiving a SIP SUBSCRIBE request for the “xcap-diff” event package, the Group XDMS shall perform the necessary authorization checks on the originator's identity. If the authorization is successes, it shall create a subscription dialog to "xcap-diff" event package to provide the changes of the data identified by the body of SUBSCRIBE request, and return “200 OK” to the subscriber.

4) The SIP/IP Core network forwards the “200 OK” response to the originator of the SIP SUBSCRIBE request, i.e. sip:joe.bloggs@example.com.

5) The Group XDMS generates and sends an initial SIP NOTIFY containing initial references to the XDM Document.

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP sharedgroupxdms1.home1.net;branch=z9hG4bK332b23.1
Max-Forwards: 70
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 102 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: application/xcap-diff+xml; charset="utf-8"
Contact: <sip:sharedgroupxdms1.home1.net>
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"

    xcap-root="http://xcap.example.com/"
    <document new-etag="7ahggs"
     sel="org.openmobilealliance.groups/users/sip:joe.bloggs@example.com/joebloggs_friends"/>
```

```
   </xcap-diff>
```

6) The SIP/IP Core network forwards the SIP NOTIFY request to the appropriate XDMC. If the XDMC does not yet have local copies of XDM Document it may retrieve them.

7) The XDMC responds with a "200 OK".

8) The SIP/IP Core network forwards the "200 OK" to the Group XDMS.

9) After some updates in the XDM Document, the Group XDMS sends the diff part in SIP NOTIFY to the XDMC, in this example, a new "new-friend@example.com" entry was added to the list.

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP sharedgroupxdms1.home1.net;branch=z9hG4bK332b23.1
Max-Forwards: 70
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 112 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: application/xcap-diff+xml; charset="utf-8"
Contact: <sip:sharedgroupxdms1.home1.net>
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xmlns:l="urn:oma:xml:poc:list-service" xcap-
   root="http://xcap.example.com">
    <document previous-etag="7ahggs"
   sel="org.openmobilealliance.groups/users/sip:joe.bloggs@example.com/joebloggs_friends"
     new-etag="ffds66a">
      <change-log>
        <add sel="l:group/l:list-service/l:list">
          <l:entry l:uri="sip:new-friend@example.com">
        </add>
      </change-log>
    </document>
   </xcap-diff>
```

10) The SIP/IP Core network forwards the SIP NOTIFY request to the appropriate XDMC.

11) The XDMC responds with a "200 OK" and updates the old content identified with older eTag, if any exists, according to [IETF-XCAP_Diff].

12) The SIP/IP Core network forwards the "200 OK" to the Group XDMS.

C.5.2 Subscription Using Subscription Proxy

The precondition for this example is that XDMC was provisioned with the SIP URI of the Subscription Proxy and that the Subscription Proxy in domain "example.com" is preconfigured with the address of Subscription Proxy in domain "other_domain.com".

For simplification, SIP headers not relevant for this example (Via, Route, Record-route, …) are not mentioned in the example SIP messages.

C.5.2.1 Initial Subscription

Figure C.6 is an example that demonstrates how an XDMC subscribes to changes in several XDM Documents.

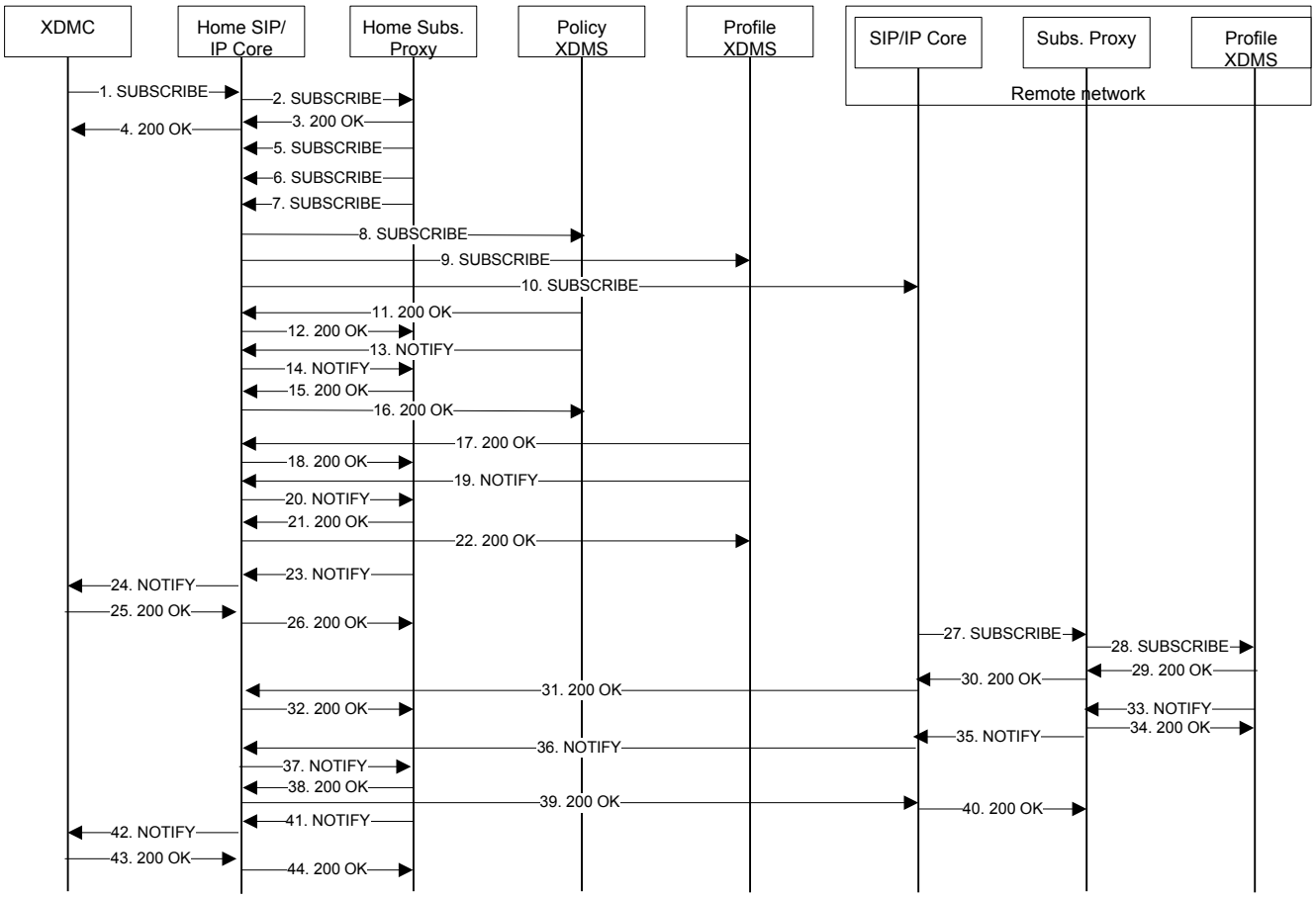

Figure C.6 - XDMC subscribes to changes in XDM Documents.

1) XDMC (XUI=joe.bloggs@example.com) subscribes to:

- his User Access Policy XDM Document because he uses multiple devices and wants to keep it updated;
- User Profile of his friends - user1@example.com, and user2@other_domain.com.

XDMC sends SIP SUBSCRIBE request to the Subscription Proxy via the SIP/IP Core. The Request URI of the SIP SUBSCRIPTION request is set to the SIP address of the Subscription Proxy as obtained using provisioning. The body of the SIP SUBSCRIBE request contains the resource list with three entries:

- AUID “org.openmobilealliance.access-rules”
- URI pointing to User Profile of user user1@example.com
- URI pointing to User Profile of user “user2@other_domain.com”

Because the request is targeted to the Subscription Proxy, in addition to the Accept header field with value “application/xcap-diff+xml”, the XDMC includes the Accept header fields with values “multipart/related” and “application/rlmi+xml” as specified in section 6.1.2.1.2.

```
SUBSCRIBE sip:subscription_proxy@example.com SIP/2.0
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:subscription_proxy@example.com>
Event: xcap-diff; diff-processing=aggregate
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 1 SUBSCRIBE
```

```
P-Preferred-Identity: "Joe Bloggs" <sip:joe.bloggs@example.com>
Expires: 600000
Accept: application/xcap-diff+xml
Accept: multipart/related
Accept: application/rlmi+xml
Content-Type: application/resource-lists+xml; charset="UTF-8"
Content-Length: ..

<?xml version="1.0" encoding="UTF-8"?>
<resource-list xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list>
    <entry uri="org.openmobilealliance.access-rules/"/>
    <entry uri="org.openmobilealliance.user-profile/users/sip:user1@example.com/user-profile"/>
    <entry uri="org.openmobilealliance.user-profile/users/sip:user2@other_domain.com/user-
   profile"/>
  </list>
</resource-list>
```

2) The SIP/IP Core network forwards the SIP SUBSCRIBE request to the Subscription Proxy. When the SIP/IP Core network corresponds to 3GPP IMS or 3GPP2 MMD, the subscriber's preferred Public User Identity is inserted in P-Asserted-Identity header.

```
SUBSCRIBE sip:subscription_proxy@example.com SIP/2.0
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:subscription_proxy@example.com>
Event: xcap-diff; diff-processing=aggregate
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 1 SUBSCRIBE
P-Asserted-Identity: "Joe Bloggs" <sip:joe.bloggs@example.com>
Expires: 600000
Accept: application/xcap-diff+xml
Accept: multipart/related
Accept: application/rlmi+xml
Content-Type: application/resource-lists+xml; charset="UTF-8"
Content-Length: ..

<?xml version="1.0" encoding="UTF-8"?>
<resource-list xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list>
    <entry uri="org.openmobilealliance.access-rules/"/>
    <entry uri="org.openmobilealliance.user-profile/users/sip:user1@example.com/user-profile"/>
    <entry uri="org.openmobilealliance.user-profile/users/sip:user2@other_domain.com/user-
   profile"/>
  </list>
</resource-list>
```

3) Upon receiving a SIP SUBSCRIBE request for the “xcap-diff” event package, the Subscription Proxy creates a subscription dialog to "xcap-diff" event package to provide the changes of the data identified by the body of SUBSCRIBE request, and return “200 OK” to the subscriber.

4) The SIP/IP Core network forwards the “200 OK” response to the originator of the SIP SUBSCRIBE request, i.e. sip:joe.bloggs@example.com.

5) Based on the received initial subscription, the Subscription Proxy generates SIP SUBSCRIBE requests for back-end subscriptions. The first back-end subscription is targeted to Policy XDMS. The Subscription Proxy replaces the entry “org.openmobilealliance.access-rules/” with “org.openmobilealliance.access-rules/users/sip:joe.bloggs@example.com/” and sets the Request URI to “sip:joe.bloggs@example.com;auid=org.openmobilealliance.access-rules”. Such SIP SUBSCRIBE request is sent to Policy XDMS via the SIP/IP Core.

```
SUBSCRIBE sip:joe.bloggs@example.com;auid=org.openmobilealliance.access-rules SIP/2.0
From: <sip:subscription_proxy@example.com>;tag=31514
To: <sip:joe.bloggs@example.com;auid=org.openmobilealliance.access-rules>
Event: xcap-diff; diff-processing=aggregate
Call-ID: b89rjhnedlrfjflslj50b333
CSeq: 1 SUBSCRIBE
```

```
P-Asserted-Identity: "Joe Bloggs" <sip:joe.bloggs@example.com>
Expires: 600000
Accept: application/xcap-diff+xml
Content-Type: application/resource-lists+xml; charset="UTF-8"
Content-Length: ..

<?xml version="1.0" encoding="UTF-8"?>
<resource-list xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list>
    <entry uri="org.openmobilealliance.access-rules/users/sip:joe.bloggs@example.com/"/>
  </list>
</resource-list>
```

6) The next back-end subscription is targeted to Profile XDMS. The Subscription Proxy sets the Request URI to the value "sip:user1@example.com;auid=org.openmobilealliance.user-profile". Such SIP SUBSCRIBE request is sent to Profile XDMS via the SIP/IP Core.

```
SUBSCRIBE sip:user1@example.com;auid=org.openmobilealliance.user-profile SIP/2.0
From: <sip:subscription_proxy@example.com>;tag=31514
To: <sip:user1@example.com;auid=org.openmobilealliance.user-profile>
Event: xcap-diff; diff-processing=aggregate
Call-ID: b89rjhnedlrfjflslj50b334
CSeq: 1 SUBSCRIBE
P-Asserted-Identity: "Joe Bloggs" <sip:joe.bloggs@example.com>
Expires: 600000
Accept: application/xcap-diff+xml
Content-Type: application/resource-lists+xml; charset="UTF-8"
Content-Length: ..

<?xml version="1.0" encoding="UTF-8"?>
<resource-list xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list>
    <entry uri="org.openmobilealliance.user-profile/users/sip:user1@example.com/user-profile"/>
  </list>
</resource-list>
```

7) The last back-end subscription is targeted to Profile XDMS in the Remote Network. The Subscription Proxy sets the Request URI to the preconfigured SIP URI of Subscription Proxy in the Remote Network. Such SIP SUBSCRIBE request is sent to Subscription Proxy in the Remote Network via the SIP/IP Core and SIP/IP Core in the Remote Network.

Because the request is targeted to the Subscription Proxy, in addition to the Accept header field with value "application/xcap-diff+xml", the XDMC includes the Accept header fields with values "multipart/related" and "application/rlmi+xml" as specified in section 6.1.2.1.2.

```
SUBSCRIBE sip:subscription_proxy@other_domain.com SIP/2.0
From: <sip:subscription_proxy@example.com>;tag=31514
To: <sip:subscription_proxy@other_domain.com>
Event: xcap-diff; diff-processing=aggregate
Call-ID: b89rjhnedlrfjflslj50b335
CSeq: 1 SUBSCRIBE
P-Asserted-Identity: "Joe Bloggs" <sip:joe.bloggs@example.com>
Expires: 600000
Accept: application/xcap-diff+xml
Accept: multipart/related
Accept: application/rlmi+xml
Content-Type: application/resource-lists+xml; charset="UTF-8"
Content-Length: ..

<?xml version="1.0" encoding="UTF-8"?>
<resource-list xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list>
    <entry uri="org.openmobilealliance.user-profile/users/sip:user2@other_domain.com/user-
   profile"/>
  </list>
</resource-list>
```

8) The SIP/IP Core network forwards the SIP SUBSCRIBE request targeted to "sip:joe.bloggs@example.com;auid=org.openmobilealliance.access-rules" to the Policy XDMS.

