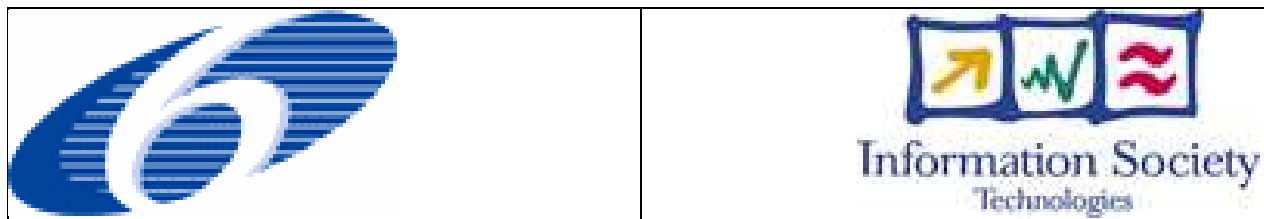


**Sixth Framework Programme
Priority IST 2.5.12
Information Society Technologies**



Integrated Project



Contract No.: 033564

**Specification of the Policy Management and Authorisation
Service**

Version 1.1

Due date of deliverable: 31/05/2009

Internal release date: 24/03/2009
Actual submission date: 24/03/2009

Document Control Page

Title	Specification of the Policy Management and Authorisation Service	
Creator	EIG	
Editor	Pascal Dihé (EIG)	
Description	This document defines a platform specific implementation specification of the Policy Management and Authorisation Service for the SANY Web Services Platform.	
Publisher	SANY Consortium	
Contributors	Julian Fischer (EIG) Thomas Berlinghoff (EIG) Thorsten Herter (EIG) Nils Steinbiß (EIG) Wenjie Ma (EIG)	
Type	Text	
Format	MS Word	
Language	EN-GB	
Creation date	2006-08-10	
Version number	1.1	
Version date	2008-02-13	
Last modified by	EIG	
Rights	Copyright "SANY Consortium". During the drafting process, access is generally limited to the SANY Partners.	
Audience	<input type="checkbox"/> internal <input checked="" type="checkbox"/> public <input type="checkbox"/> restricted, access granted to:	
Review status	<input type="checkbox"/> Draft <input checked="" type="checkbox"/> WP Manager accepted <input type="checkbox"/> SP Manager accepted <input type="checkbox"/> MB quality controlled <input type="checkbox"/> Co-ordinator accepted	Where applicable: <input type="checkbox"/> Accepted by the GA <input type="checkbox"/> Accepted by the GA as public document
Action requested	<input type="checkbox"/> to be revised by Partners involved in the preparation of the Project Deliverable <input type="checkbox"/> to be revised by all SANY Partners <input type="checkbox"/> for approval of the WP Manager <input checked="" type="checkbox"/> for approval of the SP Manager <input type="checkbox"/> for approval of the Quality Manager <input type="checkbox"/> for approval of the Project Co-ordinator <input type="checkbox"/> for approval of the General Assembly	
Requested deadline	<<dd/mm/yyyy >>	

Copyright © 2009, SANY Consortium

The SANY Consortium (www.sany-ip.eu) grants third parties the right to use and distribute all or parts of this document, provided that the SANY project and the document are properly referenced.

THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Table of Contents

1. Introduction	8
2. Overview and Architecture Outline.....	8
2.1. Role and Scope of the Policy Management and Authorisation Service.....	8
2.2. Service Specification Summary.....	9
3. Context of the Policy Management and Authorisation Service.....	11
3.1. Relations to Standards.....	11
3.2. Relations to Information Models.....	11
3.3. Relations to other Service Specifications	11
4. Specification of the Service Capabilities Interface.....	12
4.1. getCapabilities Operation.....	12
4.1.1 OA_GetCapabilitiesRequest Type	14
4.1.2 OA_GetCapabilitiesResponse Type.....	14
5. Specification of the Authorisation Interface	16
5.1. authorise Operation.....	16
5.1.1 AuthoriseRequest Type.....	17
5.1.2 ResponseDocument Type.....	19
6. Specification of the PolicyManagement Interface	20
6.1. getPolicy Operation	20
6.1.1 getPolicyRequest Type	21
6.1.2 SequenceOfPolicy Type.....	21
6.2. getPoliciesOperation	23
6.2.1 getPoliciesRequest Type	24
6.2.2 SequenceOfPolicy Type.....	24
6.3. createPolicyOperation	25
6.3.1 createPolicyRequest Type	26
6.4. deletePolicyOperation	27
6.4.1 deletePolicyRequest Type	28
6.5. updatePolicyOperation	29
6.5.1 updatePolicyRequest Type	30
7. References.....	31
8. Appendix A: XML Schema and WSDL Documents.....	33
8.1. XML Schema Documents.....	33
8.1.1 policy_author_exceptions.xsd	33
8.2. policy_author_types.xsd.....	36
8.3. policy_author_requests.xsd.....	37
8.4. Capabilities Document Template	39
8.5. WSDL Document.....	45

Table of Acronyms

LDAP	Lightweight Directory Access Protocol
OASIS	1) Open Advanced System for Disaster and Emergency Management 2) Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
ORCHESTRA	Open Architecture and Spatial Data Infrastructure for Risk Management
OSI	ORCHESTRA Service Instance
PDP	Policy Decision Point
PEP	Policy Enforcement Point
RBAC	Role Based Access Control
SAML	Security Assertion Markup Language
SensorSA	Sensor Service Architecture
UAA	User Management, Authentication and Authorisation
WSDL	Web Services Description Language
WSS	Web Services Security
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

Tables

Table 1: Specification of the getCapabilities Operation	13
Table 2: Specification of the authorise Operation.....	16
Table 3: Specification of the getPolicy Operation	21
Table 4: Specification of the getPolicy Operation	23
Table 5: Specification of the createPolicy Operation.....	25
Table 6: Specification of the deletePolicy Operation.....	27
Table 7: Specification of the updatePolicy Operation	29

Diagrams

Diagram 1: Simplified Class Diagram of the Policy Management and Authorisation Service	10
---	----

Figures

Figure 1: getCapabilities Operation	13
Figure 2: OA_GetCapabilitiesRequest.....	14
Figure 3: OA_GetCapabilitiesResponse	15
Figure 4: authorise operation.....	17
Figure 5: Authorise Request Type	18
Figure 6: AuthoriseResponse Type	19
Figure 7: getPolicy operation	21
Figure 8: getPolicyRequest Type	21
Figure 9: SequenceOfPolicy.....	22
Figure 10: getPolicies operation.....	24
Figure 11: getPoliciesRequest Type.....	24
Figure 12: SequenceOfPolicy.....	24
Figure 13: createPolicy operation	26
Figure 14: createPolicyRequest Type	26
Figure 15: deletePolicy operation	28
Figure 16: deletePolicyRequest Type	28
Figure 17: updatePolicy operation	30
Figure 18: updatePolicyRequest Type	30

1. Introduction

The present version 1.1 of the Policy Management and Authorisation Service is an advancement of the former specification of the Authorisation Service. It has been heavily reworked and consequently enhanced. The former specification of the Authorisation Service was originally based on

- the ORCHESTRA abstract specification of the Authorisation Service, Version 1.2 and
- the ORCHESTRA implementation specification of the Authorisation Service, Version 1.0

Since the Authorisation Service has originally been defined in the ORCHESTRA project as an integral part of the ORCHESTRA UAA concept, knowledge of this concept was essential for the understanding of the former specification. The new specification of the Policy Management and Authorisation Service does not have this restriction.

This document specifies the interfaces implemented by the Policy Management and Authorisation Service as well as its main purpose. It provides a formal description of the implemented operations in WSDL and XML-Schema in conformance to the guidelines and rules of the SANY W3C Web Services Platform defined in the SensorSA [SANY D2.3.3]. The overall Access Control Services Pattern applied in SANY is also described in detail in SensorSA.

The changes to the original specification of the Authorisation Service, the relations to SANY technical requirements and progress of the specification work with respect of the individual project phases are documented in the D2.4.x Deliverables.

2. Overview and Architecture Outline

The Policy Management and Authorisation Service is one of the four Access Control Services of the SensorSA.

The PolicyManagement Interface is responsible for the management of XACML policies (create, update, delete) while the Authorisation Interface is responsible for the evaluation of an authorisation request.

2.1. Role and Scope of the Policy Management and Authorisation Service

The Policy Management and Authorisation Service acts as an external policy decision point (PDP) and decides whether some identity (e.g. a user or a service) is authorised to access a certain resource. A resource in the context of the Policy Management and Authorisation Service can be an arbitrary service, a service chain, an information model or a concrete data set. Furthermore it allows the management (create, update, delete) of XACML policies.

The Authorisation Interface evaluates an authorisation request of a policy enforcement point (PEP) and returns the authorisation decision. The authorisation decision is based on the

authorisation context passed from the PEP or a security-enabled service. The authorisation context comprises the authenticated identities of the service requestor including all relevant attributes (e.g. authenticated attributes of an LDAP profile) as well as, for example, individual state variables of the service. The authorisation decision is currently provided as a compliance value indicating how to treat the request (e.g. permit or deny).

The Policy Management Interface is responsible for the management of access policies and thus plays the role of a policy information point (PIP). Access policies can be expressed in the XACML access control policy language. XACML defines also a processing model which describes how the policies shall be interpreted. The evaluation of the policies is delegated to a Policy Decision Point (PDP) which is typically a software component that can be invoked by the service implementation.

XACML allows the definition of very flexible policies that can be evaluated against any kind of environment attributes. Such environment attributes may be derived from boundary conditions of a service request as well as from the underlying data source. Environment attributes can in most cases only be determined by a service or a component that is directly involved in the invocation of a request to a secured service. This is typically the role of a PEP.

