XML Document Management (XDM) Specification

Candidate Version 2.0 – 24 Jul 2007

Open Mobile Alliance

OMA-TS-XDM_Core-V2_0-20070724-C

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 functional entities that are needed to provide XDM services to other enablers. Such enablers can utilize this specification to support any required application-specific usages.

2. References

2.1 Normative References

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

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

Note: Work in progress, awaiting IETF drafts

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

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

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

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

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

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

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

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

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

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

[Dict] "Dictionary for OMA Specifications", Version 2.4, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_4, URL: http://www.openmobilealliance.org/

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

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

[IETF-MSRP] IETF draft-ietf-simple-message-sessions-19 "The Message Session Relay Protocol", B. Campbell, R. Mahy, C. Jennings, February 2007, URL: http://www.ietf.org/internet-drafts/draft-ietf-simple-message-sessions-19.txt

Note: Work in progress

[IETF-XCAP_Diff] IETF draft-ietf-simple-xcap-diff-05 "An Extensible Markup Language (XML) Document Format for Indicating Changes in XML Configuration Access Protocol (XCAP) Resources", J. Rosenberg, March, 2007, URL: http://www.ietf.org/internet-drafts/draft-ietf-simple-xcap-diff-05.txt
Note: Work in progress

[IETF-XCAP_Diff_Event] IETF draft-urpalainen-sip-xcap-diff-event-00 "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Diff Event Package ", J, Urpalainen, Dec 7, 2006.
URL: http://www.ietf.org/internet-drafts/draft-urpalainen-sip-xcap-diff-event-00.txt

Note: Work in progress

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

[PoC_CP] “Push to talk Over Cellular (PoC) - Control Plane Specification”, Draft Version 2.0, Open Mobile Alliance™, OMA-TS-PoC-ControlPlane-V2_0, URL: http://www.openmobilealliance.org/”

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[RLS_XDM] “Resource List Service (RLS) XDM Specification”, Version 1.0, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE_RLS_XDM-V1_0_1, URL:http://www.openmobilealliance.org/

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

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

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

[XDM_AD] “XML Document Management Architecture”, Version 2.0, Open Mobile Alliance™, OMA-AD-XDM-V2_0, URL: http://www.openmobilealliance.org/

[XDM_ERELD-V1_0] “Enabler Release Document for XDM”, Version 1.0, Open Mobile Alliance™, OMA-ERELD-XDM-V1_0, URL:http://www.openmobilealliance.org/

[XDM_ERELD-V2_0] “Enabler Release Document for XDM”, Version 2.0, Open Mobile Alliance™, OMA-ERELD-XDM-V2_0, URL:http://www.openmobilealliance.org/

[XDM_Group] “Shared Group XDM Specification”, Version 1.0, Open Mobile Alliance™, OMA-TS-XDM_Shared_Group-V1_0, URL: http://www.openmobilealliance.org/

[XDM_List] “Shared List XDM Specification”, Version 2.0, Open Mobile Alliance™, OMA-TS-XDM_Shared_List-V2_0, URL: http://www.openmobilealliance.org/

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

[XDM_RD] “XML Document Management Requirements”, Version 2,0, Open Mobile Alliance™, OMA-RD-XDM-V2_0, URL:http://ww.openmobilealliance.org/

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

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

[XSD_search] “XML Schema Definition: “XDM – Search”, Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_search-V2_0, URL: http://www.openmobilealliance.org/

[XSD_xcapDir] “XML Schema Definition: “XDM – XCAP Directory”, Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_xdm_xcapDirectory-V2_0, URL: http://www.openmobilealliance.org/

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

2.2 Informative References

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

[XDM_Profile] “Shared Profile XDM Specification”, Version 1.0, Open Mobile Alliance™, OMA-TS-XDM_Shared_Profile-V1_0, 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

3GPP IMS A IP Multimedia Subsystem defined by 3GPP

Application Server A functional entity that implements the service logic for SIP Sessions (e.g. PoC Server or IM Server).

Application Unique ID A unique identifier within the namespace of application unique IDs created by this specification that differentiates XCAP Resources accessed by one application from XCAP resources accessed by another. (Source: [RFC4825])

Application Usage Detailed information on the interaction of an application with an XCAP server. (Source: [RFC4825])

Document Selector A sequence of path segments, with each segment being separated by a "/", that identify the 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])

Event header Used to indicate the which event or class of events the message contains or subscribes to, defined in [RFC3265]

Global Document A document placed under the Global Tree that applies to all users of that Application Usage.

Global Tree A URI that represents the parent for all Global Documents for a particular Application Usage within a particular XCAP Root. (Source: [RFC4825])

Group A Group is a predefined set of Users together with its policies and attributes. A Group is identified by a SIP URI.

Group Identity The SIP URI of the Group.

HTTP URI An HTTP Request-URI as defined by [RFC2616]

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])

Presence List URI A Service URI as defined by [RLS_XDM]

Primary Principal The Primary Principal is the user associated with the XCAP User Identity, which defines where the document resides. (Source: [XDM_RD])

Principal An entity that has an identity, that is capable of providing consent and other data, and to which authenticated actions are done on its behalf. Examples of principals include an individual user, a group of individuals, a corporation, service enablers/applications, system entities and other legal entities (Source: [Dict])

Public URI A Public User Identity as defined by [3GPP-TS_23.003]

Request-URI A part of the start line of a request using the SIP protocol as defined by [RFC3261]

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])

Search Request A request to perform a search operation towards XCAP Resources.

Service Provider A legal or administrative entity that provides a service to its clients or customers. Typically it is (but is not restricted to) a network operator.

SIP NOTIFY The SIP method NOTIFY as defined by [RFC3265]

SIP SUBSCRIBE The SIP method SUBSCRIBE as defined by [RFC3265]

SIP URI A communication resource as defined by [RFC3261]

Tel URI A globally unique identifier used to describe a resource identified by a telephone number as defined by [RFC3966]

URI A Uniform Resource Identifier as defined by [RFC3986]

User A User is any entity that uses the described features through the User Equipment.

User Address A User Address identifies a User. The User Address can be used by one User to request communication with other Users. If the SIP/IP Core is 3GPP/3GPP2 IMS, the User Address is a public user identity.

Users Tree A URI that represents the parent for all user documents for a particular Application Usage within a particular XCAP Root.

XCAP Client An HTTP client that understands how to follow the naming and validation constraints defined in this specification. (Source: [RFC4825]) (“This specification” refers to [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]) In this specification meaning all documents in all XDMSs accessible via the Aggregation Proxy.

XCAP Root URI An HTTP URI that represents the XCAP Root. Although a valid URI, the XCAP Root URI does not correspond to an actual resource. (Source:[RFC4825])

XCAP Server An HTTP server that understands how to follow the naming and validation constraints defined in this specification. (Source: [RFC4825])

XCAP URI An HTTP URI that represents an XCAP Resource.

XCAP User Identifier The XUI is a string, valid as a path element in an HTTP URI, that is associated with each user served by the XCAP Server. (Source: [RFC4825])

3.3 Abbreviations

ABNF	Augmented Backus-Naur Form
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
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extension
MMD	MultiMedia Domain
OMA	Open Mobile Alliance

OMNA	OMA Naming Authority
SCR	Static Conformance Requirement
SIP	Session Initiation Protocol
TLS	Transport Layer Security
UE	User Equipment
URI	Uniform Resource Identifier
XCAP	XML Configuration Access Protocol
XDM	XML Document Management
XDMC	XDM Client
XDMS	XDM Server
XML	Extensible Markup Language
XUI	XCAP User Identifier

4. Introduction

Various OMA enablers such as, Presence, Push to talk Over Cellular (PoC), Instant Messaging (IM), etc. need support for access to and manipulation of certain information that are needed by these enablers. Such information is expressed as 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 an 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 Application Usage by which XML 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 XDMC subscribes to changes to one or all documents within one Application Usage that it owns.
In order to facilitate finding certain information, limited search capabilities are defined. An XDMC can search for data stored in an XDMS by using HTTP POST requests conforming to defined Application Usages.

Basic requirements for handling of XML documents in remote domains are specified.

Common, reusable as well as enabler-specific document formats and associated Application Usages are described in separate specifications (shared ones in e.g. [XDM_List], and enabler specific ones in e.g. [IM_XDM]) that make use of the XCAP protocol specified here for their document management.

5. Common Procedures

5.1 Security Procedures

5.1.1 Authentication

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

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

If the Generic Authentication Architecture (GAA) as defined in [3GPP-TS_33.222] is not used, the XDMC and the Aggregation Proxy SHALL support and use the HTTP Digest mechanism for client authentication.

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

1. The HTTP "401 Unauthorized" error response SHALL be used;
2. The "rspauth" parameter MAY be used to provide mutual authentication;
3. The "username" parameter SHALL have the value of the XUI (i.e. the SIP URI or Tel URI) identifying the user (the public user identity);

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

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

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

5.1.2 XDM Client Identity Assertion

The XDMC SHALL deliver in an XDMC identity assertion the XDMC identity that has been successfully authenticated in a system, which SHALL thus be safely shared and used within trusted networks for authorizing the XDMC without the need for reauthentication.

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

1. SHALL insert the "X-XCAP-Asserted-Identity" header, as defined in Appendix E, to the HTTP requests after a successful HTTP Digest Authentication;
2. SHALL populate the "X-XCAP-Asserted-Identity" header with the SIP URI in quotation marks ("") provided by the "username" field in the HTTP Digest Authorization header.
3. SHALL 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 instance of the XDMC identity with which the XDMC authentication with the Aggregation Proxy was successful.

When realized in 3GPP IMS or 3GPP2 MMD networks and the GAA is present, the procedures described [3GPP-TS_24.109] SHALL be followed with the following clarifications:

The XDMC MAY insert the "X-3GPP-Intended-Identity" header as defined in [3GPP-TS_24.109] to the HTTP requests to deliver its preferred identity for XDMC identity assertion.

The Aggregation Proxy

1. SHALL act as an Authentication Proxy defined in [3GPP TS 24.109].
2. SHALL check whether an XDMC identity has been inserted in “X-3GPP-Intended-Identity” header of HTTP request;
    a. If the “X-3GPP-Intended-Identity” is included, the Aggregation Proxy SHALL check if the value in the header is allowed to be used by the authenticated identity.
    b. 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.

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

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.

5.1.3 XDM Client Identity Sharing

The XDMC authentication and identity assertion provided by the Aggregation Proxy SHALL be shared on the following reference points (see [XDM_AD]):

1. The XDM-4 reference point between the Aggregation Proxy and the Shared List XDMS, the Shared Group XDMS, the Shared Policy XDMS and the Shared Profile XDMS;
2. the XDM-6 reference point between the Aggregation Proxy and the Search Proxy;
3. the XDM-7 reference point between the Search Proxy and the Shared List XDMS, the Shared Group XDMS, the Shared Policy XDMS and the Shared Profile XDMS;
4. the Enabler Specific reference point between the Aggregation Proxy and the Enabler Specific XDMS;
5. the Enabler Specific reference point between the Search Proxy and the Enabler Specific XDMS;
6. the XDM-8 reference point between the Aggregation Proxy and the Aggregation Proxy of the Remote Network in case of a trusted remote network;
7. the XDM-9 reference point between the Search Proxy and the Aggregation Proxy of the Remote Network in case of a trusted remote network.

Further details of the security mechanisms for the above listed reference points are out of scope of this specification.

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

5.1.4 Integrity and Confidentiality Protection

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

1. The XDM-3 reference point between the XDMC and the Aggregation Proxy;
2. the XDM-5 reference point between the XDMC and the Aggregation Proxy;
3. the XDM-8 reference point between the Aggregation Proxy and the Aggregation Proxy of Remote Network;
4. the XDM-9 reference point between the Search Proxy and the Aggregation Proxy of Remote Network.

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

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

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

The XDM-8 and XDM-9 reference point SHALL protect HTTP traffic by enabling the TLS transport security mechanism or other inter-network domain security mechanism. When using TLS, the TLS resumption procedure SHALL be used as specified in [RFC2818]. When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the protection of the HTTP traffic between trusted domains MAY be implemented using Network Domain Security as defined in [3GPP-TS_33.210] and [3GPP2-S.S0086] respectively.

