Policy Evaluation, Enforcement and Management – Management Interface (PEM-2) Technical Specification

Approved Version 1.0 – 24 Jul 2012

Open Mobile Alliance

OMA-TS-PEEM_PEM2-V1_0-20120724-A

Contents

Tables

1. Scope

This document describes the Policy Management interface (PEM-2) specification, and is part of a group of documents defining the Policy Evaluation, Enforcement and Management (PEEM) enabler specifications. The PEM-2 interface is used by other resources (management requestors) to make a request for policy management, where Policy Management is the act of creating, updating, deleting, and viewing policies. The specification is extensible, in the sense that other operations may be added, if required in support of specific policies defined by other enabler releases. The PEM-2 specification is loosely coupled with the Policy Expression Language (PEL) specification, in the sense that PEM-2 needs to support management operations for policies written using PEL specification. PEM-2 is independent of the PEM-1 specification.

2. References

2.1 Normative References

[PEEM AD] "Policy Evaluation, Enforcement and Management Architecture", Open Mobile Alliance™, OMA-AD_Policy_Evaluation_Enforcement_Management-V1_0, URL: http://www.openmobilealliance.org/

[PEEM RD] "Policy Evaluation, Enforcement and Management Requirements", Open Mobile Alliance™, OMA-RD_Policy_Evaluation_Enforcement_Management-V1_0, URL: http://www.openmobilealliance.org/

[PEEM_Callable_Policy_Interface] "Policy Evaluation, Enforcement and Management Callable Interface (PEM-1) Technical Specification", Open Mobile Alliance™, OMA-TS-PEEM_PEM1-V1_0, URL: http://www.openmobilealliance.org/

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

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

[RFC4234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL:http://www.ietf.org/rfc/rfc4234.txt

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

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

2.2 Informative References

[BPEL] "Business Process Execution Language", OASIS,
URL: Web Services Business Process Execution Language Version 2.0
http://docs.oasis-open.org/wsbpel/2.0/CS01/wsbpel-v2.0-CS01.pdf

[OMADICT] "Dictionary for OMA Specifications", Version 2.6, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_6, URL:http://www.openmobilealliance.org/

[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

[RFC 3060] "Policy Core Information Model -- Version 1 Specification", B. Moore et al, February 2001, URL: http://www.ietf.org/rfc/rfc3060.txt

[RFC 3198] "Terminology for Policy-Based Management", A. Westerinen et al, November 2001, URL: http://www.ietf.org/rfc/rfc3198.txt

[RFC 3460] "Policy Core Information Model (PCIM) Extensions", B. Moore, Ed., January 2003, URL: http://www.ietf.org/rfc/rfc3460.txt

[RFC 4745] "Common Policy: A Document Format for Expressing Privacy Preferences, H.Schulzrinne et al, IETF RFC 4745, February 2007, URL: http://www.rfc-editor.org/rfc/rfc4745.txt

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

[XSD_BPEL] OASIS, WS-BPEL 2.0 XSD, http://docs.oasis-open.org/wsbpel/2.0/CS01/process/executable/ws-bpel_executable.xsd

[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/

3. Terminology and Conventions

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

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

3.1 Conventions

3.2 Definitions

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: [RFC 4825])
Application Usage	Detailed information on the interaction of an application with an XCAP server. (Source: [RFC 4825])
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: [RFC 4825])
Document URI	The HTTP URI containing the XCAP Root and Document Selector, resulting in the selection of a specific document. (Source: [RFC 4825])
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: [RFC 4825])
HTTP URI	An HTTP Request-URI as defined by [RFC 2616]
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 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])
Policy	An ordered combination of policy rules that defines how to administer, manage, and control access to resources [Derived from [RFC 3060], [RFC 3198] and [RFC 3460]].
Policy Action	Action (e.g. invocation of a function, script, code, workflow) that is associated to a policy condition in a policy rule and that is executed when its associated policy condition results in "true" from the policy evaluation step.
Policy Condition	A condition is any expression that yields a Boolean value.
Policy Enforcement	The process of executing actions, which may be performed as a consequence of the output of the policy evaluation process or during the policy evaluation process.
Policy Evaluation	The process of evaluating the policy conditions and executing the associated policy actions up to the point that the end of the policy is reached.
Policy Management	The act of describing, creating, updating, deleting, provisioning and viewing policies.
Policy Processing	Policy evaluation or policy evaluation and enforcement
Policy Rule	A combination of a condition and actions to be performed if the condition is true
Request	An articulation of the need to access a resource (e.g. asynchronous events).
Requestor	Any entity that issues a request to a resource.
Resource	Any component, enabler, function or application that can receive and process requests.
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: [RFC 4825]) (“This specification” refers to [RFC 4825])