2.2. Service Specification Summary

The specification of the Policy Management and Authorisation Service is comprised of the following interfaces that are defined in distinct interface type specifications:

- The Service Capabilities Interface
- The Authorisation Interface
- The Policy Management Interface

The new Policy Management Interface completely replaces the former Permission Management Interfaces. Associations between policies and identities are stored in the policy document itself.

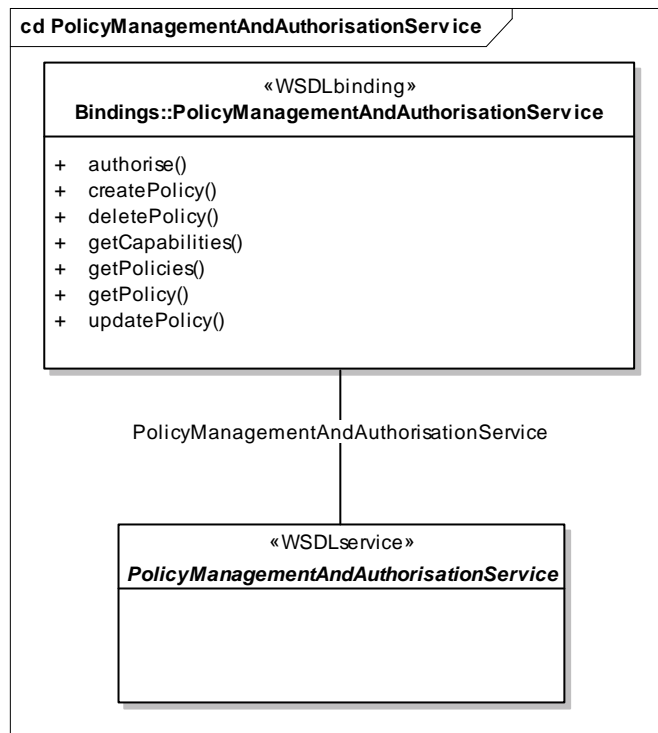


Diagram 1: Simplified Class Diagram of the Policy Management and Authorisation Service

The formal platform-specific description of the Policy Management and Authorisation Service can be found in Appendix A: XML Schema and WSDL Documents.

3. Context of the Policy Management and Authorisation Service

3.1. Relations to Standards

The Policy Management and Authorisation Service supports XACML 2.0 for the definition of policies.

- OASIS eXtensible Access Control Markup Language (XACML) TC
http://www.oasis-open.org/committees/workgroup.php?wg_abbrev=xacml

3.2. Relations to Information Models

There are no relations to information models.

3.3. Relations to other Service Specifications

Policies refer to identities which have been defined and authenticated using an Identity Management and Authentication Service.

During an authorisation request a Policy Management and Authorisation Service retrieves an authorisation request containing authenticated identities and runtime information of the service requesting the authorisation decision which is typically a policy enforcement point (PEP).

4. Specification of the Service Capabilities Interface

The Service Capabilities Interface has originally been described in the ORCHESTRA Specification of the OA Basic Service. To maintain backward compatibility to ORCHESTRA Services (e.g. the Catalogue Service), this interface remains unchanged.

4.1. getCapabilities Operation

The specification of the getCapabilities operation has been copied without any modification from the respective ORCHESTRA interface specification. SANY specific changes are only required on the level of the capabilities document itself.

The mandatory getCapabilities operation informs the client of the capabilities of an service instance. This operation takes into account that in addition to capabilities that may be common to all services in a service network a service may provide a specific set of capabilities. Furthermore, this operation allows the capabilities to be delivered according to different service meta-information schemas. This implementation specification therefore does not prescribe a certain schema to be used for the service capabilities. One or several schemas to be supported as well as a default schema have to be defined in the context of a service network.

A service meta-information document is returned to the requesting client, either complete or including selected parts according to the given sections in the request.

A request to perform the getCapabilities operation shall include the parameters listed and defined in Table 1. This table also specifies the data type (Type), the obligation [optional | mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted. Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.

Compliance	Identical to ORCHESTRA Specification			
Overrides	Not applicable			
Preconditions	none			
Post conditions	Service meta-information document returned to requesting client, either complete or including selected parts according to the given sections in the request			
Use	mandatory			
Receives	Name	Type	Use	Description
	request	OA_GetCapabilities Request	mandatory	Specifies the parts of the meta-information to be returned. If absent, all parts shall be returned using the default schema.

Returns	Type	Description
	OA_CapabilitiesDocument	Service capabilities of the specific service instance for the specific Service.
Throws	Type	Cause
	OA_InvalidParameterValue	Operation request contains an invalid parameter value. Return the name of the parameter with invalid value.
	OA_MissingParameterValue	Operation request does not include a parameter value. Return the name of the missing parameter.
	OA_NoApplicableCode	No other basic or service-specific exception type applies.
	OA_InternalError	A problem occurred in the runtime environment (e.g. out of memory).
	OA_VersionNegotiationFailed	List of versions in acceptSpecVersions parameter value in the getCapabilities request did not include any version supported by the service instance.
	OA_UnsupportedSchema	The sections parameter in the getCapabilities request referred to a schema unsupported by the service instance.

Table 1: Specification of the getCapabilities Operation

The formal platform-specific specification of the getCapabilities operation can be found in XML Schema Documents.

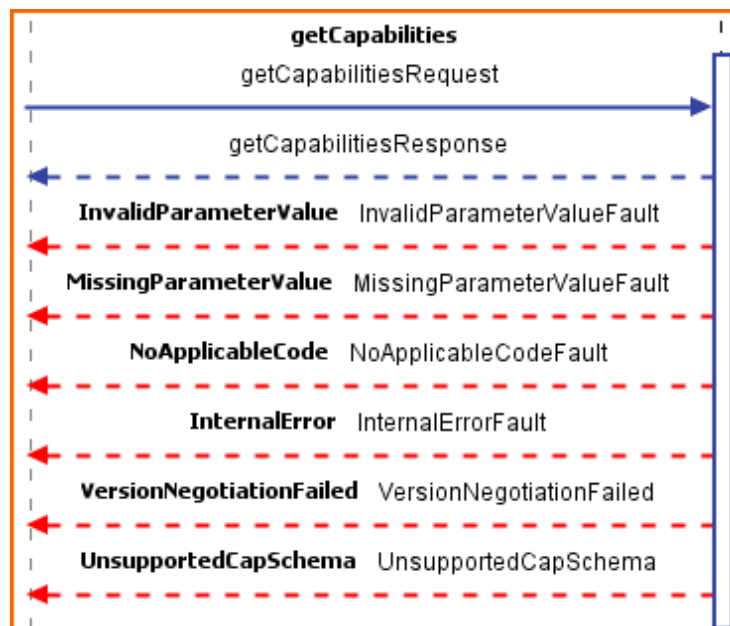


Figure 1: getCapabilities Operation

4.1.1 OA_GetCapabilitiesRequest Type

The OA_GetCapabilitiesRequest Type consists of the following elements:

- **acceptFormats:** Optional parameter containing a prioritized sequence of zero or more response formats desired by the caller, with preferred formats listed first. The formats are to be expressed in terms of MIME types. Independent of the contents of this element, the returned capabilities can always be formatted according to the default MIME type which is defined as “text/xml”.
- **acceptSpecVersions:** Optional parameter containing a prioritized sequence of zero or more specification versions of the service accepted by the client, with the preferred versions listed first. If no versions are included in the request, the highest version that the service supports shall be used.
- **sections:** Optional parameter specifying the schema for service meta-information according to which the capabilities shall be structured. In addition the names of requested sections in the complete set of meta-information elements can be listed. If absent, the complete service capabilities shall be returned structured according to a default schema.



Figure 2: OA_GetCapabilitiesRequest

4.1.2 OA_GetCapabilitiesResponse Type

The OA_GetCapabilitiesResponse Type consists of the following elements:

- **capabilitySections**: Capabilities as meta-information document which is internally structured according to the indicated format and the indicated schema. It is either complete or includes only selected parts according to the given sections in the request.
- **format**: MIME type of the format in which the capabilities are returned as value of the **capabilitySections** parameter. The value of this attribute may always denote the default MIME Type “text/xml” indicating that the capabilities are formatted in XML.
- **schemaName**: Schema according to which the capabilities are returned. The value is the same as the one contained in the **sections** parameter of the request. If not explicitly included in the request, the value indicates the default schema.
- **version**: Version number of the specification to which the delivered service meta-information document is conform.



Figure 3: OA_GetCapabilitiesResponse

A template for the service specific response of the **getCapabilities** operation is specified in chapter 8.4.

5. Specification of the Authorisation Interface

The Authorisation Interface defines operations to decide whether some identity (e.g. a user or a service) is authorised to access a certain resource.

5.1. authorise Operation

The mandatory authorise operation requests an authorisation decision for a given authorisation query. The response is generated by a XACML policy decision point (PDP) and contains a status code indicating whether the authorisation was successful or not.

A request to perform the authorise operation shall include the parameters listed and defined in Table 2. This table also specifies the data type (Type), the obligation [optional|mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted. Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.

Compliance	Identical			
Overrides	Not applicable			
Preconditions	None			
Post conditions	None			
Use	Mandatory			
Receives	Name	Type	Use	Description
	request	Authorise Request	mandatory	It contains the request document with the authorisation query.
Returns	Type		Description	
	ResponseDocument		Compliance value representing the advice on how to treat a certain service request.	
Throws	Type	Cause		
	OA_InternalError	A problem occurred in the runtime environment (e.g. out of memory).		
	OA_PermissionDenied Exception	The service requestor does not have permissions needed to perform the requested operation.		

Table 2: Specification of the authorise Operation

The formal platform-specific specification of the authorise operation can be found in Appendix A: XML Schema and WSDL Documents.