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

5.1.5 Authorization

For the authorization of HTTP requests, the XDMS SHALL check that the identity of the requesting XDMC has been granted access rights to perform the requested operations: The XDMS SHALL use the information in the “X-XCAP-Asserted-Identity” HTTP header provided by the Aggregation Proxy to determine the identity of the XDMC. When realized in 3GPP IMS or 3GPP2 MMD networks and the GAA is present, the identity of the requesting XDMC SHALL be obtained from the “X-3GPP-Asserted-Identity” or the “X-3GPP-Intended-Identity” HTTP header.

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

For XCAP Resources in the Users Tree, Application Usages MAY define their own authorization policies. In the absence of an Application Usage specific authorization policy, the default SHALL be as follows:

1. The Primary Principal SHALL have permission to perform all operations defined in Sections 6.1.1.*”Document Management”* and 6.1.2 *“Subscribing to changes in the XML documents*”;
2. Principals other than the Primary Principal SHALL NOT have permissions to perform operations defined in 6.1.1*”Document Management”* and 6.1.2, *“Subscribing to changes in the XML documents”*;

NOTE: Local policy may allow trusted applications to be granted some or all of the permissions defined in Sections 6.1.1*”Document Management”* and 6.1.2, *“Subscribing to changes in the XML documents”*.

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

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

5.2 Common Extensions

5.2.1 URI Lists defined in Shared List XDMS

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

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

The attribute “anchor” of the <external> element SHALL be percent-encoded as defined by the procedures in [RFC4825] section 6 before it is inserted into a document.

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

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

1) 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.
2) 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 a Document URI or Node URI that has already been resolved SHALL be ignored.

5.2.2 Authorization Rules

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

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

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

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

5.2.2.1 Structure

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

1) MAY include the <identity> condition element as defined in [RFC4745];
2) MAY include the <external-list> condition element;
3) MAY include the <anonymous-request> condition element;
4) MAY include the <other-identity> condition element;
5) MAY include the <media-list> condition element;
6) MAY include the <service-list> condition element.

NOTE: According to [RFC4745], a rule is applicable to a request only if all expressions in the conditions part of the rule evaluate to TRUE. This means that the Application Server ignores rules that contain a <conditions> child element that it does not understand or support.

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

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

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

1) an <all-media-except> element or;
2) a list of one or more media elements selected from the list of possible media elements below.

List of possible media elements:

1) The <message-session> media element indicating session based messaging as defined in [IETF-MSRP];
2) The <pager-mode-message> media element indicating pager mode message requests as defined in [RFC3428];
3) The <file-transfer> media element indicating file transfer as defined in [IM_TS];
4) The <audio> media element indicating a streaming media type as defined in [RFC3840];
5) The <video> media element indicating a streaming media type as defined in [RFC3840];
6) The <poc-speech> media element indicating a PoC speech media type as defined in [PoC_CP];
7) Any elements from any other namespaces defining a media element.

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

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

1) MAY include the <full-duplex> element indicating that media can be exchanged in both directions simultaneously;
2) MAY include the <half-duplex> element indicating that media can be exchanged in only one direction at a time.

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

1) a list of one or more <service> elements or;
2) an <all-services-except> element.

The <all-services-except> element MAY include a list of one or more <service> elements.

The <service> element:

1) MAY include an attribute "enabler" including a string defining an Enabler ;
2) MAY include attributes from any other namespace for the purpose of extensibility to have other service identification definitions than using the "enabler" attribute;
3) MAY include any other elements from any other namespaces for purpose of extensibility to have other service identification definitions than using attributes.

5.2.2.2 Data Semantics

If present in any rule, the <external-list> element SHALL match those identities that are part of a URI List.

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

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

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

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

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

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

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

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

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

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

The <service> element SHALL be used to define a service.

The attribute “enabler” SHALL specify the enabler defining the service. The “enabler” attribute SHALL be used only for Open Mobile Alliance defined Enablers. The Enabler SHALL use the OMNA registered Enabler XML schema domain token as the value of the “enabler” attribute (e.g. “poc” for the Push to talk over Cellular Enabler and “im” for the IM SIMPLE Enabler).

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

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

5.2.2.3 XML Schema

The authorization policy document SHALL conform to the XML schema described in [RFC4745] Section 13 with the extensions described in [XSD_commPol], in [XSD_ext], in XDM unique extensions to [RFC4745] and in enabler unique extensions to [RFC4745].

5.2.2.4 Combining Permissions

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

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

2. Those rules matching the URI value against the <identity> element SHALL take precedence over those rules based on matching it against an <external-list> or an <other-identity> element. That is, if there are applicable rules based on <identity> matches, only those will be used for the evaluation of the combined permission.
3. Those rules containing an <other-identity> element SHALL be used for the evaluation of the combined permission only if there are no other matching rules.

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

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

5.2.3 Detailed Conflict Reports

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

The XDMC 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.

5.2.3.1 Structure

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

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

The <local-constraint-failure> element:

1) MAY include the “phrase” attribute;

2) MAY include one or more <alt-value> elements with the mandatory “field” attribute, providing one or more alternate values for the element or attribute indicated by the “field” attribute;

3) MAY include one or more <description> elements with an optional “lang” attribute, providing one or more descriptions documenting the local constraint failure, possibly in different languages.

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

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

5.2.3.2 XML Schema

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

5.3 Common Application Usages

5.3.1 XCAP Server Capabilities

The XCAP Server Capabilities Application Usage allows an XDMC to determine what extensions, Application Usages, or namespaces an XDMS supports before making a request. The XDMS SHALL support the XCAP Server Capabilities Application Usage, as defined in [RFC4825] “*XCAP Server Capabilities*”. The XDMC MAY support the XCAP Server Capabilities Application Usage.

5.3.2 XML Documents Directory

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

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

The XDMS SHALL support the XML Documents Directory Application Usage, as defined below. The XDMC MAY support the XML Documents Directory Application Usage.

The 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 XDMS supports and for which there are documents in the Users Tree 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 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 XDMC 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 XDMC 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 XDMC 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 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 XDMSs 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” document back to the XDMC. The content type of the returned “directory.xml” document SHALL be “application/vnd.oma.xcap-directory+xml” as defined in section 5.3.2.2.

When the Aggregation Proxy receives an HTTP “200 OK” response with XML content it SHALL include a <folder> element in the composite “directory.xml” document with the content.

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” document.
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.

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.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 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 XDMC.

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

5.3.2.1 Application Unique ID

The AUID SHALL be "org.openmobilealliance.xcap-directory" AUID.

5.3.2.2 MIME Type

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

5.3.2.3 Default Namespace

The default namespace SHALL be "urn:oma:xml:xdm:xcap-directory"

5.3.2.4 XML Schema

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

5.3.2.5 Additional Constraints

None.

5.3.2.6 Data Semantics

See section 5.3.2.

5.3.2.7 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".

5.3.2.8 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 1: This does not imply that the server must actually store this "directory" document. All that is required is that the XDMS is 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 2: This implies that conditional operations are not supported against the "directory" document. The XDMC should always refresh any cached copy.

5.3.2.9 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 5.1.5.

5.4 Common Content Types

5.4.1 Search Document

The XDMC SHALL support the Search document if the XDMC supports the search feature as described in subclause 6.1.3 "*Searching forData in XML Documents*".

The XDMS SHALL support the Search document if the XDMS supports the search feature as described in subclause 6.2.3 "*Searching forData in XML Documents*".

5.4.1.1 Mime Type

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

5.4.1.2 XML Schema

The Search document SHALL conform to the XML schema described in [XSD_SEARCH].

5.4.1.3 Structure

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

The <search> element:

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

- MAY include the "max-results" attribute with the positive integer value indicating the maximum number of results requested by the XDMC.
- MAY include any other attribute for the purposes of extensibility

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

The XQuery expression SHALL include one input function – collection. The collection of the data to be searched is created as a set of all documents stored in the Users Tree of an appropriate Application Usage, as a set of all documents in the particular User's home directory of an appropriate Application Usage, or as a particular document in the particular User's home directory of an appropriate Application Usage. As such, the parameter of the collection function SHALL be either the string of "[AUID]/users/", that of "[AUID]/users/[XUI]", or that of "[AUID]/users/[XUI]/<document_name>". For example,

collection("org.openmobilealliance.user-profile/users/")

represents all User Profile documents stored in the Users Tree on Shared Profile XDMS to which the Search Request is targeted;

collection("org.openmobilealliance.groups/users/sip:joebloggs@example.com")

represents all Group documents stored in the home directory of "sip:joebloggs@example.com" on Shared Group XDMS to which the Search Request is targeted;

collection ("resource-lists/users/sip:joebloggs@example.com/index")

represents the URI List document with the name "index" stored in the home directory of "sip:joebloggs@example.com" on Shared List XDMS to which the Search Request is targeted.

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

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

In addition, each Application Usage that supports the Search feature SHALL define one or more basic XQuery expressions that are supported by the Application Usage. Such basic XQuery expressions allows the Application Usage to restrict the data that can be searched and also restrict the results provided to the XDMC.

5.5 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 5.3.1*"XCAP Server Capabilities"*) describing the Application Usages supported by an XDMS.

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

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

6. Description of Procedures at XDM Functional Entities

6.1 Procedures at the XDM Client

An XDM Client (XDMC) is an entity that accesses an 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].

The XDMC SHALL support the following security functions:

1. authentication function described in section 5.1.1 *"Authentication"*;
2. client identity assertion function described in section 5.1.2 *"XDM Client Identity Assertion"*;
3. HTTP traffic protection function described in section 5.1.4 *"Integrity and Confidentiality Protection"*.

The XDMC SHALL, when generating HTTP requests, include "User-Agent" HTTP header as defined in [RFC2616] with the value set to "XDM-client/OMA2.0" to indicate that the XDMC is compliant with this specification.

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

6.1.1 Document Management

The XDMC SHALL support document management as described in this subsection.

6.1.1.1 XDM URI Construction

An XCAP URI represents an XML document , an element within an XML document or an attribute of an element with an XML document stored in an XDMS. The rules for constructing such XCAP URIs SHALL follow the rules described in [RFC4825] Section 6 with the clarifications given in this sub-clause.

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

The XCAP Root URI SHALL include host address of the Aggregation Proxy in the XDMC's home domain. The XDMC that resides in an UE SHALL use the XCAP Root URI provisioned to the XDMC as described in Appendix D "XDMC Provisioning". The XDMC that resides in an AS 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 XDMC SHALL compare whether the XCAP Root URI of any XCAP Resource to be accessed is the same as the XCAP Root URI that has been provisioned or preconfigured. If the validation fails, the XDMC SHALL replace the XCAP Root URI with the provisioned/preconfigured XCAP Root URI.

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

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

NOTE 3: If the User has multiple User Addresses available, each single User Address constitutes an independent and unrelated XUI. For example, if a user has two UserAddresses 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 User Addresses of a user, allowing e.g. interchangeable XUI usage, is out of the scope of this specification.

If the user has both a Tel URI and its associated SIP URI then the XDMC SHALL use the SIP URI in preference to the Tel URI as an XUI. Here the term 'associated' means that the Tel URI can be translated to the SIP URI and vice versa, for interchangeable usage in the SIP / IP Core. Both the translation and the interchangeable usage are out of the scope of this specification.

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

6.1.1.2 XDM Operations

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

The XDMC SHALL construct the Request-URI based on its knowledge of the Application Usage governing that XML document.

An XDMC MAY implement the conditional operations of [RFC4825] section 7.11.

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

6.1.1.2.1 Create or Replace a Document

Creating or replacing an 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. XDMCs 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.

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. XDMCs 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 or attribute in an XML document SHALL follow the procedures described in [RFC4825] Section 7.10

6.1.2 Subscribing to changes in the XML documents

The XDMC MAY support subscription to changes in XML documents as described in this subsection.

6.1.2.1 Initial subscription

If the XDMC subscribes to changes in XML documents, then it SHALL be done by sending a SIP SUBSCRIBE request according to [RFC3265] and [IETF-XCAP_Diff_Event] with the clarifications given in the sub-clause 6.1.2.1.1 for a XDMC resided in an Application Server and sub-clause 6.1.2.1.2 for a XDMC resided in an UE.

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

NOTE: The XDMC 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.