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: [RFC 4825])
XCAP Root	A context that includes all of the documents across all Application Usages and users that are managed by a server. (Source: [RFC 4825]) 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:[RFC 4825])
XCAP Server	An HTTP server that understands how to follow the naming and validation constraints defined in this specification. (Source: [RFC 4825])
XCAP URI	An HTTP URI that represents an XCAP Resource.
XCAP User Identifier	The XCAP User Identifier (XUI) is a string, valid as a path element in an HTTP URI, which is associated with each user served by the XCAP Server. (Source: [RFC 4825])

3.3 Abbreviations

AUID	Application Unique Identifier
HTTP	Hyper Text Transfer Protocol
BPEL	Busines Process Execution Language
MIME	Multipurpose Internet Mail Extension
SCR	Static Conformance Requirement
PEEM	Policy Evaluation, Enforcement and Management
PEL	(PEEM) Policy Expression Language
PEM-2	PEEM Management interface
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

The specification of the Policy Evaluation, Enforcement and Management (PEEM) enabler is driven by the need to reduce management complexity whilst introducing consistent new subscriber services with the same or reduced time to market. The PEEM enabler processes policies, and provides means to manage policies. Policies are applied to requests to, or responses from resources or, when explicitly called by a resource. Policy Management is the act of creating, updating, deleting, and viewing policies. Various management actors such as service provider, network operator, enterprise, and end-user that may manage policies, via applications, must be supported. Such actors are called Management Requestors. The PEEM architecture [PEEM AD] introduced the PEM-2 interface, used by authorized principals to manage policies related to a resource. This interface is therefore also referred to as PEEM management interface. The PEEM requirements with respect to policy management are captured in the [PEEM RD]. The PEM-2 specification defines the input/output messages and parameters exchanged over the PEM-2 interface and the protocols used to exchange those messages. The PEM-2 interface specification is independent of the PEL used by the policies exchanged across the interface. For a request to operate properly, the policies are expected to be defined using a PEL supported by the PEEM implementation.

In a typical PEEM management flow, an authorized principal issues a request for Policy Management to the PEEM enabler, through the PEM-2 interface. Upon reception of the request, the PEEM enabler identifies the type of policy management request (e.g. create, delete, view, modify), executes the appropriate function and returns the results to the authorized principal.

5. PEM-2 Interface definition

PEM-2 Interface SHALL be compliant to the IETF XML Configuration Access Protocol (XCAP) specification [RFC 4825]. The policy management operations requests (Client side) and responses (Server side) specify how to apply the [RFC 4825]. The policy management operations defined in PEM-2 handle entire policies (normatively), as well as policy elements and elements' attributes within a policy (optionally). Hence PEM-2 full compliance to [RFC 4825] will allow implementations to support those additional features in implementations.

It is also possible that PEM-2 may be implemented as part of a broader XDM implementation [XDMSPEC], since XDM is also based on [RFC 4825]. However, in order to meet PEM-2 requirements, not all of the currently defined [XDMSPEC] features need to be supported; hence the relationship to XDM is addressed in the informative Appendix C. PEM-2 dependency on XDM, rather than XCAP may be reconsidered, should future versions of [XDMSPEC] identify the subset needed by PEM-2, as a more granular XDM profile. The following sections define the PEM-2 conformance to [RFC 4825].