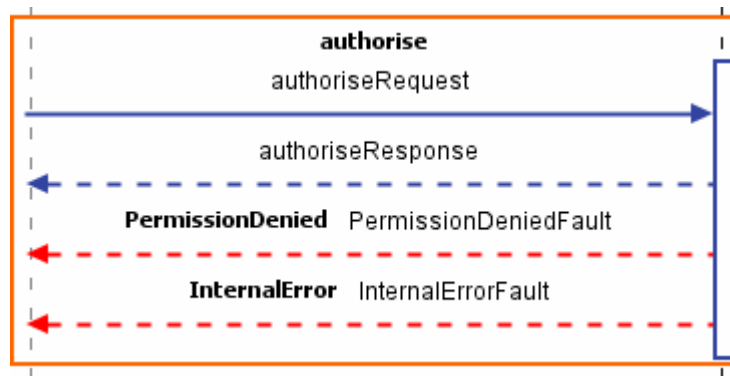


Figure 4: authorise operation

5.1.1 AuthoriseRequest Type

The AuthoriseRequest Type consists of a XACMLAuthzDecisionQuery which is a SAML Query that extends the SAML Protocol Schema. It is defined in [OASIS-SAML-Profile]. The authorisation request comprises the authenticated identities of the service requestor including all relevant attributes (e.g. authenticated attributes of an LDAP profile) as well as specific environment attributes, for example individual state variables of the service.

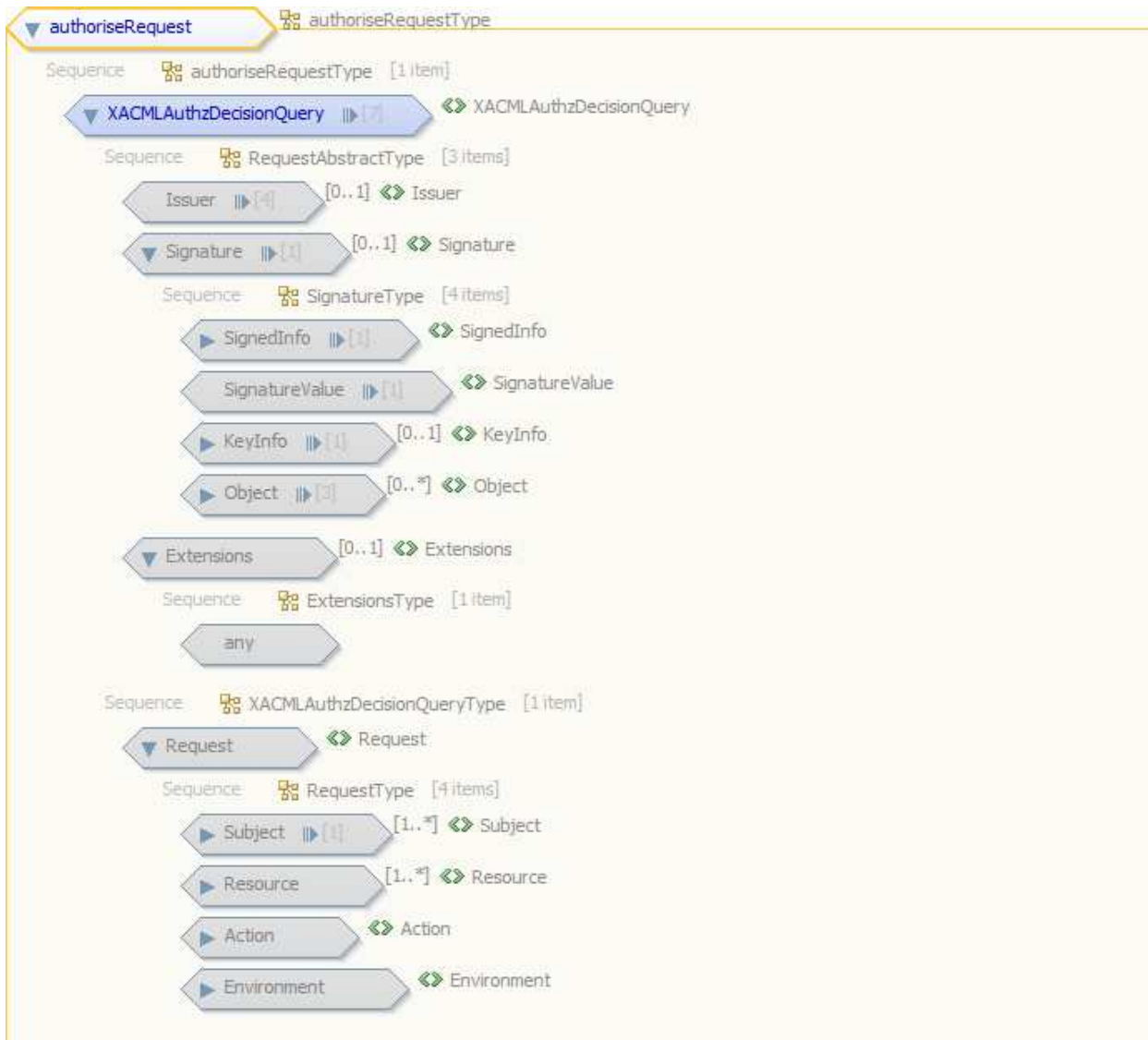


Figure 5: Authorise Request Type

5.1.2 ResponseDocument Type

The AuthoriseResponse consists of a XACML Response Document that contains the result of the authorisation decision which is currently permit or deny. It is specified in [OASIS-XACML].



Figure 6: AuthoriseResponse Type

6. Specification of the PolicyManagement Interface

The Policy Management Interface is an adaptation of the former ORCHESTRA Authorisation Management Interface. It defines operations to create and maintain policies.

6.1. getPolicy Operation

The mandatory getPolicy operation retrieves a policy specified by an id. If the id is unknown to the service instance an empty document is returned.

A request to perform the getPolicy operation shall include the parameters listed and defined in Table 3. This table also specifies the data type (Type), the obligation [optional|mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted. Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.

Compliance	Not applicable			
Overrides	Not applicable			
Preconditions	Policy has to exist.			
Post conditions	None			
Use	Mandatory			
Receives	Name	Type	Use	Description
	request	getPolicyRequest	mandatory	It contains the id of the policy to be retrieved.
Returns	Type		Description	
	SequenceOfPolicy		It contains the policy matching the given id.	
Throws	Type		Cause	
	OA_InvalidParameterValue		Operation request contains an invalid parameter value. Returns the name of the parameter with invalid value.	
	OA_MissingParameterValue		Operation request does not include a parameter value. Returns the name of the missing parameter.	
	OA_InternalError		A problem occurred in the runtime environment (e.g. out of memory).	

	OA_PermissionDeniedException	The service requestor does not have permissions needed to perform the requested operation.
--	------------------------------	--

Table 3: Specification of the getPolicy Operation

The formal platform-specific specification of the getPolicy operation can be found in Appendix A: XML Schema and WSDL Documents.

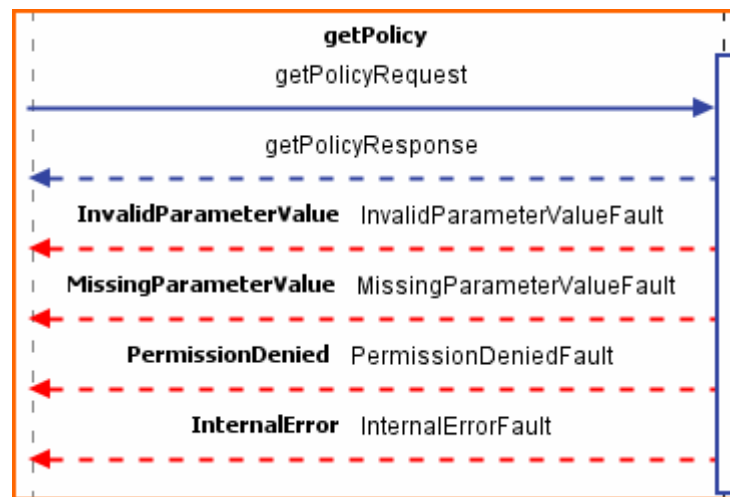


Figure 7: getPolicy operation

6.1.1 getPolicyRequest Type

The getPolicyRequest Type consists of the following elements:

- PolicyIdReference: the id of the policy to be retrieved.



Figure 8: getPolicyRequest Type

6.1.2 SequenceOfPolicy Type

The SequenceOfPolicy Type consists of the following elements:

- Policies: This object consists of a sub-element “Sequence” containing the policy as selected through the query of the request parameter (see above). The Policy Type is defined in [OASIS-XACML].