6.1.2.1.1 XDMC residing in an Application Server

When the XDMC resides in an Application Server:

1. SHALL set the Request-URI to the public SIP URI or TEL URI identifying the Primary Principal, or to the SIP URI or TEL URI identifying the service instance (e.g. Group Identity, or Presence List URI);
2. SHALL include the Document Selector of the document to be watched in the “path” Event header parameter in case a specific document is to be watched.
3. SHALL include the part of the Document Selector that identifies the folder (in form of “/AUID/users/XUI/”) to be watched in the “path” Event header parameter in case of all documents under the AUID owned by the User identified by the XUI are to be watched;
4. SHALL include all Document Selectors of the folders or the documents from a single AUID and owned by a single Primary Principal only in case that more documents or folders are subscribed in a single subscription;
5. SHALL in case the service instance SIP URI or TEL URI is set as Request-URI of the SIP SUBSCRIBE request, then set the path Event header parameter to specify the relevant document or the relevant element inside the document stored in the “global” tree for this service instance SIP URI or TEL URI.

   NOTE: For example, if the Request-URI SIP URI identifying the service instance is “sip:my_friends@example.com” stored in Shared Group XDMS, the “path” parameter has to be set to “path= “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 SHALL include the value "application/xcap-diff+xml" to indicate support for partial XML updates described in [IETF-XCAP_Diff];
7. SHALL send the SIP SUBSCRIBE request towards the SIP/IP Core according to the procedures of the SIP/IP Core.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD an AS acting as the XDMC SHALL use 3GPP IMS or 3GPP2 MMD requirements respectively, mechanisms and procedures as defined in chapter 5.7.3 [3GPP-TS_24.229] /[3GPP2-X.S0013-004] with the clarifications given in the respective sub clauses.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD, and the XDMC 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] section 5.7.3 and setting its public SIP URI or the Public SIP URI of the user on which the Application Server is acting on behalf on in the "P-Asserted-Identity" header.

6.1.2.1.2 XDMC residing in an UE

When the XDMC resides in the UE:

1. SHALL set the Request-URI to the public SIP URI or TEL URI identifying the Primary Principal that it is subscribing to;
2. SHALL in case a specific document is to be watched, then set the "path" Event header parameter to the Document Selector of the relevant document;

   NOTE: The mechanism used by the XDMC to retrieve the SIP URI of the Primary Principal and the Document Selector of the document to be watched is out of scope of the present specification.
3. SHALL include the part of the Document Selector that identifies the folder (in form of "/AUID/users/XUI/") to be watched in the "path" Event header parameter in case of all documents under the AUID owned by the User identified by the XUI are to be watched;
4. SHALL include all Document Selectors of the folders or the documents from a single AUID and owned by a single Primary Principal only in case that more documents or folders are subscribed in a single subscription;
5. SHALL include an Accept header to indicate acceptable content-type for notifications. The Accept header SHALL include the value "application/xcap-diff+xml" to indicate support for partial XML updates described in [IETF-XCAP_Diff];
6. SHALL send the SIP SUBSCRIBE request towards the SIP/IP Core according to the procedures of the SIP/IP Core.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD, a UE acting as the XDMC SHALL use 3GPP IMS or 3GPP2 MMD requirements respectively, mechanisms and procedures as defined in chapter 5.1 in [3GPP-TS_24.229] / [3GPP2-X.S0013-004].

6.1.2.2 SIP NOTIFY processing

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

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

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD, the XDMC SHALL use 3GPP IMS or 3GPP2 MMD requirements respectively, mechanisms and procedures as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] with the clarifications given in this sub-clause.

6.1.3 Searching for Data in XML Documents

The XDMC MAY support searching for data in XML documents using Limited XQuery over HTTP as described in this subsection.

When performing a search operation, the XDMC SHALL generate the Search Request by using HTTP POST request containing a Search document as defined in chapter 5.4.1 "*Search Document*".

The HTTP Request-URI for the Search Request SHALL be constructed as http://[XCAP Root URI]/org.openmobilealliance.search. For routing purposes, the HTTP Request-URI of the Search Request SHALL include the mandatory query parameter of "target" and whose value is equal to the parameter of the collection input function of the XQuery request in the Search document that identifies the document(s) to be searched as described in section 5.4.1 *"Search Document"*. When the search target is the set of all documents stored in the Users Tree of an appropriate Application Usage, there needs to specify which domain is to be searched for. For the identification of target search domain, the HTTP Request-URI of the Search Request MAY include the optional query parameter of "domain" and whose value includes 'home' to request home domain search, 'all' to request to expand the search to all possible remote domains, or target domain names to request the particular domain to be searched. Multiple values are separated using the percent encoded whitespace (i.e., "%20") as specified in [RFC3986]. The default interpretation in the absence of "domain" query parameter SHALL be home domain search.

When using the "target" and "domain" query parameter, the HTTP Request-URI for the Search Request SHALL be constructed as http://[XCAP Root URI]/org.openmobilealliance.search?target=[value of collection input function]&domain=[home, all, or target domains].

> Example: http://xcap.example.com/org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/&domain=all

> Example: http://xcap.example.com/org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/&domain=home%20example.com%20example2.com

The Search Request SHALL include the XML body of the content type "application/vnd.oma.search +xml" as defined in chapter 5.4.1.1*"MIME Type"*.

The XQuery expression in the Search Request SHALL conform to the constraints as defined by the target Application Usage to be searched for.

The XDMC MAY limit the number of Search results using the optional "max-results" attribute of the <search> element in the Search document.

6.2 Procedures at the XDM Server

An XDMS is a HTTP origin server that manipulates XCAP Resources according to the conventions described in [RFC4825], and processes Search Requests.

An XDMS SHALL authorize the requests as described in section 5.1.5 *"Authorization"*.

An XDMS receiving an HTTP POST request containing an HTTP Request-URI of the form http://[XCAP Root URI]/ org.openmobilealliance.search SHALL be processed as described in chapter 6.2.3. All other HTTP POST requests SHALL be rejected with an HTTP "405 Method not allowed" response.

An XDMS receiving an XCAP Request SHALL process the request as described in section 6.2.1 "*Document Management*".

The XDMS MAY, when generating HTTP responses towards XDMC, include "Server" HTTP header as defined in [RFC2616] with the value set to "XDM-serv/OMA2.0" to indicate that the XDMS is compliant with this specification.

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

6.2.1 Document Management

The XDMS SHALL support document management as described in this subsection.

An XDMS SHALL conform to [RFC4825] section 8.5 for the management of Etags.

An XDMS SHALL implement the conditional operations of [RFC4825] section 7.11.

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

6.2.1.1 PUT handling

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

Additional validation constraints might be applied which may result in a HTTP “409 Conflict” error response. An HTTP “409 Conflict” error response SHALL include a document in the HTTP body that conforms to that defined in [RFC4825] Section 11 and the extensions defined in this specification Section 5.2.3 *“Detailed Conflict Reports*”.
For additional details of the handling of those, see [RFC4825] Section 8.2.5 and this specification Section 5.2.3 *“Detailed Conflict Reports”*. Other specifications MAY define the value of the “phrase” attribute, which contains text for rendering to a human user, that is optionally present in an error element identifying an error condition.

6.2.1.2 GET handling

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

6.2.1.3 DELETE handling

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

6.2.2 Subscriptions to changes in the XML documents

The XDMS MAY support subscription to changes in XML documents as described in this subsection. If subscription to changes is not supported the XDMS SHALL return appropriate error response upon reception of a SIP SUBSCRIBE request for the “xcap-diff” event defined in [IETF-XCAP_Diff_Event].

6.2.2.1 Initial subscription

Upon receiving a SIP SUBSCRIBE request for the “xcap-diff” event defined in [IETF-XCAP_Diff_Event] the XDMS:

1. SHALL perform necessary checks on the XCAP resources listed in the “path” Event header parameter. In case that any resource is not recognized as resource from appropriate Application usage, the XDMS SHALL return the SIP “404 Not found” error response;
2. SHALL perform the necessary authorization checks on the originator. When the SIP/IP Core corresponds to 3GPP IMS or 3GPP2 MMD the XDMS SHALL use the "P-Asserted-Identity" as defined in [3GPP-TS_24.229]/[3GPP2-X.S0013-004] to ensure that this particular XDMC is authorized to track the document changes. If the authorization check fails, the XDMS SHALL return the SIP "403 Forbidden" error response;
    a. For documents in the Users tree, by default the Primary Principal of the document SHALL be authorized to subscribe to the “xcap-diff” event package;
    b. For documents in the Global Tree, other principals (e.g. XDMCs residing in the UE and Application Servers) MAY be authorised to subscribe based on local policy or other enabler-specific policy;
    c. Additional authorization policy MAY be defined for an Application Usage in the respective application-specific XDM Technical Specifications.
3. SHALL create a subscription to changes of XML data identified by Event header parameters as described in [IETF-XCAP_Diff_Event];

4. SHALL send a SIP "200 OK" in accordance with [RFC3265], [IETF-XCAP_Diff_Event], and the procedures of the SIP/IP Core;
5. SHALL generate and send an initial SIP NOTIFY request as specified in sub-clause 6.2.2.2 "*Generating a SIP NOTIFY request*".

When a change in the subscribed document occurs, the XDMS SHOULD generate and send a SIP NOTIFY request as specified in sub-clause 6.2.2.2 "*Generating a SIP NOTIFY request*".

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD, the XDMS SHALL use 3GPP IMS or 3GPP2 MMD requirements respectively, mechanisms and procedures as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] with the clarifications given in this sub-clause.

6.2.2.2 Generating a SIP NOTIFY request

If the "xcap-diff" event is supported the XDMS SHALL generate a SIP NOTIFY request as described in the [RFC3265] and [IETF-XCAP_Diff_Event] with the clarifications given in this sub-clause.

The XDMS

1. SHALL include an "application/xcap-diff+xml" body as defined in [IETF-XCAP_Diff];
2. SHALL send the SIP 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 SIP NOTIFY request SHALL be handled in accordance with [RFC3265], [IETF-XCAP_Diff_Event], and the procedures of the SIP/IP Core.

When the SIP/IP Core corresponds with 3GPP IMS or 3GPP2 MMD, the XDMS SHALL use 3GPP IMS or 3GPP2 MMD requirements respectively, mechanisms and procedures as defined in [3GPP-TS_24.229] / [3GPP2-X.S0013-004] with the clarifications given in this sub-clause.

6.2.3 Searching for Data in XML Documents

The XDMS MAY support searching for data in XML documents using Limited XQuery over HTTP as described in this subsection.

The Search Request SHALL contain a Search document as defined in chapter 5.4.1 "*Search Document*".

Upon receiving the Search Request, the XDMS:

1. SHALL verify whether the Search document included in the body of the Search Request conforms to the structure defined in chapter 5.4.1.3 "*Structure*";
2. SHALL get the AUID from the "collection" input function of the XQuery and based on this AUID validate the XQuery expression included in the body of the Search Request against the XQuery restrictions as defined by the corresponding target Application Usage of the XDMS.

When the XQuery expression fits to the defined restrictions, the XDMS SHALL execute the query over all XML documents stored in the Users Tree of the corresponding Application Usage included in the "collection" input function of the XQuery request. The XDMS SHALL based on the results of the query generate a response.

In case that "max-results" attribute is included in the Search Request, the XDMS SHALL include in the response only the number of results of the Search up to and including the value specified in the "max-results" attribute.

The XDMS MAY restrict the number of results of the Search based on local policy.

In addition, each Application Usage that supports the Search feature SHALL define one or more basic XQuery expressions that are supported by the Application Usage. Such basic XQuery expressions allows the Application Usage to restrict the data that can be searched and also restrict the results provided to the XDMC.

Each request violating defined restrictions SHALL be responded with HTTP "409 Conflict" error response with the <constraint-failure> error condition element defined in [RFC4825]. If the basic XQuery expressions as defined by the corresponding Application Usage do not allow:

- the Search operation as requested in the included XQuery expression, the "phrase" attribute, if it is included, SHOULD be set to "Search request not allowed";
- the types of Search Result as requested in the included XQuery expression, the "phrase" attribute, if it is included, SHOULD be set to "Search result types not allowed".

6.3 Procedures at the Aggregation Proxy

The Aggregation Proxy is the contact point for XDMC implemented in a UE to access XCAP Resources stored in XDMS. The Aggregation Proxy is also the contact point for XDMC implemented in an Enabler Specific Server to access XCAP resources stored in an XDMS of a remote network.

The Aggregation Proxy:

1. SHALL act as an HTTP Proxy defined in [RFC2616] and be configured as an HTTP Reverse Proxy [RFC 3040];
2. SHALL, upon receiving an XCAP or HTTP request targeted to the Aggregation Proxy, authenticate the originating XDMC implemented in a UE as specified in the subclause 5.1.1 *"Authentication"*;
3. SHALL, upon the successful authentication, assert the identity of the originating XDMC as described in the subclause 5.1.2 *"XDM Client Identity Assertion"*;
4. SHALL forward the requests as described in the subclause 6.3.1 *"HTTP Request Handling"*.