5.1 PEM-2 relationship to XCAP

XCAP defines a protocol that can be used to manipulate XML documents, on a per-principal basis. This section introduces terminology and aspects of XCAP needed to define the PEM-2 interface (see [RFC 4825] for details).

XCAP includes a set of conventions for mapping XML documents and document components into HTTP URIs, rules for how the modification of one resource affects another, data validation constraints, and authorization policies associated with access to those resources and normal HTTP primitives can be used to manipulate the data.

Specific usages of XCAP are referred to as XCAP applications. The Application Usage defines the XML schema for the data used by the application, along with other key pieces of information. XCAP specifies how clients read, write, modify, create, and delete pieces of that data, through operations supported using HTTP/1.1 [RFC 2616]. An XCAP server acts as a repository for collections of XML documents, stored for each XCAP application. Within each application, documents are stored for each user, who can have multiple documents for a particular application. To access some component of one of those documents, XCAP defines an algorithm for constructing a URI that can be used to reference that component (e.g. the document itself, or elements or elements' attributes within the document). Each document managed via XCAP follows the XCAP document URI (XCAP URI) construction specification, which includes an XCAP Root and a Document Selector, and optionally a Node Selector (see [RFC 4825] for details). An XCAP Root URI identifies the XCAP Root, a Document URI identifies the document, and a Node URI identifies the node (an element or an element's attribute) within the document. Document Selectors may include a Global Tree and a Users Tree, and a document in the Users Tree may include an XCAP User Identifier (XUI).

Any HTTP resource that follows the naming conventions and validation constraints defined here is called an XCAP resource. Since XCAP resources are also HTTP resources, they can be accessed using HTTP methods. Reading an XCAP resource is accomplished with HTTP GET, creating or modifying one is done with HTTP PUT, and removing one of the resources is done with an HTTP DELETE. POST operations to HTTP URIs representing XCAP resources are not defined.

Each Application Usage is associated with an Application Unique ID (AUID), which uniquely identifies the Application Usage within the namespace of Application Usages, and is different from AUIDs used by other applications. AUIDs may be registered in an IETF namespace or maybe defined in a vendor-proprietary namespace.

An XCAP Server needs to validate the content of each XCAP resource when an XCAP Client tries to modify one, and XCAP Clients need to know how to construct valid requests. Application Usage is documented in a specification that conveys the following information:

- Application Unique ID (AUID): If the application usage is meant for general use on the Internet, the application usage MUST register the AUID into the IETF tree.
- XML Schema
- Default Document Namespace
- MIME Type

- Validation Constraints
- Data Semantics
- Naming Conventions
- Resource Interdependencies
- Authorization Policies

5.1.1 PEM-2 Application Usage

Policy management via PEM-2 interface has to support policies that may be written conforming to different XML schemas, corresponding respectively to the specific Policy Expression Language used. PEM-2 specification defines 2 Application Usages, conforming to the 2 PEL options supported by PEEM specifications. Application Usage is extensible. For example, OMA enablers, Service Providers and Vendors MAY extend the specified Application Usage with additional constraints, data semantics, naming conventions, resource interdependencies and authorization policies, or add entirely new additional Application Usages, under new AUIDs. However, while PEM-2 specifies a certain Application Usage, and therefore an implementation has to support such Application Usage, it does so that the implementations have a further choice to use it to validate documents being created or replaced if so desired, and not because validation of the passed documents is mandated as part of the PEM-2 specification. PEM-2 specification explicitly defines any validations that PEEM may perform, based on the Application Usage defined, as out-of-scope for the PEM-2 specification. PEEM implementations may provide tools to enable or disable validation of incoming policies, based on the specified Application Usage.

5.1.1.1 Application Unique ID

The AUID for policies using PEL option for ruleset framework SHALL be “org.openmobilealliance.policy-commonpol”.

The AUID for policies using PEL option for business process SHALL be “org.openmobilealliance.policy-bpel”.