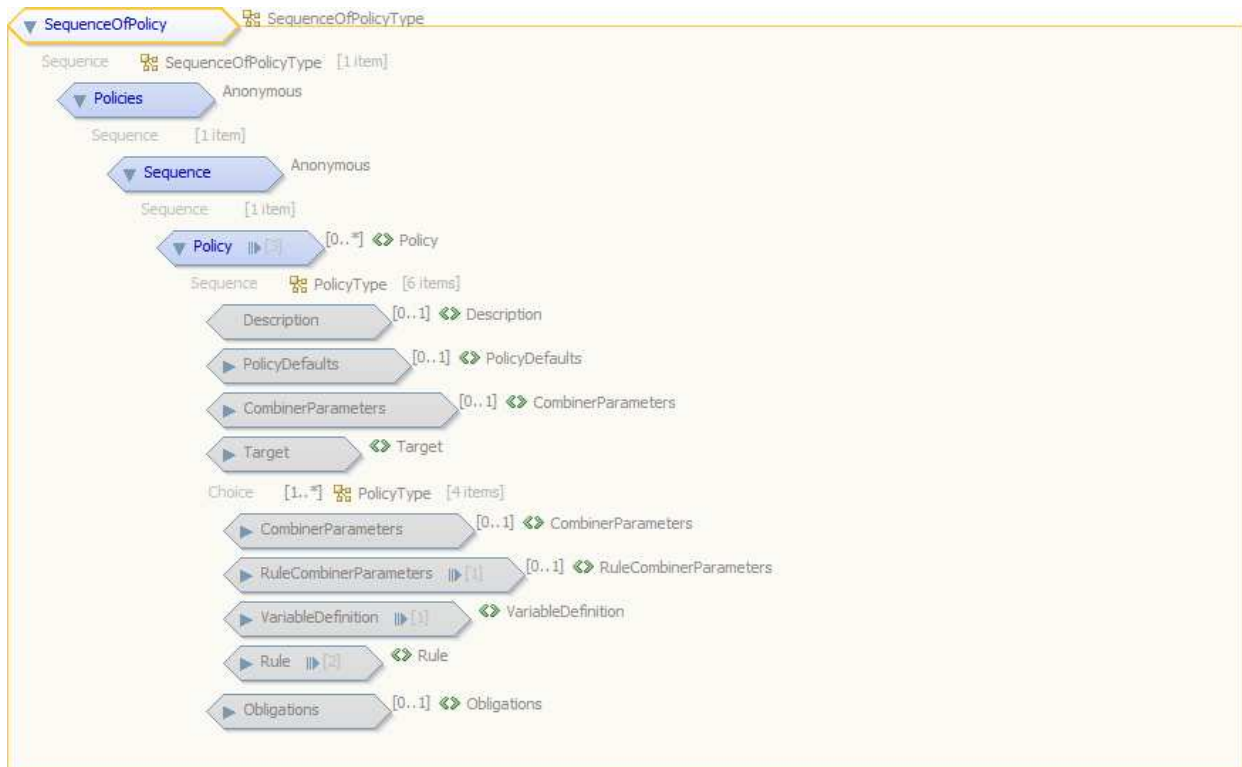


Figure 9: SequenceOfPolicy

6.2. getPoliciesOperation

The mandatory getPolicies operation retrieves all policies available in the current Policy Management Service Instance. If no policies are defined an empty document is returned.

A request to perform the getPolicies operation shall include the parameters listed and defined in Table 4 This table also specifies the data type (Type), the obligation [optional|mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted. Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.

Compliance	Not applicable			
Overrides	Not applicable			
Preconditions	Policy has to exist.			
Post conditions	None			
Use	Mandatory			
Receives	Name	Type	Use	Description
	request	getPolicies Request	mandatory	An empty request document.
Returns	Type		Description	
	SequenceOfPolicy		It contains all available policies.	
Throws	Type		Cause	
	OA_InvalidParameterValue		Operation request contains an invalid parameter value. Returns the name of the parameter with invalid value.	
	OA_MissingParameterValue		Operation request does not include a parameter value. Returns the name of the missing parameter.	
	OA_InternalError		A problem occurred in the runtime environment (e.g. out of memory).	
	OA_PermissionDeniedException		The service requestor does not have permissions needed to perform the requested operation.	

Table 4: Specification of the getPolicy Operation

The formal platform-specific specification of the getPolicy operation can be found in Appendix A: XML Schema and WSDL Documents.

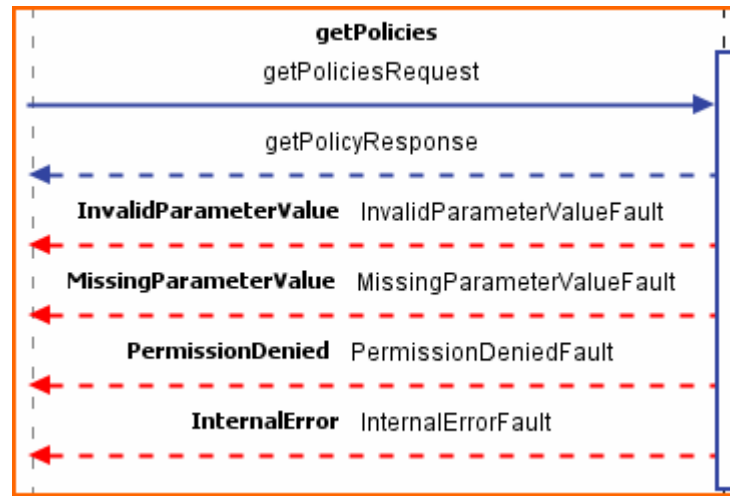


Figure 10: getPolicies operation

6.2.1 getPoliciesRequest Type

The getPoliciesRequest Type is just an empty placeholder.



Figure 11: getPoliciesRequest Type

6.2.2 SequenceOfPolicy Type

The SequenceOfPolicy Type consists of the following elements:

- Policies: This object consists of a sub-element “Sequence” containing all available policies. The Policy Type is defined in [OASIS-XACML].

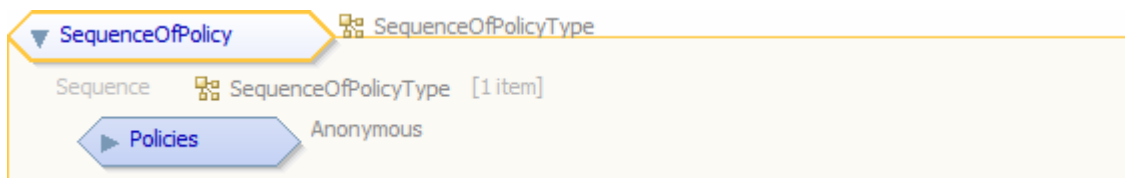


Figure 12: SequenceOfPolicy

6.3. createPolicyOperation

The mandatory createPolicy operation creates a new policy. To update an existing policy the updatePolicy operation shall be used. The policy to be created must be a valid XACML policy defining at least one resource and target section. The Policy Management Interface instance shall assign a unique id to the policy.

A request to perform the createPolicy operation shall include the parameters listed and defined in Table 5. This table also specifies the data type (Type), the obligation [optional|mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted. Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.

Compliance	Not applicable			
Overrides	Not applicable			
Preconditions	None			
Post conditions	None			
Use	Mandatory			
Receives	Name	Type	Use	Description
	request	createPolicy Request	mandatory	It contains the new policy to be created.
Returns	Type		Description	
	Not applicable		Not applicable	
Throws	Type		Cause	
	OA_InvalidParameterValue		Operation request contains an invalid parameter value. Returns the name of the parameter with invalid value.	
	OA_MissingParameterValue		Operation request does not include a parameter value. Returns the name of the missing parameter.	
	OA_InternalError		A problem occurred in the runtime environment (e.g. out of memory).	
	OA_PermissionDeniedException		The service requestor does not have permissions needed to perform the requested operation.	

Table 5: Specification of the createPolicy Operation

The formal platform-specific specification of the createPolicy operation can be found in Appendix A: XML Schema and WSDL Documents.



Figure 13: createPolicy operation

6.3.1 createPolicyRequest Type

The createPolicyRequest Type contains a Policy element which is specified in [OASIS-XACML].



Figure 14: createPolicyRequest Type

6.4. deletePolicyOperation

The mandatory deletePolicy operation deletes a policy specified by an id. If the policy to be deleted does not exist, the operation returns normally.

A request to perform the deletePolicy operation shall include the parameters listed and defined in Table 6. This table also specifies the data type (Type), the obligation [optional|mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted. Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.

Compliance	Not applicable			
Overrides	Not applicable			
Preconditions	None			
Post conditions	None			
Use	Mandatory			
Receives	Name	Type	Use	Description
	request	deletePolicy Request	mandatory	It contains the id of the policy to be deleted.
Returns	Type		Description	
	Not applicable		Not applicable	
Throws	Type		Cause	
	OA_InvalidParameterValue		Operation request contains an invalid parameter value. Returns the name of the parameter with invalid value.	
	OA_MissingParameterValue		Operation request does not include a parameter value. Returns the name of the missing parameter.	
	OA_InternalError		A problem occurred in the runtime environment (e.g. out of memory).	
	OA_PermissionDeniedException		The service requestor does not have permissions needed to perform the requested operation.	

Table 6: Specification of the deletePolicy Operation

The formal platform-specific specification of the deletePolicy operation can be found in Appendix A: XML Schema and WSDL Documents.



Figure 15: deletePolicy operation

6.4.1 deletePolicyRequest Type

The deletePolicyRequest Type consists of the following elements:

- PolicyIdReference: the id of the policy to be deleted.

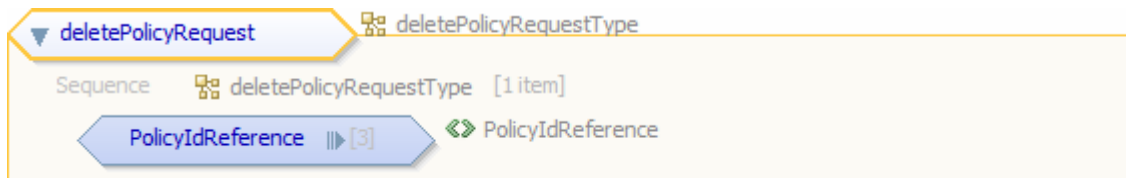


Figure 16: deletePolicyRequest Type

6.5. updatePolicyOperation

The mandatory updatePolicy operation updates an existing policy. The policy to be updated is identified by a unique id. If the id specified is unknown to the Policy Management and Authorisation Service instance an Invalid Parameter Exception is thrown.

A request to perform the getPolicy operation shall include the parameters listed and defined in Table 7. This table also specifies the data type (Type), the obligation [optional|mandatory] (Use) and a short description (Description) of each listed parameter. Furthermore the “Description” shall state the consequences for service instances if the correspondent parameter is optional and omitted. Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.

Compliance	Not applicable			
Overrides	Not applicable			
Preconditions	The policy has to exist.			
Post conditions	None			
Use	Mandatory			
Receives	Name	Type	Use	Description
	request	getPolicysRequest	mandatory	It contains the policy to be updated.
Returns	Type		Description	
	Not applicable		Not applicable	
Throws	Type		Cause	
	OA_InvalidParameterValue		Operation request contains an invalid parameter value. Returns the name of the parameter with invalid value.	
	OA_MissingParameterValue		Operation request does not include a parameter value. Returns the name of the missing parameter.	
	OA_InternalError		A problem occurred in the runtime environment (e.g. out of memory).	
	OA_PermissionDeniedException		The service requestor does not have permissions needed to perform the requested operation.	