6.3.1 HTTP Request Handling

6.3.1.1 General

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

1) SHALL check whether the domain of the XUI matches with the domain of the Aggregation Proxy;

2) SHALL perform one of the following:

   a) If the domain of the XUI matches with the domain of the Aggregation Proxy, forward the XCAP request to the corresponding XDMS based on the AUID in the HTTP Request-URI; or

   b) If the domain of the XUI does not match the domain of the Aggregation Proxy and the identified domain is trusted, forward the XCAP request to the domain of the XUI's Aggregation Proxy of the remote network with the XCAP Root URI set to the XCAP Root URI of the domain of the XUI.

Upon receiving an HTTP POST request containing an HTTP Request-URI of the form http://[XCAP Root URI]/ org.openmobilealliance.search, the Aggregation Proxy SHALL forward the HTTP POST request to the Search Proxy.

Upon receiving the responses to the XCAP Request, the Aggregation Proxy SHALL aggregate and forward responses back to the XDMC.

Upon receiving the responses to the HTTP POST request, the Aggregation Proxy SHALL forward the responses back to the XDMC.

The Aggregation Proxy MAY, when generating HTTP responses to XDMC (e.g., when challenging the XDMC for authentication), include the "Server" HTTP header [RFC2616] with the value set to "XDM-proxy/OMA2.0" to indicate that the Aggregation Proxy is compliant with this specification.

NOTE: It is out of scope of this specification how the Aggregation Proxy to handle the received "Server" HTTP headers included in the received HTTP responses, when aggregating and forwarding those HTTP responses to XDMC.

The Aggregation Proxy SHALL protect the HTTP traffic between the XDMC and the Aggregation Proxy and between the Aggregation Proxy and the Aggregation Proxy of Remote Network as specified in section 5.1.4 *"Integrity and Confidentiality Protection"*.

6.3.1.2 Error Cases

If the Aggregation Proxy receives an XCAP Request where the domain of the XUI does not match with the domain of the Aggregation Proxy and the domain of the XUI is not a domain of a trusted remote network, the Aggregation Proxy SHALL reject the request with an HTTP "403 Forbidden" error response.

If the Aggregation Proxy receives an HTTP request targeted at an XCAP Resource whose Application Usage is not recognized or understood, the Aggregation Proxy or XDMS SHALL reject the request with an HTTP "404 Not Found" error response

Upon receiving an HTTP request containing an HTTP Request-URI of the form http://[XCAP Root URI]/ org.openmobilealliance.search where HTTP Method is different from POST, the Aggregation Proxy SHALL reject the request with an HTTP "405 Method not allowed" error response.

Upon receiving an HTTP POST request that does not contain an HTTP Request-URI of the form http://[XCAP Root URI]/ org.openmobilealliance.search, the Aggregation Proxy SHALL reject the request with an HTTP "405 Method not allowed" error response.

6.3.1.3 XCAP Server Capabilities retrieval

Upon receiving an XCAP GET request for the "xcap-caps" AUID (described in section 5.3.1), the Aggregation Proxy:

1. SHALL act as an HTTP Reverse Proxy;
2. SHALL obtain XCAP Server Capabilities from all XDMSs that serve the request originator. To perform this operation the Aggregation Proxy SHALL forward the XCAP request to all XDMSs that serve the request originator and if the target XDMSs respond with HTTP "200 OK" response, collect the <auid>, <extension> 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 any other XCAP requests than XCAP GET for an "xcap-caps" document, the Aggregation Proxy SHALL respond with an HTTP "405 Method Not Allowed" response.

6.3.1.4 XCAP Directory retrieval

Upon receiving an XCAP GET request for the "org.openmobilealliance.xcap-directory" AUID (described in section 5.3.2), the Aggregation Proxy:

1. SHALL act as an HTTP Reverse Proxy;
2. SHALL obtain the requested XCAP Directory from the corresponding XDMSs that serve the request originator. To perform this operation the Aggregation Proxy:
    a. SHALL forward the XCAP request either to all XDMSs that serve the request originator if the request is targeted at the directory document, or to the XDMS serving the specific AUID if the request is targeted at a specific AUID as specified by the Node Selector;

   b. SHALL, if the target XDMSs responded with HTTP "200 OK" response, collect the <folder> elements.

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 any other XCAP requests for an "org.openmobilealliance.xcap-directory" document than XCAP GET, the Aggregation Proxy SHALL respond with an HTTP "405 Method Not Allowed" response.

6.3.2 Compression

The Aggregation Proxy MAY support compression using content encoding.

If the Aggregation Proxy supports compression it SHALL follow the procedures defined in [RFC2616].

6.4 Procedures at the Search Proxy

The Search Proxy performs request forwarding / response aggregation procedure for HTTP traffic carrying Search Requests / Search Responses as described in this subclause.

The Search Proxy SHALL share the XDMC authentication and its identity assertion provided by the Aggregation Proxy as described in section 5.1.3 *"Authenticated Identity Sharing."*

The Search Proxy SHALL protect the HTTP traffics as described in section 5.1.4 *"Integrity and Confidentiality Protection"*.

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

6.4.1 Search Request Forwarding

Upon receiving the HTTP Search Request, the Search Proxy:

1. SHALL get the AUID from the "target" query parameter included in the HTTP URI;
2. SHALL get the target search domain information from the optional "domain" query parameter included in the HTTP URI. If the "domain" query parameter does not exist, the target search domain SHALL be home domain only;
3. SHALL forward the Search Request to appropriate XDMS based on the AUID if the target search domain is home domain, or to appropriate Aggregation Proxies of remote networks with the XCAP Root URI set to the XCAP Root URI of the remote network if inter domain search is requested and supported. When forwarding, the "domain" query parameter, if it exists, it SHALL be removed from the HTTP URI.

6.4.1.1 Error Cases

If the Search Proxy receives an HTTP Search Request where:

- The HTTP Method is different from POST, the Search Proxy SHALL reject the request with an HTTP "405 Method Not Allowed" error response.
- The value in "target" query parameter is not recognized as known Application Usage, the Search Proxy SHALL reject the request with an HTTP "409 Conflict" error response with the <constraint-failure> error condition element defined in [RFC4825]. If included, the "phrase" attribute SHOULD be set to "Search not supported for indicated Application Usage".
- The value in "domain" query parameter is not recognized as a known user domain, the Search Proxy SHALL reject the request with an HTTP "409 Conflict" error response with the <constraint-failure> error condition element defined in [RFC4825]. If included, the "phrase" attribute SHOULD be set to "Search towards indicated domain not supported".

6.4.2 Search Response Aggregation

Upon receiving the responses for the Search Requests, the Search Proxy:

- SHALL forward the response back to the originator in case that the corresponding Search Request was forwarded to single XMDS.
- SHALL aggregate and forward responses back to the originator in case that the corresponding Search Request was forwarded to multiple XDMSs in different domains.

When the responses from multiple XDMSs in different domains are aggregated, the Search Proxy SHALL ensure that the total amount of results do not exceed the value of "max-results" attribute in corresponding Search Request if included. The mechanism of the selection of the subset of results in case that total amount of aggregated results is higher than requested by the XDMC is out of scope of this specification.

NOTE: It is out of scope of this specification how the Search Proxy to handle the received "Server" HTTP headers included in the received HTTP responses, when aggregating and forwarding those HTTP responses towards XDMC.

6.5 Procedures at the Aggregation Proxy of Remote Network

The Aggregation Proxy of Remote Network SHALL act as an HTTP Proxy defined in [RFC2616] with the following clarifications. Upon receiving XCAP requests or Search Requests from remote networks, the Aggregation Proxy of Remote Network:

1. SHALL be configured as an HTTP Reverse Proxy (see [RFC3040]);
2. SHALL share the XDMC authentication and its identity assertion with the originating network as described in section 5.1.3 *"XDM Client Identity Sharing"* if they are XCAP requests or Search Requests from originating networks of trusted domains;
3. SHALL verify whether the Aggregation Proxy of Remote Network is responsible for the target domain of the received XCAP requests or Search Requests;
4. SHALL forward XCAP requests to the corresponding XDMSs that store the targeted XML documents and forward Search Requests to the Search Proxy;
5. SHALL aggregate XCAP responses from XDMSs as appropriate, then forward those back to the originating network or SHALL forward Search Request responses from the Search Proxy back to the originating network ;
6. SHALL protect the HTTP traffics as described in section 5.1.4 *"Integrity and Confidentiality Protection"*.

If the Aggregation Proxy of Remote Network receives an XCAP request or a Search Request for a target domain that it is not responsible for, the Aggregation Proxy of Remote Network SHALL reject the request with an HTTP "404 Not Found" error response.

If it receives an XCAP request or a Search Request from an untrusted remote network, the Aggregation Proxy of Remote Network SHALL reject the request with an HTTP "403 Forbidden" error response.

Appendix A. Change History (Informative)

A.1 Approved Version History

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

A.2 Draft/Candidate Version 2.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-TS-XDM_Core-V2_0	12 Apr 2006	6.2.2.1	Incorporated OMA-PAG-2006-0193
	26 Jun 2006	6.2.2.2 2.1, 6.1.2.1 B.5, 3.2 6.1.1.2.10 6.2.1, 6.6.3, 6.7.2 B.1, B.2, B.3, B.4, C.2 6.4.1, A, B 6.8, 5, 6.1, 6.2, 6.9	Incorporated CRs: OMA-PAG-2006-0224R01 OMA-PAG-2006-0233 OMA-PAG-2006-0234 OMA-PAG-2006-0243R01 OMA-PAG-2006-0254 OMA-PAG-2006-0300R01 OMA-PAG-2006-0334R03 Document history moved as Appendix A.
	02 Aug 2006	2, 6.1.2.1, 6.2.2.2	Incorporated CRs: OMA-PAG-2006-0389R02
	01 Sep 2006	6.3.3.4 6.7.3, C.5, C.6, 6.9, 2, D, D.1, D.3 6.1.2, 6.2.2,	Incorporated CRs: OMA-PAG-2006-0422R02 OMA-PAG-2006-0423R02 OMA-PAG-2006-0442R01 OMA-PAG-2006-0461R01 OMA-PAG-2006-0469R01
	10 Nov 2006	2.1, 6.4.1, 6.6.2.1, 6.6.2.2 6.6.3.2, 6.7.2.2 6.7.2.3 6.7.2.4 6.7.3.4 C.3	Incorporated CRs: OMA-PAG-2006-0518R02 OMA-PAG-2006-0536 OMA-PAG-2006-0552R03 OMA-PAG-2006-0553 OMA-PAG-2006-0560 OMA-PAG-2006-0594
	24 Nov 2006	C.2, C.3, C.4, C.5, 6.6.1, 6.6.2.1, 6.1.1.2.6, 6.1.1.2.9, 6.1.1.1, 6.4.1, 6.4.3, C.1, C.2, C.4, C.5, B	Incorporated CRs: OMA-PAG-2006-0671 OMA-PAG-2006-0672 OMA-PAG-2006-0674R01 OMA-PAG-2006-0676 OMA-PAG-2006-0677 OMA-PAG-2006-0678 OMA-PAG-2006-0692R02 OMA-PAG-2006-0710 OMA-PAG-2006-0715R01 OMA-PAG-2006-0775
	25 Nov 2006	C.2, C.3	Incorporated CR: OMA-PAG-2006-0611 OMA-PAG-2006-0612R01