5.1.1.2 MIME Type

The MIME type for a policy using PEL option for ruleset framework SHALL be “application/vnd.oma.policy-commonpol+xml”.

The MIME type for a policy using PEL option for business process SHALL be “application/vnd.oma.policy-bpel+xml”.

5.1.1.3 Default Namespace

The default namespace for policies using PEL option for ruleset framework SHALL be “urn:oma:xml:xdm:policy-commonpol”.

The default namespace for policies using PEL option for business process SHALL be “urn:oma:xml:xdm:policy-bpel”.

5.1.1.4 XML Schema

Policies using PEL options SHALL conform to the respective policy schemas supported by the PEEM implementation. Extensions to PEL options MAY come from work in OMA or outside OMA.

For example, in this version of PEM-2 TS, the policies using PEL option for ruleset framework (based on RFC 4745) SHALL conform to the XML schema described in [RFC 4745], and the extensions added in OMA described in [XSD_commPol] and [XSD_ext].

For example, the policies using PEL option for business process (based on BPEL) SHALL conform to the XML schema described in [XSD_BPEL].

5.1.1.5 Additional Constraints

None.

5.1.1.6 Data Semantics

None.

5.1.1.7 Naming Conventions

None.

5.1.1.8 Data Interdependencies

None.

5.1.1.9 Authorization Policies

None.

5.2 Procedures at the PEEM management requestor (client side)

A PEEM management requestor is a resource that uses the PEM-2 interface to issue policy management requests. A PEEM management requestor acts like an XCAP Client, and SHALL follow the procedures described in as described in [RFC 4825].

5.2.1 PEEM policy identifier parameter

A PEEM policy SHALL be encapsulated in an XML document, and a Policy identifier SHALL be an XCAP URI. The construction of a policy identifier SHALL follow the procedures described in [RFC 4825], that apply to creation of an XCAP URI for an XML document. The XML document has to conform to the XML schema, and to data constraints described under the Application Usage definition, used both by PEEM management requestor (XCAP Client) and PEEM component (XCAP Server).

5.2.2 Policy Management Operations

The PEM-2 interface SHALL support the operations of Create Policy, Modify Policy, Delete Policy and View Policy. These operations SHALL re-use interface messages specified in [RFC 4825]. For policy management operations, the procedures in the referred sections SHALL apply, that are relevant to handling XCAP URIs that represent XML document (i.e. policy identifiers). The procedures that are relevant to handling of XCAP URIs that are tags internal to XML documents MAY optionally be supported. The PEEM management requestor (XCAP Client) MUST be able to handle PEEM component responses to policy management operations, including error responses, which may be issued by the PEEM component (see section 5.2.3 for details).

5.2.2.1 Create Policy

The Create Policy operation SHALL follow the procedures described in [RFC 4825] for Creating a policy document, and MAY in addition optionally be used for Creating elements that are part of such a document. For this request, the HTTP PUT method is being used, where the XCAP URI parameter is a new PEEM policy identifier parameter constructed as described in section 5.1.1, identifying the location where the policy document is to be placed. An XCAP URI Node Selector MAY be used to identify elements (i.e. tags) inside the policy document. The MIME content type MUST be the type defined by the Application Usage. A successful response is represented by a 201 Created response, accompanied by an entity tag and optionally a Location header field for the document. For errors handling see section 5.2.3.

5.2.2.2 Modify Policy

The Modify Policy operation SHALL follow the procedures described in [RFC 4825] for Replacing a policy document, and MAY in addition optionally be used for Modifying elements that are part of such a document. For this request, the HTTP PUT method is being used, where the XCAP URI parameter is an existing PEEM policy identifier parameter constructed as described in section 5.1.1, identifying the location of the policy document to be replaced. An XCAP URI Node Selector MAY be used to identify elements (i.e. tags) inside the policy document. The MIME content type MUST be the type defined

by the Application Usage. A successful response is represented by a 200 OK response, and no other content. For errors handling see section 5.2.3.