9) The SIP/IP Core network forwards the SIP SUBSCRIBE request targeted to "sip:user1@example.com;auid=org.openmobilealliance.user-profile" to the Profile XDMS.

10) The SIP/IP Core network forwards the SIP SUBSCRIBE request targeted to "sip:subscription_proxy@other_domain.com" to the SIP/IP Core in the Remote Network.

11) The Policy XDMS performs the authorization checks and responds the received SIP SUBSCRIBE request with "200 OK" response sent back to SIP/IP Core.

12) The SIP/IP Core forwards the "200 OK" response from the Policy XDMS to the Subscription Proxy.

13) The Policy XDMS generates and sends an initial SIP NOTIFY containing initial reference to XDM Document listed in the body of SIP SUBSCRIBE request.

```
NOTIFY sip:subscription_proxy@example.com:1357 SIP/2.0
From: <sip:joe.bloggs@example.com;auid=org.openmobilealliance.access-rules>;tag=31415
To: <sip:subscription_proxy@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj50b333
CSeq: 1 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: application/xcap-diff+xml; charset="UTF-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.example.com/">
  <document new-etag="7ahggs"
    sel="org.openmobilealliance.access-rules/users/sip:joe.bloggs@example.com/access-rules"/>
</xcap-diff>
```

14) The SIP/IP Core network forwards the SIP NOTIFY request to the Subscription Proxy.

15) The Subscription Proxy responds with a "200 OK".

16) The SIP/IP Core network forwards the "200 OK" to the Policy XDMS.

17) In parallel to messages 11 to 16, the Profile XDMS performs the authorization checks of received SIP SUBSCRIBE request (message 9) and responds the received SIP SUBSCRIBE request with "200 OK" response sent back to SIP/IP Core.

18) The SIP/IP Core forwards the "200 OK" response from the Profile XDMS to the Subscription Proxy.

19) The Profile XDMS generates and sends an initial SIP NOTIFY containing initial reference to XDM Document listed in the body of SIP SUBSCRIBE request.

```
NOTIFY sip:subscription_proxy@example.com:1357 SIP/2.0
From: <sip:user1@example.com;auid=org.openmobilealliance.user-profile>;tag=31415
To: <sip:subscription_proxy@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj50b334
CSeq: 1 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: application/xcap-diff+xml; charset="UTF-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.example.com/">
  <document new-etag="8asw5r"
```

```
    sel="org.openmobilealliance.user-profile/users/sip:user1@example.com/user-profile"/>
</xcap-diff>
```

20) The SIP/IP Core network forwards the SIP NOTIFY request to the Subscription Proxy.

21) The Subscription Proxy responds with a “200 OK”.

22) The SIP/IP Core network forwards the “200 OK” to the Profile XDMS.

23) In this example, it is assumed that initial notifications from XDMSs in the same domain are received without any significant delay. The Subscription Proxy, after the predefined interval, generates initial notification to the XDMC.

Until this state, the initial notifications from Policy XDMS and Profile XDMS in the same network were received and there was no response from the Remote Network yet. The initial notification contains the body indicating this state. The SIP NOTIFY request is sent to the SIP/IP Core.

```
NOTIFY sip:joe.bloggs@example.com:5060 SIP/2.0
From: <sip:subscription_proxy@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 1 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: multipart/related;type="application/rlmi+xml"; charset="UTF-8";
   start="abc@sp.example.com"; boundary=”Abcdefgh”
Content-Length: (...)

--Abcdefgh
Content-Transfer-Encoding: binary
Content-ID: abc@sp.example.com
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<list xmlns="urn:ietf:params:xml:ns:rlmi" uri="example.com" version="0" fullState="true">
  <resource uri="sip:joe.bloggs@example.com;auid=org.openmobilealliance.access-rules">
    <instance id="juwigmtboe" state="active" cid="12345@sp.example.com"/>
  </resource>
  <resource uri="sip:user1@example.com;auid=org.openmobilealliance.user-profile">
    <instance id="hqzsuxtfyq" state="active" cid="67890@sp.example.com"/>
  </resource>
  <resource uri="other_domain.com"/>
</list>

--Abcdefgh
Content-Transfer-Encoding: binary
Content-ID: 12345@sp.example.com
Content-Type: application/xcap-diff+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.example.com/">
  <document new-etag="7ahggs"
    sel="org.openmobilealliance.access-rules/users/sip:joe.bloggs@example.com/access-rules"/>
</xcap-diff>

--Abcdefgh
Content-Transfer-Encoding: binary
Content-ID: 67890@sp.example.com
Content-Type: application/xcap-diff+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.example.com/">
  <document new-etag="8asw5r"
    sel="org.openmobilealliance.user-profile/users/sip:user1@example.com/user-profile"/>
</xcap-diff>
```

24) The SIP/IP Core network forwards the SIP NOTIFY request to the XDMC.

25) XDMC responds with "200 OK" response.

26) The SIP/IP Core network forwards the "200 OK" to the Subscription Proxy

27) In parallel to previous procedure, the SIP/IP Core in the Remote Network forwards the SIP SUBSCRIBE request obtained in message 10 to the Subscription Proxy in the Remote Network.

28) Based on the received subscription, the Subscription Proxy in the Remote Network generates SIP SUBSCRIBE requests for back-end subscriptions. In this case there is only one back-end subscription targeted to Profile XDMS in the Remote Network. The Subscription Proxy sets the Request URI to "sip:user2@other_domain.com;auid=org.openmobilealliance.user-profile". Such SIP SUBSCRIBE request is sent to Profile XDMS in Remote Network.

NOTE: The request can be sent via SIP/IP Core in the Remote Network or directly. For simplification, direct sending is used in this example.

```
SUBSCRIBE sip:user2@other_domain.com;auid=org.openmobilealliance.user-profile SIP/2.0
From: <sip:subscription_proxy@other_domain.com>;tag=31514
To: <sip:user2@other_domain.com;auid=org.openmobilealliance.user-profile>
Event: xcap-diff; diff-processing=aggregate
Call-ID: b89rjhnedlrfjflslj50b3458
CSeq: 1 SUBSCRIBE
P-Asserted-Identity: "Joe Bloggs" <sip:joe.bloggs@example.com>
Expires: 600000
Accept: application/xcap-diff+xml
Content-Type: application/resource-lists+xml; charset="UTF-8"
Content-Length: ..

<?xml version="1.0" encoding="UTF-8"?>
<resource-list xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list>
    <entry uri="org.openmobilealliance.user-profile/users/sip:user2@other_domain.com/user-
   profile"/>
  </list>
</resource-list>
```

29) The Profile XDMS in Remote Network performs the authorization checks of received SIP SUBSCRIBE request and responds the received SIP SUBSCRIBE request with "200 OK" response sent back to Subscription Proxy in Remote Network.

30) The Subscription proxy in Remote Network responds the received SIP SUBSCRIBE request obtained in message 27 with "200 OK" response sent back to SIP/IP Core in the Remote Network.

31) The SIP/IP Core in the Remote Network forwards the "200 OK" response back to the SIP/IP Core in the home network.

32) The SIP/IP Core forwards the "200 OK" response to the Subscription Proxy.

33) Based on the received SIP SUBSCRIBE request, the Profile XDMS in the Remote Network generates the initial SIP NOTIFY containing initial reference to XDM document listed in the body of SIP SUBSCRIBE request.

```
NOTIFY sip:subscription_proxy@other_domain.com:1357 SIP/2.0
From: <sip:user2@other_domain.com;auid=org.openmobilealliance.user-profile>;tag=31415
To: <sip:subscription_proxy@other_domain.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj50b3458
CSeq: 1 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: application/xcap-diff+xml; charset="UTF-8"
```

```
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.other_domain.com/">
  <document new-etag="0rte3w"
    sel="org.openmobilealliance.user-profile/users/sip:user2@other_domain.com/user-profile"/>
</xcap-diff>
```

34) The Subscription Proxy in the Remote Network responds the SIP NOTIFY request with “200 OK” response.

35) Based on received SIP NOTIFY from back-end subscription to Profile XDMS in the Remote Network, the Subscription Proxy in the Remote Network generates SIP NOTIFY for the back-end subscription obtained from the Subscription Proxy in the home network.

```
NOTIFY sip:subscription_proxy@example.com:1357 SIP/2.0
From: <sip:subscription_proxy@other_domain.com>;tag=31415
To: <sip:subscription_proxy@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj50b335
CSeq: 1 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: multipart/related;type="application/rlmi+xml"; charset="UTF-8";
   start="def@sp.other_domain.com"; boundary=”Ijklmnop”
Content-Length: (...)

--Ijklmnop
Content-Transfer-Encoding: binary
Content-ID: def@sp.other_domain.com
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<list xmlns="urn:ietf:params:xml:ns:rlmi" uri="other_domain.com" version="0" fullState="true">
  <resource uri="sip:user2@other_domain.com;auid=org.openmobilealliance.user-profile">
    <instance id="asdweerfd" state="active" cid="09876@sp.other_domain.com"/>
  </resource>
</list>

--Ijklmnop
Content-Transfer-Encoding: binary
Content-ID: 09876@sp.other_domain.com
Content-Type: application/xcap-diff+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.other_domain.com/">
  <document new-etag="0rte3w"
    sel="org.openmobilealliance.user-profile/users/sip:user2@other_domain.com/user-profile"/>
</xcap-diff>
```

36) The SIP/IP Core in the Remote Network forwards the SIP NOTIFY request to the SIP/IP Core in the home network.

37) The SIP/IP Core forwards the SIP NOTIFY request to the Subscription Proxy.

38) The Subscription Proxy responds the SIP NOTIFY request with “200 OK” response sent to the SIP/IP Core.

39) The SIP/IP Core forwards the response to the SIP/IP Core in the Remote Network.

40) SIP/IP Core in the Remote Network forwards the “200 OK” response to the Subscription Proxy in the Remote Network.

41) Based on received SIP NOTIFY request from the back-end subscription, the Subscription Proxy generates SIP NOTIFY request indicating the new state to the XDMC.

```
NOTIFY sip:joe.bloggs@example.com:5060 SIP/2.0
```

```
From: <sip:subscription_proxy@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 2 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: multipart/related;type="application/rlmi+xml"; charset="UTF-8";
   start="abc@sp.example.com"; boundary=”Abcdefgh”
Content-Length: (...)

--Abcdefgh
Content-Transfer-Encoding: binary
Content-ID: abc@sp.example.com
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<list xmlns="urn:ietf:params:xml:ns:rlmi" uri="example.com" version="1" fullState="false">
  <resource uri="other_domain.com">
    <instance id="mnhgtyuiop" state="active" cid="34567@sp.example.com"/>
  </resource>
</list>

--Abcdefgh
Content-Transfer-Encoding: binary
Content-ID: 34567@sp.example.com
Content-Type: multipart/related;type="application/rlmi+xml"; charset="UTF-8";
   start="def@sp.other_domain.com"; boundary=”Ijklmnop”

--Ijklmnop
Content-Transfer-Encoding: binary
Content-ID: def@sp.other_domain.com
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<list xmlns="urn:ietf:params:xml:ns:rlmi" uri="other_domain.com" version="0" fullState="true">
  <resource uri="sip:user2@other_domain.com;auid=org.openmobilealliance.user-profile">
    <instance id="asdweerfd" state="active" cid="09876@sp.other_domain.com"/>
  </resource>
</list>

--Ijklmnop
Content-Transfer-Encoding: binary
Content-ID: 09876@sp.other_domain.com
Content-Type: application/xcap-diff+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.other_domain.com/">
  <document new-etag="0rte3w"
    sel="org.openmobilealliance.user-profile/users/sip:user2@other_domain.com/user-profile"/>
</xcap-diff>
```

42) The SIP/IP Core forwards the SIP NOTIFY request to the XDMC.

43) The XDMC responds the SIP NOTIFY request with “200 OK” response sent to the SIP/IP Core.

44) The SIP/IP Core forwards the response to the Subscription Proxy.

C.5.2.2 Subsequent Notifications

Figure C.7 is an informative example that demonstrates how the notifications of changes in XDM Documents are provided to the XDMC.

The precondition for this example is successful establishment of initial subscription as described in previous section C.5.1.

The XDMC received the initial notification indicating that value of the etag of the Profile XDM Document of user “user1@example com” is “8asw5r”. Base on the further notification, the XDMC knows the value of the etag of the User

Profile XDM Document of user "user2@other_domain.com" that is "0rte3w". The XDMC compared the received values with the values of stored XDM Documents, if needed, downloaded the latest versions of these XDM Documents. Getting the latest version of an XDM Document using XDM procedures is not mentioned in this example.

