XML Document Management (XDM) Specification

Approved Version 1.1 – 27 Jun 2008

Open Mobile Alliance

OMA-TS-XDM_Core-V1_1-20080627-A

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

[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 24.109]	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 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/
[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/
[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/
[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/
[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/index.cfm NOTE: Work in progress, estimated availability January 2006
[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/index.cfm
[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/index.cfm
[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/
[OMA-XDM-AC]	“XDM Application Characteristics file of XDM V1.0”, Version 1.0, Open Mobile Alliance, OMA-SUP-AC_ap0003_xdm-v1_0, URL:http://www.openmobilealliance.org/
[Provisioning Content]	OMA – Provisioning Content V1.1, OMA-DM_ProvCont-V1_2_0, Open Mobile Alliance™, URL:http://www.openmobilealliance.com/
[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
[RFC2616]	“Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding, June 1999, URL:http://www.ietf.org/rfc/rfc2616.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
[RFC2818]	"HTTP Over TLS", Rescorla, E., RFC 2818, May 2000. URL:

URL:http://www.ietf.org/rfc/rfc2818.txt

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

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

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

[XDM_MO] "OMA Management Object for XML Document Management", Version 1.1, Open Mobile Alliance™, OMA-TS-XDM_MO-V1_1, URL:http://www.openmobilealliance.org/.

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

[XSD_XCAPDIR] "XDM – XCAP Directory", Candidate Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_xcapDirectory-V1_0, 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/

2.2 Informative References

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

[Presence_XDM] "Presence XDM Specification", Version 1.1, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE_XDM-V1_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.

[RLS_XDM] "Resource List Service (RLS) XDM Specification", Version 1.1, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE_RLS_XDM-V1_1, URL:http://www.openmobilealliance.org/.

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

[XDMAD] "XML Document Management Architecture", Version 1.1, Open Mobile Alliance™. OMA-AD-XDM-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

Application Server	A functional entity that implements the service logic for SIP Sessions (e.g. PoC Server).
Application Unique ID (AUID)	A unique identifier within the namespace of application unique IDs created by this specification that differentiates XCAP resources accessed by one application from XCAP resources accessed by another. (Source: [RFC4825])
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: [RFC4825])
Document URI	The HTTP URI containing the XCAP root and document selector, resulting in the selection of a specific document. (Source: [RFC4825])
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: [RFC4825])
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: [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	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.
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])
XCAP Client	An HTTP client that understands how to follow the naming and validation constraints defined in this specification. (Source: [RFC4825])
XCAP Resource	An HTTP resource representing an XML document, an element within an XML document, or an attribute of an element within an XML document that follows the naming and validation constraints of XCAP. (Source: [RFC4825])
XCAP Root	A context that includes all of the documents across all application usages and users that are managed by a server. (Source: [RFC4825])
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 this specification. (Source: [RFC4825])
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: [RFC4825])
XCAP Application Usage	Detailed information on the interaction of an application with an XCAP server. (Source: [RFC4825])

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

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
- MAY support 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 SHALL:

1. be configured as an HTTP reverse proxy (see [RFC3040]), forward the XCAP requests to the corresponding XDM Server, and forward the response back to the XDM Client as described in section 6.3.3; and
2. support authenticating the XDM Client as described in section 6.3.1; and
3. assert the XDM Client identity as described in section 6.3.2; and
4. protect the XCAP traffic by enabling TLS transport security mechanism as described in section 6.4.1.

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 [RFC4825].

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 [RFC4825] 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.)

The XCAP Root URI SHALL be the URI of the Aggregation Proxy in the XDMC's home domain; The XDMC that resides in an UE SHALL use the XCAP Root URI as provided by XDMC provisioning in Appendix C. The XDMC that resides in an application server SHALL use the XCAP Root URI as preconfigured. If the XDMC resides within an AS, it SHALL have the possibility to address the XDMS directly without going through the Aggregation Proxy; in this case, the XDMC SHALL be preconfigured per AUID with the host address of the XDMS, in addition to the XCAP Root URI.

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 [RFC4825].
- 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 [RFC4825] 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 [RFC4825] Section 7.1.

6.1.1.2.2 Delete a Document

Deleting an XML document SHALL follow the procedures described in [RFC4825] Section 7.2.

6.1.1.2.3 Retrieve a Document

Retrieving an XML 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 element in an XML document SHALL follow the procedures described in [RFC4825] Section 7.4.

