XML Document Management (XDM) Specification

Approved Version 1.0.1 – 28 Nov 2006

Open Mobile Alliance

OMA-TS-XDM_Core-V1_0_1-20061128-A

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

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

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

Contents

Figures

1. Scope

This document specifies common protocols, data access conventions, common data application usages and two 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

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

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

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

[RFC2616] “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding, June 1999, URL:http://www.ietf.org/rfc/rfc2616.txt

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

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

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

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

[XDM RD] “XML Document Management Requirements”, Version 1,0, Open Mobile Alliance™, OMA-RD-XDM-V1_0, URL:http://ww.openmobilealliance.org/

[XSD_COMMONPOL] “XDM – Common Policy”, Candidate Version 1.0.1, Open Mobile Alliance™, OMA-SUP-XSD_xdm_commonPolicy-V1_0_1, URL http://www.openmobilealliance.org/

[XSD_XCAPDIR] “XDM – XCAP Directory”, Candidate Version 1.0.1, Open Mobile Alliance™, OMA-SUP-XSD_xdm_xcapDirectory-V1_0_1, URL: http://www.openmobilealliance.org/

[XSD_XCAPERR] “XDM – XCAP Error”, Candidate Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_xcapError-V1_0, URL: http://www.openmobilealliance.org/

[COMMONPOL] “A Document Format for Expressing Privacy Preferences”, H. Schulzrinne, J. Morris, H. Tschofenig, J. Cuellar, J. Polk, J. Rosenberg, Aug, 2006,
URL: http://www.ietf.org/internet-drafts/draft-ietf-geopriv-common-policy-11.txt

Note: IETF Draft Work in progress.

[XCAP] “The Extensible Markup Language (XML) Configuration Access protocol (XCAP)”, J. Rosenberg, October 13, 2006, URL: http://www.ietf.org/internet-drafts/draft-ietf-simple-xcap-12.txt
Note: IETF Draft Work in progress

[XCAP_Config] “Extensions to the Session Initiation Protocol (SIP) User Agent Profile Delivery Change Notification Event Package for the Extensible Markup Language Configuration Access Protocol (XCAP)”, D. Petrie, March 20, 2006.
URL: http://www.ietf.org/internet-drafts/draft-ietf-sip-xcap-config-00.txt
(Note: this is a temporary address for this draft)
Note: IETF Draft Work in progress

[XCAP_Diff] “An Extensible Markup Language (XML) Document Format for Indicating Changes in XML Configuration Access Protocol (XCAP) Resources”, J. Rosenberg, October 17, 2006, URL: http://www.ietf.org/internet-drafts/draft-ietf-simple-xcap-diff-04.txt
Note: IETF Draft Work in progress

[SIP_UA_Prof] “A Framework for Session Initiation Protocol User Agent Profile Delivery”, D. Petrie October 3 2006. URL: http://www.ietf.org/internet-drafts/draft-ietf-sipping-config-framework-09.txt
Note: Work in progress

[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) (Release 6)”, URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.222/

[XDM_MO] “OMA Management Object for XML Document Management”, Version 1.0.1, Open Mobile Alliance™, OMA-TS-XDM_MO-V1_0_1, URL:http://www.openmobilealliance.org/.

[OMA-DM-v1-1-2] OMA Device Management, V1.1.2 (based on SyncML DM), OMA-DM-V1_1_2, Open Mobile Alliance™, URL:http://www.openmobilealliance.com/

[OMA-DM-v1-2] OMA Device Management, V1.2 (based on SyncML DM), OMA-DM-V1_2, Open Mobile Alliance™, URL:http://www.openmobilealliance.com/

[Provisioning Content] OMA – Provisioning Content V1.1, OMA-DM_ProvCont-V1_2_0, Open Mobile Alliance™,

URL:http://www.openmobilealliance.com/

[3GPP TS 24.109] 3GPP TS 24.109 “Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details (Release 6)”

URL: http://www.3gpp.org/ftp/Specs/archive/24_series/24.109/

[3GPP TS 23.228] 3GPP TS 23.228 “IP Multimedia Subsystem (IMS); Stage 2 (Release 6)”

URL: http://www.3gpp.org/ftp/Specs/archive/23_series/23.228/

[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 (Release 6)”

URL: http://www.3gpp.org/ftp/Specs/archive/24_series/24.229/

[3GPP2 X.S0013-004-A] 3GPP2 X.S0013-004-A “All-IP Core Network Multimedia Domain: IP Multimedia Call Control Protocol Based on SIP and SDP, Stage 3”, Revision A, Version 1.0, 3GPP2, 2005
URL: http://www.3gpp2.org/Public_html/specs/

[3GPP2 X.S0013-002-A] 3GPP2 X.S0013-002-A “All-IP Core Network Multimedia Domain: IP Multimedia Subsystem - Stage 2”, Revision A, Version 1.0, 3GPP2, 2005

URL: http://www.3gpp2.org/Public_html/specs/

[3GPP TS 33.141] 3GPP TS 33.141 “Presence service; Security"; (Release 6)”. URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.141/

[3GPP2 X.P0027-002-0] 3GPP2 X.P0027-002-0 “Presence Security”, Revision 0, Version 1.0, 3GPP2, 2005

URL: http://www.3gpp2.org/Public_html/specs/

Note: Work in progress, estimated availability January 2006

2.2 Informative References

[XDMAD] “XML Document Management Architecture”, Version 1.0.1, Open Mobile Alliance™. OMA-AD-XDM-V1_0_1, URL:http://www.openmobilealliance.org/.

[Shared_XDM] “Shared XDM Specification”, Version 1.0.1, Open Mobile Alliance™, OMA-TS-XDM_Shared-V1_0_1, URL:http://www.openmobilealliance.org/.

[PoC_XDM] “PoC XDM Specification”, Version 1.0.1, Open Mobile Alliance™, OMA-TS-POC_XDM-V1_0_1, URL:http://www.openmobilealliance.org/.

[RLS_XDM] “Resource List Service (RLS) XDM Specification”, Version 1.0.1, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE_RLS_XDM-V1_0_1, URL:http://www.openmobilealliance.org/.

[Presence_XDM] “Presence XDM Specification”, Version 1.0.1, Open Mobile Alliance™, OMA-TS-

Presence_SIMPLE_XDM-V1_0_1, URL:http://www.openmobilealliance.org/.

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

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

Application Unique ID (AUID)	A unique identifier that differentiates XCAP resources accessed by one application from XCAP resources accessed by another. [Source: XCAP]
Document Selector	A sequence of path segments, with each segment being separated by a "/", that identify the XML document within an XCAP root that is being selected. (Source: [XCAP])
Document URI	The HTTP URI containing the XCAP root and document selector, resulting in the selection of a specific document. (Source: [XCAP])
Global document	A document placed under the XCAP 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: [XCAP])
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 a document. (Source: [XCAP])
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: [XCAP])
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: [XCAP])
Primary Principal	The principal who has full access rights (e.g., read, write, delete) for a given document, including the right to delegate some of these rights to other principals. (Source: [XDM RD])
Reverse Proxy	A reverse proxy is 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])
XCAPApplication Usage	Detailed information on the interaction of an application with an XCAP server. (Source: [XCAP])
XCAP Client	An HTTP client that understands how to follow the naming and validation constraints defined in this specification. (Source: [XCAP])
XCAP Root	A context that includes all of the documents across all application usages and users that are managed by a server. [Source: XCAP]
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: XCAP]
XCAP Server	An HTTP server that understands how to follow the naming and validation constraints defined in this specification. (Source: [XCAP])
XCAP User Identifier (XUI)	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: XCAP]

3.3 Abbreviations

AS	Application Server
AUID	Application Unique ID

GAA	Generic Authentication Architecture
HTTP	Hyper Text Transfer Protocol
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
MMD	MultiMedia Domain
OMA	Open Mobile Alliance
OMNA	OMA Naming Authority
TLS	Transport Layer Security
UE	User Equipment
URI	Uniform Resource Identifier
XCAP	XML Configuration Access Protocol
XDM	XML Document Management
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 XML documents and stored in various document repositories in the network where such 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 XML documents by authorized principals. This specification reuses the IETF XML Configuration Access Protocol (XCAP).

XCAP defines:

- A convention for describing elements and attributes of an XML document as a HTTP resource, i.e., accessible via an HTTP URI
- A technique for using HTTP GET, PUT and DELETE methods for various document manipulation operations (e.g., retrieving/adding/deleting elements/attributes, etc.)
- The concept and structure of an XCAP Application Usage by which service or enabler specific documents can be described
- A default authorization policy for accessing and manipulating documents

This specification also defines a technique by which changes to such XML documents can be conveyed to an XCAP Client. This reuses an IETF-defined SIP event package by which an XDM Client subscribes to changes to all documents that it owns.

Common, reusable as well as enabler-specific document formats and associated XCAP application usages are described in separate specifications (e.g., [Shared_XDM] [PoC_XDM] [Presence_XDM] and [RLS_XDM]) that make use of the XCAP protocol specified here for their document management.

5. Description of Functional Elements

5.1 XDM Client

The XDM Client SHALL support the XDM Client procedures described in section 6.1, and the XCAP application usages described in Section 6.7.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the XDM Client MAY be implemented in a UE or an AS as defined in [3GPP TS 23.228] and [3GPP2 X.S0013-002-A] respectively.