Document Identifier	Date	Sections	Description
	18 Dec 2006	5.3, 5.4, 5.5, 5.6, 6.2.1.1, 6.4.3, 6.7.3.7, 6.9.1 App E,	Incorporated CRs: OMA-PAG-2006-0585R01 OMA-PAG-2006-0694 OMA-PAG-2006-0737 OMA-PAG-2006-0831R01 OMA-PAG-2006-0832 OMA-PAG-2006-0835R01
	19 Dec 2006	6.1.1.1, 6.1.3, 6.2.3, 6.3.3.1, 6.3.2.2, 6.7.3.5, 6.7.3.6, 6.8, 6.9.1, 6.10, 6.11,	Incorporated CRs: OMA-PAG-2006-0836R03 OMA-PAG-2006-0851R01 OMA-PAG-2006-0852R03 OMA-PAG-2006-0854R02 OMA-PAG-2006-0855R01
	5 Feb 2007	Most chapters	Incorporated CR: OMA-PAG-2007-0051R02
	6 Feb 2007	3.3.1, 4, 6.1, 6.1.1, 6.1.1.2.10 6.1.2, 6.1.3, 6.2.1, 6.2.2, 6.2.3, 6.3, 6.3.2, 6.3.3.1, 6.3.3.2, 6.6.1 6.6.2.1 6.6.2.3 6.6.3.2 6.7.1, 6.7.2, 6.7.3, 6.7.3.8, 6.7.3.9, 6.9, 6.110.1.1, 6.11, 6.11.1, 6.11.1.1, 6.11.2, B.3, B.4, C.6 C.5,	Incorporated CRs: OMA-PAG-2007-0027R01 OMA-PAG-2007-0030R01 OMA-PAG-2007-0031R01 OMA-PAG-2007-0032R02 OMA-PAG-2007-0033 OMA-PAG-2007-0036R01 OMA-PAG-2007-0052R01 OMA-PAG-2007-0053R02 OMA-PAG-2007-0050R01 OMA-PAG-2007-0059 OMA-PAG-2007-0067 OMA-PAG-2007-0069 OMA-PAG-2007-0068R01

Document Identifier	Date	Sections	Description
	15 Feb 2007	2.1, D, 6.1.1.1, 6.1.3, 6.3.1, 6.3.3.5, 6.4.1, 6.7.3.1, 6.7.3.5, 6.7.3.6, 6.7.3.7	Incorporated CRs: OMA-PAG-2007-0043R02 OMA-PAG-2007-0054R01 OMA-PAG-2007-0058R01 OMA-PAG-2007-0066R01 OMA-PAG-2007-0071R01 OMA-PAG-2007-0088 OMA-PAG-2007-0089R01 OMA-PAG-2007-0107
	02 Mar 2007	2.1, 6.7.2.1 6.7.2.2 6.7.2.9 C.1	Incorporated CRs: OMA-PAG-2007-00120 OMA-PAG-2007-00121 OMA-PAG-2007-00122 OMA-PAG-2007-00124 OMA-PAG-2007-00125 OMA-PAG-2007-00126
	06 Mar 2007	3.2, 6.2.3, 6.1.3, 6.3, 6.3.3.5, 6.7.3.9, 6.9.1, 6.11,	Incorporated CRs: OMA-PAG-2007-0029R02 OMA-PAG-2007-0123R02
	21 Mar 2007	6.10	Incorporated CR: OMA-PAG-2007-0153R02
	03 Apr 2007	2.1, 3.2, 6.1, 6.1.3, 6.2, 6.2.2, 6.3, 6.4, 6.5, 6.9.1, 6.9.1.1, 6.10, 6.11, B1, B2, B3, B4, B5, B6,	Incorporated CRs: OMA-PAG-2007-0127R04 OMA-PAG-2007-0144R02 OMA-PAG-2007-0154R01 OMA-PAG-2007-0155R01 OMA-PAG-2007-0156R03 OMA-PAG-2007-0169

Document Identifier	Date	Sections	Description
	23 Apr 2007	2.1, 2.2, 4, 6.1, 6.1.1 6.1.1.2 6.1.2.1, 6.1.2.2, 6.1.1.2.9, 6.1.3, 6.2, 6.2.1 6.2.1.1 6.2.1.2 6.2.1.3 6.2.1.4, 6.2.3, 6.3, 6.3.1.1, 6.3.1.2, 6.3.3.5, 6.4.3, 6.4.4, 6.6 6.6.1, 6.6.2, 6.6.2.1 6.6.2.3 6.7.2, 6.7.3, 6.7.2.8, 6.8, 6.9.1 6.9.2, 6.10, 6.11, 6.11.1.3 6.12.1.3 B.4, B.5, B.6 C.1, C.2, C.3, C.4, C.5, C.6 D.1,	Incorporated CRs: OMA-PAG-2007-0161R02 OMA-PAG-2007-0190R01 OMA-PAG-2007-0191R01 OMA-PAG-2007-0200R01 OMA-PAG-2007-0204 OMA-PAG-2007-0205R01 OMA-PAG-2007-0206R04 OMA-PAG-2007-0208R02 OMA-PAG-2007-0213 OMA-PAG-2007-0218 OMA-PAG-2007-0225R02 OMA-PAG-2007-0228R01 OMA-PAG-2007-0236 OMA-PAG-2007-0238R01 OMA-PAG-2007-0239 OMA-PAG-2007-0243R01OMA-PAG-2007-0253R01 OMA-PAG-2007-0255 OMA-PAG-2007-0258R02 OMA-PAG-2007-0259 OMA-PAG-2007-0260 OMA-PAG-2007-0261R01 OMA-PAG-2007-0262R02 OMA-PAG-2007-0265 OMA-PAG-2007-0266R01 OMA-PAG-2007-0269 OMA-PAG-2007-0274R01
	10 May 2007	6.2.1.1, 6.3.3.2, 6.4.1, 6.4.5, 6.5	Incorporated CRs: OMA-PAG-2007-0296R01 OMA-PAG-2007-0299R01
	05 Jun 2007	All	Incorporated CRs: OMA-PAG-2007-0317R01 OMA-PAG-2007-0330R03 OMA-PAG-2007-0300R04 OMA-PAG-2007-0357

Document Identifier	Date	Sections	Description
	14 Jun 2007	2.1, 3.2, 5.1.3, 5.1.4, 5.2.2, 5.2.2.1, 6.1, 6.1.1.1, 6.1.2.1.1, 6.1.2.1.2, 6.1.3, 6.2, 6.2.2.1, 6.2.2.2, 6.3, B.1.1, B.1, B.2, B.3, B.4, C.1, C.2, C.5, D.1, D.2, D.3	Incorporated CRs: OMA-PAG-2007-0360R02 OMA-PAG-2007-0401R01 OMA-PAG-2007-0417R01 OMA-PAG-2007-0440R01 OMA-PAG-2007-0446 OMA-PAG-2007-0449 OMA-PAG-2007-0450 OMA-PAG-2007-0457R03
	15 Jun 2007	5.1.2 5.4.1.1	Editorial corrections
	30 Jun 2007	2.1, 5.1.1, 5.2.2.2, 6.1, 6.1.2.1.1, 6.1.3, 6.3	Editorials corrections: [XDM_Group] ref. not in alphabetized order. Style/indention of the numbered bullets 3 paragraphs indented, should be left justified. Broken cross ref. " that is in blue font The first example should not be a link (in blue) Broken cross ref.
Candidate Version OMA-TS-XDM_Core-V2_0	24 Jul 2007	n/a	Status changed to Candidate by TP (2007-07-11 to 2007-07-24) TP ref # OMA-TP-2007-0284-INP_XDM_V2_0_ERP_for_Candidate_approval

Appendix B. Static Conformance Requirements (normative)

The SCRs [SCRRULES] defined in the following tables include SCR for:

- Aggregation Proxy
- XDMS
- XDMC
- Search Proxy
- Aggregation Proxy of Remote Network

Each SCR table MUST have a title and MUST have only the following columns [SCRRULES]:

- Item: Identifier for a feature. It MUST be of type ScrItem in the dependency grammar described below.
- Function: Short description of the feature.
- Reference: Section(s) of the specification(s) with more details on the feature.
- Requirement: Other features required by this feature, independent of whether those other features are mandatory or optional. The notation in the dependency grammar, as described below, MUST be used for this column when other features are required, else the column MUST be left empty.

The dependency grammar notation to be used in the Requirement column of the SCR and CCR tables using ABNF [RFC2234] is described below [SCRRULES].

```
TerminalExpression =    ScrReference
                        / NOT TerminalExpression
                        / TerminalExpression LogicalOperator TerminalExpression
                        / “(“ TerminalExpression “)”

ScrReference =          ScrItem
                        / ScrGroup

ScrItem =               SpecScrName “–“ GroupType “–“ DeviceType “–“ NumericId “-”
                        Status
                        / SpecScrName “–“ DeviceType “–“ NumericId “-” Status

ScrGroup =              SpecScrName “:” FeatureType
                        / SpecScrName “– “ GroupType “–“ DeviceType “–” FeatureType

SpecScrName = 1*Character;

GroupType = 1*Character;

DeviceType = “C” / “S”; C – client, S – server

NumericId = Number Number Number

Status = “M” / “O”; M - Mandatory, O - Optional

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

FeatureType = “MCF” / “OCF” / “MSF” / “OSF”;

Character = %x41-5A ; A-Z

Number = %x30-39 ; 0-9
```

The following tags are used in the Function column to identify the relationship of the requirements in this enabler release [XDM_ERELD-V2_0] with the requirements of the previous enabler release [XDM_ERELD-V1_0]:

- XDMv1.0 – Requirement that is the same in this enabler release [XDM_ERELD-V2_0] , as in the previous enabler release [XDM_ERELD-V1_0].
- XDMv2.0 – Requirement that is new in this enabler release [XDM_ERELD-V2_0] .
- XDMv1.0mod – Requirement that exists in the previous enabler release [XDM_ERELD-V1_0], but is modified in this enabler release [XDM_ERELD-V2_0] .

B.1 XDM Client

B.1.1 XDMC implemented in a UE

Item	Function	Reference	Requirement
XDM_Core-XOP-C-001-M	Support rules for constructing XDM URIs (XDMv1.0)	6.1.1.1	
XDM_Core-XOP-C-002-M	Including “User-Agent” HTTP header with the required value (XDMv2.0)	6.1	
XDM_Core-XOP-C-003-M	Support for XDM Operations (XDMv1.0)	6.1.1.2	
XDM_Core-SUB-C-001-O	Initial Subscription using the SIP SUBSCRIBE message (XDMv1.0)	6.1.2.1	XDM_Core-SUB-C-002-O
XDM_Core-SUB-C-002-O	Processing Received SIP NOTIFY Request (XDMv1.0)	6.1.2.2	XDM_Core-SUB-C-001-O
XDM_Core-SEC-C-001-M	Support HTTP Digest authentication (XDMv1.0)	6.1, 5.1.1	
XDM_Core-SEC-C-002-M	Support HTTP over TLS using the required cipher suite (XDMv1.0)	6.1, 5.1.4	
XDM_Core-SEC-C-003-O	Support other cipher suites defined in RFC2246 (XDMv1.0)	6.1, 5.1.4	
XDM_Core-HCOM-C-001-O	Support HTTP Compression (XDMv1.0)	6.1.1.2	
XDM_Core-SRC-C-001-O	Searching for XML documents (XDMv2.0)	6.1.3	XDM_Core-SRC-C-002-O
XDM_Core-SEC-C-004-O	Support GAA (XDMv1.0)	6.1, 5.1.1	
XDM_Core-CAPS-C-001-O	Support Application Usage “xcap-caps” (XDMv1.0)	5.3.1	
XDM_Core-DIR-C-001-O	Support Application Usage “org.openmobilealliance.xcap-directory” (XDMv1.0)	5.3.2	
XDM_Core-SRC-C-002-O	Support Search document (XDMv2.0)	5.4.1	XDM_Core -SRC-C-001-O

Item	Function	Reference	Requirement
XDM_Core-SEC-C-005-M	XDM Client Identity Assertion (XDMv1.0)	6.1, 5.1.2	
XDM_Core-ERR-C-001-M	Support types of <error-element> as required (XDMv1.0 – SCR item was missing)	5.2.3	

B.1.2 XDMC implemented in an AS

Item	Function	Reference	Requirement
XDM_Core-XOP-C-004-M	Support rules for constructing HTTP URIs (XDMv1.0)	6.1.1.1	
XDM_Core-XOP-C-005-M	Support for XDM Operations (XDMv1.0)	6.1.1.2	
XDM_Core-SUB-C-003-O	Initial Subscription using the SIP SUBSCRIBE message (XDMv1.0)	6.1.2.1	XDM_Core-SUB-C-004-O
XDM_Core-SUB-C-004-O	Processing Received SIP NOTIFY Request (XDMv1.0)	6.1.2.2	XDM_Core-SUB-C-003-O
XDM_Core-HCOM-C-002-O	Support HTTP Compression (XDMv1.0)	6.1.1.2	