6.1.1.2.5 Delete an Element

Deleting an element in an XML document SHALL follow the procedures described in [RFC4825] Section 7.5.

6.1.1.2.6 Retrieve an Element

Retrieving an element in an XML document SHALL follow the procedures described in [RFC4825] 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 [RFC4825] 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 [RFC4825] 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 [RFC4825] Section 7.9.

6.1.1.2.10 Fetch Namespace Bindings

Fetching namespace bindings of an element in a XML document SHALL follow the procedures described in [RFC4825] Section 7.10.

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 [RFC4825].

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 [RFC4825] section 8.5 for the management of Etags.

An XDM server SHALL implement the conditional operations of [RFC4825] 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 [RFC4825] Section 8.1.

6.2.1.2 PUT handling

HTTP PUT requests targeted at an XDM resource SHALL be processed as described in [RFC4825] Section 8.2.

6.2.1.3 GET handling

HTTP GET requests targeted at an XDM resource SHALL be processed as described in [RFC4825] Section 8.3.

6.2.1.4 DELETE handling

HTTP DELETE requests targeted at an XDM resource SHALL be processed as described in [RFC4825] Section 8.4.

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

Authentication SHALL be performed over the XDM-3 reference point (see [XDMAD]). The Aggregation Proxy SHALL fulfill the functions described in section 6.4.1.

6.3.2 XDM Client identity assertion

When the 3GPP/3GPP2 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 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. The Aggregation Proxy SHALL check the existence of the X-3GPP-Intended-Identity header of the HTTP request:

1. If included, the Aggregation Proxy SHALL check the value in the header is allowed to be used by 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.

NOTE: The Enabler Specific Server should also provide the XDMC identity assertion when the Enabler Specific Server generates a HTTP request to XDMS on behalf of a User. In this case, as the Aggregation Proxy does, the Enabler Specific Server should use the "X-XCAP-Asserted-Identity" HTTP header, or the "X-3GPP-Asserted-Identity" HTTP header in 3GPP/3GPP2 realization, to carry the identity of the user for whom it generates 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;
    c. prepend the XCAP Root URI to the received "uri" attribute value if it contains only the Document Selector.

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 between the XDMC and the Aggregation Proxy (see [XDMAD]) SHALL provide mutual authentication.

When realized with 3GPP IMS or 3GPP2 MMD networks, the XDM-3 corresponds to the Ut reference point. In this case the Aggregation Proxy SHALL act as an Authentication Proxy defined in [3GPP TS 33.222] and the authentication between the XDM 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 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 user identity 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 or TEL URI) identifying the user (the public user identity);

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

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.

The authenticated identity provided by the Aggregation Proxy SHALL be shared on the following reference points (see [XDMAD]):

1. The XDM-4 reference point between the Aggregation Proxy and the Shared XDMS;
2. the Enabler Specific reference point between the Aggregation Proxy and the Enabler Specific XDMS;

When realized with 3GPP IMS or 3GPP2 MMD networks, the above listed reference points SHALL use the security mechanisms as defined in the corresponding 3GPP/3GPP2 specifications.

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 [RFC4825] Section 9 and the extensions defined in Section 6.6.3. For additional details of the handling of those, see [RFC4825] 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 if the HTTP request by the XDMC fails to get authorized by the XDMS per the authorization policy defined by the target Application Usage.

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 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 [RFC4825] 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 [RFC4745].

Application usages that utilize the <external> element SHALL resolve a XCAP 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.

6.6.2 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;

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 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

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;
3) MAY include the <anonymous-request> condition element; and
4) MAY include the <other-identity> condition element.

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.

If present in any rule, the <external-list> element SHALL match those identities that are part of a URI List. 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 SHALL match those incoming requests that have been identified as anonymous.

NOTE: 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 6.6.2.3 "*Combining Permissions*".

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] to identify the source of the anonymous request.

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.

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 [RFC4825] 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 [RFC4745] Section 13 with the extension described in [XSD_COMMONPOL].

6.6.2.3 Combining Permissions

When evaluating any authorization policy document based on [RFC4745] 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 <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.

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 [RFC4825], 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 [RFC4825] 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 [RFC4825] and this section.