5.2 Aggregation Proxy

The Aggregation Proxy is the contact point for the XDM Client implemented in an UE to access XML documents stored in any XDMS.

The Aggregation Proxy SHALL act as an HTTP Proxy defined in [RFC2616] with the following clarifications. The Aggregation Proxy:

1. SHALL be configured as an HTTP reverse proxy (see [RFC3040]);
2. SHALL support authenticating the XDM Client; in case the GAA is used according to [3GPP TS 33.222], the mutual authentication SHALL be supported; or SHALL assert the XDM Client identity by inserting the X-XCAP-Asserted-Identity extension header to the HTTP requests after a successful HTTP Digest Authentication as defined in Section 6.3.2, in case the GAA is not used.
3. SHALL forward the XCAP requests to the corresponding XDM Server, and forward the response back to the XDM Client;
4. SHALL protect the XCAP traffic by enabling TLS transport security mechanism. The TLS resumption procedure SHALL be used as specified in [RFC2818].

When realized with 3GPP IMS or 3GPP2 MMD networks, the Aggregation Proxy SHALL act as an Authentication Proxy defined in [3GPP TS 33.222] with the following clarifications. The Aggregation Proxy: SHALL check whether an XDM Client identity has been inserted in X-3GPP-Intended-Identity header of HTTP request.

- If the X-3GPP-Intended-Identity is included , the Aggregation Proxy SHALL check the value in the header is allowed to be used by the authenticated identity.
- If the X-3GPP-Intended-Identity is not included, the Aggregation Proxy SHALL insert the authenticated identity in the X-3GPP-Asserted-Identity header of the HTTP request.

6. Description of procedures

6.1 Procedures at the XDM Client

An XDM Client is an entity that accesses a XCAP resource in an XML Document Management Server (XDMS). Such XCAP resources correspond to elements and attributes of an XML document. An XCAP resource is identified via an HTTP URI following the conventions for constructing URIs in [XCAP].

6.1.1 Document Management

6.1.1.1 XDM URI Construction

An HTTP URI represents each element and attribute of an XML document in a XDM respository. The rules for constructing such URIs SHALL follow the rules described in [XCAP] Section 6 with the clarifications given in this sub-clause.

Therefore, for example, a generic XCAP URI would be of the form [XCAP Root URI]/[AUID]/users/[XUI]/… (See Appendix B for examples.)

Note: In the case where the XDMC resides within an UE the DNS lookup of the hostname of [XCAP Root URI] shall resolve to the address of the Aggregation Proxy.

The path segment corresponding to the XUI SHALL either be a Public SIP URI of form sip: user@domain or TEL URI, e.g., tel:+1720-555-1212, identifying the document owner. If the XDMC resides within an AS (within a trusted environment) it SHALL have the possibility to address the XDMS directly without going through the Aggregation Proxy.

If a user has multiple Public URIs available, each single Public URI constitutes an independent and unrelated XUI. For example, if a user has two Public SIP URIs 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. Any relationship between Public URIs of a user, allowing e.g. interchangeable XUI usage, is out of the scope of this specification.

If a user has both a Public 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 system. Both the translation and the interchangeable usage are out of the scope of this specification.

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

- The Node Selector Separator SHALL convey the meaning as defined in [XCAP].
- The Node Selector Separator SHALL appear only once, as a URI separator (i.e. in the form of “/~~/”).
- The Node Selector Separator SHOULD NOT be percent-encoded according to the procedures defined in [RFC 3986].

Note: 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

An XDM Client manipulates an XML document by invoking certain HTTP operations (defined in sub-sections below) on the XDM resource identified in the Request-URI of the HTTP header.

The client SHALL construct the Request-URI based on its knowledge of the application usage governing that XML document.

An XDM client MAY implement the conditional operations of [XCAP] section 7.10.

An XDM client MAY support HTTP compression using content encoding. If the XDM client 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 XML document SHALL follow the procedures described in [XCAP] Section 7.1.

6.1.1.2.2 Delete a Document

Deleting an XML document SHALL follow the procedures described in [XCAP] Section 7.2.

6.1.1.2.3 Retrieve a Document

Retrieving an XML document SHALL follow the procedures described in [XCAP] Section 7.3.

6.1.1.2.4 Create or Replace an Element

Creating or replacing an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.4.

6.1.1.2.5 Delete an Element

Deleting an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.5.

6.1.1.2.6 Retrieve an Element

Retrieving an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.6.

Note: When an XML-fragment is received as a result of a retrieve operation, the XML-fragment does not always contain all needed namespace bindings. XDM clients 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 attribute of an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.7.

6.1.1.2.8 Delete an Attribute

Deleting an attribute of an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.8.

6.1.1.2.9 Retrieve an Attribute

Retrieving an attribute of an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.9.

Note: When an XML-fragment is received as a result of a retrieve operation, the XML-fragment does not always contain all needed namespace bindings. XDM clients 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.

6.1.1.2.10 Fetch Namespace Bindings

Fetching namespace bindings of an element in a XML document SHALL follow the procedures described in [XCAP] Section 7.10.

6.1.2 Subscribing to changes in the XML documents

6.1.2.1 Initial subscription

If the XDM Client subscribes to changes in XML documents, then it SHALL be done by sending a SUBSCRIBE request according to [RFC3265] and [SIP_UA_Prof]/[XCAP_Config] with the clarifications given in this sub-clause.

When the XDM Client resides in an Application Server:

1. SHALL set the Request-URI to the public SIP URI identifying the document owner, or to the SIP URI identifying the service instance (e.g. PoC Group URI, or Presence List URI);
2. SHALL include value “application” in the “profile-type” Event header parameter. It is beyond the scope of the present specification the value set for the “vendor”, “model” and “version” event header parameters that their use is mandated in [SIP_UA_Prof].
3. SHALL include the AUID to be watched in the “auid” Event header parameter even in case the “document” event header parameter is present. In case the “document” header is not being used it specifies that a subscription will be placed to all the documents under the AUID owned by the user identified in the Request-URI.
4. SHALL include the XCAP URI of the document to be watched in the “document” Event header parameter in case a specific document is to be watched.
5. In case the service instance SIP URI is set as Request-URI of the SIP SUBSCRIBE request, then the “document” Event header parameter SHALL be set to specify the relevant document or the relevant element inside the document stored in the “global” tree for this service instance SIP URI.

   Note: For example, if the Request-URI SIP URI identifying the service instance is “sip:my_friends@example.com” stored in PoC XDMS, the document parameter has to be set to “document= global/index/~~/group/list-service[@uri=sip:my_friends@example.com]”.
6. SHALL include an Accept header to indicate acceptable content-type for notifications. The Accept header
   a. MAY include the value “application/xcap-diff+xml” to indicate support for partial XML updates described in [XCAP_Diff];
   b. MAY include the value “message/external-body” to indicate support for content indirection described in [RFC4483];
7. SHALL send the SUBSCRIBE request towards the SIP/IP Core according to the procedures of the SIP/IP core.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS an AS acting as the XDM Client SHALL use 3GPP/3GPP2 IMS requirements, mechanisms and procedures as defined in chapter 5.7.3 [3GPP TS 24.229] / [3GPP2 X.S0013-004-A] with the clarifications given in the respective sub clauses.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, and the XDM Client resides in an Application Server (e.g. PoC Server) the mechanisms of the “Application Server acting as originating User Agent” SHALL be applied as defined in [3GPP TS 24.229] / [3GPP2 X.S0013-004-A] section 5.7.3 and setting its public SIP URI in the “P-Asserted-Identity” header.

When the XDM Client resides in the UE

1. SHALL set the Request-URI to the public SIP URI identifying the owner of the document(s) that is subscribing to;
2. SHALL include value “application” in the “profile-type” Event header parameter. It is beyond the scope of the present specification the value set for the “vendor”, “model” and “version” event header parameters, that their use is mandated in [SIP_UA_Prof].
3. SHALL include the AUID to be watched in the “auid” Event header parameter even in case the “document” event header parameter is present. In case the “document” header is not being used it specifies that a subscription will be placed to all the documents under the AUID owned by the user identified in the Request-URI.
4. In case a specific document is to be watched, then the “document” Event header parameter SHALL be set to the XCAP URI of the relevant document

   Note: The mechanism used by the XDM Client to retrieve the SIP URI of the document owner and the XCAP URI of the document to be watched is out of scope of the present specification.
5. SHALL include an Accept header to indicate acceptable content-type for notifications. The Accept header

   a. MAY include the value “application/xcap-diff+xml” to indicate support for partial XML updates described in [XCAP_Diff];

   b. MAY include the value “message/external-body” to indicate support for content indirection described in [RFC4483];

6. SHALL send the SUBSCRIBE request towards the SIP/IP Core according to the procedures of the SIP/IP core.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, a UE acting as the XDM Client SHALL use 3GPP/3GPP2 IMS requirements, mechanisms and procedures as defined in chapter 5.1 in [3GPP TS 24.229] / [3GPP2 X.S0013-004-A].