Table 7: Specification of the updatePolicy Operation

The formal platform-specific specification of the updatePolicy operation can be found in Appendix A: XML Schema and WSDL Documents.



Figure 17: updatePolicy operation

6.5.1 updatePolicyRequest Type

The updatePolicyRequest Type contains a Policy element which is specified in [OASIS-XACML].



Figure 18: updatePolicyRequest Type

7. References

- OASIS-SAML** OASIS Standard: Security Assertion Markup Language (SAML) v2.0, published 03/2005, available from: <<http://www.oasis-open.org/specs/index.php#samlv2.0>>
- OASIS-XACML** OASIS Standard: eXtensible Access Control Markup Language TC v2.0, published 02/2005, available from: <<http://www.oasis-open.org/specs/index.php#xacmlv2.0>>
- OASIS-SAML-Profile** OASIS Standard: SAML 2.0 profile of XACML v2.0, published 01/2005, available from: <http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf>
- ORCH-Authentication-ImplSpec** Implementation Specification of the Authentication Service (Version 1.1), ORCHESTRA Consortium, Editor: Environmental Informatics Group (EIG), available from: <<http://www.eu-orchestra.org/publications.shtml#OAImplspecs>>
- ORCH-Authentication-AbstractSpec** Service Specification of the Authentication Service (Version 1.3), ORCHESTRA Consortium, Editor: Environmental Informatics Group (EIG), available from: <<http://www.eu-orchestra.org/publications.shtml#OAspecs>>
- ORCH-User-Mgmt-AbstractSpec** Service Specification of the User Management Service (Version 1.8), ORCHESTRA Consortium, Editor Environmental Informatics Group (EIG), available from: <<http://www.eu-orchestra.org/publications.shtml#OAspecs>>
- SANY D2.3.3, 2009** SANY D2.3.3 “Specification of the Sensor Service Architecture V2”, SANY IP document; available from sany website: <<http://sany-ip.eu/biblio>>
- SANY-D2.4.3, 2009** SANY D2.4.3 “Sensor Service Specification V2”, SANY IP document; available from SANY website: <<http://sany-ip.eu/biblio>>

SANY-Authentication

Specification of the Identity Management and Authentication Service (Version 1.1), SANY Consortium, Editor: Environmental Informatics Group (EIG), available from SANY website: <<http://www.sany-ip.eu/biblio>>

SANY-PEP

Specification of the Policy Enforcement Service (Version 1.1), SANY Consortium, Editor: Environmental Informatics Group (EIG), available from SANY website: <<http://www.sany-ip.eu/biblio>>

SANY-ProfileManagement

Specification of the Profile Management Service (Version 1.1), SANY Consortium, Editor: Environmental Informatics Group (EIG), available from SANY website: <<http://www.sany-ip.eu/biblio>>

8. Appendix A: XML Schema and WSDL Documents

8.1. XML Schema Documents

The following XML Schema Documents define the data types of this service.

The XML schema documents of the used data types are bundled in a zip file with the present document and can be downloaded at

<http://repository.sany-ip.eu/svn/schemas/v2/schema/security/> and from the SANY website

<http://www.sany-ip.eu/>.

In addition to XML Schema Documents specified in this appendix, this specification requires several normative XML Schema Documents. These XML Schema Documents define the common data types, e.g. common Exception Types, Basic Data Types and the GML Profile.

This file contains the XML Schema documents of the OA Basic Service and Policy Management and Authorisation Service.

The namespaces

<http://eu-orchestra.org/OA/OABasicService/types/1.0>,

<http://eu-orchestra.org/OA/OABasicService/exceptions/1.0>,

<http://www.enviomatics.net/WS/PolicyManagementAndAuthorisationService>

<http://www.enviomatics.net/WS/PolicyManagementAndAuthorisationService/requests/2.0>

<http://www.enviomatics.net/WS/PolicyManagementAndAuthorisationService/exceptions/2.0>

are used.

8.1.1 policy_author_exceptions.xsd

```
<?xml version="1.0" encoding="windows-1252"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"

xmlns:pa_exc="http://www.enviomatics.net/WS/PolicyManagementAndAuthorisation
Service/exceptions/2.0"
xmlns:oab_exc="http://eu-
orchestra.org/OA/OABasicService/exceptions/1.0"

targetNamespace="http://www.enviomatics.net/WS/PolicyManagementAndAuthorisat
ionService/exceptions/2.0"
elementFormDefault="qualified"
version="1.0">
<import namespace="http://eu-
orchestra.org/OA/OABasicService/exceptions/1.0"
```

```
schemaLocation="http://www.enviromatics.net/WS/OrchestraArchive/oa_basic_ex.xsd"/>
```

```

    <element name="PermissionDeniedException"
type="pa_exc:PermissionDeniedExceptionType"
substitutionGroup="oab_exc:OA_AbstractException"/>
    <complexType name="PermissionDeniedExceptionType">
        <complexContent>
            <extension base="oab_exc:OA_AbstractException">
                <sequence/>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="PermissionDeniedExceptionPropertyType">
        <sequence minOccurs="0">
            <element ref="pa_exc:PermissionDeniedException"/>
        </sequence>
    </complexType>

    <element name="NoSuchAssociationException"
type="pa_exc:NoSuchAssociationExceptionType"
substitutionGroup="oab_exc:OA_AbstractException"/>
    <complexType name="NoSuchAssociationExceptionType">
        <complexContent>
            <extension base="oab_exc:OA_AbstractException">
                <sequence/>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="NoSuchAssociationExceptionPropertyType">
        <sequence minOccurs="0">
            <element ref="pa_exc:NoSuchAssociationException"/>
        </sequence>
    </complexType>

    <element name="AssociationAlreadyExistsException"
type="pa_exc:AssociationAlreadyExistsExceptionType"
substitutionGroup="oab_exc:OA_AbstractException"/>
    <complexType name="AssociationAlreadyExistsExceptionType">
        <complexContent>
            <extension base="oab_exc:OA_AbstractException">
                <sequence/>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="AssociationAlreadyExistsExceptionPropertyType">
        <sequence minOccurs="0">
            <element ref="pa_exc:AssociationAlreadyExistsException"/>
        </sequence>
    </complexType>

    <element name="PermissionNotFoundException"
type="pa_exc:PermissionNotFoundExceptionType"
substitutionGroup="oab_exc:OA_AbstractException"/>
    <complexType name="PermissionNotFoundExceptionType">
        <complexContent>
            <extension base="oab_exc:OA_AbstractException">
                <sequence/>
            </extension>
        </complexContent>
    </complexType>

```

```
        </extension>
      </complexContent>
    </complexType>
    <complexType name="PermissionNotFoundExceptionPropertyType">
      <sequence minOccurs="0">
        <element ref="pa_exc:PermissionNotFoundException"/>
      </sequence>
    </complexType>

    <element name="PermissionSetNotFoundException"
type="pa_exc:PermissionSetNotFoundExceptionType"
substitutionGroup="oab_exc:OA_AbstractException"/>
    <complexType name="PermissionSetNotFoundExceptionType">
      <complexContent>
        <extension base="oab_exc:OA_AbstractException">
          <sequence/>
        </extension>
      </complexContent>
    </complexType>
    <complexType name="PermissionSetNotFoundExceptionPropertyType">
      <sequence minOccurs="0">
        <element ref="pa_exc:PermissionSetNotFoundException"/>
      </sequence>
    </complexType>
```

8.2. policy_author_types.xsd

```

<?xml version="1.0" encoding="windows-1252"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"

xmlns:author_types="http://www.enviromatics.net/WS/PolicyManagementAndAuthori
sationService/types/2.0"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"

targetNamespace="http://www.enviromatics.net/WS/PolicyManagementAndAuthorisat
ionService/types/2.0"
elementFormDefault="qualified"
version="1.0">
  <import namespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
schemaLocation="http://docs.oasis-
open.org/xacml/2.0/access_control-xacml-2.0-policy-schema-os.xsd" />

  <element name="SequenceOfPolicy"
type="author_types:SequenceOfPolicyType" />
  <complexType name="SequenceOfPolicyType">
    <sequence>
      <element name="Policies">
        <complexType>
          <sequence>
            <element name="Sequence">
              <complexType>
                <sequence>
                  <element ref="xacml:Policy" minOccurs="0"
maxOccurs="unbounded" />
                </sequence>
              </complexType>
            </element>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</schema>

```

8.3. policy_author_requests.xsd

```

<?xml version="1.0" encoding="windows-1252"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"

xmlns:ia_types="http://www.enviroomatics.net/WS/IdentityManagementAndAuthentic
ationService/types/2.0"

xmlns:pa_requests="http://www.enviroomatics.net/WS/PolicyManagementAndAuthoris
ationService/requests/2.0"
    xmlns:oab_types="http://eu-orchestra.org/OA/OABasicService/types/1.0"
    xmlns:xprofp="urn:oasis:xacml:2.0:saml:protocol:schema:os"
    xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"

targetNamespace="http://www.enviroomatics.net/WS/PolicyManagementAndAuthorisat
ionService/requests/2.0"
    elementFormDefault="qualified"
    version="1.0">
    <import
namespace="http://www.enviroomatics.net/WS/IdentityManagementAndAuthentication
Service/types/2.0"

schemaLocation="http://www.enviroomatics.net/WS/IdentityManagementAndAuthentic
ationService/types/2.0/identity_authen_types.xsd"/>
        <import namespace="http://eu-orchestra.org/OA/OABasicService/types/1.0"

schemaLocation="http://www.enviroomatics.net/WS/OrchestraArchive/oa_basic.xsd"
/>
            <import namespace="urn:oasis:xacml:2.0:saml:protocol:schema:os"
schemaLocation="http://docs.oasis-
open.org/xacml/2.0/access_control-xacml-2.0-saml-protocol-schema-os.xsd" />
                <import namespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
schemaLocation="http://docs.oasis-
open.org/xacml/2.0/access_control-xacml-2.0-policy-schema-os.xsd" />
                    <import namespace="urn:oasis:names:tc:xacml:2.0:context:schema:os"
schemaLocation="http://docs.oasis-
open.org/xacml/2.0/access_control-xacml-2.0-context-schema-os.xsd" />

            <element name="createPolicyRequest"
type="pa_requests:createPolicyRequestType"/>
                <complexType name="createPolicyRequestType">
                    <sequence>
                        <element ref="xacml:Policy"/>
                    </sequence>
                </complexType>
            <element name="deletePolicyRequest"
type="pa_requests:deletePolicyRequestType"/>
                <complexType name="deletePolicyRequestType">
                    <sequence>
                        <element ref="xacml:PolicyIdReference"/>
                    </sequence>
                </complexType>

        <!--TODO alternative Request without ReferenceID so the client needn't
read