6.6.3.1 Structure

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

a. MAY include the “phrase” attribute;

b. 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;

c. 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 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-failure> 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 [RFC4825] 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 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>. It SHALL consist 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 SHALL consist of a number of <entry> elements or an <error-code> element. Each <entry> element SHALL contain a number of attributes, which are:

1. uri: this attribute SHALL be the Document URI or Document Selector 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 Status-Code and Reason-Phrase retrieved from the Status-Line of the received HTTP response message returned by an XDMS (see [RFC2616]).

For an XCAP GET request targeted at the “directory.xml” 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 the “directory.xml” document containing a <folder> element for each supported AUID providing a list of all XML 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, it SHALL include the Document URI as the value of the “uri” attribute returned to the Aggregation Proxy; otherwise, it SHALL include the Document Selector.

The Aggregation Proxy SHALL aggregate responses from all XDMSs before sending the composite “directory.xml” document back to the XDM Client. The content type of the returned “directory.xml” document SHALL be “application/vnd.oma.xcap-directory+xml” as defined in section 6.7.2.2. When the Aggregation Proxy receives an HTTP “200 OK” it SHALL include all returned <folder> elements in the composite “directory.xml” document with the content.

The Aggregation Proxy SHALL prepend the XCAP Root URI to the received “uri” attribute value if it contains only the Document Selector.

When an XDMS response is received with an error message, the Aggregation Proxy SHALL insert one <folder> element containing an <error-code> child element with the error message included for every corresponding AUID.

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

6.7.2.1 Application Unique ID

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

6.7.2.2 MIME Type

The MIME type for this document SHALL be “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 SHALL be only one XCAP directory document per XUI in each XDMS. The name of the XCAP directory document SHALL be “directory.xml”.

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 XDMS SHALL be the only Principal allowed to create and modify the “directory.xml” document. Thus, the Primary Principal SHALL only be allowed to retrieve this document.

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

6.8 Global Documents

[RFC4825] 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 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.

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	
XDM-XDMC-C-002	Support for XDM Operations	6.1.1.2	M	

Item	Function	Reference	Status	Requirement
XDM-XDMC-C-003	Void			
XDM-XDMC-C-004	Void			
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	
XDM_XDMC-C-009	Support GAA	6.4.1	O	
XDM_XDMC-C-010	Support Application Usage “xcap-caps”	6.7	O	
XDM_XDMC-C-011	Support Application Usage “org.openmobilealliance.xcap-directory”	6.7	O	
XDM_XDMC-C-012	XDM Client Identity Assertion	6.4.3	M	
XDM_XDMC-C-013	Support types of <error-element> as required	6.6.3	M	
XDM-XDMC-C-014	Support Early IMS authentication	6.4.1	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	Void			
XDM-XDMC-C-004	Void			

A.2 XDM Server

Item	Function	Reference	Status	Requirement
XDM-XDMS-S-001	Support for XCAP	6.2.1	M	
XDM-XDMS-S-002	Void			
XDM-XDMS-S-003	Void			
XDM-XDMS-S-004	Void			
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	
XDM_XDMS-S-009	Not using other types of <error-element> than what is recommended.	6.6.3	O	

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, 5.56.4.2	M	
XDM-AP-S-003	Support other cipher suites defined in RFC2246	5.56.4.2	O	
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	
XDM_AP-S-008	Sending XCAP response back	6.3.3	M	

Item	Function	Reference	Status	Requirement
XDM_AP-S-009	Support Error Handling	6.5, 6.4.1	M	
XDM_AP-S-010	Acting as an HTTP Proxy [RFC2616] and configuration as an HTTP Reverse Proxy [RFC3040]	6.3	M	
XDM_AP-S-011	XCAP Server Capabilities retrieval (Application Usage "xcap-caps")	6.3.3.2	M	
XDM_AP-S-012	XCAP Directory retrieval (Application Usage "org.openmobilealliance.xcap-directory")	6.3.3.3	M	
XDM-AP-S-013	Support Early IMS authentication	5.2, 6.3.1, 6.4.1	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".