B.2 XDM Server

Item	Function	Reference	Requirement
XDM_Core-XCAP-S-001-M	Support for XCAP (XDMv1.0)	6.2, 6.2.1	XDM_Core-XOP-S-001-M
XDM_Core-XOP-S-001-M	Processing different HTTP requests (XDMv1.0)	6.2.1.1, 6.2.1.2, 6.2.1.3	
XDM_Core-SUB-S-001-O	Support Initial Subscription when SIP SUBSCRIBE message received (XDMv1.0)	6.2.2.1	XDM_Core-SUB-S-002-O
XDM_Core-SUB-S-002-O	Generating a SIP NOTIFY request (XDMv1.0)	6.2.2.2	XDM_Core-SUB-S-001-O
XDM_Core-SEC-S-001-M	Support XDMC identity access authorization (XDMv1.0)	6.2,5.1.5 , 5.1.5	
XDM_Core- ERR-S-001-M	Support Error Handling (XDMv1.0)	5.1.1, 5.1.5, 6.2, 6.2.1.1, 6.2.2, 6.3.1.2,	

Item	Function	Reference	Requirement
XDM_Core-CAPS-S-001-M	Support Application Usage "xcap-caps" (XDMv1.0)	5.3.1	
XDM_Core-DIR-S-001-M	Support Application Usage "org.openmobilealliance.xcap-directory" (XDMv1.0)	5.3.2	
XDM_Core-SRC-S-001-O	Support Search document	5.4.1	XDM_Core -SRC-S-002-O
XDM_Core-SRC-S-002-O	Searching for XML documents (XDMv2.0)	6.2.3	XDM_Core-SRC-S-001-O
XDM_Core-XOP-S-002-O	Including "Server" HTTP header with the required value in HTTP response to XDMC (XDMv2.0)	6.2	
XDM_CoreERR-S-002-O	Not using other types of <error-element> than what is recommended. (XDMv1.0)	5.2.3	

B.3 Aggregation Proxy

Item	Function	Reference	Requirement
XDM_Core-XOP-S-003-M	Acting as an HTTP Proxy [RFC2616] and configuration as an HTTP Reverse Proxy [RFC3040] (XDMv1.0)	6.3	
XDM_Core-SEC-S-002-M	Support HTTP Digest authentication (XDMv1.0)	6.3, 5.1.1	
XDM_Core-SEC-S-003-M	Support HTTP over TLS using the required cipher suite (XDMv1.0)	6.3.1.1, 5.1.4	
XDM_Core-SEC-S-004-O	Support other cipher suites defined in RFC2246 (XDMv1.0)	6.3.1.1, 5.1.4	
XDM_Core-SEC-S-005-M	Support XDMC Identity Assertion (XDMv1.0)	6.3, 5.1.2	
XDM_Core-XOP-S-004-M	Support XCAP request forwarding (XDMv1.0)	6.3, 6.3.1	
XDM_Core-XOP-S-005-M	Sending XCAP response back (XDMv1.0)	6.3, 6.3.1.1	
XDM_Core-ERR-S-003-M	Handling error cases with appropriate HTTP error response (XDMv1.0mod)	6.3.1.2	

Item	Function	Reference	Requirement
XDM_Core-HCOM-S-001-O	Support Compression (XDMv1.0)	6.3.2	
XDM_Core-SEC-S-006-O	Support for GAA (XDMv1.0)	6.3, 5.1.1	
XDM_Core-CAPS-S-002-M	XCAP Server Capabilities retrieval (Application Usage “xcap-caps”) (XDMv1.0 – SCR item was missing)	6.3.1.3, 5.3.1	XDM_Core-XRF-S-001-M AND XDM_Core-XRF-S-002-M AND XDM_Core-XRF-S-003-M
XDM_Core-DIR-S-002-M	XCAP Directory retrieval (Application Usage “org.openmobilealliance.xcap-directory”) (XDMv1.0)	6.3.1.4, 5.3.2	XDM_Core-XRF-S-001-M AND XDM_Core-XRF-S-002-M AND XDM_Core-XRF-S-003-M
XDM_Core-ERR-S-004-M	Support Error Handling (XDMv1.0)	5.1.1, 6.3.1.3	
XDM_Core-SEC-S-007-M	XDM Client identity sharing (XDMv2.0)	5.1.3	

B.4 Search Proxy

Item	Function	Reference	Requirement
XDM_Core-SRC-S-003-M	Forwarding Search Requests (XDMv2.0)	6.4.1	XDM_Core -SEC-S-006-M
XDM_Core-SRC-S-004-M	Aggregating Search results from XDMSs and forwarding those back (XDMv2.0)	6.4.2	XDM_Core -SEC-S-006-M
XDM_Core-ERR-S-005-M	Handling error cases (XDMv2.0)	6.4.1.1	
XDM_Core-SEC-S-008-M	Integrity and Confidentiality Protection support (XDMv2.0)	6.4, 5.1.4	
XDM_Core -SEC-S-009-M	Sharing XDMC authentication and its identity assertion provided by the Aggregation Proxy (XDMv2.0)	6.4, 5.1.3	
XDM_Core -SEC-S-010-M	Integrity and confidentiality protection (XDMv2.0)	5.1.4	

B.5 Aggregation Proxy of Remote Network

Item	Function	Reference	Requirement
XDM_Core-XOP-S-006-M	Acting as an HTTP Proxy [RFC2616] and configuration as an HTTP Reverse Proxy [RFC3040] (XDMv2.0)	6.5	

Item	Function	Reference	Requirement
XDM_Core-XOP-S-007-M	Forwarding XCAP requests (XDMv2.0)	6.5	
XDM_Core-XOP-S-008-M	Aggregating and forwarding responses back to trusted domains (XDMv2.0)	6.5	
XDM_Core-SEC-S-011-M	Protecting HTTP traffic (XDMv2.0)	6.5, 5.1.4	
XDM_Core -ERR-S-006-M	Reject a request from untrusted remote network with an HTTP "403 Forbidden" error response (XDMv2.0)	6.5	
XDM_Core-ERR-S-007-M	Reject a request for a target domain that is not responsible for with an HTTP "404 Not Found" error response (XDMv2.0)	6.5	
XDM_Core -SEC-S-012-M	Sharing the XDMC identity assertion with the originating Aggregation Proxy (XDMv2.0)	6.5, 5.1.3	

Appendix C. Examples (informative)

C.1 Sample XCAP Operation

Figure C.1 describes how an XCAP operation is performed in 3GPP IMS or 3GPP2 MMD. The “resource-list” Application Usage (see [XDM_List]) i.e. the manipulation of a URI List is used in this specific example, but the same types of messages apply for other Application Usages (although the HTTP body content would, of course, be different). In this example is the XDMC residing in a UE in the same domain as the Shared List XDMS. It is also assumed that the address of Aggregation Proxy is “xcap.example.com” and the XCAP Root URI is xcap.example.com/”.

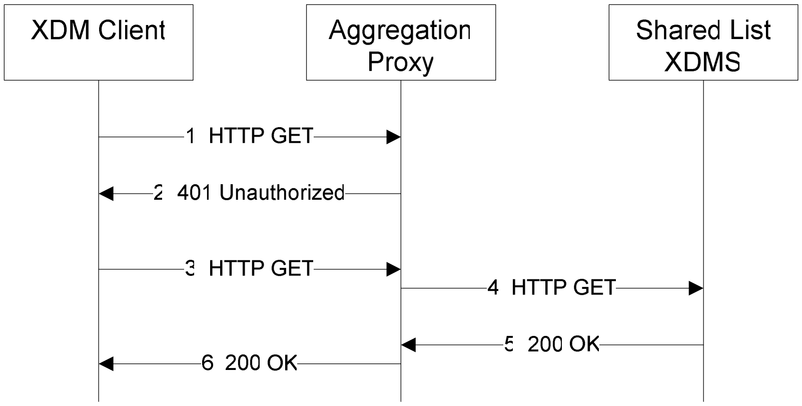

Figure C.1- Sample XCAP operation

The details of the flows are as follows:

1) The user “sip:joebloggs@example.com” wants to obtain an XML document. For this purpose the XDMC sends an HTTP GET request to the Aggregation Proxy.

```
GET /resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.0
Date: Thu, 08 Jan 2007 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
```

2) Upon receiving an unauthorized HTTP GET the Aggregation Proxy chooses to authenticate the XDMC.

```
HTTP/1.1 401 Unauthorized
Server: XDM-proxy/OMA2.0
Date: Thu, 08 Jan 2007 10:50:35 GMT
WWW-Authenticate: Digest realm="xcap.example.com", nonce="47364c23432d2e131a5fb210812c", qop=auth-
   int
Content-Length: 0
```

3) The XDMC sends a HTTP GET request including the Authorization header.

```
GET /resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.0
Date: Thu, 08 Jan 2007 10:50:37 GMT
Authorization: Digest realm="xcap.example.com", nonce="47364c23432d2e131a5fb210812c",
   username="sip:joebloggs@example.com", qop=auth-int,
   uri="/resource-lists/users/sip:joebloggs@example.com/index",
   response="2c8ee200cec7f6e966c932a9242554e4", cnonce="dcd99agsfgfsa8b7102dd2f0e8b1", nc=00000001
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
```

4) Based on the AUID the Aggregation Proxy forwards the request to appropriate XDMS.

```
GET /resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: xcap.example.com
Via: HTTP/1.1 proxy.example.com (Apache/1.1)
User-Agent: XDM-client/OMA2.0
Date: Thu, 08 Jan 2007 10:50:37 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
```

NOTE: If the “X-3GPP-Intended-Identity” is not included in the message (3), the Aggregation Proxy will include the “X-3GPP-Asserted-Identity” header.

5) After the XDMS has performed the necessary authorisation checks on the request originator, the XDMS sends an HTTP “200 OK” response including the requested document in the body.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.0
Date: Thu, 08 Jan 2007 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:+430123499995678"/>
  </list>
</resource-lists>
```

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

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.0
Via: HTTP/1.1 proxy.example.com (Apache/1.1)
Date: Thu, 08 Jan 2007 10:50:39 GMT
Authentication-Info: nextnonce="e966c32a924255e42c8ee20ce7f6"
Etag: "eti87"
Content-Encoding: gzip
Content-Type: application/resource-lists+xml
Content-Length: (...)

 (binary data)
```

C.2 Sample XCAP message flow

This example describes the message flows used to manipulate an XML document in an XDMS after authentication.

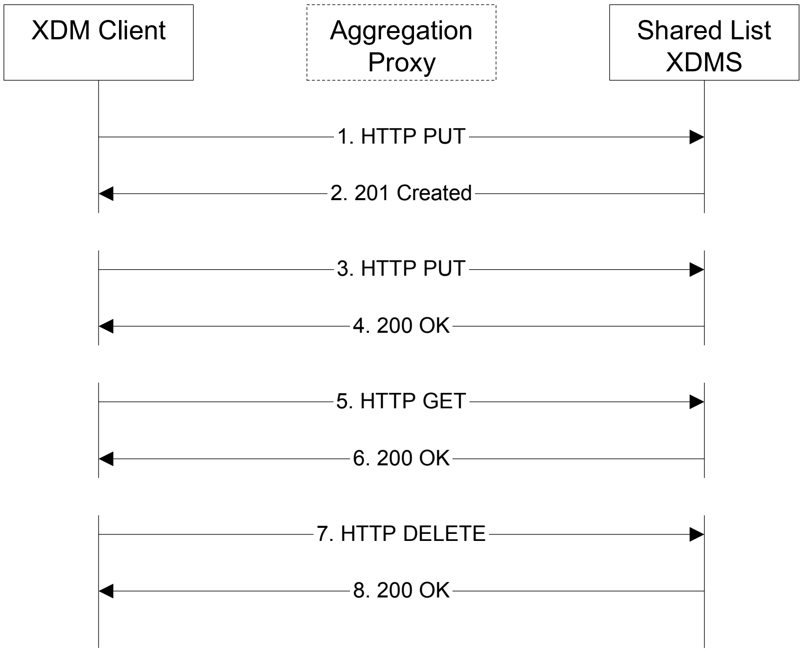

Figure C.2- XDMC manipulating an XML document

NOTE 1: The request messages (1,3,5,7) are shown in one diagram for the convenience of the reader, but there is no implication that all of them have to be performed.

NOTE 2: The Aggregation Proxy is not shown in the flow diagram as its omission does not affect the content of the exchanged messages. The flow diagram also does not show the authentication headers and other HTTP headers not necessary to illustrate the XCAP functionality.