```

```

    the Policy
  -->
  <element name="updatePolicyRequest"
type="pa_requests:updatePolicyRequestType" />
  <complexType name="updatePolicyRequestType">
    <sequence>
      <element ref="xacml:Policy" />
    </sequence>
  </complexType>
  <element name="assignPolicySetToIdentityRequest" />
  <complexType name="assignPolicySetToIdentityRequestType">
    <sequence>
      <element ref="xacml:PolicyIdReference" />
      <element name="oaidentity" type="ia_types:IdentityType" />
    </sequence>
  </complexType>
  <complexType name="unassignPolicySetFromIdentityRequestType">
    <sequence>
      <element ref="xacml:PolicyIdReference" />
      <element name="oaidentity" type="ia_types:IdentityType" />
    </sequence>
  </complexType>
  <element name="getPolicyRequest"
type="pa_requests:getPolicyRequestType" />
  <complexType name="getPolicyRequestType">
    <sequence>
      <element ref="xacml:PolicyIdReference" />
    </sequence>
  </complexType>
  <!--ToDo PolicyQuery-->
  <element name="getPoliciesRequest"
type="pa_requests:getPoliciesRequestType" />
  <complexType name="getPoliciesRequestType">
  </complexType>

  <element name="authoriseRequest"
type="pa_requests:authoriseRequestType" />
  <complexType name="authoriseRequestType">
    <sequence>
      <element ref="xprofp:XACMLAuthzDecisionQuery" />
    </sequence>
  </complexType>

  <element name="authoriseResponse"
type="pa_requests:authoriseResponseType" />
  <complexType name="authoriseResponseType">
    <sequence>
      <element ref="xacml-context:Response" />
    </sequence>
  </complexType>
</schema>

```

8.4. Capabilities Document Template

The following XML document defines a template for the capabilities document of the service:

```
<oami:OA_MI_Service_Capabilities
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://eu-orchestra.org/OAS-MI/service/1.1
http://www.enviromatics.net/WS/OrchestraArchive/oasmi.xsd
http://eu-orchestra.org/OA/OABasicService/types/1.0
http://www.enviromatics.net/WS/OrchestraArchive/oa_basic.xsd

http://www.enviromatics.net/WS/PolicyManagementAndAuthorisationService/mi/2.0
http://www.enviromatics.net/WS/PolicyManagementAndAuthorisationService/mi/2.0
/policy_author_mi.xsd
http://eu-orchestra.org/OAS-MI/service/invocation/1.1
inv.xsd"
  xmlns:oami="http://eu-orchestra.org/OAS-MI/service/1.1"
  xmlns:oabs="http://eu-orchestra.org/OA/OABasicService/types/1.0"

  xmlns:pmsami="http://www.enviromatics.net/WS/PolicyManagementAndAuthorisation
Service/mi/2.0"
  xmlns:inv="http://eu-orchestra.org/OAS-MI/service/invocation/1.1">

  <oami:serviceCommonCapabilities>
    <oami:OA_MI_Service_CommonCapabilities xmlns:oabs="http://eu-
orchestra.org/OA/OABasicService/types/1.0">
      <oami:id>Policy Management and Authorisation Service
1.0</oami:id>
      <oami:publicationDate>2009-01-13</oami:publicationDate>
      <oami:serviceDescription>The Policy Management and Authorisation
Service acts as an external policy decision point (PDP) and decides whether
some identity (e.g. a user or a service) is authorised to access a certain
resource. Furthermore it allows the management (create, update, delete) of
XACML policies.</oami:serviceDescription>
      <oami:serviceDocumentation>http://www.sany-
ip.eu</oami:serviceDocumentation>
      <oami:serviceInvocationBasic>
        <inv:OA_MI_Service_InvocationBasic>
          <inv:operation>
            <inv:OA_MI_Operation>
              <inv:accessPoints>
                <inv:OA_MI_AccessPoint>

<inv:uri>http://localhost:8080/axis2/services/PolicyManagementAndAuthorisatio
nService</inv:uri>

                </inv:OA_MI_AccessPoint>
              </inv:accessPoints>
              <inv:description>The mandatory getCapabilities
operation informs the client of the capabilities of an service instance. This
operation takes into account that in addition to capabilities that may be
common to all services in a service network a service may provide a specific
set of capabilities.</inv:description>
              <inv:name>getCapabilities</inv:name>
              <inv:parameters>
                <inv:OA_MI_OperationParameter>
                  <inv:description></inv:description>
                  <inv:direction>in</inv:direction>

```

```

        <inv:name>request</inv:name>
        <inv:optionality>>false</inv:optionality>

<inv:repeatability>>true</inv:repeatability>

<inv:valueType>OA_GetCapabilitiesRequest</inv:valueType>
    </inv:OA_MI_OperationParameter>
</inv:parameters>
<inv:parameters>
    <inv:OA_MI_OperationParameter>
        <inv:description></inv:description>
        <inv:direction>out</inv:direction>
        <inv:name>reponse</inv:name>
        <inv:optionality>>false</inv:optionality>

<inv:repeatability>>true</inv:repeatability>

<inv:valueType>OA_GetCapabilitiesResponse</inv:valueType>
    </inv:OA_MI_OperationParameter>
</inv:parameters>
</inv:OA_MI_Operation>
</inv:operation>
<inv:operation>
    <inv:OA_MI_Operation>
        <inv:accessPoints>
            <inv:OA_MI_AccessPoint>

<inv:uri>http://localhost:8080/axis2/services/PolicyManagementAndAuthorisationService</inv:uri>
            </inv:OA_MI_AccessPoint>
        </inv:accessPoints>
        <inv:description>The mandatory authorise
operation requests an authorisation decision for a given authorisation query.
The reponse is generated by a XACML policy decision point (PDP) and contains
a status code indicating whether the authorisation was successful or
not.</inv:description>
            <inv:name>authorise</inv:name>
            <inv:parameters>
                <inv:OA_MI_OperationParameter>
                    <inv:description>The request document
with the authorisation query.</inv:description>
                    <inv:direction>in</inv:direction>
                    <inv:name>request</inv:name>
                    <inv:optionality>>false</inv:optionality>

<inv:repeatability>>true</inv:repeatability>

<inv:valueType>AuthoriseRequest</inv:valueType>
    </inv:OA_MI_OperationParameter>
</inv:parameters>
<inv:parameters>
    <inv:OA_MI_OperationParameter>
        <inv:description>Compliance value
representing the advice how to treat a certain service
request.</inv:description>
            <inv:direction>out</inv:direction>
            <inv:name>response</inv:name>
            <inv:optionality>>false</inv:optionality>

```

```

<inv:repeatability>>true</inv:repeatability>

<inv:valueType>ResponseDocument</inv:valueType>
  </inv:OA_MI_OperationParameter>
  </inv:parameters>
  </inv:OA_MI_Operation>
</inv:operation>

  <inv:operation>
    <inv:OA_MI_Operation>
      <inv:accessPoints>
        <inv:OA_MI_AccessPoint>

<inv:uri>http://localhost:8080/axis2/services/PolicyManagementAndAuthorisatio
nService</inv:uri>
          </inv:OA_MI_AccessPoint>
        </inv:accessPoints>
        <inv:description>The mandatory getPolicy
operation retrieves a policy specified by an id. If the id is unknown to the
service instance an empty document is returned.</inv:description>
        <inv:name>getPolicy</inv:name>
        <inv:parameters>
          <inv:OA_MI_OperationParameter>
            <inv:description>Contains the id of the
policy to be retrieved.</inv:description>
            <inv:direction>in</inv:direction>
            <inv:name>request</inv:name>
            <inv:optionality>>false</inv:optionality>

<inv:repeatability>>true</inv:repeatability>

<inv:valueType>getPolicyRequest</inv:valueType>
  </inv:OA_MI_OperationParameter>
  </inv:parameters>
  <inv:parameters>
    <inv:OA_MI_OperationParameter>
      <inv:description>Contains the policy
matching the given id.</inv:description>
      <inv:direction>out</inv:direction>
      <inv:name>response</inv:name>
      <inv:optionality>>false</inv:optionality>

<inv:repeatability>>true</inv:repeatability>

<inv:valueType>SequenceOfPolicy</inv:valueType>
  </inv:OA_MI_OperationParameter>
  </inv:parameters>
  </inv:OA_MI_Operation>
</inv:operation>

  <inv:operation>
    <inv:OA_MI_Operation>
      <inv:accessPoints>
        <inv:OA_MI_AccessPoint>

<inv:uri>http://localhost:8080/axis2/services/PolicyManagementAndAuthorisatio
nService</inv:uri>
          </inv:OA_MI_AccessPoint>