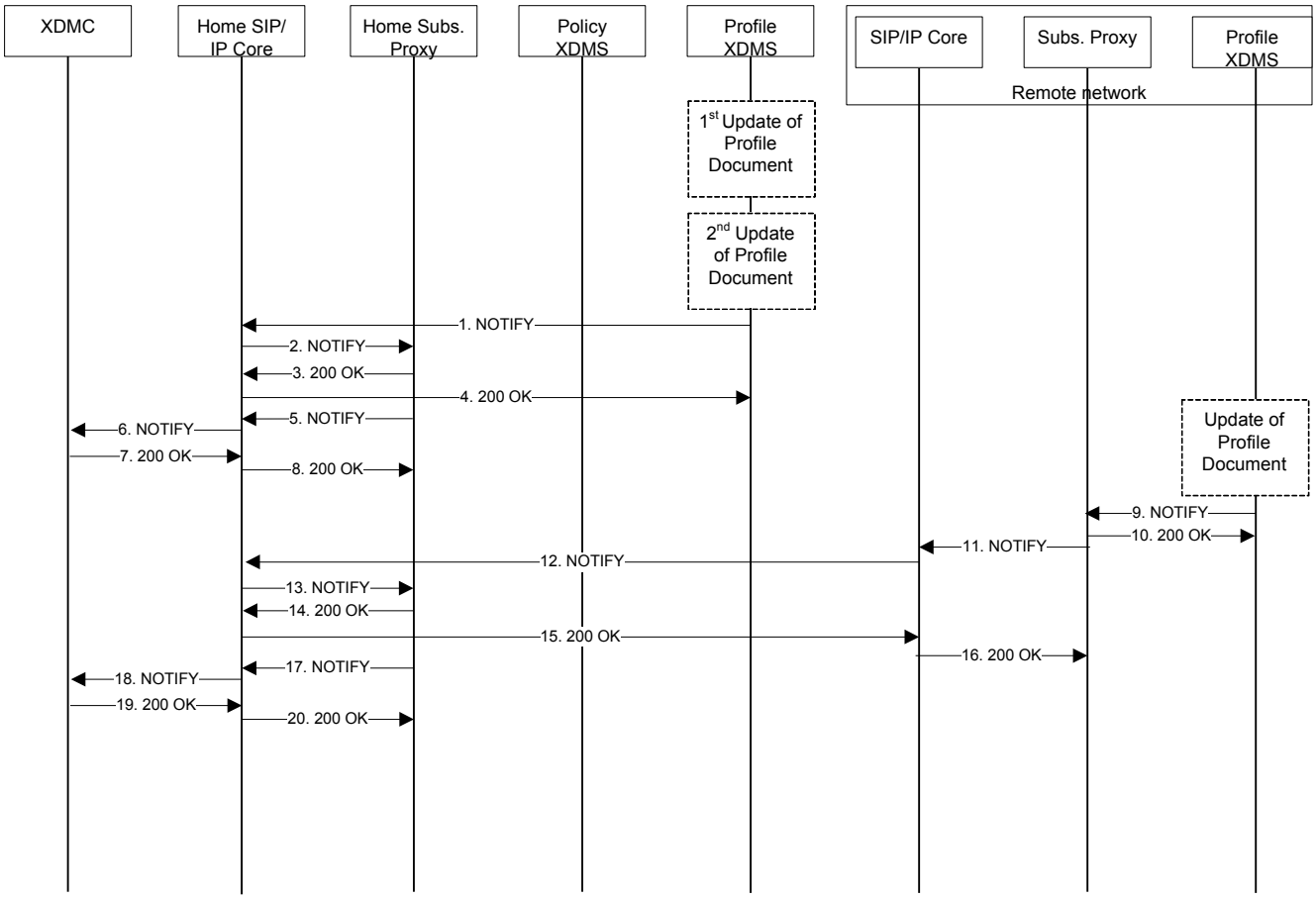

Figure C.7 - Notification of changes in XDM Documents.

First action listed in the figure C.6 is update of User Profile XDM Document of the user "user1@example.com" in the Profile XDMS.

NOTE: To keep the example simple, XDM call flow is not part of the figure C.6, the boxes "Update of User Profile Document" represent update from XDMC of appropriate user using XCAP PUT operation. The example of manipulating XDM Document can be found in Appendix C.2.

In the first update, the user "user1@example.com" updates his list of hobbies by adding a new element indicating a new hobby "football". The resulting XDM Document has a new etag value "23w2er".

After the first update of the User Profile XDM Document of user "user1@example.com", the Profile XDMS does not create notification immediately, but waits to aggregate more changes to a single SIP NOTIFY if appropriate, as was requested by the XDMC in the SIP SUBSCRIBE request.

In the second update, the user "user1@example.com" updates his communication-tyes by adding a new element indicating a new communication-type "Push to talk". The resulting XDM Document has a new etag value "87re6q".

1) The SIP NOTIFY request is generated after the second update of the Profile XDM Document of user "user1@example.com". The SIP NOTIFY request contains two updates of the XDM Document listed in the body of the message. The new-etag value is the one generated after the last update of the XDM Document.

```
NOTIFY sip:subscription_proxy@example.com:1357 SIP/2.0
From: <sip:user1@example.com;auid=org.openmobilealliance.user-profile>;tag=31415
To: <sip:subscription_proxy@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj50b334
CSeq: 2 NOTIFY
Subscription-State: active;expires=599372
Event: xcap-diff
Content-Type: application/xcap-diff+xml; charset="UTF-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.example.com/">
  <document previous-etag="8asw5r"
    sel="org.openmobilealliance.user-profile/users/sip:user1@example.com/user-profile"
    new-etag="87re6q">
    <add sel="/user-profiles/user-profile/hobbies"><hobby>football</hobby>
    </add>
    <add sel="/user-profiles/user-profile/communication-types"><comm-type>Push to talk</comm-type>
    </add>
  </document>
</xcap-diff>
```

2) The SIP/IP Core forwards the SIP NOTIFY request to the Subscription Proxy.

3) The Subscription Proxy responds the SIP NOTIFY request with "200 OK" response sent to the SIP/IP Core.

4) The SIP/IP Core forwards the response to the Profile XDMS.

5) Based on the received notification from the back-end subscription, the Subscription Proxy generates notification to the XDMC.

```
NOTIFY sip:joe.bloggs@example.com:5060 SIP/2.0
From: <sip:subscription_proxy@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 3 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: multipart/related;type="application/rlmi+xml"; charset="UTF-8";
   start="abc@sp.example.com"; boundary="Abcdefgh"
Content-Length: (...)

--Abcdefgh
Content-Transfer-Encoding: binary
Content-ID: abc@sp.example.com
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<list xmlns="urn:ietf:params:xml:ns:rlmi" uri="example.com" version="0" fullState="true">
  <resource uri="sip:user1@example.com;auid=org.openmobilealliance.user-profile">
    <instance id="hqzsuxtfyq" state="active" cid="67890@sp.example.com"/>
  </resource>
</list>

--Abcdefgh
Content-Transfer-Encoding: binary
Content-ID: 67890@sp.example.com
Content-Type: application/xcap-diff+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.example.com/">
```

```
  <document previous-etag="8asw5r"
    sel="org.openmobilealliance.user-profile/users/sip:user1@example.com/user-profile"
    new-etag="87re6q">
    <add sel="/user-profiles/user-profile/hobbies"><hobby>football</hobby>
    </add>
    <add sel="/user-profiles/user-profile/communication-types"><comm-type>Push to talk</comm-type>
    </add>
  </document>
</xcap-diff>
```

6) The SIP/IP Core forwards the SIP NOTIFY request to the XDMC.

7) The XDMC responds the SIP NOTIFY request with "200 OK" response sent to the SIP/IP Core.

8) The SIP/IP Core forwards the response to the Subscription Proxy.

In the next step, the user "user2@other_domain.com" updates his User Profile XDM Document stored in the Profile XDMS in the Remote Network. The resulting XDM Document has a new etag value "222qsv".

9) Based on the update of the User Profile XDM Document, the Profile XDMS in the Remote Network generates the SIP NOTIFY request. Although the Event header parameter of the SIP SUBSCRIBE request establishing the subscription dialog indicated the diff-processing aggregate, as defined in [IETF-XCAP_Diff_Event] the notifier may fall back to the simpler operational mode. In this example, the notifier uses the "no-patching" processing, so only the new value of the E-Tag is provided. The SIP NOTIFY request is sent to the Subscription Proxy in the Remote Network.

```
NOTIFY sip:subscription_proxy@other_domain.com:1357 SIP/2.0
From: <sip:user2@other_domain.com;auid=org.openmobilealliance.user-profile>;tag=31415
To: <sip:subscription_proxy@other_domain.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj50b3458
CSeq: 2 NOTIFY
Subscription-State: active;expires=599235
Event: xcap-diff
Content-Type: application/xcap-diff+xml; charset="UTF-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.other_domain.com/">
  <document previous-etag="0rte3w"
    sel="org.openmobilealliance.user-profile/users/sip:user2@other_domain.com/user-profile"
    new-etag="222qsv"/>
</xcap-diff>
```

10) The Subscription Proxy in the Remote Network responds the SIP NOTIFY request with "200 OK" response sent to the Profile XDMS in the Remote Network.

11) Based on received SIP NOTIFY from back-end subscription to Profile XDMS in the Remote Network, the Subscription Proxy in the Remote Network generates SIP NOTIFY for the back-end subscription obtained from the Subscription Proxy in the home network.

```
NOTIFY sip:subscription_proxy@example.com:1357 SIP/2.0
From: <sip:subscription_proxy@other_domain.com>;tag=31415
To: <sip:subscription_proxy@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj50b335
CSeq: 2 NOTIFY
Subscription-State: active;expires=599233
Event: xcap-diff
Content-Type: multipart/related;type="application/rlmi+xml"; charset="UTF-8";
   start="def@sp.other_domain.com"; boundary="Ijklmnop"
Content-Length: (...)

--Ijklmnop
```

```
Content-Transfer-Encoding: binary
Content-ID: def@sp.other_domain.com
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<list xmlns="urn:ietf:params:xml:ns:rlmi" uri="other_domain.com" version="1" fullState="false">
  <resource uri="sip:user2@other_domain.com;auid=org.openmobilealliance.user-profile">
    <instance id="asdweerfd" state="active" cid="09876@sp.other_domain.com"/>
  </resource>
</list>

--Ijklmnop
Content-Transfer-Encoding: binary
Content-ID: 09876@sp.other_domain.com
Content-Type: application/xcap-diff+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.other_domain.com/">
  <document previous-etag="0rte3w"
    sel="org.openmobilealliance.user-profile/users/sip:user2@other_domain.com/user-profile"
    new-etag="222qsv"/>
</xcap-diff>
```

12) The SIP/IP Core in the Remote Network forwards the SIP NOTIFY request to the SIP/IP Core in the home network.

13) The SIP/IP Core forwards the SIP NOTIFY request to the Subscription Proxy.

14) The Subscription Proxy responds the SIP NOTIFY request with “200 OK” response sent to the SIP/IP Core.

15) The SIP/IP Core forwards the response to the SIP/IP Core in the Remote Network.

16) SIP/IP Core in the Remote Network forwards the “200 OK” response to the Subscription Proxy in the Remote Network.

17) Based on received SIP NOTIFY request from the back-end subscription, the Subscription Proxy generates SIP NOTIFY request indicating the new state to the XDMC.

```
NOTIFY sip:joe.bloggs@example.com:5060 SIP/2.0
From: <sip:subscription_proxy@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 3 NOTIFY
Subscription-State: active;expires=59323
Event: xcap-diff
Content-Type: multipart/related;type="application/rlmi+xml"; charset="UTF-8";
   start="abc@sp.example.com"; boundary=”Abcdefgh”
Content-Length: (...)

--Abcdefgh
Content-Transfer-Encoding: binary
Content-ID: abc@sp.example.com
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<list xmlns="urn:ietf:params:xml:ns:rlmi" uri="example.com" version="2" fullState="false">
  <resource uri="other_domain.com">
    <instance id="mnhgtyuiop" state="active" cid="34567@sp.example.com"/>
  </resource>
</list>

--Abcdefgh
Content-Transfer-Encoding: binary
Content-ID: 34567@sp.example.com
Content-Type: multipart/related;type="application/rlmi+xml"; charset="UTF-8";
   start="def@sp.other_domain.com"; boundary=”Ijklmnop”

--Ijklmnop
Content-Transfer-Encoding: binary
```

```
Content-ID: def@sp.other_domain.com
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<list xmlns="urn:ietf:params:xml:ns:rlmi" uri="other_domain.com" version="1" fullState="false">
  <resource uri="sip:user2@other_domain.com;auid=org.openmobilealliance.user-profile">
    <instance id="asdweerfd" state="active" cid="09876@sp.other_domain.com"/>
  </resource>
</list>

--Ijklmnop
Content-Transfer-Encoding: binary
Content-ID: 09876@sp.other_domain.com
Content-Type: application/xcap-diff+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="http://xcap.other_domain.com/">
  <document previous-etag="0rte3w"
    sel="org.openmobilealliance.user-profile/users/sip:user2@other_domain.com/user-profile"
    new-etag="222qsv"/>
</xcap-diff>
```

18) The SIP/IP Core forwards the SIP NOTIFY request to the XDMC.

19) The XDMC responds the SIP NOTIFY request with a "200 OK" response sent to the SIP/IP Core.

20) The SIP/IP Core forwards the response to the Subscription Proxy.

C.6 Sample Search Operation

Figure C.6 describes how a Search operation is performed. The example shows searching user profile data in Profile XDMS [XDM_Profile]; the same type of messages apply for searching in other Application Usages, where content of HTTP body would be different. In this example is the XDMC in the same domain as the Profile XDMS. It is also assumed that the address of Aggregation Proxy is "xcap.example.com" and the XCAP Root URI is "xcap.example.com/".

For simplicity, search in home domain only is described in following example.