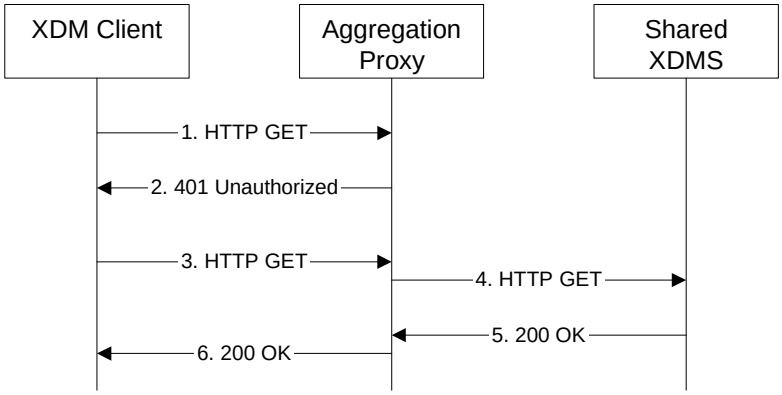

Figure B.1- Sample XCAP operation

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/OMA1.1
Date: Thu, 10 Jan 2008 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/OMA1.1
Date: Thu, 10 Jan 2008 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/OMA1.1
Date: Thu, 10 Jan 2008 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-xcap.example.com
Via: HTTP/1.1 proxy.example.com (Apache/1.1)
User-Agent: XDM-client/OMA1.1
```

```
Date: Thu, 10 Jan 2008 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/OMA1.1
Date: Thu, 10 Jan 2008 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:+33492944949"/>
  </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/OMA1.1
Via: HTTP/1.1 proxy.example.com (Apache/1.1)
Date: Thu, 10 Jan 2008 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)
```

B.2 Sample XCAP message flow

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

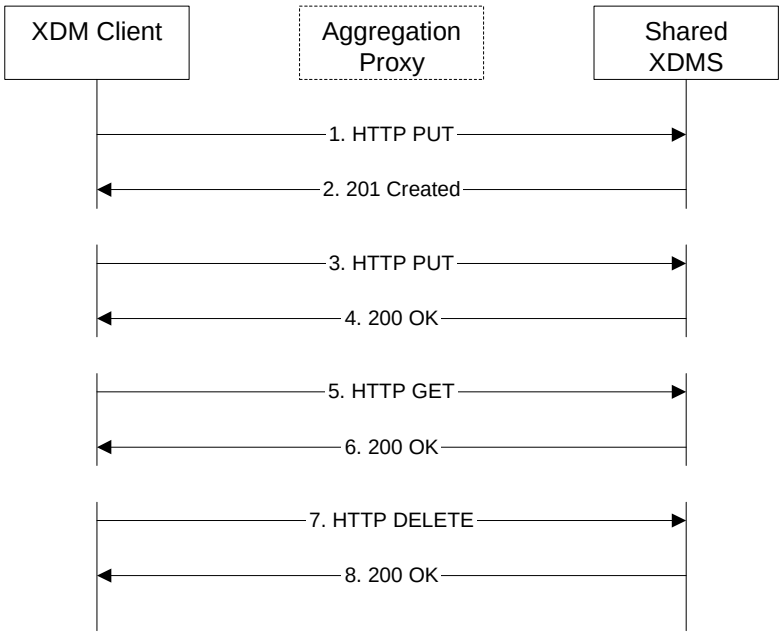

Figure B.2- XDM Client manipulating an XML document

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 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/index.xml 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 XDMS acknowledges the creation of the index 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” 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=%22-list-
   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>
```

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 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 (Shared) 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 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 Shared 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 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 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.

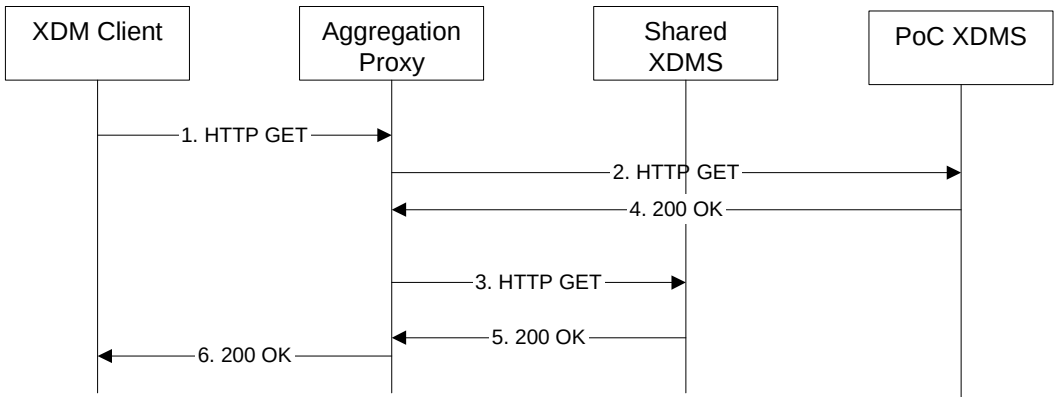

Figure B.3- Sample XCAP Directory retrieval operation

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/OMA1.1
Date: Thu, 10 Jan 2008 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 and PoC User Access Policy documents belonging to sip:joebloggs@example.com in a HTTP 200 OK response