5.2.2.3 Delete Policy

The Delete Policy operation SHALL follow the procedures described in [RFC 4825] for Deleting a policy document, and MAY in addition optionally be used for Deleting elements that are part of such a document. For this request, the HTTP DELETE method is being used, where the XCAP URI parameter is the PEEM policy identifier parameter constructed as described in section 5.1.1, identifying the location of the policy document to be deleted. An XCAP URI Node Selector MAY be used to identify elements (i.e. tags) inside the policy document. A successful response is represented by a 200 OK response. If the response includes an entity tag, it means that the document was not deleted, and only an element's attribute within the document was part of the deletion request. For error handling see section 5.2.3.

As a side effect of a successful Delete Policy operation (success being defined as a 200 OK response, and no accompanying entity tag), the XCAP URI can be later re-used.

5.2.2.4 View Policy

The View Policy operation SHALL follow the procedures described in [RFC 4825] for Fetching a policy document, and MAY in addition optionally be used for Fetching elements that are part of such a document. For this request, the HTTP GET method is being used, where the XCAP URI parameter is the PEEM policy identifier parameter constructed as described in section 5.1.1, identifying the location of the policy document to be retrieved. An XCAP URI Node Selector MAY be used to identify elements (i.e. tags) inside the policy document. A successful response is represented by a 200 OK response, accompanied by the returned policy as an XML document. If the response includes an entity tag, a successful response would be 200 OK, accompanied by an XML fragment representing the selected element or the element's attribute. For error handling see section 5.2.3.

5.2.3 PEM-2 Error Handling

PEEM management requestor (XCAP Client) MUST be able to handle any errors received in response of a PEM-2 request from a PEEM component (XCAP Server). See detailed error description in section 5.4.

5.3 Procedures at the PEEM component (server side)

The PEEM component acts as an XCAP Server, when handling policy management requests received via the PEM-2 interface. For handling of incoming requests, PEEM component SHALL follow the procedures described in [RFC 4825] section 6.2. The procedures that are relevant to handling XCAP URIs that represent XML document (i.e. policy identifiers) SHALL apply. The procedures that are relevant to handling of XCAP URIs that are tags internal to XML documents MAY optionally be supported. Errors returned by the PEEM component (XCAP Server) are described in section 5.3.

In particular:

- Upon receiving a Create Policy request, PEEM component SHALL create a new policy (or policy element) using the policy (or element) identifier received, store the policy (or element) identified by the XCAP URI, and acknowledge the success of the operation, or return an error. A Create Policy request is using the HTTP PUT method, and the semantics of PUT are specified in [RFC 2616]. If the Create Policy was successful, and the document interdependencies have been resolved, the PEEM component SHALL return a 201 Created. In this case, the response MUST include an entity tag and MAY include a Location header field for the document. If the Create Policy request processing failed, the PEEM component SHALL return an error, as defined in the next section.
- Upon receiving a Modify Policy request, PEEM component SHALL identify and replace an existing policy (or policy element) in its repository, with the policy (or element) identified by the XCAP URI received, and acknowledge the success of the operation, or return an error. A Modify Policy request is using the HTTP PUT method, and the semantics of PUT are specified in [RFC 2616]. If the Modify Policy was successful, and the document interdependencies have been resolved, the PEEM component SHALL return a 200 OK, and the response MUST NOT include any other content. If a Node Selector was used to modify an element or an element's attribute within the document, then the 200 OK response MUST include the entity tag of the document. If the Modify Policy request processing failed, the PEEM component SHALL return an error, as defined in the next section.

