

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



Integrated Project



Contract No.: 033564

**Specification of the Profile Management 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 Profile Management Service	
Creator	EIG	
Editor	Pascal Dihé (EIG)	
Description	This document defines the implementation specification of the Profile Management Service for the SANY Web Service 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	2009-02-02	
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 checked="" type="checkbox"/> Draft <input 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 Profile Management Service	8
2.1.1 Profiles and Identities.....	9
2.2. Service Specification Summary.....	9
3. Context of the Profile Management 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 Profile Management Interface	16
5.1. createProfile Operation	16
5.1.1 createProfileRequest Type.....	17
5.1.2 createProfileResponse Type.....	18
5.2. deleteProfile Operation.....	20
5.2.1 deleteProfileRequest Type	21
5.3. updateProfile Operation	22
5.3.1 updateProfileRequest Type.....	23
5.4. getProfiles Operation.....	24
5.4.1 getProfilesRequest Type	25
5.4.2 getProfilesResponse Type	26
5.5. addIdentityToProfile Operation.....	27
5.5.1 addIdentityToProfile Operation.....	28
5.6. removeIdentityFromProfile Operation.....	30
5.6.1 removeIdentityFromProfile Type	31
6. References	32
7. Appendix A: XML Schema and WSDL Documents.....	33
7.1. XML Schema Documents.....	33
7.1.1 profilemanagement_exceptions.xsd.....	33
7.1.2 profilemanagement_requests.xsd	35
7.1.3 profilemanagement_types.xsd	37
7.2. WSDL Document.....	41
7.3. Capabilities Document Template	50
8. Appendix B: Default LDAP Schema.....	57

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
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
XML	eXtensible Markup Language

Tables

Table 1: Specification of the getCapabilities Operation	13
Table 2: Specification of the createProfile Operation	17
Table 3: Specification of the deleteProfile Operation	21
Table 4: Specification of the updateProfile Operation	22
Table 5: Specification of the getProfiles Operation	25
Table 6: Specification of the addIdentityToProfile Operation	28
Table 7: Specification of the removeIdentityFromProfile Operation	31
Table 8: Default LDAP Schema	63

Figures

Figure 1: getCapabilities Operation	13
Figure 2: OA_GetCapabilitiesRequest	14
Figure 3: OA_GetCapabilitiesResponse	15
Figure 4: createProfile Operation	17
Figure 5: createProfileRequest	18
Figure 6: KeyVectorProfileAttributesType	18
Figure 7: createProfileResponse	19
Figure 8: deleteProfile Operation	21
Figure 9: deleteProfileRequest	21
Figure 10 : updateProfile operation	23
Figure 11 : updateProfileRequest	23
Figure 12: getProfiles operation	25
Figure 13: getProfilesRequest	25
Figure 14: getProfilesResponse	26
Figure 15: addIdentityToProfile operation	28
Figure 16: addIdentityToProfileRequest	29
Figure 17: removeIdentityFromProfile operation	31
Figure 18: removeIdentityFromProfileRequest	31

Diagrams

Diagram 1: Simplified class diagram of the Profile Management Service	10
---	----

1. Introduction

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

- the ORCHESTRA abstract specification of the User Management Service, Version 1.8 and
- the ORCHESTRA implementation specification of the User Management Service, Version 1.0.

Since the User Management 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. Although the new specification of the Profile Management Service does not have this restriction, a basic understanding of the concept of profiles and identities is required. Therefore those concepts and also the collaboration of the SANY Access Control Services are explained briefly in the subsequent chapters and more detailed in the Specification of the Sensor Service Architecture (SensorSA) [SANY D2.3.3].

This document specifies the interfaces implemented by the Profile Management Service as well as their main purpose and 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.

The changes to the original specification of the User Management 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

2.1. Role and Scope of the Profile Management Service

The Profile Management Service is one of the four Access Control Services of the SensorSA. Together with the Identity Management Service and the underlying information model it forms an abstract user concept that allows the separation of user profiles and identities.

The Profile Management Service is responsible for the management of profiles (create, update, delete) and their associations to certain identities. The identities themselves are stored and managed in instances of the Identity Management Interface.

The clear separation of identity management (Identity Management and Authentication Service) and profile management (Profile Management Service) enables the distributed and independent management of access rights. One user (a single profile) may have several identities and thus different access rights in disparate systems or networks.