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.1
Date: Thu, 10 Jan 2008 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.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>

  <folder auid="org.openmobilealliance.poc-rules">
    <entry uri="http://xcap.example.com/org.openmobilealliance.poc-
   rules/users/sip:joebloggs@example.com/pocrules" etag="hjk987"/>
  </folder>

</xcap-directory>
```

where <folder> element defined used AUID and <entry> element lists a document under that AUID. In this example "directory.xml" document contains sip:joebloggs@example.com's PoC Groups called "skiing" and "shopping" and PoC User Access Policy document.

5) The Shared XDMS returns the "directory.xml" document containing the URI list and Group Usage list documents belonging to sip:joebloggs@example.com in a HTTP 200 OK response

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.1
Date: Thu, 10 Jan 2008 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 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>
</xcap-directory>
```

where the folder element defined used AUID and <entry> element lists the sip:joebloggs@example.com's document under that AUID.

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/OMA1.1
Date: Thu, 10 Jan 2008 10:55:39 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="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="org.openmobilealliance.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>

  <folder auid="org.openmobilealliance.poc-rules">
    <entry uri="http://xcap.example.com/org.openmobilealliance.poc-
   rules/users/sip:joebloggs@example.com/pocrules" etag="hjk987"/>
  </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.

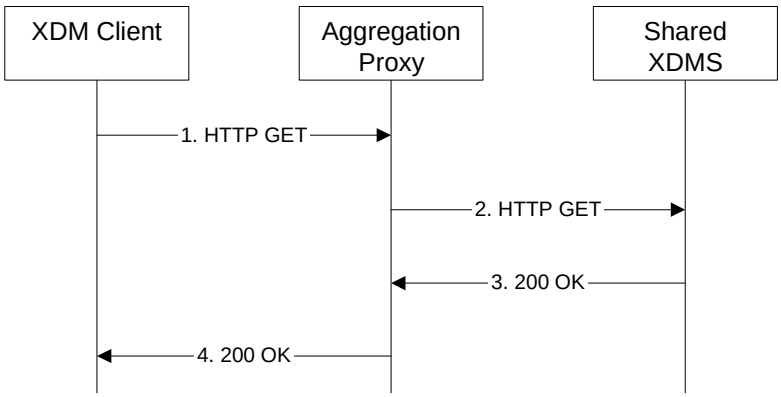

Figure B.4- Sample XCAP Directory retrieval operation from a particular XDMS

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/OMA1.1
Date: Thu, 10 Jan 2008 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/OMA1.1
Date: Thu, 10 Jan 2008 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 Shared XDMS in this example.

4) The Aggregation Proxy returns the consolidated "directory.xml" 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/OMA1.1
Date: Thu, 10 Jan 2008 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>
```

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 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

NOTE 1: The parameters ”XCAP Authentication username” and “XCAP Authentication password” are not needed if GAA is used in a 3GPP/3GPP2 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 XDM Client 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, Service URI of a Presence List), 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. the characters ":" , "@" and ";"

c. 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 using only unreserved characters according to [RFC3986].

<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 3: the XUI is a Public 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>;poc-group=<id>	sip:joe@example.com;poc-group=abc123
<xui>;poc-group=<id>	tel:+1720-555-1212;poc-group=abc123
<xui>;pres-list=<id>	sip:joe@example.com;pres-list=abc123

Table 1: 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	"ap0003"
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

The XDM Application Characteristics file for the OMA XDM 1.1 enabler is defined in [OMA-XDM-AC].

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-2006-0734 OMA-PAG-2006-0735 OMA-PAG-2006-0736R03 OMA-PAG-2006-0739 OMA-PAG-2006-0740R01
OMA-TS-XDM_Core-V1_0_1-20080627-A	27 Jun 2008	Status changed to Approved by TP TP ref# OMA-TP-2008-0244-INP_XDM_V1_1_ERP_for_Final_Approval