- Upon receiving a Delete Policy request, PEEM component SHALL identify and delete an existing policy (or policy element) in its repository, using the policy (or element) identified by the XCAP URI received, and acknowledge the success of the operation, or return an error. A Delete Policy request is using the HTTP DELETE method, and the semantics of DELETE are specified in [RFC 2616]. If the Delete Policy was successful, the PEEM component SHALL return a 200 OK, and the response MUST NOT include any other content, if the entire document is deleted. If a Node Selector was used to delete an element or an element's attribute within the document, but the document continues to exist at the completion of this request, then the 200 OK response MUST include the entity tag of the document. If the Delete Policy request processing failed, the PEEM component SHALL return an error, as defined in the next section.
- Upon receiving a View Policy request, PEEM component SHALL identify and retrieve an existing policy (or policy element) from its repository, using the policy (or element) identified by the XCAP URI received, and acknowledge the success of the operation, or return an error. A View Policy request is using the HTTP GET method, and the semantics of GET are specified in [RFC 2616]. If the View Policy request processing was successful, the PEEM component SHALL return a 200 OK. The MIME type of the body of the 200 OK response MUST be the MIME type defined by that Application Usage. If the View Policy failed, the PEEM component SHALL return an error, as defined in the next section. See [RFC 4825] for handling responses when the XCAP URI includes a Node Selector.

5.4 PEM-2 errors

The PEEM component acts as an XCAP Server in response to requests issued via PEM-2. As such, the errors it will return will be errors that an XCAP Server returns when handling HTTP requests for document creation, modification, deletion or retrieving (see table below).

HTTP Error Code	**HTTP Error Description**	**Received in response to PEM-2 request**	**Error explanation**	**Handling by PEEM management requestor (client side)**
400	Bad Request	Any PEM-2 request	This error is issued when any qualified names are present that use a namespace prefix, and that prefix is not defined in an xmlns() expression in the query component of the request URI.	Check [RFC 4825] for details.
404	Not Found	Any PEM-2 request	This error can be issued in one of the following cases: 1. The URI in the PEM-2 request refers to an Application Usage not understood by the PEEM component. 2. The URI in the PEM-2 request refers to a user (identified by an XUI) that is not recognized by the PEEM component. 3. The URI in the PEM-2 request includes extension-selectors that the PEEM component does not understand.	Check for the possible conditions, correct and re-issue PEM-2 request.
404	Not Found	View Policy (via HTTP GET) OR Delete Policy (via HTTP DELETE)	This error can be issued in one of the following cases: 1. The URI in the PEM-2 request contains only a document selector, but the document cannot be found. 2. the URI in the PEM-2 request contains a Node Selector, and: a. The document pointed to by the document selector cannot be found, OR b. The document pointed to by the document selector exists, but the Node Selector is a no-match or invalid (see [RFC 4825] for details.	Check for the possible conditions, correct and re-issue PEM-2 request.

405	Method Not Allowed	Invalid PEM-2 operation (via HTTP POST)	This error is issued when a PEEM component receives an HTTP POST request. HTTP POST operations are not defined in XCAP, hence not defined in PEM-2. Note: While [RFC 4825] does not define the use of HTTP POST for Creating, Replacing, Deleting or Fetching of XML documents, [XDMSPEC] specifies the use of HTTP POST for Search Operations at an XDM Aggregation Proxy. This is an XDM extension, and out of scope for PEM-2.	This is out-of-scope for XCAP and PEM-2. HTTP POST should not be used for PEM-2 operations.
405	Method Not Allowed	Create Policy (via HTTP PUT) OR Modify Policy (via HTTP PUT) OR Delete Policy (via HTTP DELETE) OR	If the request URI contained a namespace-selector, the server MUST reject the request with a 405 (Method Not Allowed) and MUST include an Allow header field including a list of valid methods for the requested resource (see [RFC 4825] and [RFC 2616] for details).	Check for the possible conditions, correct using the provided methods, and re-issue PEM-2 request.
409	Conflict	Create or Modify Policy (via HTTP PUT)	This error can be issued in several situations: 1. If the parent URI has no node selector separator, it is referring to the directory into which the document should be inserted. In normal XCAP operations, this will be either the user's home directory or the global directory, which will always exist on the server. However, if an application usage is making use of subdirectories (despite the fact that this is not recommended), it is possible that the directory into which the document should be inserted does not exist. In this case, the server MUST return a 409 response, and SHOULD include a detailed conflict report including the <no-parent> element. Detailed conflict reports are discussed in the next section. If the directory does exist, the server checks to see if there is a document with the same filename as the target node. If there is none, the operation is the creation operation. If there is such a document, the operation is the modification operation. The 409 error may be a result of the following conditions: a. The document is not a well-formed document. The error 409 will be issued, accompanied by a detailed conflict report including the <not-well-formed> element. b. The document is not UTF-8 encoded. The error 409 will be issued,	Check for the possible conditions, correct and re-issue PEM-2 request.