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, and the XDM Client resides in an Application Server (e.g. PoC Server) the mechanisms of the “Application Server acting as originating User Agent” SHALL be applied as defined in [3GPP TS 24.229] / [3GPP2 X. S0013-004-A] section 5.7.3 and setting its public SIP URI in the “P-Asserted-Identity” header.

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

Note: The XDM Client is not able to subscribe for changes in multiple documents stored under different AUIDs in a single subscription. This functionality has been postponed for a future release.

6.1.2.2 NOTIFY processing

Upon receiving an incoming NOTIFY request that is part of the same dialog as the previously sent SUBSCRIBE request the XDM Client

1. SHALL handle the request according to [RFC3265], [SIP_UA_Prof], and the procedures of the SIP/IP core;
2. SHOULD update the stored XML document based on the information in the NOTIFY request.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, the XDM Client SHALL use 3GPP/3GPP2 IMS requirements, mechanisms and procedures as defined in [3GPP TS 24.229] / [3GPP2 X.S0013-004-A] with the clarifications given in this sub-clause.

6.2 Procedures at the XDM Server

A XDM Server (XDMS) is a HTTP origin server that manipulates XML resources according to the conventions described in [XCAP].

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the XDM Server SHALL be implemented in an AS as defined in [3GPP TS 23.228] and [3GPP2 X.S0013-002-A] respectively.

6.2.1 Document Management

An XDM Server receiving an HTTP request targeted at an XCAP resource identified by the HTTP Request-URI follows the following procedures based on the method requested.

An XDM server SHALL conform to [XCAP] section 8.5 for the management of Etags.

An XDM server SHALL implement the conditional operations of [XCAP] section 7.11.

If the XDM Server implements parallel processing of requests, it SHALL ensure the integrity of the resulting document.

6.2.1.1 POST handling

HTTP POST requests targeted at an XDM resource SHALL be rejected with an HTTP 405 “Method not allowed” response as described in [XCAP] Section 8.1.

6.2.1.2 PUT handling

HTTP PUT requests targeted at an XDM resource SHALL be processed as described in [XCAP] Section 8.2.

6.2.1.3 GET handling

HTTP GET requests targeted at an XDM resource SHALL be processed as described in [XCAP] Section 8.3.

6.2.1.4 DELETE handling

HTTP DELETE requests targeted at an XDM resource SHALL be processed as described in [XCAP] Section 8.4.

6.2.2 Subscriptions to changes in the XML documents

6.2.2.1 Initial subscription

Upon receiving a SUBSCRIBE request for the “ua-profile” event defined in [SIP_UA_Prof] the XDM Server performs the following steps:

1. SHALL return the SIP “489 Bad Event” error response, if the “ua-profile” event is not supported. Otherwise perform the following steps.
2. SHALL use the Request-URI
    a. as an XUI identifying the owner of the document in case the “document” event header parameter is defining a document in the “users” tree or is not set;
    b. as a service instance SIP URI (e.g. PoC group) in case the “document” event header parameter is defining a document in the “global” tree;
3. SHALL perform the necessary authorization checks on the originator. When the SIP/IP Core corresponds to 3GPP/3GPP2 IMS the XDM Server SHALL use the "P-Asserted-Identity" as defined in [3GPP TS 24.229] / [3GPP2 X.S0013-004-A] to ensure that this particular XDM Client is authorized to track the document changes. If the authorization check fails, the XDM Server SHALL return the SIP "403 Forbidden" error response.
    a. By default, the initial creator of the document in the "users" tree is the primary principal of that document and SHALL be authorized to subscribe to the “ua-profile” event package as described in Section 6.4.2.

       Other principals (e.g. XDMCs residing in the UE and Application Servers) identified by their "P-Asserted-Identity" headers MAY be authorised to subscribe in the document in the “global” tree based on the service provider policy defined in each XDMS.
    b. This policy is defined in the application-specific XDM TSs.
4. SHALL create a subscription to changes of XML data identified by Event header parameters as described in [SIP_UA_Prof];
5. SHALL send a SIP “200 OK” in accordance with [RFC3265], [SIP_UA_Prof], and the procedures of the SIP/IP core.
6. SHALL generate and send an initial NOTIFY request as specified in sub-clause 6.2.2.2 “Generating a NOTIFY request”.

When a change in the subscribed document occurs, the XDM Server SHOULD generate and send a NOTIFY request as specified in sub-clause 6.2.2.2 “Generating a NOTIFY request”.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, the XDM Server SHALL use 3GPP/3GPP2 IMS requirements, mechanisms and procedures as defined in [3GPP TS 24.229] / [3GPP2 X.S0013-004-A] with the clarifications given in this sub-clause.

6.2.2.2 Generating a NOTIFY request

If the “ua-profile” event is supported the XDM Server SHALL generate a NOTIFY request as described in the [RFC3265] and [SIP_UA_Prof] with the clarifications given in this sub-clause.

The XDM Server

1. SHALL check content-types accepted by the XDM Client as indicated in the SUBSCRIBE request (see sub-clause 6.1.2.1);

    a. if both indirect and directly supplied content are acceptable, the XDM Server MAY include either alternative;

    b. if acceptable for the XDM Client, the XDM Server SHALL include an “application/xcap-diff+xml” body as defined in [XCAP_Diff];

    c. if acceptable for the XDM Client, the XDM Server SHALL include a “message/external-body” body as defined in [SIP_UA_Prof] and [RFC4483];

2. SHALL send the NOTIFY request towards the SIP/IP Core according to the procedures of the SIP/IP core.

3. When the subscription is placed to all the documents under an AUID then the notification SHALL indicate all the document(s) that have changed

The responses to the NOTIFY request SHALL be handled in accordance with [RFC3265], [SIP_UA_Prof], and the procedures of the SIP/IP core.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, the XDM Server SHALL use 3GPP/3GPP2 IMS requirements, mechanisms and procedures as defined in [3GPP TS 24.229] / [3GPP2 X.S0013-004-A] with the clarifications given in this sub-clause.

6.3 Procedures at the Aggregation Proxy

The Aggregation Proxy performs security procedures, as well as the request forwarding procedure for HTTP traffic. The first function is covered in section 6.3.1 and 6.3.2, and the request forwarding procedure is covered in section 6.3.3.

6.3.1 Authentication

The authentication function SHALL be performed over XDM-3 reference point (see [XDMAD]). The initial HTTP request from XDM Client SHALL be interrogated by the Aggregation Proxy using the HTTP Digest mechanism as specified in [RFC2617].

The Aggregation Proxy SHALL fulfill the functions described in sub-clause 6.4.1.

6.3.2 XDM Client identity assertion

When the 3GPP GAA is not present the Aggregation Proxy SHALL perform the following:

1. Insert the X-XCAP-Asserted-Identity extension header to the HTTP requests after a successful HTTP Digest Authentication ;

2. Populate the X-XCAP-Asserted-Identity with the public SIP or TEL URI in quotation marks (“”) provided by the “username” field in the HTTP Digest Authorization header.

3. Ensure that only one instance of the X-XCAP-Asserted-Identity header exists in the HTTP Requests before forwarding it. In cases where there are multiple instances, the Aggregation Proxy SHALL remove all previous instances of this header and insert its own provided that the XDM Client authentication with the Aggregation Proxy was successful

When realized in 3GPP IMS and the GAA is present, the procedures described [3GPP TS 24.109] SHALL be followed with the following clarifications:

1. The Aggregation Proxy SHALL check whether an XDM Client identity has been inserted in X-3GPP-Intended-Identity header of HTTP request. If so, the Aggregation Proxy SHALL check the value in the header is equal to the authenticated identity.
2. If the X-3GPP-Intended-Identity is not included, the Aggregation Proxy SHALL insert authenticated identity in the X-3GPP-Asserted-Identity header of the HTTP request.

6.3.3 XCAP request forwarding

6.3.3.1 General

Upon receiving an XCAP request targeted to the Aggregation Proxy, the Aggregation Proxy:

1. SHALL act as an HTTP reverse proxy;
2. SHALL forward the XCAP request to the corresponding XDM Server based on the HTTP Request URI.

The response to the XCAP request SHALL be sent back to the originator.

6.3.3.2 XCAP Server Capabilities retrieval

Upon receiving an XCAP GET request for the “xcap-caps” AUID (described in section 6.7.1), the Aggregation Proxy:

1. SHALL act as an HTTP reverse proxy;
2. SHALL obtain XCAP Server Capabilities from all XDM Servers that serve the request originator. To perform this operation the Aggregation Proxy SHALL:
    a. forward the XCAP request to all XDM Servers that serve the request originator;
    b. if the target XDM Server responded with HTTP “200 OK” response, collect the <auid>, <extension> and <namespace> elements.
3. SHALL return the HTTP “200 OK” response with the “application/xcap-caps+xml” body including all received <auid>, <extension> and <namespace> elements.

Upon receiving of other HTTP request for an “xcap-caps” document, the Aggregation Proxy shall respond with an HTTP “405 Method Not Allowed” response.