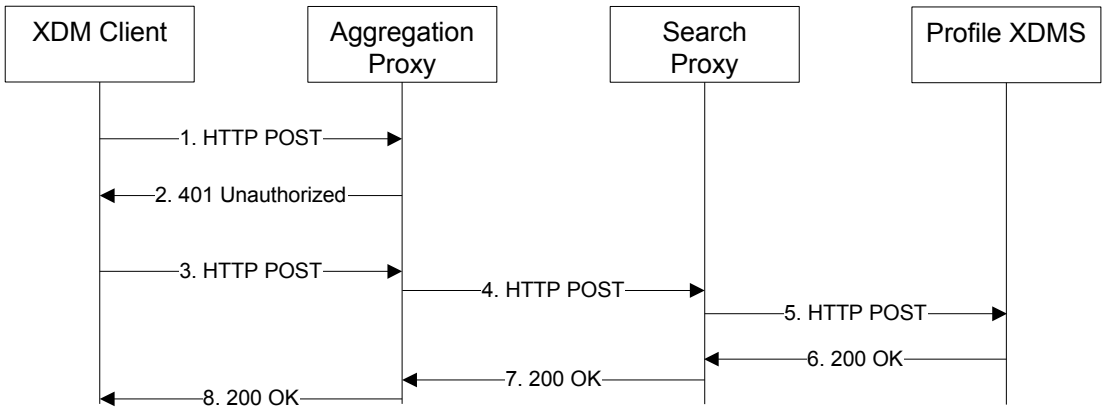

Figure C.8 - Sample Search operation

The details of the flows are as follows:

1) The user "sip:joebloggs@example.com" wants to obtain the user profile data with people from Japan and hobby football. For this purpose the XDMC sends an HTTP POST request to the Aggregation Proxy. The search is targeted to the home domain only.

```
POST /org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/ HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.1
Date: Thu, 10 Aug 2006 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
Content-Type: application/vnd.oma.search+xml; charset="utf-8"
Content-Length: …

<?xml version="1.0" encoding="UTF-8"?>
<search-set   xmlns="urn:oma:xml:xdm:search">

<search id="1234">
  <request>
    <query>
   <![CDATA[
      xquery version "1.0";
      declare default element namespace "urn:oma:xml:xdm:user-profile";

      for $u in collection("org.openmobilealliance.user-profile/users/")/user-profiles/user-
   profile
      where ($u/hobbies/hobby="football")and($u/address/country="JP")
      return <user-profile>{$u/@uri}{$u/display-name}</user-profile>
  ]]>
    </query>
  </request>
</search>

</search-set>
```

2) Upon receiving an unauthorized HTTP POST the Aggregation Proxy chooses to authenticate the XDMC.

```
HTTP/1.1 401 Unauthorized
Server: XDM-proxy/OMA2.1
Date: Thu, 10 Aug 2006 10:50:33 GMT
WWW-Authenticate: Digest realm="xcap.example.com", nonce="47364c23432d2e131a5fb210812c", qop=auth-
   int
Content-Length: 0
```

3) The XDMC sends a HTTP POST request including the Authorization header to the Aggregation Proxy.

```
POST /org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/ HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.1
Date: Thu, 10 Aug 2006 10:50:33 GMT
Authorization: Digest realm="xcap.example.com", nonce="47364c23432d2e131a5fb210812c",
   username="sip:joebloggs@example.com", qop=auth-int,
   uri="/org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/",
   response="2c8ee200cec7f6e966c932a9242554e4", cnonce="dcd99agsfgfsa8b7102dd2f0e8b1", nc=00000001
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
Content-Type: application/vnd.oma.search+xml; charset="utf-8"
Content-Length: …

<?xml version="1.0" encoding="UTF-8"?>

<search-set   xmlns="urn:oma:xml:xdm:search">

<search id="1234">
  <request>
    <query>
   <![CDATA[
      xquery version "1.0";
      declare default element namespace "urn:oma:xml:xdm:user-profile";
```

```
      for $u in collection("org.openmobilealliance.user-profile/users/")/user-profiles/user-
   profile
      where ($u/hobbies/hobby="football")and($u/address/country="JP")
   return <user-profile>{$u/@uri}{$u/display-name}</user-profile>
  ]]>
    </query>
  </request>
</search>

</search-set>
```

4) Based on the "org.openmobilealliance.search" AUID, the Aggregation Proxy forwards the Search Request to the Search Proxy.

```
POST /org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/ HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.1
Date: Thu, 10 Aug 2006 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
Content-Type: application/vnd.oma.search+xml; charset="utf-8"
Content-Length: …

<?xml version="1.0" encoding="UTF-8"?>
<search-set   xmlns="urn:oma:xml:xdm:search">

<search id="1234">
  <request>
    <query>
   <![CDATA[
      xquery version "1.0";
      declare default element namespace "urn:oma:xml:xdm:user-profile";

      for $u in collection("org.openmobilealliance.user-profile/users/")/user-profiles/user-
   profile
      where ($u/hobbies/hobby="football")and($u/address/country="JP")
      return <user-profile>{$u/@uri}{$u/display-name}</user-profile>
  ]]>
    </query>
  </request>
</search>

</search-set>
```

NOTE: If the "X-3GPP-Intended-Identity" is not included in the message (3), the Aggregation Proxy will include the X-3GPP-Asserted-Identity header.

5) Because the domain parameter is missing, the Search Proxy recognizes that the search operation is targeted to the home domain. Based on the target parameter in the Request URI, the Search Proxy forwards the Search Request to the appropriate XDMS.

```
POST /org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/ HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.1
Date: Thu, 10 Aug 2006 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
Content-Type: application/vnd.oma.search+xml; charset="utf-8"
Content-Length: …

<?xml version="1.0" encoding="UTF-8"?>
<search-set   xmlns="urn:oma:xml:xdm:search">

<search id="1234">
  <request>
    <query>
```

```
   <![CDATA[
      xquery version "1.0";
      declare default element namespace "urn:oma:xml:xdm:user-profile";

      for $u in collection("org.openmobilealliance.user-profile/users/")/user-profiles/user-
   profile
      where ($u/hobbies/hobby="football")and($u/address/country="JP")
      return <user-profile>{$u/@uri}{$u/display-name}</user-profile>
  ]]>
    </query>
  </request>
</search>

</search-set>
```

6) After the XDMS has performed the search operation, the XDMS sends an HTTP "200 OK" response including the requested results in the body.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Date: Thu, 10 Aug 2006 10:50:39 GMT
Content-Type: application/vnd.oma.search+xml; charset="utf-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<search-set xmlns="urn:oma:xml:xdm:search" xmlns:up="urn:oma:xml:xdm:user-profile">

<search id="1234">
  <response>
   <up:user-profile uri="A@example.com"><up:display-name>Alex</up:display-name></up:user-profile>
   <up:user-profile uri="B@example.com"><up:display-name>Brian</up:display-name></up:user-profile>
   <up:user-profile uri="C@example.com"><up:display-name>Chris</up:display-name></up:user-profile>
   <up:user=profile uri="D@example.com"><up:display-name>David</up:display-name></up:user-profile>
  </response>
</search>

</search-set>
```

7) The Search Proxy routes the response to the Aggregation Proxy.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Date: Thu, 10 Aug 2006 10:50:39 GMT
Content-Type: application/vnd.oma.search+xml; charset="utf-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<search-set xmlns="urn:oma:xml:xdm:search" xmlns:up="urn:oma:xml:xdm:user-profile">

<search id="1234">
  <response>
   <up:user-profile uri="A@example.com"><up:display-name>Alex</up:display-name></up:user-profile>
   <up:user-profile uri="B@example.com"><up:display-name>Brian</up:display-name></up:user-profile>
   <up:user-profile uri="C@example.com"><up:display-name>Chris</up:display-name></up:user-profile>
   <up:user-profile uri="D@example.com"><up:display-name>David</up:display-name></up:user-profile>
  </response>
</search>

</search-set>
```

8) The Aggregation Proxy encodes (optionally) the content and routes the response back to the XDMC.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.1
Date: Thu, 10 Aug 2006 10:50:39 GMT
Content-Type: application/vnd.oma.search+xml; charset="utf-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<search-set xmlns="urn:oma:xml:xdm:search" xmlns:up="urn:oma:xml:xdm:user-profile">

<search id="1234">
  <response>
```

```
   <up:user-profile uri="A@example.com"><up:display-name>Alex</up:display-name></up:user-profile>
   <up:user-profile uri="B@example.com"><up:display-name>Brian</up:display-name></up:user-profile>
   <up:user-profile uri="C@example.com"><up:display-name>Chris</up:display-name></up:user-profile>
   <up:user-profile uri="D@example.com"><up:display-name>David</up:display-name></up:user-profile>
  </response>
</search>

</search-set>
```

C.7 Examples of Access Permissions Documents

C.7.1 Administrator Controlled Access Permission Document

This example shows an Access Permissions Document for a user with XUI “sip:joe@example.com”. A user with the identity “sip:bob@example.com” is the administrator of the Access Permissions in the User Directory. The user “sip:joe@example.com” is allowed to do any changes in his XDM Documents apart from changing the Access Permissions Document as a <directory-rule> element is missing and therefore the default Access Permissions apply, allowing the Primary Principal access to any XDM operation.

```
<?xml version="1.0" encoding="UTF-8"?>
<ap-rules xmlns="urn:oma:xml:xdm:ap"
    xmlns:cp="urn:ietf:params:xml:ns:common-policy"
    xmlns:ocp="urn:oma:xml:xdm:common-policy">
<access-permissions-document-rule>
 <cp:ruleset>
  <cp:rule id="ap-admin">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:bob@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
    </cp:ruleset>
</access-permissions-document-rule>
</ap-rules>
```

C.7.2 Administrator Controlled User Directory

This example shows an Access Permissions Document for a user with XUI “sip:joe@example.com”. The user with the identity “sip:bob@example.com” is the administrator of all XDM Document in this User Directory. The user “sip:joe@example.com” can read his XDM Documents as a Primary Principal always has read access to its User Directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<ap-rules xmlns="urn:oma:xml:xdm:ap"
    xmlns:cp="urn:ietf:params:xml:ns:common-policy"
    xmlns:ocp="urn:oma:xml:xdm:common-policy">
<access-permissions-document-rule>
 <cp:ruleset>
  <cp:rule id="ap-admin">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:bob@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
```

```
    </cp:rule>
    </cp:ruleset>
</access-permissions-document-rule>
<directory-rule>
 <cp:ruleset>
  <cp:rule id="allow-all">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:bob@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
    </cp:ruleset>
</directory-rule>
</ap-rules>
```

C.7.3 Granting only a View of the Access Permissions Document to all users

This example shows an Access Permission Document for a user with XUI “sip:joe@example.com”. The user “sip:bob@example.com” is the administrator of all XDM Document is this User Directory. The user “sip:joe@domain can still read his XDM Documents as a Primary Principal always has read access to its User Directory. The user “sip:john@example.com” is allowed to retrieve all XDM Documents in “sip:joe@example.com”’s User Directory. When the user sip:john@example.com is retrieving the Access Permission Document he will get an Access Permissions Document with only the “ap- own-many” rule and the “allow-retrieve” rule as the “allow-retrieve-own” rule grants him and any other user access to an Access Permissions Document which only contains rules related to the requesting user. The “allow-retrieve-own” rule controls also that a user’s Access Permissions List Document is updated by the XDMS when this user is granted access to the User Directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<ap-rules xmlns="urn:oma:xml:xdm:ap"
    xmlns:cp="urn:ietf:params:xml:ns:common-policy"
    xmlns:ocp="urn:oma:xml:xdm:common-policy">
<access-permissions-document-rule>
 <cp:ruleset>
  <cp:rule id="ap-admin">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:bob@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
<cp:rule id="ap-own-many">
    <cp:conditions>
         <cp:identity>
            <cp:many/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-retrieve-own-data/>
     </cp:actions>
    </cp:rule>
    </cp:ruleset>
</access-permissions-document-rule>
<directory-rule>
 <cp:ruleset>
  <cp:rule id="allow-all">
    <cp:conditions>
         <cp:identity>
```

```
            <cp:one id="sip:bob@example.com"/>
          </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
  <cp:rule id=" allow-retrieve">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:john@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-retrieve/>
     </cp:actions>
    </cp:rule>
    </cp:ruleset>
</directory-rule>
</ap-rules>
```

C.7.4 Blocking a single User to retrieve the Access Permissions Document

This example shows an Access Permission Document for a user with XUI “sip:joe@example.com”. The user with the identity “sip:bob@example.com” is the administrator of all XDM Document is this User Directory. The user “sip:joe@ example.com” can still read his XDM Documents as a Primary Principal always has read access to his User Directory. The user “sip:john@ example.com” is allowed to retrieve all XDM Documents in the user “sip:joe@ example.com”’s User Directory. The user “sip:alice@example.com” is blocked to retrieve any data about her Access Permissions to this User Directory. If the user “sip:alice@example.com” tries to retrieve the Access Permissions Document, she will get a “Forbidden” response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ap-rules xmlns="urn:oma:xml:xdm:ap"
    xmlns:cp="urn:ietf:params:xml:ns:common-policy"
    xmlns:ocp="urn:oma:xml:xdm:common-policy">
<access-permissions-document-rule>
 <cp:ruleset>
  <cp:rule cp:id="ap-admin">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:bob@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
<cp:rule cp:id="ap-many">
    <cp:conditions>
         <cp:identity>
            <cp:many>
              <cp:except id="sip:alice@example.com"/>
            </cp:many>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-retrieve-own-data/>
     </cp:actions>
    </cp:rule>
    </cp:ruleset>
</access-permissions-document-rule>
<directory-rule>
 <cp:ruleset>
  <cp:rule cp:id="allow-all">
```

```
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:bob@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
  <cp:rule cp:id="allow-retrieve">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:john@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-retrieve/>
     </cp:actions>
    </cp:rule>
    </cp:ruleset>
</directory-rule>
</ap-rules>
```

C.7.5 Access Permissions Document with a filter

This example shows an Access Permission Document for the “group” Application Usage. The user with the identity “sip:bob@example.com” is the administrator of all XDM Document is this User Directory. The user “sip:carl@example.com” is allowed to forward part of the group document “mysoccerteam” as described by the <filter set> element below.

```
<?xml version="1.0" encoding="UTF-8"?>
<ap-rules xmlns="urn:oma:xml:xdm:ap"
    xmlns:cp="urn:ietf:params:xml:ns:common-policy"
    xmlns:ocp="urn:oma:xml:xdm:common-policy"
    xmlns:fi="urn:ietf:params:xml:ns:simple-filter">
<access-permissions-document-rule>
 <cp:ruleset>
  <cp:rule cp:id="ap-admin">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:bob@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
</cp:ruleset>
</access-permissions-document-rule>
<directory-rule>
 <cp:ruleset>
  <cp:rule cp:id="allow-all">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:bob@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
  </cp:ruleset>
</directory-rule>
<document-rule path="mysoccerteam">
 <cp:ruleset>
   <cp:rule cp:id=" forward">
     <cp:conditions>
```

```
        <cp:identity>
           <cp:one id="sip:carl@example.com"/>
        </cp:identity>
      </cp:conditions>
    <cp:actions>
        <allow-forward/>
    </cp:actions>
    <cp:transformations>
      <fi:filter-set>
        <fi:ns-bindings>
          <fi:ns-binding prefix="gr" urn="urn:oma:xml:poc:list-service"/>
        </fi:ns-bindings>
        <fi:filter id="ap-include">
          <fi:what>
            <fi:include type="xpath">//gr:list-service/gr:display-name</fi:include>
            <fi:include type="xpath">//gr:list-service/gr:list</fi:include>
            <fi:include type="xpath">//gr:list-service/gr:invite-members</fi:include>
           </fi:what>
        </fi:filter>
     </fi:filter-set>
    </cp:transformations>
   </cp:rule>
 </cp:ruleset>
</document-rule>
</ap-rules>
```

C.7.6 Administrator and Primary Principal Controlled User Directory.