			accompanied by a detailed conflict report including the <not-utf-8> element. c. The document is not compliant with the schema provided in the data constraints. The error 409 will be issued, accompanied by a detailed conflict report including the <schema-validation-error> element. d. The document does not meet element uniqueness constraints provided in data constraints. The error 409 will be issued, accompanied by a detailed conflict report including the <uniqueness-failure> element. e. The document does not meet URI constraints and/or other non-schema data constraints. The error 409 will be issued, accompanied by a detailed conflict report including the <constrain-failure> element. f. Issues with attempts of creating or replacing elements or elements' attributes. These are out-of-scope for PEM-2, and may occur as a result of implementation extensions. See [RFC 4825] for explanations.	
415	Unsupported Media Type	Create or Modify Policy (via HTTP PUT)	This error is issued if the MIME type in the Content-Type header field of the PEM-2 request is not equal to the MIME type defined for the application usage.	Correct the MIME type in the Content-Type header field, and re-issue PEM-2 request.

Table 1: PEM-2 errors issued by a PEEM component (XCAP Server)

.See [RFC 4825] and [RFC 2616] for additional details, and see next section for Detailed Conflict Reports.

5.4.1 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 [RFC 4825].

The PEEM management requestor (XCAP Client) SHALL support the types of <error-element>. Other types of <error-element> elements MAY be ignored by the PEEM management requestor. It is thus RECOMMENDED that the PEEM component (XCAP Server) does not use other types of <error-element> elements than those defined in [RFC 4825].

See Appendix C for the case when PEM-2 is implemented as part of a broader XDM implementation.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-PEEM_PEM2-V1_0-20120724-A	24 Jul 2012	Status changed to Approved by TP Ref TP Doc# OMA-TP-2012-0278-INP_PEEM_V1_0_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for PEM-2 Client

Item	Function	Reference	Requirement
PEEM-PEM2-C-001-M	Capability to format Policy identification as XCAP URI	5.2.1	PEEM-PEM2-C-002-M AND PEEM-PEM2-C-003-M AND PEEM-PEM2-C-004-M
PEEM-PEM2-C-002-M	Capability to include in Policy identification URI the PEEM specified AUID	5.1.1.1	
PEEM-PEM2-C-003-M	Capability to include in Policy identification URI the PEEM specified MIME Type	5.1.1.2	
PEEM-PEM2-C-004-M	Capability to include in Policy identification URI the PEEM Default Namespace	5.1.1.3	
PEEM-PEM2-C-005-M	Issue a Create Policy request	5.1.2, 5.1.2.1	PEEM-PEM2-C-001-M
PEEM-PEM2-C-006-M	Issue a Modify Policy request	5.1.2, 5.1.2.2	PEEM-PEM2-C-001-M
PEEM-PEM2-C-007-M	Issue a Delete Policy request	5.1.2, 5.1.2.1	PEEM-PEM2-C-001-M
PEEM-PEM2-C-008-M	Issue a View Policy request	5.1.2, 5.1.2.2	PEEM-PEM2-C-001-M