6.3.3.3 XCAP Directory retrieval

Upon receiving an XCAP GET request for the “org.openmobilealliance.xcap-directory” AUID (described in section 6.7.2), the Aggregation Proxy:

1. SHALL act as an HTTP reverse proxy;
2. SHALL obtain the requested XCAP Directory from the corresponding XDM Servers that serve the request originator. To perform this operation the Aggregation Proxy SHALL:
    a. forward the XCAP request either to all XDM Servers that serve the request originator if the request is targeted at the directory document, or to the XDM Server serving the specific AUID if the request is targeted at a specific AUID as specified by the node selector;
    b. if the target XDM Server responded with HTTP “200 OK” response, collect the <folder> elements.
3. SHALL return the HTTP “200 OK” response either with the “application/vnd.oma.xcap-directory+xml” body that contains xcap-directory document including all received <folder> elements if the request was targeted at the 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 of other HTTP request for an “org.openmobilealliance.xcap-directory” document, the Aggregation Proxy shall respond with an HTTP “405 Method Not Allowed” response.

6.3.4 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 Security Procedures

6.4.1 Authentication

The XDM-3 reference point (see [XDMAD]) SHALL provide mutual authentication.

For a 3GPP/3GPP2 realisation, the XDM-3 corresponds to the Ut reference point. In this case the authentication between theXDM Client and the Aggregation Proxy SHALL be performed according to [3GPP TS 33.141] / [3GPP2 X.S0027-002-0].

If the Generic Authentication Architecture (GAA) as defined in [3GPP TS 33.222] is not used, the XDM Client and the Aggregation Proxy (see [XDMAD]) SHALL support the HTTP Digest mechanism for client authentication.

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

- The HTTP server (“401 Unauthorized”) SHALL be used;
- the “rspauth” parameter MAY be used to provide mutual authentication;
- the “username” parameter SHALL contain the XUI (i.e. the SIP or TEL URI) identifying the user (the public user identity);

NOTE: The “username” is a part of the Device Provisioning parameters (see Appendix C).The XDM client shall use the “username” received without any modification.

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

For a 3GPP/3GPP2 realization, the reference points between the Aggregation Proxy and any XDMS or an Application Server and any XDMS uses the security mechanisms defined in 3GPP/3GPP2 that are out of scope of this specification.

6.4.2 Integrity and Confidentiality protection

The XDM Client and the Aggregation Proxy SHALL support the TLS as specified in [RFC2246] with the following clarifications:

- The following cipher suite SHALL be supported:
  - TLS_RSA_WITH_3DES_EDE_CBC_SHA

  other cipher suites defined in [RFC2246] MAY be supported.

When the SIP/IP Core corresponds with 3GPP IMS, the XDM Client and the Aggregation Proxy SHALL support the TLS version and profile as specified in clause 5.3 of [3GPP TS 33.222].

6.4.3 Authorization

The XDMS SHALL check that the identity of the requesting XDMC has been granted access rights to perform the requested operations. Application usages MAY define their own policies for accessing different XCAP resources (e.g. global documents).

The XDMS SHALL use the information in the X-XCAP-Asserted-Identity header provided by the Aggregation Proxy to determine the identity of the XDM Client.

When realized in 3GPP IMS and the GAA is present, the identity of the requesting XDMC is obtained from the X-3GPP-Asserted-Identity or the X-3GPP-Intended-Identity.

By default, the initial creator of a document is its primary principal. The primary principal SHALL have permission to perform all operations defined in Sections 6.1.1 and 6.1.2. In this release, it will not be possible to change the primary principal. Additionally, it will not be possible to assign permissions to access or manipulate a document to anyone except for the primary principal or trusted applications.

Any application usage defining the use of any global documents SHALL specify the authorization policy associated with the use of such documents.

6.5 Error cases

If the Aggregation Proxy or XDM server receives an HTTP request targeted at an XCAP resource whose application usage is not recognized or understood, the Aggregation Proxy or XDM Server SHALL reject the request with an HTTP 404 (Not Found) response.

Additional validation constraints might be applied which may result in a HTTP 409 Response. An HTTP 409 error response SHALL include a document in the HTTP body that conforms to that defined in [XCAP] Section 9 and the extensions defined in Section 6.6.3.

For additional details of the handling of those, see [XCAP] Section 8.2.5 and Section 6.6.3.

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.

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

6.6 Common Extensions

6.6.1 Lists defined in Shared XDMS

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

The <external> element SHALL contain either an XCAP document URI pointing to a “resource-lists” document in the Shared XDMS or an XCAP node URI pointing to a <list> element within a “resource-lists” document in the Shared XDMS.

If the <external> element contains an XCAP node URI, then the node selector part SHALL be percent-encoded as defined by the procedures in [XCAP] section 6 before it is inserted into an XCAP document.

NOTE: There is an <external-list> condition element defined in section 6.6.2. It points to URI Lists in the Shared XDMS, against which the authorization rules are specified according to [COMMONPOL].

Application usages that utilize the <external> element will resolve it to a set of URIs according to the following procedures:

- If the <external> element contains a XCAP document URI, then it SHALL be resolved to all the URIs contained within the “resource-lists” document that is pointed to.
- If the <external> element contains a XCAP node URI, then it SHALL be resolved 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 which points to an XCAP document URI or XCAP node URI that has already been resolved SHALL be ignored.

6.6.2 Authorization Rules

Authorization rules (also called authorization policies) are based on the common policy framework described in [COMMONPOL], 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 [COMMONPOL];
- 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;

Every authorization policy SHALL support the extensions to [COMMONPOL] defined in this sub-clause.

Note 1: Individual enablers may also define extensions to [COMMONPOL] to meet application-specific needs. Such extensions must not change or cause to change the semantics of the common extensions defined in section 6.6.2.1 or the evaluation algorithm for combining permissions defined in section 6.6.2.3.

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

6.6.2.1 Structure

Every rule in an authorization policy document SHALL support the following extensions to [COMMONPOL]:

- the <external-list> condition element (as defined in section 6.6.2.2);
- the <anonymous-request> condition element (as defined in section 6.6.2.2);
- the <other-identity> condition element (as defined in section 6.6.2.2).

If present in any rule, the <external-list> element allows for matching those identities that are part of a URI List (as defined in section 6.6.2.2). If the <external-list> element is empty (i.e. there are no child elements), or if all the child elements resolve to URI Lists that are empty, then the corresponding rule does not match for any user.

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

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, an AS SHALL use the procedures as defined in chapter 5.7.1.4 in [3GPP TS 24.229] / [3GPP2 X.S0013-004-A] how to identify the source of the request anonymous.

Note: If the authorization policy document includes a rule having an <anonymous-request> condition element, an XDM client 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, matches all identities that are not referenced in any rule. It allows for specifying a default policy.

The <conditions> element of a rule SHALL contain no more than one of <identity>, <external-list>, <anonymous-request> or <other-identity>.

If the <external-list> element contains an XCAP node URI, then the node selector part SHALL be percent-encoded as defined by [XCAP] section 6 before it is inserted into an XCAP document.

6.6.2.2 XML Schema

The authorization policy document SHALL conform to the XML schema described in [COMMONPOL] Section 13 with the extension described in [XSD_COMMONPOL].

6.6.2.3 Combining Permissions

When evaluating any authorization policy document based on [COMMONPOL] together with the extensions described in section 6.6.2.1 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 <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 shall be used for the evaluation of the combined permission.
2. 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 [COMMONPOL].

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

6.6.3 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 [XCAP], and extended by OMA defined common extensions in order to handle violations of constraints defined by local policy appropriately.

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

6.6.3.1 Structure

The detailed error report provided in a 409 Conflict error response SHALL support the following extension to the <extension> element defined in the xcap-error namespace in [XCAP]:

- the <local-constraint-failure> error element (as defined in section 6.6.3.2).

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

The <local-constraint-failure> MAY contain the “phrase” attribute and MAY contain the following child elements:

a. zero or more <alt-value> elements with the mandatory “field” attribute, providing zero or more alternate values for the element or attribute indicated by the “field” attribute;

b. zero or more <description> elements with an optional “lang” attribute, providing zero or more descriptions documenting the local constraint failure, possibly in different languages.

Note: 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 [XCAP], SHALL be used for this, unless specified otherwise by the application usage.

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

6.6.3.2 XML Schema

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

6.7 Common Application Usage

6.7.1 XCAP Server Capabilities

Every XDM server SHALL support the Application Usage “xcap-caps”, which defines the capabilities of the server, as defined in [XCAP] Section 11.

The single document in the “global” tree corresponding to the “xcaps-caps” Application Usage SHALL be available to all principals as a part of the global URI tree.

6.7.2 XML Documents Directory

The XML Documents Directory application usage allows an XDM Client (corresponding to a given XUI) to fetch:

1. the list of all XCAP managed documents corresponding to that XUI across all XDMSes, or
2. the list of all documents for a given AUID corresponding to that XUI stored in an XDMS.

An XDMS SHALL support an application usage named “org.openmobilealliance.xcap-directory” and SHALL maintain one document in the “users” tree per XUI named “directory.xml”.

The structure of the “directory.xml” document is as follows: it is a well-formed and valid XML document encoded in UTF-8 that begins with the root element <xcap-directory>. It consists of a number of <folder> elements.

Each <folder> element SHALL have an attribute “auid”, whose value corresponds to an AUID that the XCAP server supports and for which there are documents in the “users” tree corresponding to.a given XUI.