This example shows an Access Permissions Document for a user with XUI “sip:joe@example.com”. The user with the identity “sip:bob@example.com” is the administrator of all XDM Documents in this User Directory. The user “sip:joe@example.com”, the Primary Principal, can also handle his own XDM Documents apart from the Access Permissions Document.

```
<?xml version="1.0" encoding="UTF-8"?>
<ap-rules xmlns="urn:oma:xml:xdm:ap"
    xmlns:cp="urn:ietf:params:xml:ns:common-policy"
    xmlns:ocp="urn:oma:xml:xdm:common-policy">
<access-permissions-document-rule>
 <cp:ruleset>
  <cp:rule id="ap-admin">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:bob@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
    </cp:ruleset>
</access-permissions-document-rule>
<directory-rule>
 <cp:ruleset>
   <cp:rule id="ap-xui">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:joe@example.com"/>
         </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
  <cp:rule id="allow-all">
    <cp:conditions>
         <cp:identity>
            <cp:one id="sip:bob@example.com"/>
```

```
        </cp:identity>
    </cp:conditions>
    <cp:actions>
        <allow-any-operation/>
     </cp:actions>
    </cp:rule>
    </cp:ruleset>
</directory-rule>
</ap-rules>
```

For more examples with the <document-rule> element in an Access Permissions Documents see [XDM_Group] and [XDM_List].

C.8 Examples of XDCP Operations

C.8.1 Differential Read - No Filter

This example uses data similar to that found in Annex C.9.1. In this example flow, the XDMC uses differential read "catch-up" to the current state of the Request History Information Document of a List XDM Document. This example does not use a <filter-set> element; for an example of differential read operation using a filter, refer to the subsequent section of this annex.
.

In the first figure below, the XDMC has the history document up to requests for Bob and Alice. Therefore, the XDMC has the E-Tag of the Request History Information Document of the associated List XDM Document post the requests from Bob and Alice (i.e., after the time of "time2"). After "time2", Ted and Carol make requests against this document, which the XDMS has recorded in the history document.

```
<?xml version="1.0" encoding="UTF-8"?>
<request-history xmlns="urn:oma:xml:xdm:request-history">
 <document id="index">
  <requestor id="sip:bob@domain">
    <last-requests>
      <request type="retrieve" result="unauthorized" timestamp="time1"/>
    </last-requests>
  </requestor>
  <requestor id="sip:alice@domain">
    <last-requests>
      <request type="delete" result="unauthorized" timestamp="time2" counter="1"/>
      </request>
   </last-requests>
  </requestor>
  <requestor id="sip:ted@domain">
    <last-requests>
      <request type="retrieve" result="authorized" timestamp="time3" counter="1"/>
    </last-requests>
  </requestor>
  <requestor id="sip:carol@domain">
    <last-requests>
      <request type="retrieve" result="unauthorized" timestamp="time4" counter="1"/>
    </last-requests>
  </requestor>
 </document>
</request-history>
```

Figure C.9 - Example Request History Information Document

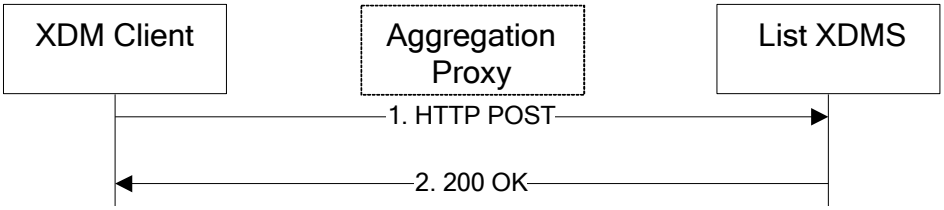

Figure C.10 - XDCP Differential Read Request operation - no filter

The abbreviated flow diagram does not show the Aggregation proxy nor authentication headers nor other HTTP headers. The Differential Read involves an HTTP POST Request and Response with XDCP Document <request> and <response> elements, respectively. The details of the flow are as follows:

1) The XDMC sends an HTTP POST request to retrieve the Request History Information Document associated with the Resource-List Document of the Principal "sip:joebloggs@example.com"'. The <etag> element contains a string that is the value of the E-Tag of the XDM document version that the XDMC currently possesses (= "etag1"):

```
POST /resource-lists/users/sip:joebloggs@example.com/oma_requests/history HTTP/1.1
Host xcap.example.com

Content-Type: application/xdcp-document+xml; charset="utf-8"
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <request>
      <diff-read>
          <etag>etag1</etag>
      </diff-read>
  </request>
</xdcp-document>
```

Figure C.11 - XDCP Differential Read Request - No Filter

2) The XDMS returns a multipart MIME in an HTTP "200 OK" response. This MIME has two parts. One part is an XDCP Document containing a <response> element that has the child element <done> with the value "etag2", which is the current E-TAG of the document. The other MIME part is an XCAP-Diff MIME containing a <document> child element, which in turn contains an <add> child element conveying the added entries to the history document that have a result attribute equal to "unauthorized".

```
HTTP/1.1 200 OK
…
Content-Type: multipart/mixed boundary="boundary1
Content-Length: (…)

--boundary1
Content-Type: application/xdcp-document+xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
   <response><done/></response>
</xdcp-document>

--boundary1
<?xml version="1.0" encoding="UTF-8"?>
<xd:xcap-diff
          xmlns:xd="urn:ietf:params:xml:ns:xcap-diff"
          xcap-root="http://xcap.example.com/root">
    <xd:document
          previous-etag="etag1"
```

```
          sel="resource-lists/users/sip:joe@domain/oma_requests/history"
          new-tag="etag2">
    <xd:add sel="/request-history/document[@id='index']">
      <requestor id="sip:ted@domain">
          <last-requests>
             <request type="retrieve" result="authorized" timestamp="time3" counter="1"/>
          </last-requests>
      </requestor>
      <requestor id="sip:carol@domain">
          <last-requests>
              <request type="retrieve" result="unauthorized" timestamp="time4" counter="1"/>
          </last-requests>
      </requestor>
    </xd:add>
  </xd:document>
</xd:xcap-diff>
```

Figure C.12 - XDCP Differential Read Response - No Filter

C.8.2 Differential Read - With Filter

In this example, the XDMC uses differential read with a filter to obtain the identities of XDM Users who have made a failed request against an application document, i.e., were not authorized. The idea is that an XDM User may not be interested to see all requests made against the document, and so this approach reduces the total amount of information being transmitted, stored, etc. If the XDMC has not previously read the Request History Information Document with this filter, then it must do so first (the following example of this annex has this case). Note: If the XDMC should later need the entire request history, the XDMC needs to read the entire request document without the filter.

Based on the differential read response, the XDMC can inform the use XDM User, who can then make a determination whether to change the access permissions to allow Carol, in the case below, to access the List Document.

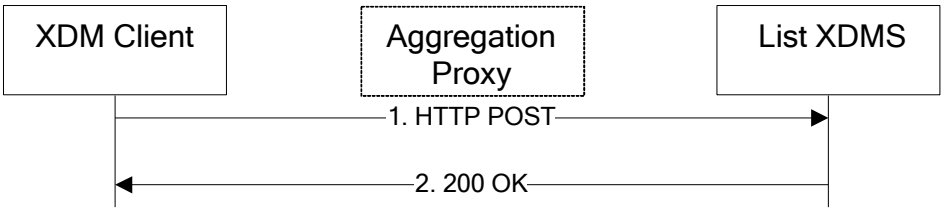

Figure C.13 - XDCP Differential Read Request operation - with Filter

The details of the flow are as follows:

1) The XDMC sends an HTTP POST Request to retrieve the Request History Information Document associated with the Resource-List XDM Document of the Principal “sip:joebloggs@example.com”’. The <etag> element contains the value of the E-Tag of the XDM document version that the XDMC currently possesses (= "etag1"). The <filter-set> [RFC4661] child element contains a <filter> child element with an "id". Similar to use of filters for SIP based subscriptions (referring to section 3.4 of [RFC4661]), the "uri" attribute of the <filter> element does not have to be included as the XDCP Request applies to the resource identified in the Request URI of the HTTP POST Request. The remainder of the filter selects "requestor" elements that contain a child node "last-requests" that in turn contain a <request> child element with a "result" attribute equal to "not-authorized".

```
POST /resource-lists/users/sip:joebloggs@example.com/oma_requests/history HTTP/1.1
Host xcap.example.com

Content-Type: application/xdcp-document+xml; charset="utf-8"
Content-Length: (…)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xdcp:xdcp-document
        xmlns:xdcp="urn:oma:xml:xdm:xdcp-document"
        xmlns:fs="urn:ietf:params:xml:ns:simple-filter>
  <xdcp:request>
     <xdcp:diff-read>
        <xdcp:etag>etag1</xdcp:etag>
        <fs:filter-set>
           <fs:ns-binding prefix="rh" urn="urn:oma:xml:xdm:request-history"/>
           <fs:filter id="not-authorized-filter">
               <fs:what>
                  <fs:include type="xpath">
                     /rh:request-history/rh:document[@id='index']/rh:requestor
                          [/rh:last-requests/rh:request/@result='unauthorized']
                  </fs:include>
               </fs:what>
           </fs:filter>
        </fs:filter-set>
     </xdcp:diff-read>
  </xdcp:request>
</xdcp:xdcp-document>
```

Figure C.14 - XDCP Differential Read Request - with Filter

2) The XDMS returns a multipart MIME in an HTTP "200 OK" Response. This MIME has two parts. One part is an XDCP Document containing a <response> element that has the child element <done>. The other part is an XCAP-Diff MIME containing a <document> child element, which in turn contains an <add> child element conveying the added entries to the history document that match the filter.

```
HTTP/1.1 200 OK
…
Content-Type: multipart/mixed boundary="boundary1
Content-Length: (…)

--boundary1
Content-Type: application/xdcp-document+xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <response><done/></response>
</xdcp-document>

--boundary1
<?xml version="1.0" encoding="UTF-8"?>
<xd:xcap-diff xmlns:xd="urn:ietf:params:xml:ns:xcap-diff"
      xcap-root="http://xcap.example.com/root">
  <xd:document
         previous-etag="etag1"
         sel="resource-lists/users/sip:joe@domain/oma_requests/history"
         new-tag="etag2">
    <xd:add sel="/request-history/document[@id='index']">
      <requestor id="sip:carol@domain">
        <last-requests>
            <request type="retrieve" result="unauthorized" timestamp="time4" counter="1"/>
        </last-requests>
      </requestor>
    </xd:add>
  </xd:document>
</xd:xcap-diff>
```

Figure C.15 - XDCP Differential Read Response - with Filter

C.8.3 "Reactive Authorization" via Differential Read and Write with Filter

In this example, the XDMC uses differential read and write operations to perform some parts of the steps described in Appendix H. when handling the Request History Information Document of a List XDM Document.

The following figure depicts the Request History Information Document for a List Document involving authorized and unauthorized requests from Bob, Alice, Ted, and Carol.

```
<?xml version="1.0" encoding="UTF-8"?>
<request-history xmlns="urn:oma:xml:xdm:request-history">
 <document id="index">
  <requestor id="sip:bob@domain">
    <last-requests>
      <request type="retrieve" result="not-authorized" timestamp="time1"/>
    </last-requests>
  </requestor>
  <requestor id="sip:alice@domain">
    <last-requests>
      <request type="delete" result="unauthorized" timestamp="time2" counter="1"/>
      </request>
   </last-requests>
  </requestor>
  <requestor id="sip:ted@domain">
    <last-requests>
      <request type="retrieve" result="authorized" timestamp="time3" counter="1"/>
    </last-requests>
  </requestor>
  <requestor id="sip:carol@domain">
    <last-requests>
      <request type="retrieve" result="unauthorized" timestamp="time4" counter="1"/>
    </last-requests>
  </requestor>
 </document>
</request-history>
```

Figure C.16 - Starting State of Request History Information Document on the XDMS

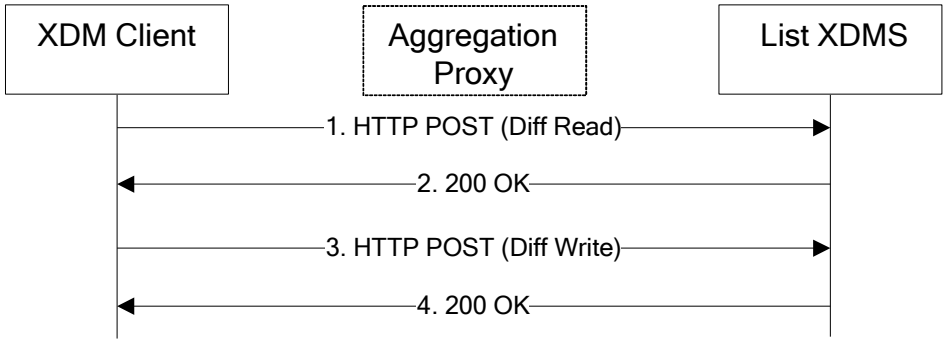

Figure C.17 - "Reactive Authorization" with XDCP Differential Read and Write Request

The example assumes the XDMC fetches all "unauthorized" entries, e.g., to determine whether to authorize certain XDM Users for the particular document. The XDMC treats some of the unauthorized users by updating e.g., the Access Permissions document (not shown in the sequence).