2.1.1 Profiles and Identities

A Profile is an abstraction of a real-world user (or service) and represents an entity that needs to be authenticated in order to perform an access controlled action (e.g. a service request). However, profiles are not authenticated themselves but rather represent a point of contact and management feature for authentication and authorisation purposes. By separating identities from profiles a profile is decoupled from authentication. This enables one user (a single profile) to have several identities (multiple identities) and thus different access rights in disparate systems or networks.

Identities are created in instances of Identity Management Services. A profile may be associated with identities in arbitrary Identity ManagementService instances.

A profile is composed of several profile attributes and is bound to one or more identities. The profile attributes correspond to the properties of a user profile (name, organisation, email etc.) and follow a certain schema (for example LDAP). One may define a common federated user profile with a set of default properties that are common across all authentication domains (systems, networks and organisation). Since different instances of the Profile Management Service may support different types of user profiles, it is possible to develop application specific user profiles. Those application specific user profiles have to include the attributes of the federated profile, if security enabled cross application interaction is required. The specification of such a federated profile is beyond the scope of this document.

2.2. Service Specification Summary

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

- The Service Capabilities Interface
- The Profile Management Interface

The new Profile Management Interface completely replaces the former User Management Interface.

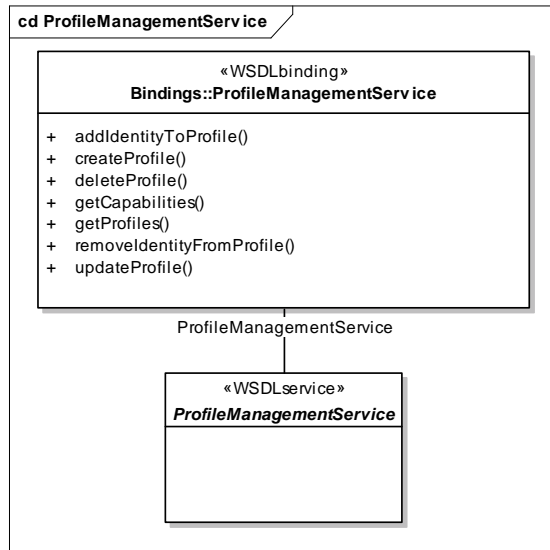


Diagram 1: Simplified class diagram of the Profile Management Service

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

3. Context of the Profile Management Service

3.1. Relations to Standards

The concept of separating profiles and identities has been taken from the Java Authentication and Authorization Service (JAAS) API.

The Profile Management Service supports Profile Attributes compatible to LDAP core schemas. RFC 4519 Lightweight Directory Access Protocol (LDAP): Schema for User Applications LDAP is widely spread in current identity management systems. Similar to the Profile Management Service these identity management systems need to store user profiles (Profile Attributes). The RFC 4519 is part of a set of LDAP RFC standard and contains schemas describing object classes including their attributes. These object classes can serve as a template for the definition of profile attribute types.

The Profile Management Service is designed to inherently support access control and therefore specifies session information at interface level. Session Information is encoded in the SAML v2.0 OASIS Standard. To learn more about the concept of the integrated PEP and the usage of SAML in the SANY Access Control Services please consult the specifications of the Policy Management and Authentication Service and the Generic Policy Enforcement Service.

3.2. Relations to Information Models

Apart from the parameter types described in this document, the Profile Management Service relies on additional information models for the description of user profiles and session information. As stated in the previous chapter, open standards are being used to define those information models.

The information model for user profiles can be defined per instance which means that different instances of the Profile Management Service may provide different types of user profiles. For interoperability reasons however there exists a default user profile that has to be supported by any instance of the Profile Management Service. This default or federated profile is specified in Appendix B: Default LDAP .

3.3. Relations to other Service Specifications

The Profile Management Service has a strong relation to the other SANY Access Control Services since its operations are per definition access controlled, except for the getCapabilities operation.

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 Appendix A: XML Schema and WSDL Documents.