1) The XDMC sends an HTTP PUT request to create a new URI list document “index” for the user with a public SIP URI of “sip:joebloggs@example.com” in the Shared List XDMS in the example.com domain.

```
PUT /resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: xcap.example.com
…
Content-Type: application/resource-lists+xml
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 Shared List 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=%22My_friends%22%5d/entry%5b@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>
```

4) The Shared List XDMS acknowledges the addition of new elements to the list with an HTTP "200 OK" reply.

```
HTTP/1.1 200 OK
Etag: "efefefef"

…
Content-Length: 0
```

5) The XDMC sends an HTTP GET request to retrieve "sip:joebloggs@example.com"'s "friends" list from the Shared List XDMS.

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

6) The Shared List XDMS returns the list to the XDMC in the body of an 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 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 List 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 Shared List XDMS, after checking the privileges of the Principal, performs the deletion.

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

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

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

Figure C.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. In this example is the XDMC residing in a UE in the same domain as Shared List XDMS and Shared Group XDMS. It is also assumed that the address of Aggregation Proxy is “xcap.example.com” and the XCAP Root URI is xcap.example.com/”.

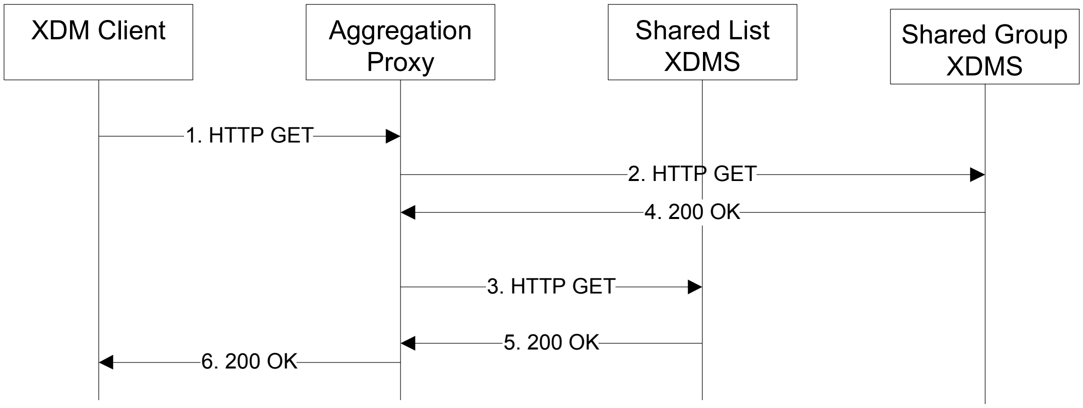

Figure C.3- Sample XCAP Directory retrieval operation

The details of the flows are as follows:

1) The user “sip:joebloggs@example.com” wants to obtain a list of all his documents stored in all XDMSes. For this purpose the XDMC sends a HTTP GET request to the Aggregation Proxy.

```
GET /org.openmobilealliance.xcap-directory/users/sip:joebloggs@example.com/directory.xml HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.0
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 Group XDMS.

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

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

```
HTTP/1.1 200 OK
```

```
Server: XDM-serv/OMA2.0
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="groups">
    <entry
   uri=”http://xcap.example.com/org.openmobilealliance.groups/users/sip:joebloggs@example.com/skiin
   g” etag=”abc123”/>
    <entry
   uri="http://xcap.example.com/org.openmobilealliance.groups/users/sip:joebloggs@example.com/shopp
   ing" etag=”def456”/>
  </folder>
</xcap-directory>
```

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

5) The Shared List 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/OMA2.0
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/OMA2.0
Date: Thu, 08 Jan 2004 10:55:39 GMT
Etag: "eti101"
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>
  <folder auid="groups">
    <entry
   uri="http://xcap.example.com/org.openmobilealliance.groups/users/sip:joebloggs@example.com/skiin
   g" etag=”abc123”/>
    <entry
   uri="http://xcap.example.com/org.openmobilealliance.groups/users/sip:joebloggs@example.com/shopp
   ing" etag=”def456”/>
  </folder>
</xcap-directory>
```

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

Figure C.4 describes how an XCAP operation is performed to retrieve all of a user’s documents corresponding to a particular Application Usage. For simplicity, the authentication steps are omitted.

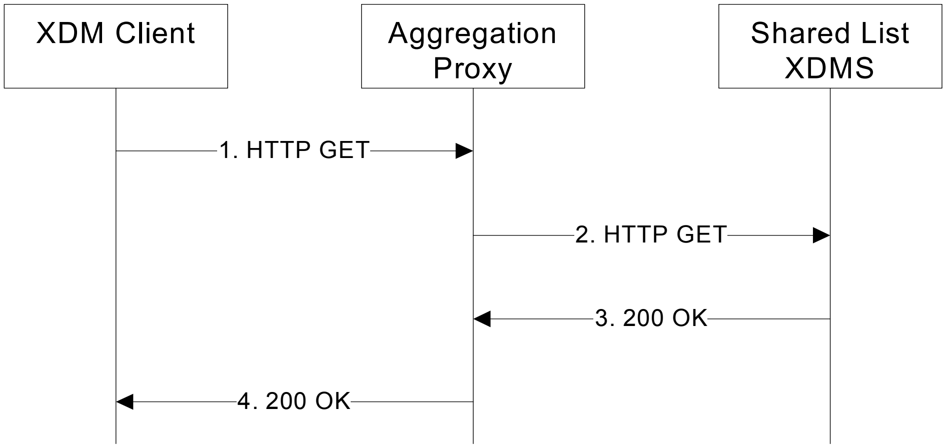

Figure C.4- Sample XCAP Directory retrieval operation from a particular XDMS

The details of the flows are as follows:

1) The user “sip:joebloggs@example.com” wants to obtain a list of all his documents (URI lists) stored in the Shared List XDMS. For this purpose the XDMC sends a HTTP GET request to the Aggregation Proxy.

```
GET /org.openmobilealliance.xcap-directory/users/sip:joebloggs@example.com/directory.xml/~~/xcap-
   directory/folder%5b@auid=%22resource-lists%22%5d HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/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 List XDMS.

3) The Shared List XDMS responds with a HTTP 200 OK including the <folder> element containing the URI List document belonging to sip:joebloggs@example.com

```
HTTP/1.1 200 OK
Server: XDM-serv/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.

C.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 residing in a UE requests to subscribe to changes in the Group document. Note the procedure is identical no matter an XDMC is subscribing to an XML belonging to himself or others.

Figure C.5 is an example that demonstrates how an XDMC residing in a UE subscribes to changes in a Group document.

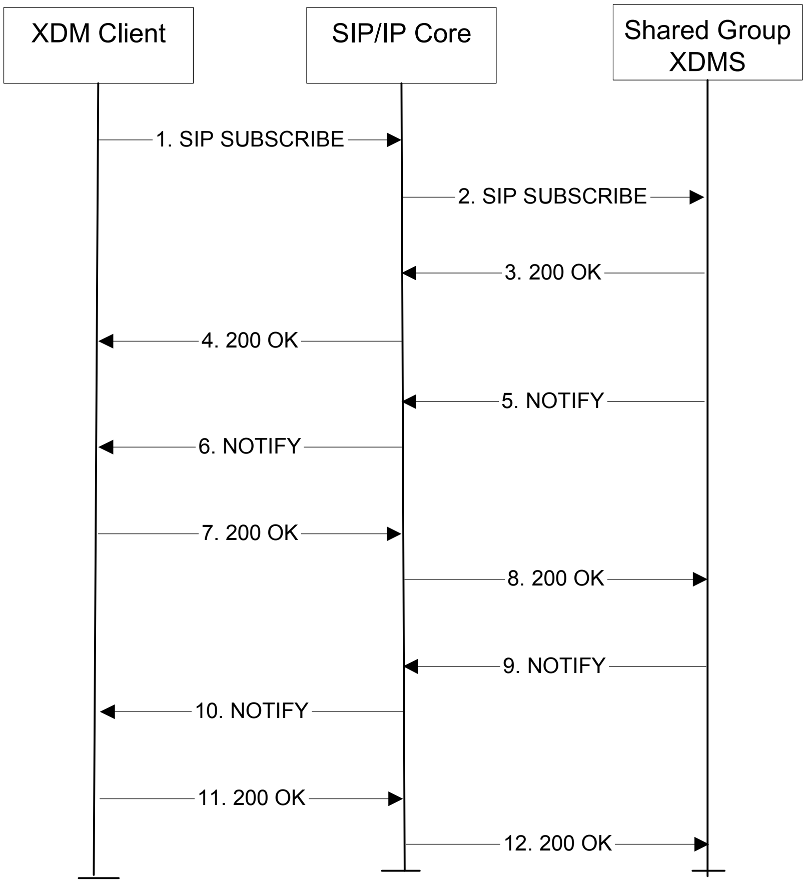

Figure C.5 - XDM Client subscribes to changes in XML documents.

1) XDMC (XUI=joe.bloggs@example.com) subscribes to his Group document named as 'joebloggs_friends', with the contact SIP URI ‘sip:joe.bloggs@example.com’, because he uses multiple devices and wants to keep them updated.

```
SUBSCRIBE sip:joe.bloggs@example.com SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 70
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>
Event: xcap-
   diff;path="org.openmobilealliance.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
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 Shared Group 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:scscf1.home1.net;lr> <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
Route: <sip:sharedgroupxdms1.home1.netlr>
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>
Event: xcap-
   diff;path="org.openmobilealliance.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
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

3) Upon receiving a SIP SUBSCRIBE request for the “xcap-diff” event package, the Shared Group XDMS shall perform the necessary authorization checks on the originator's identity. If the authorization is successes, it shall create a subscription dialog to "xcap-diff" event package to provide the changes of the data identified by the "Event" header path parameter, and return 200 OK to the subscriber.

4) The SIP/IP Core network forwards the 200 OK response to the originator of the SIP SUBSCRIBE request, i.e. sip:joe.bloggs@example.com.

5) The Shared Group 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 sharedgroupxdms1.home1.net;branch=z9hG4bK332b23.1
Max-Forwards: 70
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 102 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: application/xcap-diff+xml
Contact: <sip:sharedgroupxdms1.home1.net>
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"

    xcap-root="http://xcap.example.com"
    <document new-etag="7ahggs"
     doc-
   selector="org.openmobilealliance.groups/users/sip:joe.bloggs@example.com/joebloggs_friends">
   previous-etag="7ahggs"

    </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 Shared Group XDMS.

9) After some updates in the XDM document, the Shared Group XDMS sends the diff part in SIP NOTIFY to the XDMC, in this example, a new “new-friend@example.com” entry was added to the list.

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP sharedgroupxdms1.home1.net;branch=z9hG4bK332b23.1
Max-Forwards: 70
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From: <sip:joe.bloggs@example.com>;tag=31415
To: <sip:joe.bloggs@example.com>;tag=151170
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 112 NOTIFY
Subscription-State: active;expires=600000
Event: xcap-diff
Content-Type: application/xcap-diff+xml
Contact: <sip:sharedgroupxdms1.home1.net>
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xmlns:l="urn:oma:xml:poc:list-service" xcap-
   root="http://xcap.example.com">
    <document previous-etag="7ahggs" doc-
   selector="org.openmobilealliance.groups/users/sip:joe.bloggs@example.com/joebloggs_friends"
     new-etag="ffds66a">
      <change-log>
        <add sel="l:group/l:list-service/l:list">
          <l:entry l:uri="sip:new-friend@example.com">
        </add>
      </change-log>
    </document>
   </xcap-diff>
```

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

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

12) The SIP/IP Core network forwards the 200 OK to the Shared Group XDMS.

C.6 Sample Search Operation

Figure C.6 describes how a Search operation is performed. The example shows searching user profile data in Shared Profile XDMS [XDM_Shared_Profile]; the same type of messages apply for searching in other Application Usages, where content of HTTP body would be different. In this example is the XDMC residing in a UE in the same domain as the Shared Profile XDMS. It is also assumed that the address of Aggregation Proxy is “xcap.example.com” and the XCAP Root URI is “xcap.example.com/”.

For simplicity, search in home domain only is described in following example.