B.2 SCR for PEM-2 Server

Item	Function	Reference	Requirement
PEEM-PEM2-S-001-M	Validate format of Policy identification as XCAP URI	5.2.1	PEEM-PEM2-S-002-M AND PEEM-PEM2-S-003-M AND PEEM-PEM2-S-004-M
PEEM-PEM2-S-002-M	Validate format of Policy identification URI - AUID	5.1.1.1	
PEEM-PEM2-S-003-M	Validate format of Policy identification URI – MIME Type	5.1.1.2	
PEEM-PEM2-S-004-M	Validate format of Policy identification URI – Default Namespace	5.1.1.3	
PEEM-PEM2-S-005-M	Handle Create Policy request (create new policy, or return error)	5.3, 5.4	PEEM-PEM2-S-001-M
PEEM-PEM2-S-006-M	Handle Modify Policy request (modify policy, or return error)	5.3, 5.4	PEEM-PEM2-S-001-M
PEEM-PEM2-S-007-M	Handle Delete Policy request (delete policy, or return error)	5.3, 5.4	PEEM-PEM2-S-001-M
PEEM-PEM2-S-008-M	Handle View Policy request (return policy, or return error)	5.3, 5.4	PEEM-PEM2-S-001-M

Appendix C. XDM compliant implementation implications (informative)

XDM has added specific extensions (e.g. support for Aggregation Proxy, XDM Client notification, a new Application Usage, specific constraints, detailed conflicts reports and specific authorization policies).

PEM-2 is in fact compliant to a subset (profile) of XDM, but such profile has not been clearly defined as part of [XDMSPEC]. However, it is possible that PEM-2 may be implemented as part of a broader XDM implementation. In such a case, with respect to PEM-2, a PEEM management requestor acts as an XDM Client, and PEEM component acts as an XDM Server. PEM-2 definition also needs to consider that, when implemented within a broader XDM implementation, such an implementation may include an XDM Aggregation Proxy, and therefore PEM-2 responses may include errors related to the handling of its requests by an XDM Aggregation Proxy.

C.1 Security Procedures

When implemented as part of a complete XDM implementation, authentication and authorization procedures have to be compliant to the provisions in [XDMSPEC], and authorization policies defined in [XDMSPEC] have to be supported.

Authentication between a PEEM management requestor and an entity that handles a PEM-2 request from the PEEM management requestor (e.g. the PEEM component or an XDM Aggregation Proxy) SHALL conform to the authentication specification in [XDMSPEC].

For the authorization of HTTP requests, the PEM-2 SHALL conform to the provisions in [XDMSPEC].

For XCAP Resources, Application Usages MAY define their own authorization policies. In the absence of an Application Usage specific authorization policy, the default SHALL be as indicated in [XDMSPEC].

As part of an XDM implementation, PEEM management requestor (XDM Client) MUST be able to handle any errors that may arrive to an XDM Client, since errors since PEM-2 will in most cases be implemented as part of a broader XDM implementation. That includes errors related to security procedures that may be issued by an Aggregation Proxy deployed between the PEEM management requestor (XDM Client) or by an XDM Server that incorporates the PEEM component. These additional errors are documented in the table below:

HTTP Error Code	HTTP Error Description	Received in response to PEM-2 request	Error explanation	Handling by PEEM management requestor (client side)
401	Unauthorized	Any PEM-2 request	This error may be received at an XDM Client when HTTP Digest mechanism is used for XDM Client authentication against an Aggregation Proxy or XDM Server.	See Security Procedures in [XDMSPEC] and [RFC 2617] for details.
403	Forbidden	Any PEM-2 request	This error may be received at an XDM Client after one or more failed responses to a challenge. The interpretation is that the XDM Client failed to get authorized by the XDM Server, per authorization policy defined by the target Application Usage. The exact count of challenges is decided by local policy.	See Security Procedures in [XDMSPEC] for details.

Table 2: Authentication/Authorization errors issued by an Aggregation Proxy or XDM Server

C.2 Common XDM extensions

XDM has added defined extensions to XCAP, including:

- URI lists defined in Shared List XDM Server (XDMS).
- Authorization policies
- XCAP Server Capabilities Application Usage
- Additional detailed conflict reports
- Common (OMA) content types
- Guideline on the use of Global Documents
- XDM Client notification

When implemented as part of a broader XDM implementation, PEEM management requestor and PEEM component implementations MAY need to consider the listed XDM extensions in order to be consistent with the overall implementation and/or provide additional functionality to enhance the use of PEM-2 within a broader implementation.