Every <folder> element consists of a number of <entry> elements or an <error-code> element. Each <entry> element containing a number of attributes, which are:

1. uri: this attribute SHALL be the Document URI for a document corresponding to the “auid” attribute value in the parent <folder> element and for the given XUI.
2. etag: this attribute SHALL contain the server computed etag value of the current instance of the XML document identified by the “uri” attribute value. (This allows the XCAP client to determine whether the locally cached copy of a document is up-to-date.
3. last-modified: this attribute is OPTIONAL. When present, it SHALL contain the date and time the document identified as above was last modified. (This allows the XCAP client to determine if whether a document has changed recently or not.)
4. size: this attribute is OPTIONAL. When present, it SHALL contain the size, in octets, of the document as identified above. (This can help an XCAP client determine if it wants to upload the entire document or a fragment, as appropriate based on any resource limitation such as bandwidth.)

The <error-code> element shall contain the error message returned by an XDMS.

For a XCAP GET request targeted at the directory document belonging to a user, for example, URI http://[XCAP Root URI]/ org.openmobilealliance.xcap-directory /users/sip:joe@example.com/directory.xml, all XDMSes should return to the Aggregation Proxy a list of all XML documents associated with all supported AUIDs for the user identified by sip:joe@example.com.

The Aggregation Proxy SHALL aggregate responses from all XDMSs before sending the composite “directory.xml” back to the XDM Client. When an XDMS response is received with an error message, the Aggregation Proxy shall insert a <folder> element for the corresponding AUID and an <error-code> element with the error message included. When the Aggregation Proxy receives an HTTP 200 OK response with no XML content or no <folder> elements, it SHALL not include a <folder> element in the composite “directory.xml”.

For a XCAP GET request targeted 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/~~/xcap-directory/folder[@auid="org.openmobilealliance.poc-groups"], the XDMS serving the AUID should return to the Aggregation Proxy a <folder> element containing a list of all XML documents associated with the AUID for the user. The list in this example would be a list of all documents for PoC Group belonging to sip:joe@example.com. The content type SHALL be "application/xcap-el+xml".

The Aggregation Proxy SHALL forward the response from the serving XDMS and send it back to the XDM Client.

Note: The character escaping SHALL be applied in HTTP URI representation according to [XCAP] Section 6.3.

6.7.2.1 Application Unique ID

This specification defines the "org.openmobilealliance.xcap-directory" AUID.

6.7.2.2 MIME Type

The MIME type for this document is "application/vnd.oma.xcap-directory+xml"

6.7.2.3 Default Namespace

The default namespace SHALL be:

"urn:oma:xml:xdm:xcap-directory"

6.7.2.4 XML Schema

The XCAP directory document SHALL conform to the XML schema described in [XSD_XCAPDIR].

6.7.2.5 Additional Constraints

None.

6.7.2.6 Data Semantics

See section 6.7.2.

6.7.2.7 Naming Conventions

There is only one XCAP directory document per XUI in each XDMS. Therefore, the XDMS SHOULD assign the directory document the name "directory.xml".

To retrieve such a directory document, the XCAP Client SHALL always use this same name.

6.7.2.8 Data Interdependencies

For every 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 "directory.xml" document corresponding to that XUI.

NOTE: This does not imply that the server must actually store this "directory" document. All that is required is that the XDMS be able to serve an up-to-date version of such a document when requested.

The XDMS SHOULD NOT generate an etag value for the "directory" document.

NOTE: This implies that conditional operations are not supported against the "directory" document. The XCAP Client should always refresh any cached copy.

6.7.2.9 Authorization Policies

The “directory.xml” document is created and modified only by the XDMS. Thus, authorized principals are only allowed to retrieve this document.

The authorization policies for retrieving a “directory.xml” document SHALL conform to those described in [XDM_Spec] section 6.4.3.

6.8 Global Documents

[XCAP] specifies a global tree which is used to place documents applicable to a particular application usage but which are not specific to any particular user. An example of this is the "xcap-caps" document (see section 6.7.1) describing the application usages supported by an XDMS.

If used, each application usage describes how each global document is constructed and any associated authorization policy.

Appendix A. Static Conformance Requirements (normative)

The SCRs defined in the following tables include SCR for:

- Aggregation Proxy
- XDM Server
- XDM Client

Each SCR table identifies a list of supported features as:

Item: Identifier for a feature.

Function: Short description of the feature.

Reference: Section(s) of this specification with more details on the feature.

Status: Whether support for the feature is mandatory or optional. MUST use "M" for mandatory support and "O" for optional support in this column.

Requirement: This column identifies other features required by this feature. If no other features are required, this column is left empty.

This section describes the dependency grammar notation to be used in the Requirement column of the SCR and CCR tables using ABNF [RFC2234].

TerminalExpression = ScrReference / NOT TerminalExpression / TerminalExpression LogicalOperator TerminalExpression / "(" TerminalExpression ")"

ScrReference = ScrItem / ScrGroup

ScrItem = SpecScrName "–" GroupType "–" DeviceType "–" NumericId / SpecScrName "–" DeviceType "–" NumericId

ScrGroup = SpecScrName ":" FeatureType / SpecScrName "– " GroupType "–" DeviceType "–" FeatureType

SpecScrName = 1*Character;

GroupType = 1*Character;

DeviceType = "C" / "S"; C – client, S – server

NumericId = Number Number Number

LogicalOperator = "AND" / "OR"; AND has higher precedence than OR and OR is inclusive

FeatureType = "MCF" / "OCF" / "MSF" / "OSF"; See Section A.1.6

Character = %x41-5A ; A-Z

Number = %x30-39 ; 0-9

A.1 XDM Client

A.1.1 XDM Client implemented in a UE

Item	Function	Reference	Status	Requirement
XDM-XDMC-C-001	Support rules for constructing HTTP URIs	6.1.1.1	M	

Item	Function	Reference	Status	Requirement
XDM-XDMC-C-002	Support for XDM Operations	6.1.1.2	M	
XDM-XDMC-C-003	Initial Subscription using the SUBSCRIBE message	6.1.2.1	O	XDM-XDMC-C-004
XDM-XDMC-C-004	Processing Received NOTIFY Request	6.1.2.2	O	XDM-XDMC-C-003
XDM-XDMC-C-005	Support HTTP Digest authentication	6.4.1	M	
XDM-XDMC-C-006	Support HTTP over TLS using the supported cipher suite	6.4.1	M	
XDM-XDMC-C-007	Support other cipher suites defined in RFC2246	6.4.1	O	
XDM-XDMC-C-008	Support HTTP Compression	6.1.1.2	O	

A.1.2 XCAP Client implemented in an AS

Item	Function	Reference	Status	Requirement
XDM-XDMC-C-001	Support rules for constructing HTTP URIs	6.1.1.1	M	
XDM-XDMC-C-002	Support for XDM Operations	6.1.1.2	M	
XDM-XDMC-C-003	Initial Subscription using the SUBSCRIBE message	6.1.2.1	O	XDM-XDMC-C-004
XDM-XDMC-C-004	Processing Received NOTIFY Request	6.1.2.2	O	XDM-XDMC-C-003

Item	Function	Reference	Status	Requirement
XDM-XDMC-C-008	Support HTTP Compression	6.1.1.2	O	

A.2 XDM Server

Item	Function	Reference	Status	Requirement
XDM-XDMS-S-001	Support for XCAP	6.2.1	M	
XDM-XDMS-S-002	Support Initial Subscription when SUBSCRIBE message received	6.2.2.1	O	
XDM-XDMS-S-003	“Not Implemented” Error Handling or SUBSCRIBE request Handling	6.2.2.1	M	
XDM-XDMS-S-004	Generating a NOTIFY request	6.2.2.2	O	
XDM-XDMS-S-005	Support XDMC identity access authorization	6.4.3	M	
XDM-XDMS-S-006	Support Error Handling	6.5	M	
XDM-XDMS-S-007	Support Application Usage “xcap-caps”	6.7.1	M	
XDM-XDMS-S-008	Support Application Usage “xcap-directory”	6.7.2	M	

A.3 Aggregation Proxy

Item	Function	Reference	Status	Requirement
XDM-AP-S-001	Support HTTP Digest authentication	5.2, 6.3.1, 6.4.1	M	
XDM-AP-S-002	Support HTTP over TLS using the supported cipher suite	5.2, 9.5126.4.2	M	
XDM-AP-S-003	Support other cipher suites defined in RFC2246	9.06.4.2	O	

Item	Function	Reference	Status	Requirement
XDM-AP-S-004	Support XDM Client Identity Assertion	5.2, 6.3.2	M	
XDM-AP-S-005	Support XCAP request forwarding	6.3.3	M	
XDM-AP-S-006	Support Compression	6.3.4	O	
XDM-AP-S-007	Support for GAA	6.3, 6.4	O	

Appendix B. Examples (informative)

B.1 Sample XCAP Operation

Figure B.1 describes how an XCAP operation is performed in 3GPP/3GPP2 IMS. The “resource-list” application usage (see [Shared_XDMS]) 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 body content would, of course, be different). It is also assumed that the address of Aggregation Proxy is “xcap.example.com” and the XCAP Root URI is xcap.example.com”.