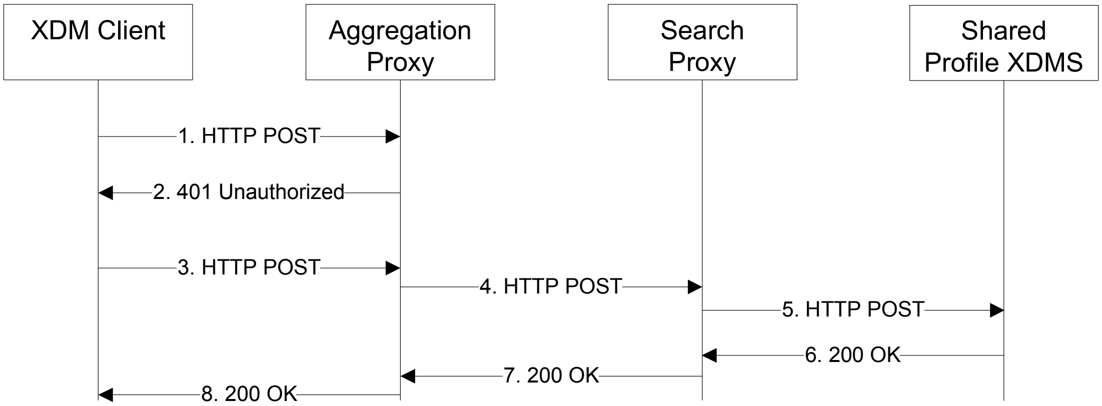

Figure C.6 - Sample XCAP operation

The details of the flows are as follows:

1) The user “sip:joebloggs@example.com” wants to obtain the user profile data with people from Japan and hobby football. For this purpose the XDMC sends an HTTP POST request to the Aggregation Proxy.

```
POST /org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/ HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.0
Date: Thu, 10 Aug 2006 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
Content-Type: application/vnd.oma.search+xml
Content-Length: …

<?xml version="1.0" encoding="UTF-8"?>
<search-set   xmlns="urn:oma:xml:xdm:search">

<search id="1234">
  <request>
    <query>
   <![CDATA[
      xquery version "1.0";
      declare default element namespace "urn:oma:xml:xdm:user-profile";

      for $u in collection("org.openmobilealliance.user-profile/users/")/user-profiles/user-
   profile
      where ($u/hobbies/hobby="football")and($u/address/country="JP")
      return <user-profile>{$u/@uri}{$u/display-name}</user-profile>
  ]]>
    </query>
  </request>
</search>

</search-set>
```

2) Upon receiving an unauthorized HTTP POST the Aggregation Proxy chooses to authenticate the XDMC.

```
HTTP/1.1 401 Unauthorized
Server: XDM-proxy/OMA2.0
Date: Thu, 10 Aug 2006 10:50:33 GMT
WWW-Authenticate: Digest realm="xcap.example.com", nonce="47364c23432d2e131a5fb210812c", qop=auth-
   int
Content-Length: 0
```

3) The XDMC sends a HTTP POST request including the Authorization header to the Aggregation Proxy.

```
POST /org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/ HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.0
Date: Thu, 10 Aug 2006 10:50:33 GMT
Authorization: Digest realm="xcap.example.com", nonce="47364c23432d2e131a5fb210812c",
   username="sip:joebloggs@example.com", qop=auth-int,
   uri="/org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/",
   response="2c8ee200cec7f6e966c932a9242554e4", cnonce="dcd99agsfgfsa8b7102dd2f0e8b1", nc=00000001
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
Content-Type: application/vnd.oma.search+xml
Content-Length: …

<?xml version="1.0" encoding="UTF-8"?>

<search-set   xmlns="urn:oma:xml:xdm:search">

<search id="1234">
  <request>
    <query>
   <![CDATA[
      xquery version "1.0";
      declare default element namespace "urn:oma:xml:xdm:user-profile";

      for $u in collection("org.openmobilealliance.user-profile/users/")/user-profiles/user-
   profile
      where ($u/hobbies/hobby="football")and($u/address/country="JP")
   return <user-profile>{$u/@uri}{$u/display-name}</user-profile>
  ]]>
    </query>
  </request>
</search>

</search-set>
```

4) Based on the “org.openmobilealliance.search” AUID, the Aggregation Proxy forwards the Search Request to the Search Proxy.

```
POST /org.openmobilealliance.search?target=org.openmobilealliance.user-profile/users/ HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.0
Date: Thu, 10 Aug 2006 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
Content-Type: application/vnd.oma.search+xml
Content-Length: …

<?xml version="1.0" encoding="UTF-8"?>
<search-set   xmlns="urn:oma:xml:xdm:search">

<search id="1234">
  <request>
    <query>
   <![CDATA[
      xquery version "1.0";
      declare default element namespace "urn:oma:xml:xdm:user-profile";

      for $u in collection("org.openmobilealliance.user-profile/users/")/user-profiles/user-
   profile
      where ($u/hobbies/hobby="football")and($u/address/country="JP")
      return <user-profile>{$u/@uri}{$u/display-name}</user-profile>
  ]]>
    </query>
  </request>
</search>

</search-set>
```

NOTE 1: 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) Based on the target parameter in the Request URI, the Search Proxy forwards the Search Request to the appropriate XDMS. When forwarding, the Search Proxy removes the “target” query parameter from the HTTP URI.

```
POST /org.openmobilealliance.search HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA2.0
Date: Thu, 10 Aug 2006 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
Content-Type: application/vnd.oma.search+xml
Content-Length: …

<?xml version="1.0" encoding="UTF-8"?>
<search-set   xmlns="urn:oma:xml:xdm:search">

<search id="1234">
  <request>
```

```
    <query>
   <![CDATA[
      xquery version "1.0";
      declare default element namespace "urn:oma:xml:xdm:user-profile";

      for $u in collection("org.openmobilealliance.user-profile/users/")/user-profiles/user-
   profile
      where ($u/hobbies/hobby="football")and($u/address/country="JP")
      return <user-profile>{$u/@uri}{$u/display-name}</user-profile>
  ]]>
    </query>
  </request>
</search>

</search-set>
```

6) After the XDMS has performed the search operation, the XDMS sends an HTTP "200 OK" response including the requested results in the body.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.0
Date: Thu, 10 Aug 2006 10:50:39 GMT
Content-Type: application/vnd.oma.search+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<search-set xmlns="urn:oma:xml:xdm:search" xmlns:up="urn:oma:xml:xdm:user-profile">

<search id="1234">
  <response>
   <up:user-profile uri="A@example.com"><up:display-name>Alex</up:display-name></up:user-profile>
   <up:user-profile uri="B@example.com"><up:display-name>Brian</up:display-name></up:user-profile>
   <up:user-profile uri="C@example.com"><up:display-name>Chris</up:display-name></up:user-profile>
   <up:user=profile uri="D@example.com"><up:display-name>David</up:display-name></up:user-profile>
  </response>
</search>

</search-set>
```

7) The Search Proxy routes the response to the Aggregation Proxy.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.0
Date: Thu, 10 Aug 2006 10:50:39 GMT
Content-Type: application/vnd.oma.search+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<search-set xmlns="urn:oma:xml:xdm:search" xmlns:up="urn:oma:xml:xdm:user-profile">

<search id="1234">
  <response>
   <up:user-profile uri="A@example.com"><up:display-name>Alex</up:display-name></up:user-profile>
   <up:user-profile uri="B@example.com"><up:display-name>Brian</up:display-name></up:user-profile>
   <up:user-profile uri="C@example.com"><up:display-name>Chris</up:display-name></up:user-profile>
   <up:user-profile uri="D@example.com"><up:display-name>David</up:display-name></up:user-profile>
  </response>
</search>

</search-set>
```

8) The Aggregation Proxy encodes (optionally) the content and routes the response back to the XDMC.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA2.0
Date: Thu, 10 Aug 2006 10:50:39 GMT
Content-Type: application/vnd.oma.search+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<search-set xmlns="urn:oma:xml:xdm:search" xmlns:up="urn:oma:xml:xdm:user-profile">

<search id="1234">
```

```
  <response>
   <up:user-profile uri="A@example.com"><up:display-name>Alex</up:display-name></up:user-profile>
   <up:user-profile uri="B@example.com"><up:display-name>Brian</up:display-name></up:user-profile>
   <up:user-profile uri="C@example.com"><up:display-name>Chris</up:display-name></up:user-profile>
   <up:user-profile uri="D@example.com"><up:display-name>David</up:display-name></up:user-profile>
  </response>
</search>

</search-set>
```

Appendix D. XDMC Provisioning (Normative)

This appendix specifies the parameters that are needed for initiation of XDM service by the XDMC, as well as continuous provisioning by Service Provider. These parameters are specified in Client Provisioning Application Characteristics document (AC file) [CP_ProvCont] and Device Management Management Objects (DM MOs) [DMStdObj]. Existing parameters in [CP_ProvCont] and [DMStdObj] are re-used; those without corresponding parameters are defined and to be registered in OMNA through OMA official registration process.

The AC file or DM MOs MAY be used for initial provisioning of parametersas specified in [DM_ERELD], and the DM MOs SHOULD be used for continuous provisioning of parameters according to [DM_ERELD], if required by the Service Provider to update service configurations.

D.1 Provisioned XDMC Parameters

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

ID	Name	Description	Mandatory (M) /Optional (O)
1	Application identity	Uniquely identifies the application	M
2	Application name	User displayable name for the XDM service	M
3	Provider–ID	Identity of the XDM Service Provider	O
4	Network Access Definitions	Reference to the connection used for the XCAP traffic.	M
5	XDM reference to SIP/IP Core	Reference to the SIP/IP Core for accessing an XDMS using the referenced SIP/IP Core.	M
6	XCAP Root URI	The root of all XCAP resources (which points to the Aggregation Proxy address). This is used when accessing via XCAP.	M
7	XCAP Authentication user name	HTTP digest “username”, for accessing an XDMS using the XCAP protocol	O
8	XCAP Authentication password	HTTP digest password	O
9	XCAP Authentication type	Authentication method for XDMS over XCAP	O
10	Conference-URI Template	A template used by the XDMC to propose a Conference URI when creating a Group document.	O

NOTE: The parameters ”XCAP Authentication username” and “XCAP Authentication password” are not needed if GAA is used in a 3GPP IMS or 3GPP2 MMD 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 Group), so that the XDMC can autonomously generate a URI that complies with local policy and uniqueness constraints. It is up to separate specifications to define provisioned parameters that make use of a URI Template.

A URI Template SHALL describe a URI as defined in [RFC3986]. The template contains a sequence, in any order, of:

a. unreserved characters according to [RFC3986],

b. the characters ":" , "@" and ";"

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

The XDMC SHALL support the following substitution tags:

<id> : The XDMC SHALL replace this tag with a unique identifier, generated by the XDMC using only unreserved characters according to [RFC3986].

<user> : The XDMC SHALL replace this tag with the user part of the XUI if the XUI is a Public SIP URI. If the XUI is a Tel URI [RFC 3966] then the XDMC SHALL replace the <user> tag with the "global-number-digits"/"local-number-digits" part of the Tel URI. Any "visual-separator" or "+" SHALL be removed from the "global-number-digits" before the replacement takes place.

<xui> : The XDMC SHALL replace this tag with the XUI.

NOTE 1: the XUI is a Public SIP URI [RFC3261] or Tel URI [RFC3966].

NOTE 2: 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>;group=<id>	sip:joe@example.com;group=abc123
<xui>;group=<id>	tel:+1720-555-1212;group=abc123

Table X: Example usages of URI Templates

D.2 Application Characteristics

The Application characteristics (AC) file for XDM 2.0 service [XDM_ERELD-V2_0] MAY be used for initial provisioning of the XDMC.

This chapter describes the provisioning document structure as described in [CP_ProvCont].

The following table lists the parameters available in an instance of the XDM Application Characteristic

Parameter Name	Req / Opt	Instances	Default
Standard Application Characteristic fields as defined in [CP_ProvCont]			
APPID	Required	1	"ap0007"
PROVIDER-ID	Optional	0 or 1	None
TO-APPREF	Required	1	None
NAME	Required	1	None
TO-NAPID	Required	1 or more	None
URI	Required	1	None
AAUTHNAME	Optional	0 or 1	None
AAUTHSECRET	Optional	0 or 1	None
AAUTHTYPE	Optional	0 or 1	None

CONF-URI-TMPLT	Optional	0 or 1	None

The Application Characteristics file for XDM 2.0 service is defined in [XDM_AC].

D.3 Management Objects

The Management Objects (MOs) for XDM 2.0 service [XDM_ERELD-V2_0] MAY be used for initial provisioning of the XDMC and SHOULD be used for continuous provisioning by Service Provider.

The Management Objects (MOs) for XDM 2.0 service is defined in [XDM_MO].

Appendix E. OMA specific 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].

E.1 X-XCAP-Asserted-Identity Extension-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 5.1.2 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.