The XDMC now seeks to delete the unauthorized request entries from the request history associated with this List document that has been treated. The example assumes the XDMC does not maintain a locally stored copy of the Request History Information Document, and so the XDMC first reads the request history associated with the List Document using the filter to select unauthorized request entries. To cause the XDMS to return all entries matching the filter, the XDMC SHALL NOT include an <etag> element in the XDCP document of the XDCP Request.

The details of the flow are as follows:

1) The XDMC sends an HTTP POST Request to the Request History Information Document associated with the Resource-List XDM Document of the Principal “sip:joebloggs@example.com”’. The <request> element contains the child

element <diff-read>. The <diff-read> element contains a <filter-set> [RFC4661] child element containing a <filter> child element with the "id" attribute associated with the filter. The "uri" attribute of the <filter> element does not have to be included as the XDCP Request applies to the resource identified in the Request URI of the HTTP POST Request. The rest of the filter selects "requestor" elements that contain a child node "last-requests" that in turn contain a <request> child element with a "result" attribute equal to "unauthorized".

```
POST /resource-lists/users/sip:joebloggs@example.com/oma_requests/history HTTP/1.1
Host xcap.example.com

Content-Type: application/xdcp-document+xml; charset="utf-8"
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<xd:xdcp-document
        xmlns:xdcp="urn:oma:xml:xdm:xdcp-document"
        xmlns:fs="urn:ietf:params:xml:ns:simple-filter>
  <xdcp:request>
     <xdcp:diff-read>
        <fs:filter-set>
           <fs:ns-binding prefix="rh" urn="urn:oma:xml:xdm:request-history"/>
           <fs:filter id="not-authorized-filter">
               <fs:what>
                  <fs:include type="xpath">
                     /rh:request-history/rh:document[@id='index']/rh:requestor
                          [/rh:last-requests/rh:request/@result='unauthorized']
                  </fs:include>
               </fs:what>
           </fs:filter>
        </fs:filter-set>
     </xdcp:diff-read>
  </xdcp:request>
</xdcp:xdcp-document>
```

Figure C.18 - XDCP Differential Read Request - Filter & Null E-Tag

2) The XDMS returns a multipart MIME in an HTTP “200 OK” response. This MIME has two parts. One part is an XDCP Document containing a <response> element that has the child element <done>. The second MIME part is an XCAP-Diff MIME containing a <document> child element, which in turn, contains a <replace> child element conveying entries in the request history document for the List XDMS document with id="index" that have a "result" attribute value of "unauthorized". The current E-TAG of the history document is conveyed via the XCAP-Diff MIME "new-etag" attribute.

```
HTTP/1.1 200 OK
…
Content-Type: multipart/mixed boundary="boundary1
Content-Length: (…)

--boundary1
Content-Type: application/xdcp-document+xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
   <response><done/></response>
</xdcp-document>

--boundary1
<?xml version="1.0" encoding="UTF-8"?>
<xd:xcap-diff
          xmlns:xd="urn:ietf:params:xml:ns:xcap-diff"
          xcap-root="http://xcap.example.com/root">
    <xd:document
          sel="resource-lists/users/sip:joe@domain/oma_requests/history"
          new-tag="etag2">
   <xd:replace> sel=”/”>
    <request-history xmlns="urn:oma:xml:xdm:request-history">
```

```
    <document id="index>
     <requestor id="sip:bob@domain">
       <last-requests>
         <request type="retrieve" result="unauthorized" timestamp="time1"/>
       </last-requests>
     </requestor>
     <requestor id="sip:alice@domain">
       <last-requests>
         <request type="delete" result="unauthorized" timestamp="time2" counter="1"/>
       </last-requests>
     </requestor>
     <requestor id="sip:carol@domain">
       <last-requests>
         <request type="retrieve" result="unauthorized" timestamp="time4" counter="1"/>
       </last-requests>
     </requestor>
    </request-history>
    </document>
</xd:replace>
</xd:xcap-diff>
```

Figure C.19 - XDCP Differential Read Response - Filter & Null E-Tag

The XDMC now has the following locally stored Request History Information Document:

```
<?xml version="1.0" encoding="UTF-8"?>
<request-history xmlns="urn:oma:xml:xdm:request-history">
 <document id="index">
  <requestor id="sip:bob@domain">
    <last-requests>
      <request type="retrieve" result="unauthorized" timestamp="time1"/>
    </last-requests>
  </requestor>
  <requestor id="sip:alice@domain">
    <last-requests>
      <request type="delete" result="unauthorized" timestamp="time2" counter="1"/>
      </request>
   </last-requests>
  </requestor>
  <requestor id="sip:carol@domain">
    <last-requests>
      <request type="retrieve" result="unauthorized" timestamp="time4" counter="1"/>
    </last-requests>
  </requestor>
 </document>
</request-history>
```

Figure C.20 - XDMC Locally Stored Request History Information Document

The authorized entry is not present because it does not match the filter.

3) When the XDMC has treated the three unauthorized users by e.g., giving them access to the “index” document, the XDMC sends an HTTP POST Request to the Request History Information Document associated with the Resource-List XDM Document of the Principal “sip:joebloggs@example.com” to remove them from the Request History Information Document. The <filter-set> is the same as in the previous step. The XCAP-Diff MIME body part contains patch-ops <remove> operation elements that remove the three entries returned in the previous step. The current E-TAG value ("etag2") of the history document is conveyed in the "previous-etag" attribute of the XCAP-Diff MIME.

```
POST /resource-lists/users/sip:joebloggs@example.com/oma_requests/history HTTP/1.1
Host xcap.example.com
```

```
…
Content-Type: multipart/mixed boundary="boundary1
Content-Length: (…)

--boundary1

Content-Type: application/xdcp-document+xml; charset="utf-8"
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<xd:xdcp-document
        xmlns:xdcp="urn:oma:xml:xdm:xdcp-document"
        xmlns:fs="urn:ietf:params:xml:ns:simple-filter>
  <xdcp:request>
     <xdcp:diff-write>
        <fs:filter-set>
           <fs:ns-binding prefix="rh" urn="urn:oma:xml:xdm:request-history"/>
           <fs:filter id="not-authorized-filter">
               <fs:what>
                  <fs:include type="xpath">
                     /rh:request-history/rh:document[@id='index']/rh:requestor
                           [/rh:last-requests/rh:request/@result='unauthorized']
                  </fs:include>
               </fs:what>
           </fs:filter>
        </fs:filter-set>
     </xd:diff-write>
  </xdcp:request>
</xdcp:xdcp-document>

--boundary1
<?xml version="1.0" encoding="UTF-8"?>
<xd:xcap-diff xmlns:xd="urn:ietf:params:xml:ns:xcap-diff"
      xcap-root="http://xcap.example.com/root">
  <xd:document
         previous-etag="etag2"
         sel="resource-lists/users/sip:joe@domain/oma_requests/history">
    <xd:remove sel="/request-history/document[@id='index']/requestor[@id='sip:bob@domain']"/>
    <xd:remove sel="/request-history/document[@id='index']/requestor[@id='sip:alice@domain']"/>
    <xd:remove sel="/request-history/document[@id='index']/requestor[@id='sip:carol@domain']"/>
  </xd:document>
</xd:xcap-diff>
```

Figure C.21 - XDCP Differential Write Request - Filter

4) The XDMS returns a multipart MIME in an HTTP “200 OK” Response with an XDCP Document containing a <response> element that has the child element <done-new-etag> with the new E-Tag value ("etag3") of the Request Information History Document.

```
HTTP/1.1 200 OK
…
Content-Type: multipart/mixed boundary="boundary1
Content-Length: (…)

--boundary1
Content-Type: application/xdcp-document+xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <response><done-new-etag>etag3</done-new-etag></response>
</xdcp-document>
```

Figure C.22 - XDCP Differential Write Response - Filter

The Request History Information Document as stored on the XDMS is:

```
<?xml version="1.0" encoding="UTF-8"?>
<request-history xmlns="urn:oma:xml:xdm:request-history">
 <document id="index">
  <requestor id="sip:ted@domain">
    <last-requests>
      <request type="retrieve" result="authorized" timestamp="time3" counter="1"/>
    </last-requests>
  </requestor>
 </document>
</request-history>
```

Figure C.23 - Final State of Request History Information Document on the XDMS

C.8.4 Set Document Reference

This example shows an XDCP Documents used to set a Document Reference to another XDM Document. The user in this example has two user identities “sip:alice@example.com and alice.swansson@example.com and wants to use the same User Access Policy Document for both identities.

The user has already created a User Access Policy Document using the identity “sip:alice.swansson@example.com as XUI. The User Access Policy Document for the identity sip:alice@example.com does not exist.

The user selects the device that can handle the user identity “sip:alice@example.com” and selects that the User Access Policy Document for this identity shall be the same as the User Access Policy Document for the user identity “sip:alice.swansson@example.com”. The XDMC issues the following XDCP Request.

```
POST / org.openmobilealliance.access-rules/users/sip:alice@example.com/access-rules HTTP/1.1
Host xcap.example.com
…
Content-Type: application/xdcp-document+xml; charset="utf-8"
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <request>
      <set-doc-ref>
          <reference>org.openmobilealliance.access-
   rules/users/sip:alice.swansson@example.com/access-rules
          </reference>
          <display-name>Alice Swansson’s User Access Policy</display-name>
      </set-doc-ref>
  </request>
</xdcp-document>
```

The XDMS receives the XDCP request and check the Access Permissions Document related to the identity “sip:alice@example.com” to authorize the XDCP request. The XDMS checks also that it is possible to retrieve the User Access Policy Document referenced in the <reference> element. In this example both checks are positive and the XDMS returns the following XDCP Response to the XDMC:

```
HTTP/1.1 201 Created
…
Content-Type: application/xdcp-document+xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <response>
   <done/>
  </response>
</xdcp-document>
```

C.8.5 Retrieve Document Reference

This example shows how Document Reference information related to a User Access Policy Document can be retrieved. “sip:alice@example.com” is the XUI of the User Access Policy Document in this example. The XDMC issues the following XDCP Request:

```
POST / org.openmobilealliance.access-rules/users/sip:alice@example.com/access-rules HTTP/1.1
Host xcap.example.com
…
Content-Type: application/xdcp-document+xml; charset="utf-8"
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <request>
      <retrieve-doc-ref/>
  </request>
</xdcp-document>
```

The XDMS receives the XDCP Request and checks the Access Permissions corresponding to the identity “sip:alice@example.com. The requesting user is allowed to retrieve information about the User Access Policy Document and therefore the XDMS returns the following XDCP response:

```
HTTP/1.1 200 OK
…
Content-Type: application/xdcp-document+xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <response>
   <retrieve-doc-ref-result>
       <reference>org.openmobilealliance.access-rules/users/sip:alice.swansson@example.com/access-
   rules
       </reference>
       <display-name>Alice Swansson’s User Access Policy</display-name>
   </retrieve-doc-ref-result>
  </response>
</xdcp-document>
```

C.8.6 Remove Document Reference

This example shows how Document Reference information can be deleted. Document Reference information related to the identity “sip:alice@example.com” User Access Policy Document is removed. The XDMC issues the following XDCP request:

```
POST / org.openmobilealliance.access-rules/users/sip:alice@example.com/access-rules HTTP/1.1
Host xcap.example.com
…
Content-Type: application/xdcp-document+xml; charset="utf-8"
…
<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <request>
      <remove-doc-ref/>
  </request>
</xdcp-document>
```

The XDMS receives the XDCP Request and checks the Access Permissions corresponding to the identity "sip:alice@example.com. The requesting user is allowed to remove Document Reference information about the User Access Policy Document and therefore the XDMS returns the following XDCP response:

```
HTTP/1.1 200 OK
…
Content-Type: application/xdcp-document+xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <response>
   <done/>
  </response>
</xdcp-document>
```

C.8.7 Forwarding XDM Resources

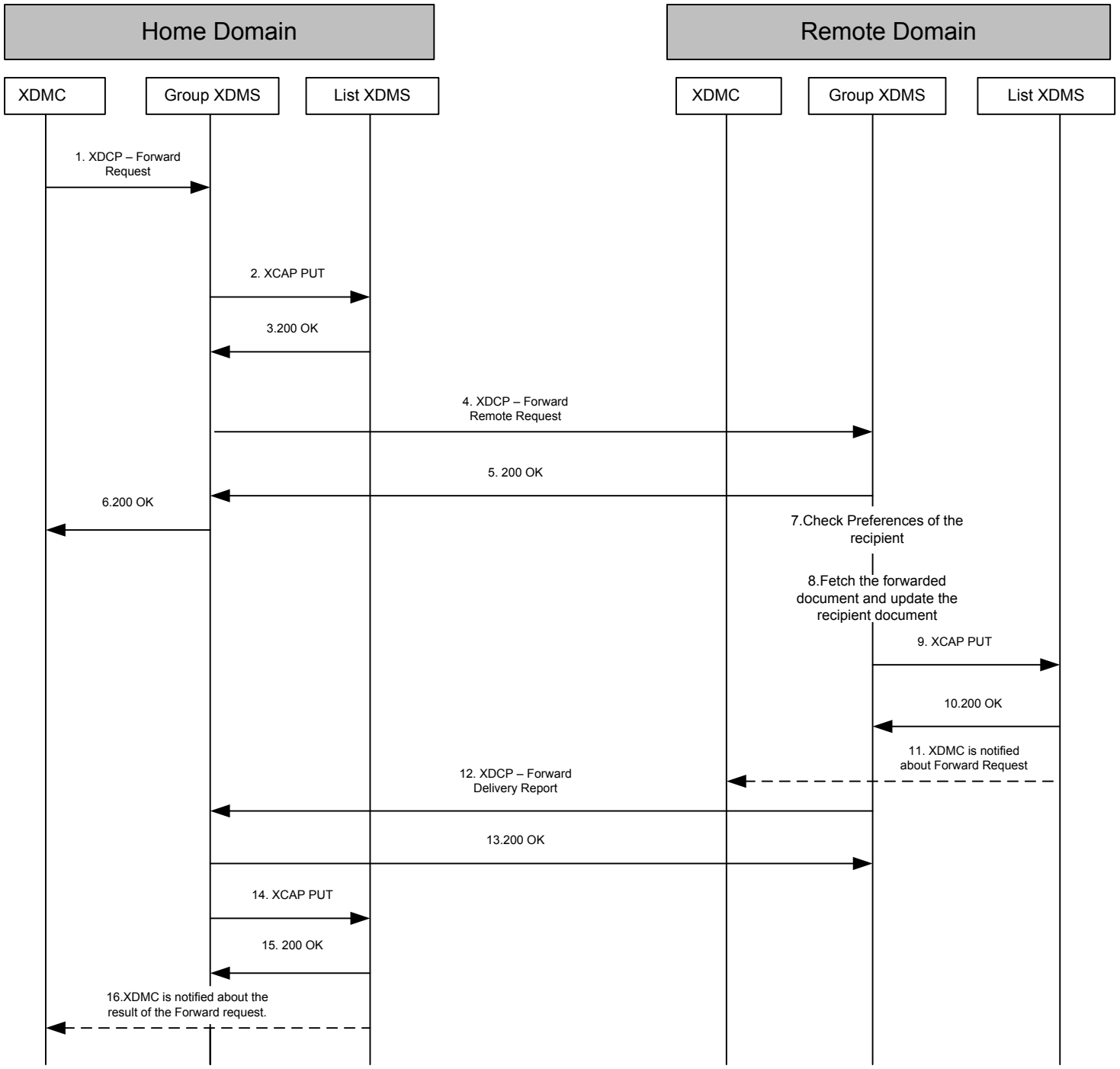

Figure C.24 - XDM Forward Example Flows

NOTE: For the sake of brevity, the abbreviated flow diagram does not show the flows involved with Aggregation Proxy, Cross-network Proxy and also the HTTP headers shown are minimal. The details of the flow are as follows:

1) The XDMC of the User ”joe@example.com” sends an HTTP POST request to forward the group document name “my_buddies” to the recipient “sip:alice@foo.com”

```
POST
   /org.openmobilealliance.xdcp/org.openmobilealliance.groups/users/sip:joe@example.com/my_buddies
   HTTP/1.1
Host xcap.example.com

Content-Type: application/xdcp-document+xml; charset="utf-8"
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <request>
      <forward>
          <recipients-list>
              <list>
                  <entry uri="sip:alice@foo.com"/>
              </list>
          </recipients-list>
          <note> Here comes my buddies </note>
          <delivery-report>true</delivery-report>
          <request-id>fjkl9078</request-id>
  </request>
</xdcp-document>
```

2) The Group XDMS on receiving the above request updates the Forwarding Notification List of the User Joe with the XCAP PUT. The content carried in the XCAP PUT body is as follows:

```
<delivery-notification request-id="fjkl9078">
  <document-
   uri>http://xcap.example.com/org.openmobilealliance.groups/users/sip:joe@example.com/my_buddies</
   document-uri>
  <recipients-list>
    <entry uri="sip:alice@foo.com" status="pending"/>
  </recipients-list>
</delivery-notification>
```

3) The List XDMS updates the Forwarding Notification List by adding the <delivery-notification> entry received and sends 200 OK response.

4) Since the recipient Alice is in different domain the Group XDMS creates the XDM Resource to be forwarded in a temporary storage and does XDCP Forward Remote request as follows:

```
POST /org.openmobilealliance.xdcp/org.openmobilealliance.groups/users/sip:alice@foo.com HTTP/1.1
Host xcap.foo.com

Content-Type: application/xdcp-document+xml; charset="utf-8"
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <request>
      <forward-remote>
          <document-uri>http://xcap.example.com/org.openmobilealliance.groups/temp/mybuddies.xml
   </document-uri>
          <recipients-list>
              <list>
                  <entry uri="sip:alice@foo.com"/>
              </list>
          </recipients-list>
          <note> Here comes my buddies </note>
          <delivery-report>true</delivery-report>
          <request-id>fjkl9078</request-id>
          <size>800</size>
          <expiration-time>Thu, 01 Jul 2010 16:00:00 GMT</expiration-time>
          <content-type>application/vnd.oma.poc.groups+xml</content-type>
```

```
  </request>
</xdcp-document>
```

5) The Group XDMS of Alice on receiving the above remote forward request checks whether the recipients listed in the request are in its domain and then creates the 200 OK response as follows and send to the Group XDMS of Joe:

```
HTTP/1.1 200 OK
…
Content-Type: application/xdcp-document+xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <response>
      <remote-forward-result/>
  </response>
</xdcp-document>
```

6) The Group XDMS of Joe on receiving the above remote forward response creates 200 OK response to the forward request received from XDMC of Joe.

```
HTTP/1.1 200 OK
…
Content-Type: application/xdcp-document+xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <response>
      <forward-result/>
  </response>
</xdcp-document>
```

7) The Group XDMS of Alice checks the preferences of the Alice for handling the Forward request. Here the case shown is that Alice wants to accept the XDM Resource received from Joe.

8) The Group XDMS fetches the XDM Resource using the URI received in the Forward Remote request and stores the Group Document in Alice's User Tree.

9) The Group XDMS of Alice updates the Forwarding Notification List of Alice with the details of the received remote forward request. The entry added is as follows:

```
<request-notification-list auid="org.openmobilealliance.groups">
  <request document-uri="http://www.example.com/forward-data/mybuddies.xml">
      <sender-identity>"sip:joe@example.com"</sender-identity>
      <status>delivered</status>
      <content-type>application/vnd.oma.poc.groups+xml</content-type>
      <time-stamp>Tue, 29 Jul 2010 16:00:00 GMT </timestamp>
      <note> Here comes my buddies </note>
  </request>
</request-notification>
```

10) The List XDMS of Alice updates the Forwarding Notification List of Alice with the above entry and sends 200 OK response.

11) Alice would be notified about the details of the received forward request provided she has subscribed to the document changes of the Forwarding Notification List Document.

12) Since Joe has requested for the Delivery Report the Group XDMS of Alice generates the Forward Delivery Report request as follows:

```
POST /org.openmobilealliance.xdcp/org.openmobilealliance.groups HTTP/1.1
Host xcap.example.com

Content-Type: application/xdcp-document+xml; charset="utf-8"
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<xdcp-document xmlns="urn:oma:xml:xdm:xdcp-document">
  <request>
```

```
      <forward-delivery-report>
          <request-id>fjkl9078</request-id>
          <recipient-uri>"sip:alice@foo.com"</recipient-uri>
          <status>delivered</status>
   </request>
</xdcp-document>
```

13) The Group XDMS of Alice sends 200 OK response with the XDCP Response containing <done/> element.

14) The Group XDMS of Joe updates the Forwarding Notification List entry created in the Step 2 of this flow by changing the status attribute value to “delivered”.

15) The list XDMS sends 200 OK response.

16) The User Joe would get notified about the delivery status provided he has the subscription to document changes for his Forwarding Notification List document.

C.9 Examples of History Information Documents

C.9.1 Request History Information Document Example

This example shows a Request History Information Document for a user with XUI “sip:joe@example.com” for the URI List Application Usage where the users with the identities “sip:bob@example.com” and “sip:alice@example.com” have made some XDM operations towards the XDM Document with the Document Selector “resource-lists/users/sip:joe@example.com/index”. “sip:bob@example.com” has tried to retrieve the whole “index” XDM Document and “sip:alice@example.com” has tried to delete a <list> element with the “name” attribute “list-c” twice. Both requests failed. The local policy states the XDMS shall not store any requests in the <request-log> element (“N”=0).

```
<?xml version="1.0" encoding="UTF-8"?>
<request-history xmlns="urn:oma:xml:xdm:request-history">
 <document id="index">
  <requestor id="sip:bob@example.com">
    <last-requests>
      <request type="retrieve" result="unauthorized" timestamp="2010-03-28T22:20:00Z"/>
    </last-requests>
  </requestor>
  <requestor id="sip:alice@example.com">
    <last-requests>
      <request type="modify" result="unauthorized" timestamp="2010-03-28T20:20:00Z" counter="1">
         <node-selector>/resource-lists/list%5B@name=list-c%5D</node-selector>
      </request>
   </last-requests>
  </requestor>
 </document>
</request-history>
```

C.9.2 Modification History Information Document Examples

Editors’s Note: Examples of Modification History Information Documents are to be added.

C.10 XDM Preferences Document Examples

The following table shows sample XDM Preferences Document of the User “adam@example.com” for managing the History and Forward related preferences for his Group documents stored in the Group XDMS.

Following are the preferences set by the User Adam:

1. Activate History Information Recording for all his Group Documents except “my_family” Group Document.
2. Record the details of all the Authorized and Un-Authorized Modification Requests performed on his Group Documents by other Users.
3. Record the details of all the Un-Authorized Retrieval Requests towards his Group Documents.
4. Reject all the XDM Forward Requests received from the Users listed in his URI List named “black-list” stored in the List XDMS.
5. Confirm all the XDM Forward Requests received from the Users listed in his URI List named “white-list” stored in the List XDMS.

```
<?xml version="1.0" encoding="UTF-8"?>
<preferences xmlns="urn:oma:xml:xdm:xdm-prefs"
    xmlns:cp="urn:ietf:params:xml:ns:common-policy"
    xmlns:ocp="urn:oma:xml:xdm:common-policy">

  <history-prefs>
   <history-info state="on">
      <except>
      <document path="/org.openmobilealliance.groups/users/sip:adam@example.com/my_family"/>
      </except>
   </history-info>

      <filter id="sh99hu" state="on">
          <cp:conditions>
              <operation-type>
                  <modify/>
              </operation-type>
              <operation-result>
                  <authorized/>
                  <un-authorized/>
              </operation-result>
              <cp:identity>
                  <cp:many>
                      <cp:except id="sip:adam@example.com"/>
                  </cp:many>
              </cp:identity>
          </cp:conditions>
          <cp:actions>
              <store-changelog> true </store-changelog>
              <store-requestlog> true </store-requestlog>
          </cp:actions>
      </filter>

      <filter id="kbs78sh" state="on">
          <cp:conditions>
              <operation-type>
                  <retrieve/>
                  <modify/>
              </operation-type>
              <operation-result>
                  <un-authorized/>
              </operation-result>
          </cp:conditions>
          <cp:actions>
              <store-requestlog> true </store-requestlog>
          </cp:actions>
      </filter>

  </history-prefs>
  <forward-prefs>
      <cp:rule id="sj904j">
          <cp:conditions>
              <ocp:external-list>
                  <ocp:entry anc="http://xcap.example.org/resource-
   list/users/sip:adam@example.com/index/~~/resource-lists/list%5B@name=%22black-list%22%5D"/>
            </ocp:external-list>
          </cp:conditions>
          <cp:actions>
```

```
            <reject/>
        </cp:actions>
    </cp:rule>

    <cp:rule id="fji787">
        <cp:conditions>
            <ocp:external-list>
                <ocp:entry anc="http://xcap.example.org/resource-
  list/users/sip:adam@example.com/index/~~/resource-lists/list%5B@name=%22white-list%22%5D"/>
          </ocp:external-list>
        </cp:conditions>
        <cp:actions>
            <confirm/>
        </cp:actions>
    </cp:rule>

  </forward-prefs>
</preferences>
```

Appendix D. XDMC Provisioning (Normative)

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

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

D.1 Provisioned XDMC Parameters

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

ID	Name	Description	Mandatory (M) /Optional (O)
1	Application identity	Uniquely identifies the application	M
2	Application name	User displayable name for the XDM service	M
3	Provider–ID	Identity of the XDM Service Provider	O
4	Network Access Definitions	Reference to the connection used for the XCAP traffic.	M
5	XDM reference to SIP/IP Core	Reference to the SIP/IP Core for accessing an XDMS using the referenced SIP/IP Core.	M
6	XCAP Root URI	The root of all XCAP resources (which points to the Aggregation Proxy address). This is used when accessing via XCAP.	M
7	XCAP Authentication user name	HTTP digest “username”, for accessing an XDMS using the XCAP protocol	O
8	XCAP Authentication password	HTTP digest password	O
9	XCAP Authentication type	Authentication method for XDMS over XCAP	O
10	Conference-URI Template	A template used by the XDMC to propose a Conference URI when creating a Group XDM Document.	O
11	Subscription Proxy URI	A SIP URI identifying the Subscription Proxy, used for subscription for notification of changes in XDM Documents.	O

NOTE 1: The parameters “XCAP Authentication username” and “XCAP Authentication password” are not needed if GAA is used in a 3GPP IMS or 3GPP2 MMD realization.

NOTE 2: The parameters “XCAP Authentication username”, “XCAP Authentication password” and “XCAP Authentication type” are not needed for a 3GPP/3GPP2 early IMS realization.

In addition, there may be Enabler-specific parameters related to the XDMC that are described in separate specifications.

One type of provisioned parameter having a reusable structure is a URI Template. A URI Template is used to describe a single syntax for a URI (e.g. Conference URI of a Group), so that the XDMC can autonomously generate a URI that complies with local policy and uniqueness constraints. It is up to separate specifications to define provisioned parameters that make use of a URI Template.

A URI Template SHALL describe a URI as defined in [RFC3986]. The template contains a sequence, in any order, of:

a. unreserved characters according to [RFC3986],

b. the characters ":" , "@" and ";"

c. substitution tags enclosed in "< >"brackets.

The XDMC SHALL support the following substitution tags:

<id> : The XDMC SHALL replace this tag with a unique identifier, generated by the XDMC using only unreserved characters according to [RFC3986].

<user> : The XDMC SHALL replace this tag with the user part of the XUI if the XUI is a Public User Identity. If the XUI is a Tel URI [RFC 3966] then the XDMC SHALL replace the <user> tag with the "global-number-digits"/"local-number-digits" part of the Tel URI. Any "visual-separator" or "+" SHALL be removed from the "global-number-digits" before the replacement takes place.

<xui> : The XDMC SHALL replace this tag with the XUI.