```

```

        </inv:accessPoints>
        <inv:description>The mandatory getPolicies
operation retrieves all policies available in the current Policy Management
Service Instance. If no policies are defined an empty document is
returned.</inv:description>
        <inv:name>getPolicies</inv:name>
        <inv:parameters>
            <inv:OA_MI_OperationParameter>
                <inv:description>An empty request
document.</inv:description>
                <inv:direction>in</inv:direction>
                <inv:name>request</inv:name>
                <inv:optionality>>false</inv:optionality>
        </inv:parameters>
        <inv:repeatability>>true</inv:repeatability>
        <inv:valueType>getPolicyRequest</inv:valueType>
            </inv:OA_MI_OperationParameter>
        </inv:parameters>
        <inv:parameters>
            <inv:OA_MI_OperationParameter>
                <inv:description>Contains all available
policies.</inv:description>
                <inv:direction>out</inv:direction>
                <inv:name>response</inv:name>
                <inv:optionality>>false</inv:optionality>
            </inv:OA_MI_OperationParameter>
        </inv:parameters>
        <inv:repeatability>>true</inv:repeatability>
        <inv:valueType>SequenceOfPolicy</inv:valueType>
            </inv:OA_MI_OperationParameter>
        </inv:parameters>
        </inv:OA_MI_Operation>
    </inv:operation>
    <inv:operation>
        <inv:OA_MI_Operation>
            <inv:accessPoints>
                <inv:OA_MI_AccessPoint>
                    <inv:uri>http://localhost:8080/axis2/services/PolicyManagementAndAuthorisatio
nService</inv:uri>
                </inv:OA_MI_AccessPoint>
            </inv:accessPoints>
            <inv:description>The mandatory createPolicy
operation creates a new policy. To update an existing policy the updatePolicy
operation shall be used. The policy to be created must be a valid XACML
policy defining at least one resource and target section. The Policy
Management Interface instance assigns a unique id with the policy any id set
in the policy document itself is ignored.</inv:description>
            <inv:name>createPolicyOperation</inv:name>
            <inv:parameters>
                <inv:OA_MI_OperationParameter>
                    <inv:description>Contains the new policy
to be created.</inv:description>
                    <inv:direction>in</inv:direction>
                    <inv:name>request</inv:name>
                    <inv:optionality>>false</inv:optionality>
                </inv:OA_MI_OperationParameter>
            </inv:parameters>
        </inv:OA_MI_Operation>
    </inv:operation>

```

```

<inv:repeatability>>true</inv:repeatability>

<inv:valueType>createPolicyRequest</inv:valueType>
  </inv:OA_MI_OperationParameter>
  </inv:parameters>
  </inv:OA_MI_Operation>
</inv:operation>

  <inv:operation>
    <inv:OA_MI_Operation>
      <inv:accessPoints>
        <inv:OA_MI_AccessPoint>

<inv:uri>http://localhost:8080/axis2/services/PolicyManagementAndAuthorisatio
nService</inv:uri>
          </inv:OA_MI_AccessPoint>
        </inv:accessPoints>
        <inv:description>The mandatory deletePolicy
operation deletes a policy specified by an id. If the policy to be deleted
does not exist, the operation returns normally.</inv:description>
        <inv:name>deletePolicy</inv:name>
        <inv:parameters>
          <inv:OA_MI_OperationParameter>
            <inv:description>Contains the id of the
policy to be deleted.</inv:description>
            <inv:direction>in</inv:direction>
            <inv:name>request</inv:name>
            <inv:optionality>>false</inv:optionality>

<inv:repeatability>>true</inv:repeatability>

<inv:valueType>deletePolicyRequest</inv:valueType>
  </inv:OA_MI_OperationParameter>
  </inv:parameters>
  </inv:OA_MI_Operation>
</inv:operation>

  <inv:operation>
    <inv:OA_MI_Operation>
      <inv:accessPoints>
        <inv:OA_MI_AccessPoint>

<inv:uri>http://localhost:8080/axis2/services/PolicyManagementAndAuthorisatio
nService</inv:uri>
          </inv:OA_MI_AccessPoint>
        </inv:accessPoints>
        <inv:description>The mandatory updatePolicy
operation updates an existing policy. The policy to be updated is identified
by a unique id. If the id specified is unknown to the Policy Management and
Authroisation Service instance an Invalid Parameter Exception is
thrown.</inv:description>
        <inv:name>updatePolicy</inv:name>
        <inv:parameters>
          <inv:OA_MI_OperationParameter>
            <inv:description>Contains policy to be
updated.</inv:description>
            <inv:direction>in</inv:direction>
            <inv:name>request</inv:name>

```

```

        <inv:optionality>>false</inv:optionality>

<inv:repeatability>>true</inv:repeatability>

<inv:valueType>updatePolicyRequest</inv:valueType>
    </inv:OA_MI_OperationParameter>
    </inv:parameters>
    </inv:OA_MI_Operation>
</inv:operation>

    </inv:OA_MI_Service_InvocationBasic>
</oami:serviceInvocationBasic>
    <oami:serviceName>Policy Management And Authorisation
Service</oami:serviceName>
    <oami:serviceSpecVersion>1.0</oami:serviceSpecVersion>
    <oami:serviceType>
        <oabs:identifier>Policy Management And Authorisation
Service</oabs:identifier>
    </oami:serviceType>
    </oami:OA_MI_Service_CommonCapabilities>
    </oami:serviceCommonCapabilities>
</oami:OA_MI_Service_Capabilities>

```

8.5. WSDL Document

The following WSDL Version 1.1 document is the formal specification of the Policy Management and Authorisation Service according to rules of the “SANY Web Services Platform”. It defines the mandatory SOAP binding.

The WSDL document is bundled in a zip file with the present document and can be downloaded at

<http://repository.sany-ip.eu/svn/schemas/v2/wSDL/security/> and from the SANY website

<http://www.sany-ip.eu/>.

```
<?xml version="1.0"?>
<wSDL:definitions xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"

  xmlns:pa="http://www.enviroomatics.net/WS/PolicyManagementAndAuthorisationService"

  xmlns:pa_requests="http://www.enviroomatics.net/WS/PolicyManagementAndAuthorisationService/requests/2.0"

  xmlns:pa_exc="http://www.enviroomatics.net/WS/PolicyManagementAndAuthorisationService/exceptions/2.0"

  xmlns:pa_types="http://www.enviroomatics.net/WS/PolicyManagementAndAuthorisationService/types/2.0"
    xmlns:oab_exc="http://eu-orchestra.org/OA/OABasicService/exceptions/1.0"
    xmlns:oab_types="http://eu-orchestra.org/OA/OABasicService/types/1.0"
    xmlns:bdt="http://eu-orchestra.org/basicTypes/1.0"
    xmlns:xprofp="urn:oasis:xacml:2.0:saml:protocol:schema:os"
    xmlns:xprofa="urn:oasis:xacml:2.0:saml:assertion:schema:os"
    name="PolicyManagementAndAuthorisationServiceInstance"

  targetNamespace="http://www.enviroomatics.net/WS/PolicyManagementAndAuthorisationService" xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype">
  <wSDL:types>
    <xs:schema
  targetNamespace="http://www.enviroomatics.net/WS/PolicyManagementAndAuthorisationService" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="qualified">
    <xs:import namespace="http://eu-orchestra.org/basicTypes/1.0"

  schemaLocation="http://www.enviroomatics.net/WS/OrchestraArchive/basic_types.xsd"/>
    <xs:import namespace="http://eu-orchestra.org/OA/OABasicService/exceptions/1.0"
```

```

schemaLocation="http://www.enviromatics.net/WS/OrchestraArchive/oa_basic_ex.x
sd"/>
    <xs:import namespace="http://eu-
orchestra.org/OA/OABasicService/types/1.0"

schemaLocation="http://www.enviromatics.net/WS/OrchestraArchive/oa_basic.xsd"
/>
    <xs:import
namespace="http://www.enviromatics.net/WS/PolicyManagementAndAuthorisationSer
vice/types/2.0"

schemaLocation="http://www.enviromatics.net/WS/PolicyManagementAndAuthorisati
onService/types/2.0/policy_author_types.xsd"/>
    <xs:import
namespace="http://www.enviromatics.net/WS/PolicyManagementAndAuthorisationSer
vice/requests/2.0"

schemaLocation="http://www.enviromatics.net/WS/PolicyManagementAndAuthorisati
onService/requests/2.0/policy_author_requests.xsd"/>
    <xs:import
namespace="http://www.enviromatics.net/WS/PolicyManagementAndAuthorisationSer
vice/exceptions/2.0"

schemaLocation="http://www.enviromatics.net/WS/PolicyManagementAndAuthorisati
onService/exceptions/2.0/policy_author_exceptions.xsd"/>
    <xs:import
namespace="urn:oasis:xacml:2.0:saml:protocol:schema:os"
        schemaLocation="http://docs.oasis-
open.org/xacml/2.0/access_control-xacml-2.0-saml-protocol-schema-os.xsd"/>
    <xs:import
namespace="urn:oasis:xacml:2.0:saml:assertion:schema:os"
        schemaLocation="http://docs.oasis-
open.org/xacml/2.0/access_control-xacml-2.0-saml-assertion-schema-os.xsd"/>
    </xs:schema>
</wsdl:types>
<wsdl:message name="getCapabilitiesRequest">
    <wsdl:part name="request"
element="oab_types:OA_GetCapabilitiesRequest"/>
</wsdl:message>
<wsdl:message name="getCapabilitiesResponse">
    <wsdl:part name="response"
element="oab_types:OA_GetCapabilitiesResponse"/>
</wsdl:message>
<wsdl:message name="authoriseRequest">
    <wsdl:part name="request" element="xprofp:XACMLAuthzDecisionQuery"/>
</wsdl:message>
<wsdl:message name="authoriseResponse">
    <wsdl:part name="response"
element="xprofa:XACMLAuthzDecisionStatement"/>
</wsdl:message>
<wsdl:message name="createPolicyRequest">
    <wsdl:part name="request" element="pa_requests:createPolicyRequest"/>
</wsdl:message>
<wsdl:message name="deletePolicyRequest">
    <wsdl:part name="request" element="pa_requests:deletePolicyRequest"/>
</wsdl:message>
<wsdl:message name="updatePolicyRequest">
    <wsdl:part name="request" element="pa_requests:updatePolicyRequest"/>