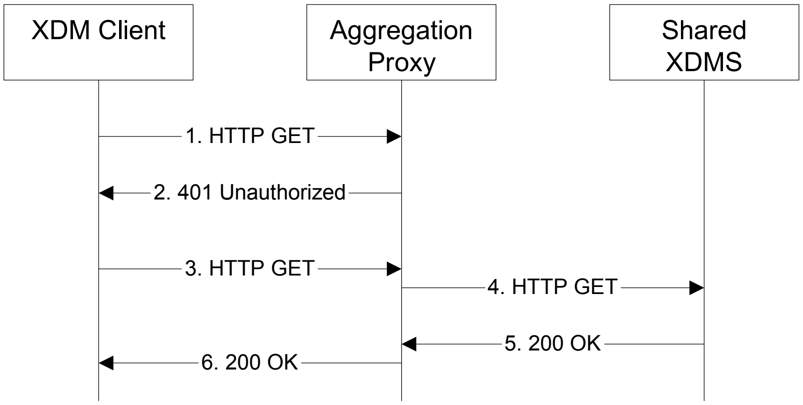

Figure B.1- Sample XCAP operation

The details of the flows are as follows:

1) The user “sip:joebloggs@example.com” wants to obtain an XML document. For this purpose the XDMC sends a 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/OMA1.0
Date: Thu, 08 Jan 2004 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/OMA1.0
Date: Thu, 08 Jan 2004 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/OMA1.0
Date: Thu, 08 Jan 2004 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: xcap.example.com
Via: HTTP/1.1 proxy.example.com (Apache/1.1)
User-Agent: XDM-client/OMA1.0
Date: Thu, 08 Jan 2004 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 document in the body.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Date: Thu, 08 Jan 2004 10:50:39 GMT
Etag: "eti87"
Content-Type: application/resource-lists+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list name="friends">
    <entry uri="sip:hermione.blossom@example.com"/>
    <entry uri="tel:5678;phone-context=+43012349999"/>
  </list>
</resource-lists>
```

6) The Aggregation Proxy encodes (optionally) the content and routes the response back to the XDM Client.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Via: HTTP/1.1 proxy.example.com (Apache/1.1)
Date: Thu, 08 Jan 2004 10:50:39 GMT
Authentication-Info: nextnonce="e966c32a924255e42c8ee20ce7f6"
Etag: "eti87"
Content-Encoding: gzip
Content-Type: application/resource-lists+xml
Content-Length: (...)

 (binary data)
```

B.2 Sample XCAP message flow

Example B.2 describes the message flows used to manipulate an XML document in an XDMS after authentication.

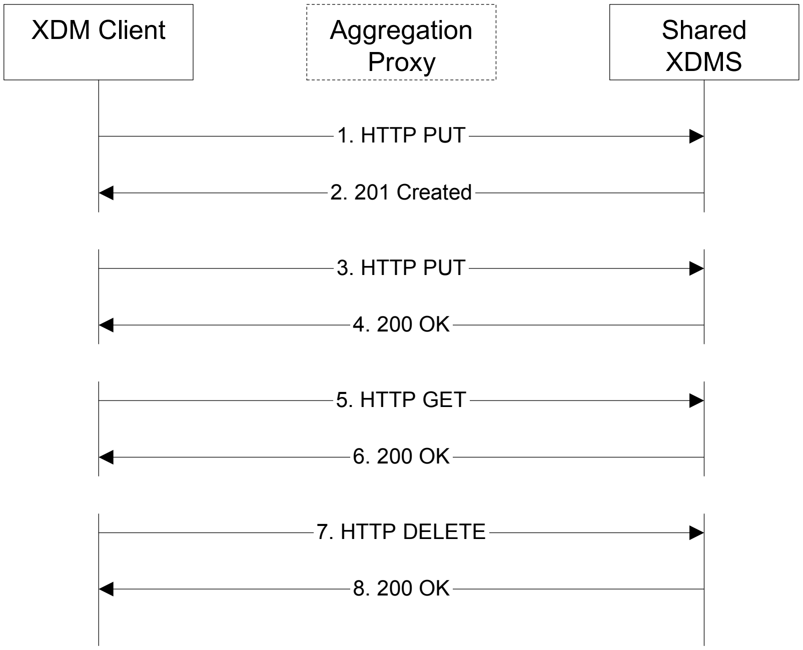

Figure B.2- XDM Client manipulating an XML document

NOTE: 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: 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 XCAP(HTTP) PUT request to create a new URI list document “index” for the user with a public SIP URI of “sip:joebloggs@example.com” in the (Shared) XDMS in the example.com domain.

```
PUT /resource-lists/users/sip:joebloggs@example.com/friends.xml HTTP/1.1
Host: xcap.example.com
…
Content-Type: application/resource-lists+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list name="My_friends">
    <entry uri="sip:friend1@example.com">
      <display-name>Friend1</display-name>
    </entry>
  </list>
</resource-lists>
```

2) The XDMS acknowledges the creation of the index document with a XCAP(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 XCAP(HTTP) PUT request to the just-created “index” document in “sip:joebloggs@example.com”’s home directory to add a new <entry> sub-element to the <list> element identified as “My_friends”.

```
PUT /resource-lists/users/sip:joebloggs@example.com/index/~~/resource-
   lists/list%5b@name=’My_friends’%5d]/entry%5d@uri=%22sip:friend2@example.com%22%5d HTTP/1.1
Host: xcap.example.com
…
Content-Type: application/xcap-el+xml
Content-Length: (…)

<entry uri="sip:friend2@example.com">
   <display-name>Friend2</display-name>
 </entry>
```

Note: The use of the Content Type “application/xcap-el+xml”.

4) The XDMS acknowledges the addition of new elements to the list with an XCAP(HTTP) “200 OK” reply.

```
HTTP/1.1 200 OK
Etag: “efefefef”

…
Content-Length: 0
```

5) The XDMC sends an XCAP(HTTP) GET request to retrieve “sip:joebloggs@example.com”’s “friends” list from the (Shared) XDMS.

```
GET /resource-lists/users/sip:joebloggs@example.com/friends.xml HTTP/1.1
Host xcap.example.com
```

6) The XDMS returns the list to the XDMC in the body of an XCAP(HTTP) “200 OK” message.

```
HTTP/1.1 200 OK
…
Etag: "ababab”
Content-Type:application/resource-lists+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
 <list name="My_friends">
  <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>
</resource-lists>
```

7) The XDMC sends an XCAP(HTTP) DELETE request to delete an <entry> identified by the URI “sip:friend2@example.com” from sip:joebloggs@example.com” ‘s “My_friends” list in the Shared XDMS.

```
DELETE /resource-lists/users/sip:joebloggs@example.com/index/~~/resource-
   lists/list%5b@name=%22My_friends%22%5d/entry%5b@uri=%22sip:friend2@example.com%22%5d HTTP/1.1
Host: xcap.example.com
```

The XDMS, after checking the privileges of the principal, performs the deletion.

8) The XDMS acknowledges the deletion of the “friend2” element from the list with an XCAP(HTTP) 200 OK.

```
HTTP/1.1 200 OK
Etag: “ghghgh”
…
Content-Length: 0
```

B.3 Sample XCAP Directory Retrieval Operation of all user documents

Figure B.3 describes how an XCAP operation is performed to retrieve all of a user’s documents for all application usages. For simplicity, only two XDMSes are shown and the authentication steps are omitted.

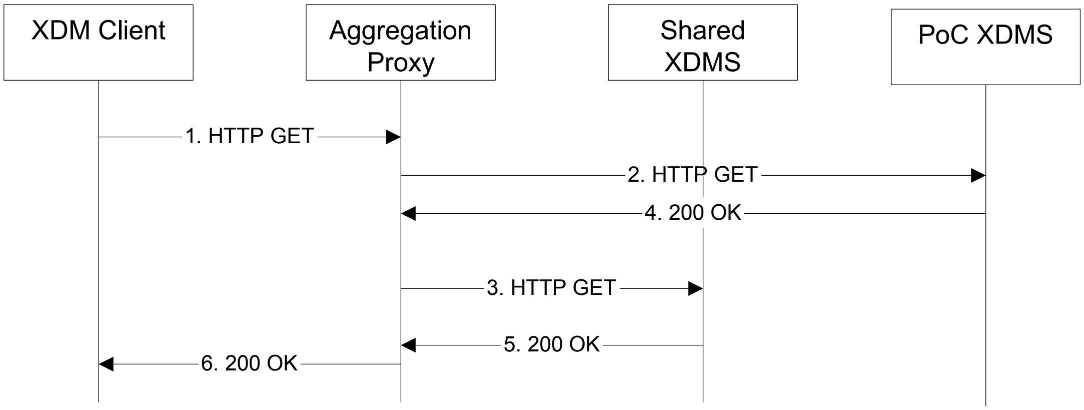

Figure B.3- Sample XCAP Directory retrieval operation

The details of the flows are as follows:

1) The user “sip:joebloggs@example.com” wants to obtain a list of all his 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/OMA1.0
Date: Thu, 08 Jan 2004 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 PoC XDMS.

3) The Aggregation proxy forwards the HTTP GET from step 1) to the Shared XDMS.