NOTE 3: the XUI is a Public User Identity (i.e SIP URI [RFC3261] or Tel URI [RFC3966]).

NOTE 4: usage of the <xui> tag in a URI Template may result in the generation of Tel URIs, which may not be valid for certain services (e.g. services that require SIP URIs).

If multiple Application Usages in a service provider domain use a URI Template, then the URI Template SHALL be different for each Application Usage in order to achieve generation of unique URIs.

Illustrative examples of URI templates are shown in Table 1.

Example URI Template	Example URI generated from template
sip:<id>@example.com	sip:abc123@example.com
sip:<id>_<user>@example.com	sip:abc123_joe@example.com
sip:<id>_<user>@example.com	sip:abc123_17205551212@example.com
<xui>;group=<id>	sip:joe@example.com;group=abc123
<xui>;group=<id>	tel:+1720-555-1212;group=abc123
<xui>;pres-list=<id>	sip:joe@example.com;pres-list=abc123

Table 1: Example usages of URI Templates

D.2 Application Characteristics

The Application characteristics (AC) file for XDM 2.0 service [XDM_ERELD-V2_0] MAY be used for initial provisioning of the XDMC.

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

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

Parameter Name	Req / Opt	Instances	Default
Standard Application Characteristic fields as defined in [CP_ProvCont]			
APPID	Required	1	"ap0007"
PROVIDER-ID	Optional	0 or 1	None
TO-APPREF	Required	1	None
NAME	Required	1	None
TO-NAPID	Required	1 or more	None
URI	Required	1	None

AAUTHNAME	Optional	0 or 1	None
AAUTHSECRET	Optional	0 or 1	None
AAUTHTYPE	Optional	0 or 1	None
CONF-URI-TMPLT	Optional	0 or 1	None
SUB-PROXY-URI	Optional	0 or 1	None

The Application Characteristics file for XDM 2.0 service is defined in [XDM_AC].

D.3 Management Objects

The Management Objects (MOs) for XDM 2.0 service [XDM_ERELD-V2_0] MAY be used for initial provisioning of the XDMC and SHOULD be used for continuous provisioning by Service Provider.

The Management Objects (MOs) for XDM 2.0 service is defined in [XDM_MO].

Appendix E. OMA Specific Uri-parameters (Normative)

This section defines the syntax of OMA specific uri-parameters.

E.1 AUID Uri-parameter

AUID uri-parameter is used to indicate the appropriate XDMS when the client subscribe for changes in XDM Documents.

The AUID uri-parameter takes form of:

auid ”=” token

where token represents the AUID of the appropriate Application Usage.

Example 1:

In this example XDM Documents stored on Group XDMS are subscribed.

sip:joe.bloggs@example.com;auid=org.openmobilealliance.groups

Appendix F. OMA XDCP Operations (Normative)

This section lists the distinct OMA XDCP Operations and the value of the <request> element of their associated XDCP Document (see section 5.4.2).

Operation	<request>	Reference
Setting a Document Reference	set-doc-ref	section 6.1.1.3.1, section 6.2.6.1
Removing a Document Reference	remove-doc-ref	section 6.1.1.3.1, section 6.2.6.1
Retrieving a Document Reference	retrieve-doc-ref	section 6.1.1.3.1, section 6.2.6.1
Forwarding an XDM Resource	forward	section 6.1.1.3.2, section 6.2.6.2
Forwarding an XDM Resource	forward-remote	section 6.1.1.3.2, section 6.2.6.2
Forward Delivery Report	forward-delivery-report	section 6.2.6.2.5
Accepting the XDM Resource received in the Forward XDCP Request	forward-accept	section 6.1.1.3.3
Rejecting the XDM Resource received in the Forward XDCP Request	forward-reject	section 6.1.1.3.3
Subscribing to changes in XDM Resources	subscribe	section 6.1.1.3.4, section 6.6.3
XDM Differential Read	diff-read	section 6.1.1.3.5
XDM Differential Write	diff-write	section 6.1.1.3.6
Restoring of XDM Resource	restore	section 6.1.1.3.7

Appendix G. "Reactive Authorization of XDM Requests using Request History Information Documents" (Informative)

This appendix contains an implementation example of how a UE can use an embedded XDMC to implement "reactive authorization" of a requesting User's Access Permission to the User's XDM Documents by combining some XDMC procedures that the XDM enabler provides. The following sequences show a simplified view of the implementation in the UE.

Activation of the "Reactive Authorizations"

1. The UE prompts the User if she/he wants to active "Reactive Authorizations" for access to information that is maintained in one or more Application Usages.
2. If "Yes" the next steps are executed by the UE if "No" the sequence stops here.
3. The UE orders the XDMC to update the XDM Preferences Document for each Application Usages in such away that the Request History Information Documents for each Application Usage are updated with information about unsuccessful XDM Requests using procedures in section 6.1.1.2 or in section 6.1.1.3.6.
4. The UE orders the XDMC to subscribe for changes to the Request History Information Documents. The XDMC can do this via SIP or via XDCP depending on the capability of the XDMC and the network. XDCP requires an embedded PUSH Client and HTTP access to the XDM Aggregation Proxy in the UE and SIP a SIP access via a SIP network to the Subscription Proxy using procedures described in section 6.1.2.1 or section 6.1.2.3.

Notification processing

1. The XDMC receives the first notification via the SIP Access or via the Push Client using procedures described in section 6.1.2.2 or in [PUSH_ERELD-V2.2].
2. The UE receives the notification from the XDMC.
3. The UE orders the XDMC to use procedures described in section 6.1.1.2.3 to fetch the Request History Information Documents indicated in the notification. The UE checks if the received documents contains old unauthorized XDM requests. If that is the case, the UE prompts the User with the list of unauthorized old XDM requests and asks what to do with them .The user is given 3 choices by the UE per request, "keep the request and take decision later", "authorize the requesting user" or "block new requests to pop up again" and continues in step 9.
4. The UE receives a new notification from the XDMC. The UE checks if it contains a new unauthenticated XDM Request and if that is the case, the UE prompts as in step 5 but with only one XDM request.
5. The UE receives the selected choices per XDM Request and does one the following:
   - "keep the request and take decision later": The UE does not have to do anything as the request remain in Request History Information Document and will be shown next time the list of old unauthorized requests are presented.
   - "authorize the requesting user": The UE orders the XDMC to update the Access Permissions Document related to the Application Usage, by using procedure described on section 6.1.1.2.4 to grant the User access to the requested XDM Document. The UE also order the XDMC to use procedures described in section 6.1.1.2.5 to delete the part in the Request History Document that contained request information about the now authorized user. This is done because the UE needs to make sure that the user is not prompted again with old unauthorized requests from e.g. other UEs that the user might have.
   - "block new requests to pop up again from this user": The UE orders the XDMC to use procedures described in section 6.1.1.2.4 or section 6.1.1.3.6 to update the XDM Preferences Document in such away that requests from this user are no longer recorded. The UE also order the XDMC to use procedures described in section 6.1.1.2.5 to delete the part in the Request History Document that contains request information about the blocked user. XDM requests from this user will no longer be recorded for this Application Usage and can therefore not pop up again.

Alternative procedures:

1. In step 4 in “Activation of the "Reactive Authorizations"” instead of subscribing for notifications, the UE can execute step 3 and 5 in “Notification processing” at regular intervals. This method only requires an HTTP Access to the Aggregation Proxy. The disadvantage with this method is that the user will not be prompted in real time,

Appendix H. " Access Permissions Change Notifications " (Informative)

This appendix describes how the XDM Version 2.1 requirement ACP-29 as described in [XDM_RD] has been solved in this specification. A UE that needs to implement a solution for the requirement is recommended to use the guidelines given in this appendix.

According to the requirement, it shall be possible to let a user be notified when any Access Permissions to an XDM Resource changed .The User must also be able to obtain information about which Access Permissions it has to a particular XDM Resource (see [XDM_RD] requirement ACP-025).

The solution for these two requirements in the XDM Core is handled through:

- a new Application Usage "Access Permissions List";
- some new procedures to the XDMS section 6.2.4;
- a new <actions> child element to section 5.6; and
- to reuse already existing XDMC and XDM Agent procedures in section 6.1.

ACP-025 is solved by giving the Admin Principal the possibility to define a permission rule that grant a user limited access to the Primary Principal's Access Permissions Document. The limitation is that a user will be able to retrieve an Access Permissions Document that is filtered to contain only the requesting Principal's permissions. The <actions> child element used is the <allow-retrieve-own-data> element as described in section 5.6. When the XDMS is executing the procedures described in section 6.2.5.2 for a retrieve request and the requested XDM Document is the Access Permissions Document, the XDMS will check for a rule with the <allow-retrieve-own-data> element and check if the <conditions> element is matching the requesting Principal. This is done by checking the <identity> and <external-list> elements. If matching occurs, the XDMS will remove all information in the Access Permissions document that does not relate to the requesting user before sending the Access Permissions Document to her. If the user is not granted access to the Access Permissions Document, the XDMS will reject the operation with an HTTP "403" Forbidden" response.

ACP-29 is solved by adding the procedure in section 6.2.4.4 and the Application Usage Access Permissions List see [XDM_List]. This Application Usage maintains a list of references to Access Permissions Documents for User Directories which the Primary Principal has been granted access to. I.e. it is a resource list for URIs to Access Permissions Documents. By subscribing to changes to the Access Permissions List Documents, a UE can be informed about changes in this list and use this information to prompt the user with e.g. "User A has granted you access to personal information". Applying the procedures in section 6.2.4.4 the XDMS will act as an XDM Agent and update a user's Access Permissions List based on two conditions; the user has been given Access Permissions to some content in another user's User Directory and; the user has Access Permissions to the Access Permissions Document of this User Directory.

To describe how the functions can be used in UE implementation, an example is shown below:

Assume that the UE needs to implement a feature "Give a contact access to my personal data and inform the contact about it".

1. The user reads about the feature in the manual in his device and activates the feature by requesting the UE to start it.
2. The UE orders the embedded XDMC to fetch the XDM Directory Documents using the procedures in section 6.1.1.2.3.
3. The UE renders some information showing what personal information the user has in stored in his User Directories based on which Application Usages and existing XDM Document URIs are returned from the XDMC.
4. The User selects which type personal information that it wants to give a contact access to and informs the UE.
5. The UE orders the XDMC to fetch the Access Permissions Documents related to this personal information and checks if the user is allowed to administrate its own Access Permissions Document for this type of personal information. If not the use case end here.

6. The UE renders a set of Access Permissions types as described in section 5.6 (e.g. allowed to retrieve, allowed to write etc) and the list of contacts that has given these types already and a possibility to select a new contact from an address book or an input field for a contact identity, or a choice to modify an existing contact's Access Permissions..
7. The user selects to grant a new contact access to the information and informs the UE.
8. The UE updates the Access Permissions Document using the modification procedures in section 6.1.1.2 or in section 6.1.1.3.6. The Access Permissions Document is updated in two places; first the rule with the <allow-retrieve-own-data> element is updated with the new contact identity to give this contact access to read what and how personal information can be accessed and second; the user is added to the rule that grants this contact retrieve access to the personal information.
9. When the second rule is update in the XDMS, it triggers the procedures in section 6.2.4.4 and the XDMS acts an XDM Agent and updates the Access Permissions List belonging to the granted contact before returning a result back to the XDMC.
10. When the Access Permissions List Document is updated and an UE, via its XDMC, has subscribed to changes in this XDM Document, the UE will be notified with the link to the contact's Access Permissions Document. This link includes the XUI value and the UE can check in its address book to map this identity to a known user name and render this information together with the identity and a text "You can access this contact's personal information." The UE can also fetch the Access Permissions Document to give more details about what the user is allowed to do.

Alternative procedures:

1. In step 5: The user selects a contact identity from the list of already granted contacts and informs the UE that its wants to inform this contact again about his Access Permissions. The UE uses the procedures in section 6.1.1.2.4 and updates the Access Permissions List directly. This will trigger a notification to the XDMC if it has subscribed to the changes to the Access Permissions List and the contact will be informed be UE via the subscribing XDMC.
2. In step 9: The XDMC has not subscribed to the changes to the Access Permissions List Document. The UE fetches instead the Access Permissions List Document at regular intervals using procedures in section 6.1.1.2.3 and informs the user about which contact personal information can be accessed.
3. In step 9: The user does not have any UE connected to the network. The UE will next time when it starts the subscription to changes again get information about the status of the Access Permissions List Document. The UE uses the procedure in section 6.1.1.3.5 to fetch what has changed since last connection with the network and render this information to the user.
4. In step 9: The contact does not want to have any Access Permissions updates from a particular user. The UE orders the XDMC to update the Access Permissions Document for the Access Permissions List Document to exclude the annoying user from the rule that grant any user access to the Access Permissions List.
5. In step 7: The user does not want to inform the contact about its new Access Permissions. The UE orders the XDMC not to add this identity to the first rule in step 7. This will prevent the XDMS from updating the Access Permissions List Document for this contact.
6. In step 7: The user want to explain more about why he wants to give the contact access to his personal document. The UE offer the user a suitable communication application (e.g. a voice or a messaging application) and sets up a communication session from the UE to the contact's communication device using the provided user address.

Appendix I. “Filter ABNF" (Normative)

This appendix describes the ABNF that SHALL be used in place of the ABNF of [RFC4661] section 5 unless an Application Usage specifies its own ABNF.

```
selection = root elem-reference  *(("/" elem-reference) / ("[" expression "]")

root = "/"

elem-reference = element / "*" / ("/" element)

expression = "[" (elem-expr / attr-expr) *(oper (elem-expr / attr-expr)) "]"

elem-expr = (elem-path / ".") compar value

elem-path = (element / "*") *("/" (element / ("/" element)))

attr-expr = [elem-path "/"] attribute compar value

oper = "and"

compar = "=" / "<" / ">" / "!="

element = [ns] string

attribute = "@" [ns] string

ns = string ":"

string = <any sequence of data supported by XML in names of XML elements, attributes,
or prefixes of namespaces>

value = <any sequence of data supported by XML as a value of the XML element or
attribute>
```