```

```

</wsdl:message>
<wsdl:message name="getPolicyRequest">
  <wsdl:part name="request" element="pa_requests:getPolicyRequest" />
</wsdl:message>
<wsdl:message name="getPolicyResponse">
  <wsdl:part name="response" element="pa_types:SequenceOfPolicy" />
</wsdl:message>
<wsdl:message name="getPoliciesRequest">
  <wsdl:part name="request" element="pa_requests:getPoliciesRequest" />
</wsdl:message>
<wsdl:message name="InvalidParameterValueFault">
  <wsdl:part name="OA_InvalidParameterValue"
element="oab_exc:OA_InvalidParameterValue" />
</wsdl:message>
<wsdl:message name="MissingParameterValueFault">
  <wsdl:part name="OA_MissingParameterValue"
element="oab_exc:OA_MissingParameterValue" />
</wsdl:message>
<wsdl:message name="NoApplicableCodeFault">
  <wsdl:part name="OA_NoApplicableCode"
element="oab_exc:OA_NoApplicableCode" />
</wsdl:message>
<wsdl:message name="InternalErrorFault">
  <wsdl:part name="OA_InternalError"
element="oab_exc:OA_InternalError" />
</wsdl:message>
<wsdl:message name="PermissionDeniedFault">
  <wsdl:part name="PermissionDeniedException"
element="pa_exc:PermissionDeniedException" />
</wsdl:message>
<wsdl:message name="VersionNegotiationFailed">
  <wsdl:part name="OA_VersionNegotiationFailed"
element="oab_exc:OA_VersionNegotiationFailed" />
</wsdl:message>
<wsdl:message name="UnsupportedCapSchema">
  <wsdl:part name="OA_UnsupportedCapSchema"
element="oab_exc:OA_UnsupportedCapSchema" />
</wsdl:message>
<wsdl:portType name="PolicyManagementAndAuthorisationService">
  <wsdl:operation name="getCapabilities">
    <wsdl:input name="capabilitiesRequest"
message="pa:getCapabilitiesRequest" />
    <wsdl:output name="capabilitiesResponse"
message="pa:getCapabilitiesResponse" />
    <wsdl:fault name="InvalidParameterValue"
message="pa:InvalidParameterValueFault" />
    <wsdl:fault name="MissingParameterValue"
message="pa:MissingParameterValueFault" />
    <wsdl:fault name="NoApplicableCode"
message="pa:NoApplicableCodeFault" />
    <wsdl:fault name="InternalError"
message="pa:InternalErrorFault" />
    <wsdl:fault name="VersionNegotiationFailed"
message="pa:VersionNegotiationFailed" />
    <wsdl:fault name="UnsupportedCapSchema"
message="pa:UnsupportedCapSchema" />
  </wsdl:operation>
  <wsdl:operation name="authorise">

```

```

        <wsdl:input name="authoriseRequest"
message="pa:authoriseRequest" />
        <wsdl:output name="authoriseRequestResponse"
message="pa:authoriseResponse" />
        <wsdl:fault name="PermissionDenied"
message="pa:PermissionDeniedFault" />
        <wsdl:fault name="InternalError"
message="pa:InternalErrorFault" />
    </wsdl:operation>
    <wsdl:operation name="createPolicy">
        <wsdl:input name="createPolicyRequest"
message="pa:createPolicyRequest" />
        <wsdl:fault name="InvalidParameterValue"
message="pa:InvalidParameterValueFault" />
        <wsdl:fault name="MissingParameterValue"
message="pa:MissingParameterValueFault" />
        <wsdl:fault name="PermissionDenied"
message="pa:PermissionDeniedFault" />
        <wsdl:fault name="InternalError"
message="pa:InternalErrorFault" />
    </wsdl:operation>
    <wsdl:operation name="deletePolicy">
        <wsdl:input name="deletePolicyRequest"
message="pa:deletePolicyRequest" />
        <wsdl:fault name="InvalidParameterValue"
message="pa:InvalidParameterValueFault" />
        <wsdl:fault name="MissingParameterValue"
message="pa:MissingParameterValueFault" />
        <wsdl:fault name="PermissionDenied"
message="pa:PermissionDeniedFault" />
        <wsdl:fault name="InternalError"
message="pa:InternalErrorFault" />
    </wsdl:operation>
    <wsdl:operation name="updatePolicy">
        <wsdl:input name="updatePolicyRequest"
message="pa:updatePolicyRequest" />
        <wsdl:fault name="InvalidParameterValue"
message="pa:InvalidParameterValueFault" />
        <wsdl:fault name="MissingParameterValue"
message="pa:MissingParameterValueFault" />
        <wsdl:fault name="PermissionDenied"
message="pa:PermissionDeniedFault" />
        <wsdl:fault name="InternalError"
message="pa:InternalErrorFault" />
    </wsdl:operation>
    <wsdl:operation name="getPolicy">
        <wsdl:input name="getPolicyRequest"
message="pa:getPolicyRequest" />
        <wsdl:output name="policyResponse"
message="pa:getPolicyResponse" />
        <wsdl:fault name="InvalidParameterValue"
message="pa:InvalidParameterValueFault" />
        <wsdl:fault name="MissingParameterValue"
message="pa:MissingParameterValueFault" />
        <wsdl:fault name="PermissionDenied"
message="pa:PermissionDeniedFault" />
        <wsdl:fault name="InternalError"
message="pa:InternalErrorFault" />
    </wsdl:operation>

```

```

        <wsdl:operation name="getPolicies">
            <wsdl:input name="getPoliciesRequest"
message="pa:getPoliciesRequest" />
            <wsdl:output name="policyResponse"
message="pa:getPolicyResponse" />
            <wsdl:fault name="InvalidParameterValue"
message="pa:InvalidParameterValueFault" />
            <wsdl:fault name="MissingParameterValue"
message="pa:MissingParameterValueFault" />
            <wsdl:fault name="PermissionDenied"
message="pa:PermissionDeniedFault" />
            <wsdl:fault name="InternalError"
message="pa:InternalErrorFault" />
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="PolicyManagementAndAuthorisationService"
type="pa:PolicyManagementAndAuthorisationService">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
        <wsdl:operation name="getCapabilities">
            <soap:operation soapAction="getCapabilities" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
            <wsdl:fault name="InvalidParameterValue">
                <soap:fault name="InvalidParameterValue" use="literal" />
            </wsdl:fault>
            <wsdl:fault name="MissingParameterValue">
                <soap:fault name="MissingParameterValue" use="literal" />
            </wsdl:fault>
            <wsdl:fault name="NoApplicableCode">
                <soap:fault name="NoApplicableCode" use="literal" />
            </wsdl:fault>
            <wsdl:fault name="InternalError">
                <soap:fault name="InternalError" use="literal" />
            </wsdl:fault>
            <wsdl:fault name="UnsupportedCapSchema">
                <soap:fault name="UnsupportedCapSchema" use="literal" />
            </wsdl:fault>
            <wsdl:fault name="VersionNegotiationFailed">
                <soap:fault name="VersionNegotiationFailed" use="literal" />
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="authorise">
            <soap:operation soapAction="authorise" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
            <wsdl:fault name="PermissionDenied">
                <soap:fault name="PermissionDenied" use="literal" />
            </wsdl:fault>
            <wsdl:fault name="InternalError">
                <soap:fault name="InternalError" use="literal" />
            </wsdl:fault>
        </wsdl:operation>
    </wsdl:binding>
</wsdl:service>

```

```

        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="createPolicy">
        <soap:operation soapAction="createPolicyRequest"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:fault name="InvalidParameterValue">
            <soap:fault name="InvalidParameterValue" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="MissingParameterValue">
            <soap:fault name="MissingParameterValue" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="PermissionDenied">
            <soap:fault name="PermissionDenied" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="InternalError">
            <soap:fault name="InternalError" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="deletePolicy">
        <soap:operation soapAction="deletePolicyRequest"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:fault name="InvalidParameterValue">
            <soap:fault name="InvalidParameterValue" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="MissingParameterValue">
            <soap:fault name="MissingParameterValue" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="PermissionDenied">
            <soap:fault name="PermissionDenied" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="InternalError">
            <soap:fault name="InternalError" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="updatePolicy">
        <soap:operation soapAction="updatePolicyRequest"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:fault name="InvalidParameterValue">
            <soap:fault name="InvalidParameterValue" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="MissingParameterValue">
            <soap:fault name="MissingParameterValue" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="PermissionDenied">
            <soap:fault name="PermissionDenied" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="InternalError">
            <soap:fault name="InternalError" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>

```

```

<wsdl:operation name="getPolicy">
  <soap:operation soapAction="getPolicyRequest" style="document"/>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:fault name="InvalidParameterValue">
    <soap:fault name="InvalidParameterValue" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="MissingParameterValue">
    <soap:fault name="MissingParameterValue" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="PermissionDenied">
    <soap:fault name="PermissionDenied" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="InternalError">
    <soap:fault name="InternalError" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getPolicies">
  <soap:operation soapAction="getPoliciesRequest"
style="document"/>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:fault name="InvalidParameterValue">
    <soap:fault name="InvalidParameterValue" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="MissingParameterValue">
    <soap:fault name="MissingParameterValue" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="PermissionDenied">
    <soap:fault name="PermissionDenied" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="InternalError">
    <soap:fault name="InternalError" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="PolicyManagementAndAuthorisationService">
  <wsdl:port name="PolicyManagementAndAuthorisationService"
binding="pa:PolicyManagementAndAuthorisationService">
    <soap:address
location="http://localhost:8080/axis2/services/PolicyManagementAndAuthorisati
onService"/>
  </wsdl:port>
</wsdl:service>
<plnk:partnerLinkType name="PolicyManagementAndAuthorisationService">
  <plnk:role name="role1"
portType="pa:PolicyManagementAndAuthorisationService"/>
</plnk:partnerLinkType>
</wsdl:definitions>

```