4) The PoC XDMS returns the “directory.xml” document containing a list of all the PoC Group documents belonging to sip:joebloggs@example.com in a HTTP 200 OK response

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Date: Thu, 08 Jan 2004 10:50:39 GMT
Content-Type: application/vnd.oma.xcap-directory+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-directory xmlns=”urn:oma:xml:xdm:xcap-directory” >
  <folder auid=poc-groups>
    <entry uri=”http://xcap.example.com/org.openmobilealliance.poc-
   groups/users/sip:joebloggs@example.com/skiing” etag=”abc123”/>
    <entry uri=”http://xcap.example.com/org.openmobilealliance.poc-
   groups/users/sip:joebloggs@example.com/shopping” etag=”def456”/>
  </folder>
</xcap-directory>
```

where each <entry> element lists a document containing one of sip:joebloggs@example.com’s PoC Groups called “skiing” and “shopping” in this example.

5) The Shared XDMS returns the “directory.xml” document containing the URI lists document belonging to sip:joebloggs@example.com in a HTTP 200 OK response

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Date: Thu, 08 Jan 2004 10:51:44 GMT
Content-Type: application/vnd.oma.xcap-directory+xml
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>
</xcap-directory>
```

where the <entry> element lists the sip:joebloggs@example.com’s URI lists index document.

6) The Aggregation Proxy returns the consolidated “directory.xml” document to the user in a HTTP 200 OK response.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Date: Thu, 08 Jan 2004 10:55:39 GMT
Content-Type: application/vnd.oma.xcap-directory+xml
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”/>
    <entry uri=”http://xcap.example.com/resource-lists/users/sip:joebloggs@example.com/colleagues”
   etag=”xyz123”/>
  </folder>
  <folder auid=poc-groups>
    <entry uri=”http://xcap.example.com/org.openmobilealliance.poc-
   groups/users/sip:joebloggs@example.com/skiing” etag=”abc123”/>
    <entry uri="http://xcap.example.com/org.openmobilealliance.poc-
   groups/users/sip:joebloggs@example.com/shopping" etag=”def456”/>
  </folder>
</xcap-directory>
```

B.4 Sample XCAP Directory Retrieval Operation of specific user documents

Figure B.4 describes how an XCAP operation is performed to retrieve all of a user's documents corresponding to a particular application usage. For simplicity, the authentication steps are omitted.

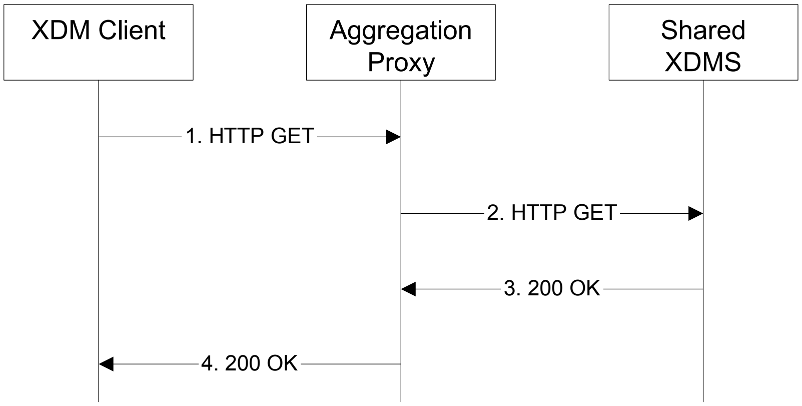

Figure B.4- Sample XCAP Directory retrieval operation from a particular XDMS

The details of the flows are as follows:

1) The user "sip:joebloggs@example.com" wants to obtain a list of all his documents (URI lists) stored in the Shared 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/OMA1.0
Date: Thu, 08 Jan 2004 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 Shared XDMS.

3) The Shared 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/OMA1.0
Date: Thu, 08 Jan 2004 10:55:39 GMT
Content-Type: application/xcap-el+xml
Content-Length: (...)

  <folder auid=”resource-lists”>
    <entry uri="http://xcap.example.com/resource-lists/users/sip:joebloggs@example.com/friends"
   etag=”pqr999”/>

  </folder>
```

4) The Aggregation proxy returns the same entity body as in step 3 to the XDMC is a HTTP 200 OK message.

B.5 Sample Subscribing to Changes in XML Documents

This is an informative section to give an illustrative example on how the subscription and notification procedures happen when XDMC located in the User Equipment requests to subscribe to changes in the PoC group document. Note the procedure is identical no matter an XDMC is subscribing to an XML belonging to himself or others.

Figure B.5 is an example that demonstrates how an XDM Client subscribes to changes in a PoC group document.

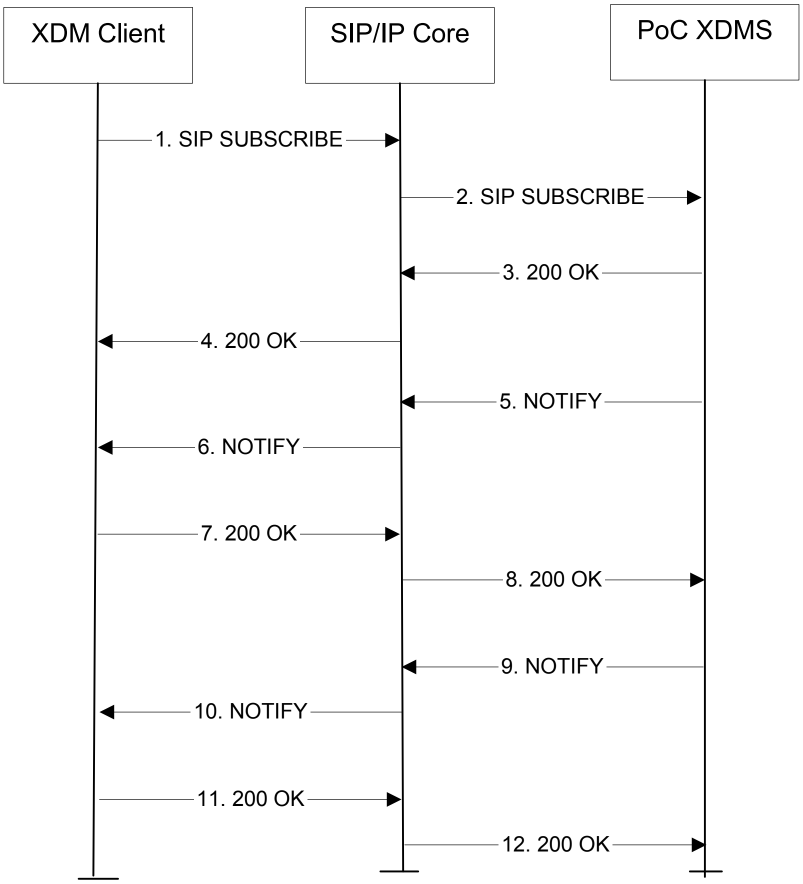

Figure B.5- XDM Client subscribes to changes in XML documents.

1) XDM Client (XUI= joe.bloggs@example.com) subscribes to his PoC group 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 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: ua-profile;profile-type="application"; auid="org.openmobilealliance.poc-
   groups";Vendor="vendor1"; Model="1";Version="1.0";document="org.openmobilealliance.poc-
   groups/users/sip:joe.bloggs@example.com/joebloggs_friends/"
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, message/external-body
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

2) The SIP/IP core network forwards the SIP SUBSCRIBE request to the corresponding PoC XDMS. When the SIP/IP Core network corresponds to 3GPP IMS or 3GPP2 MMD, the subscriber's preferred public SIP URI shall be inserted in P-Asserted-Identity header.

```
SUBSCRIBE sip:joe.bloggs@example.com 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:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
Route: <sip:pocxdms.home1.net;lr>
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>
Event: ua-profile;profile-type="application"; auid="org.openmobilealliance.poc-groups";
   Vendor="vendor1"; Model="1";Version="1.0";document="org.openmobilealliance.poc-
   groups/users/sip:joe.bloggs@example.com/joebloggs_friends/"
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, message/external-body
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

3) Upon receiving a SIP SUBSCRIBE request for the “ua-profile” event package, the PoC XDMS shall perform the necessary authorization checks on the originator's identity. If the authorization is successes, it shall create a subscription dialog to "ua-profile" event package to provide the changes of the data identified by the "Event" header parameters, and return 200OK 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 PoC XDMS generates and sends an initial SIP NOTIFY containing initial references to XDM documents.

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pocxdms1.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: ua-profile
Content-Type: application/xcap-diff+xml
Contact: <sip:pocxdms1.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/root”>
    <document new-etag=”7ahggs”
     doc-selector="org.openmobilealliance.poc-groups
     /users/sip:joe.bloggs@example.com/joebloggs_friends”>
```

```
</document>
   </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 documents it may retrieve them.

7) The XDMC responds with a 200 OK.,

8) The SIP/IP core network forwards the 200 OK to the PoC XDMS.

9) After some updates in the XDM document, the PoC XDMS sends the diff part in 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 pocxdms1.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: ua-profile
Content-Type: application/xcap-diff+xml
Contact: <sip:pocxdms1.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”
    <document previous-etag=”7ahggs” doc-selector="org.openmobilealliance.poc-
   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 uri="sip:new-friend@example.com"/>
        </add>
      </change-log>
     </document>
   </xcap-diff>
```

10) The SIP/IP core network forwards the SIP NOTIFY request to appropriate XDMC.

11) The XDMC responds with a 200 OK and update the old content identified with older eTag, if any exists, according to [XCAP_Config].

12) The SIP/IP core network forwards the 200 OK to the PoC XDMS.

Appendix C. XDMC Provisioning (Normative)

This appendix specifies the parameters that are needed by the XDM Client. Existing parameters in [Provisioning Content] and [OMA-DM-v1-1-2] are re-used; those without corresponding parameters are defined and to be registered in OMNA through OMA official registration process.

The Management Object (MO) for OMA XDM is defined in [XDM_MO]. The MO MAY be used for initial provisioning of parameters when the DM Profile is to be used (as specified on [OMA-DM-v1-2]), and the MO SHOULD be used for continuous provisioning of parameters according to [OMA-DM-v1-1-2] or [OMA-DM-v1-2], if required by the service provider to update service configurations.

C.1 Provisioned XDMC Parameters

The parameters listed in the table below are needed for XDM client provisioning:

ID	Name	Description	Mandatory (M) /Optional (O)
1	Application identity	Uniquely identifies the application	M
2	Application name	User displayable name for the XML Document Management 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. Used to access XDM servers using the referenced SIP/IP core.	M
6	XCAP Root URI	The root of all XDM 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

NOTE: The parameters ”XCAP Authentication username” and “XCAP Authentication password” are not needed if GAA is used in a 3GPP/3GPP2 realization.

In addition, there may be enabler-specific parameters related to 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 PoC Group), so that the XDM Client 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], and

b. substitution tags enclosed in "< >"brackets.

The XDM Client SHALL support the following substitution tags:

<id> : The XDM Client SHALL replace this tag with a unique identifier, generated by the XDM Client.

<user> : The XDM Client SHALL replace this tag with the user part of the XUI if the XUI is a Public SIP URI. If the XUI is a Tel URI [RFC 3966] then the XDM Client 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 XDM Client SHALL replace this tag with the XUI.

NOTE: the XUI is a Public SIP URI [RFC3261] or Tel URI [RFC3966].

NOTE: 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).

Illustrative examples of URI templates are shown in Table X.

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>;poc-group=<id>	sip:joe@example.com;poc-group=abc123
<xui>;poc-group=<id>	tel:+1720-555-1212;poc-group=abc123

Table X: Example usages of URI Templates

C.2 Initial Provisioning document

This chapter defines the provisioning document structure as described in [Provisioning Content].

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 [Provisioning Content]			
APPID	Required	1	"XDMS"
PROVIDER-ID	Optional	0 or 1	none
TO-APPREF	Required	1	n/a
NAME	Required	1	n/a
TO-NAPID	Required	1 or more	n/a
URI	Required	1	n/a
AAUTHNAME	Optional	0 or 1	n/a
AAUTHSECRET	Optional	0 or 1	n/a
AAUTHTYPE	Optional	0 or 1	n/a

The provisioning document in an AC file format

```
IDENTIFYING INFORMATION
#######################
```

```
APPID: xx.
APPID type: OMNA.
Owner: OMA Presence and Availability Working Group.
Contact: OMA Presence and Availability Working Group <TECHNICAL-
COMMENTS@MAIL.OPENMOBILEALLIANCE.ORG>.
Registration version: 1.0.
Registration timestamp: 2004-12-xx.
Application description: XDM.
Application reference: XML Document Management(XDM) enabler. OMA XDM Enabler
Release 1.0 specifications, URL:http://www.openmobilealliance.org/documents.asp.

Legal text:
Use of this document is subject to all of the terms and conditions of the Use
Agreement located at http://www.openmobilealliance.org/UseAgreement.html.
You may use this document or any part of the document for internal or educational
purposes only, provided you do not modify, edit or take out of context the
information in this document in any manner. Information contained in this document
may be used, at your sole risk, for any purposes.

You may not use this document in any other manner without the prior written
permission of the Open Mobile Alliance.  The Open Mobile Alliance authorizes you to
copy this document, provided that you retain all copyright and other proprietary
notices contained in the original materials on any copies of the materials and that
you comply strictly with these terms. This copyright permission does not constitute
an endorsement of the products or services.  The Open Mobile Alliance assumes no
responsibility for errors or omissions in this document.

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

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN
MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY
OF THE IPR’S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT
LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR
WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.
THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT,
INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES
ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION
CONTAINED IN THE DOCUMENTS.

© 2004 Open Mobile Alliance Ltd.  All Rights Reserved. Used with the permission of
the Open Mobile Alliance Ltd. under the terms set forth above.

WELL-KNOWN PARAMETERS
#####################
Characteristic/name: APPLICATION/APPID.
Status: Required.
```

```
Occurs: 1/1.
Default value: None.
Used values: xx.
Interpretation: To uniquely identify the XDM enabler.
-------
Characteristic/name: APPLICATION/PROVIDER-ID.
Status: Optional.
Occurs: 0/1.
Default value: None.
Used values: N/A.
Interpretation: Identity of the XDM service provider.
-------
Characteristic/name: APPLICATION/TO-APPREF.
Status: Required.
Occurs: 1/1.
Default value: None.
Used values: N/A.
Interpretation: It specifies the linkage between XDM and the SIP-IP-core, e.g. IMS.
-------
Characteristic/name: APPLICATION/NAME.
Status: Required.
Occurs: 1/1.
Default value: None.
Used values: N/A.
Interpretation: User displayable name for the XML Document Management enabler.
-------
Characteristic/name: APPLICATION/TO-NAPID.
Status: Required if direct use of Network Access Point supported.
Occurs: 1/*.
Default value: None.
Used values: N/A.
Interpretation: specifies the network access point used for a given application.
-------
Characteristic/parameter: RESOURCE/URI.
Status: Required.
Occurs: 1/1.
Default value: None.
Used values: A HTTP URL.
Interpretation: Identifies the ROOT URI of the documents under the global XCAP
document tree managed by the XDM server.
-------
Characteristic/parameter: RESOURCE/AAUTHNAME.
Status: Optional.
Occurs: 0/1.
Default value: None.
Used values: String.
Interpretation: HTTP user name, for accessing XDM over XCAP.
-------
Characteristic/parameter: RESOURCE/AAUTHSECRET.
Status: Optional.
Occurs: 0/1.
Default value: None.
Used values: String.
Interpretation: HTTP digest password, for accessing XDM over XCAP.
-------
Characteristic/name: RESOURCE/AAUTHTYPE.
Status: Optional.
Occurs: 0/1.
```

```
Default value: None.
Used values: "HTTP-DIGEST". Value set can be extended with new values in future.
Interpretation: Authentication method for XDM over XCAP.
----------


EXAMPLE
#######
<characteristic type="APPLICATION">
  <parm name="APPID" value="XDM"/>
  <parm name=”PROVIDER-ID” value="Best"/>
  <parm name="NAME" value="XDM"/>
  <parm name="TO-APPREF" value="SIP-IP-CORE"/>
  <parm name="TO-NAPID" value="IMS-NAP"/>
  <characteristic type="RESOURCE">
    <parm name="URI" value="http://xcap.example.com/"/>
    <parm name="AAUTHNAME" value="httpusername"/>
    <parm name="AAUTHSECRET" value="httpdigestpasswd"/>
    <parm name="AAUTHTYPE" value="HTTP-DIGEST"/>
  </characteristic>
</characteristic>
###END###
```

Appendix D. OMA specific extensions to HTTP entity header fields (Normative)

This section defines the syntax of OMA specific extension headers to HTTP entity header fields introduced in this document in Augmented Backus-Naur form as defined in [RFC 2234].

D.1 X-XCAP Asserted-Identity Extensions-Header

When 3GPP GAA is not present, the "X-XCAP-Asserted-Identity" header is used by Aggregation Proxy to deliver the HTTP Digest authenticated user identity. It contains the user identity surrounded by quotation marks (") as provided by the “username” field in the HTTP Digest Authorization header. (See section 6.3.2 for details.) The type of the user identity SHALL be either public SIP URI or TEL URI in this document.

The following is ABNF definition for “X-XCAP-Asserted-Identity”:

```
X-XCAP-Asserted-Identity = "X-XCAP-Asserted-Identity" ":" DQUOTE identity DQUOTE
identity = *(%x20-21 / %x23-7E)
```

In the syntax definition the rule 'identity' refers to the user identity and it is defined as a string of printable characters and spaces but excluding quotation marks.

Appendix E. Change History (Informative)

E.1 Approved Version History

Reference	Date	Description
OMA-TS-XDM_Core-V1_0-20060612-A	12 Jun 2006	Status changed to approved OMA-TP-2006-0196R03-INP_XDM_V1_0_for_final_approval.
OMA-TS-XDM_Core-V1_0_1-20061128-A	28 Nov 2006	Incorporated CRs: OMA-PAG-2006-0382R02 OMA-PAG-2006-0394R02 OMA-PAG-2006-0466R01 OMA-PAG-2006-0478 OMA-PAG-2006-0573 OMA-PAG-2006-0713R02 OMA-PAG-2006-0735 OMA-PAG-2006-0736R03 OMA-PAG-2006-0739 OMA-PAG-2006-0740R01