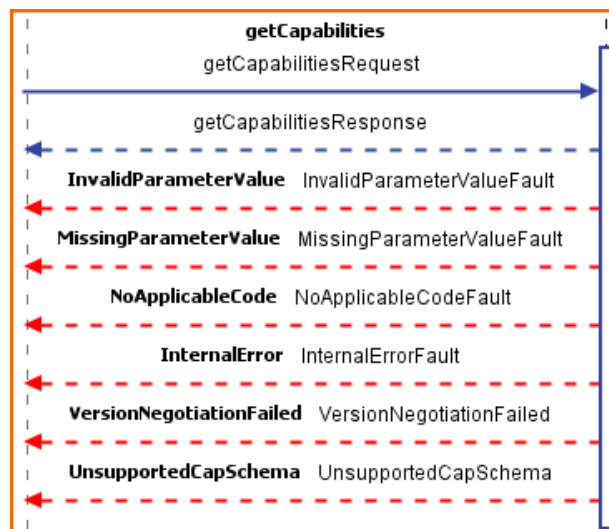


Figure 1: getCapabilities Operation

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_GetCapabilitiesRequest 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 7.3.

5. Specification of the Profile Management Interface

The Profile Management Interface is an adaptation of the former ORCHESTRA User Management Interface. It provides operations to create and maintain (user) profiles and their associations to identities.

5.1. createProfile Operation

The mandatory createProfile operation is used to add new profiles to a Profile Management Service instance. New profiles may but must not be populated with attributes. The operation returns the newly created profile including a unique id and the url of the Profile Management Service instance.

A request to perform the createProfile 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	Not applicable			
Overrides	Not applicable			
Preconditions	None			
Post conditions	None			
Use	Mandatory			
Receives	Name	Type	Use	Description
	request	createProfile Request	mandatory	It contains the profile to be created.
Returns	Type		Description	
	OA_Profile		The created profile including a recently added unique id (unique within the service instance) that can be used for a later reference to this profile.	
Throws	Type		Cause	
	OA_InvalidParameter Value		Operation request contains an invalid parameter value.	
	OA_MissingParameter Value		Operation request does not include a parameter value.	
	OA_NoApplicable Code		No other basic or service-specific exception type applies.	
	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 createProfile Operation

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

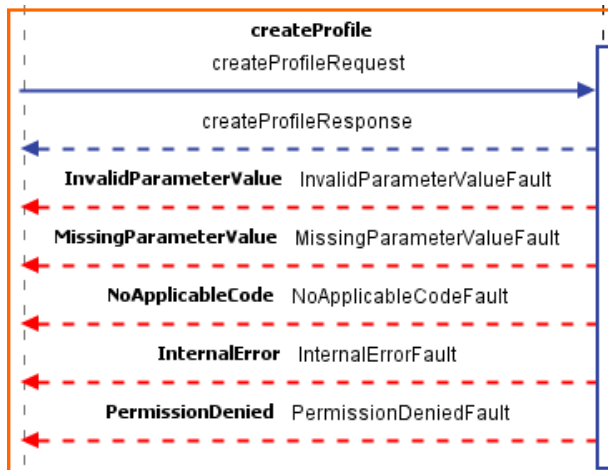


Figure 4: createProfile Operation

5.1.1 createProfileRequest Type

The createProfileRequest Type consists of the following elements:

- newProfile: The profile that shall be created. The object does not need to have any values set.

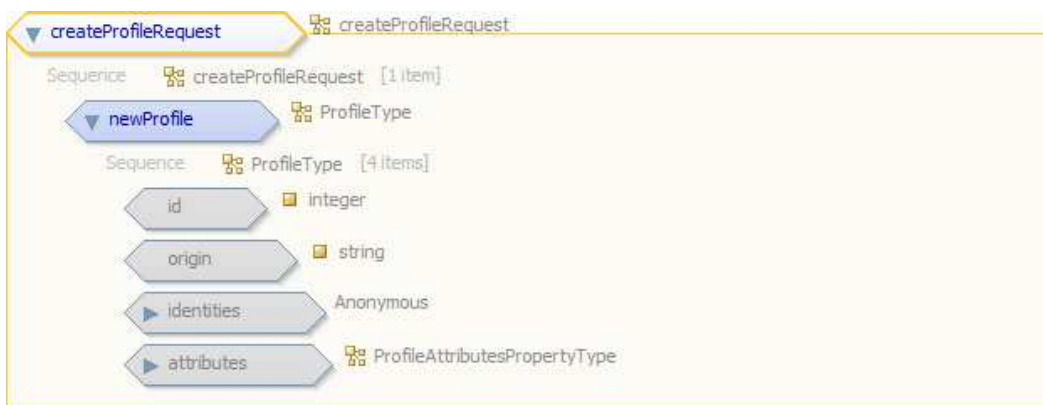


Figure 5: createProfileRequest

- attributes: The currently supported profile attributes are of type KeyVectorPairProfileAttributes where several values may be associated to a single key.

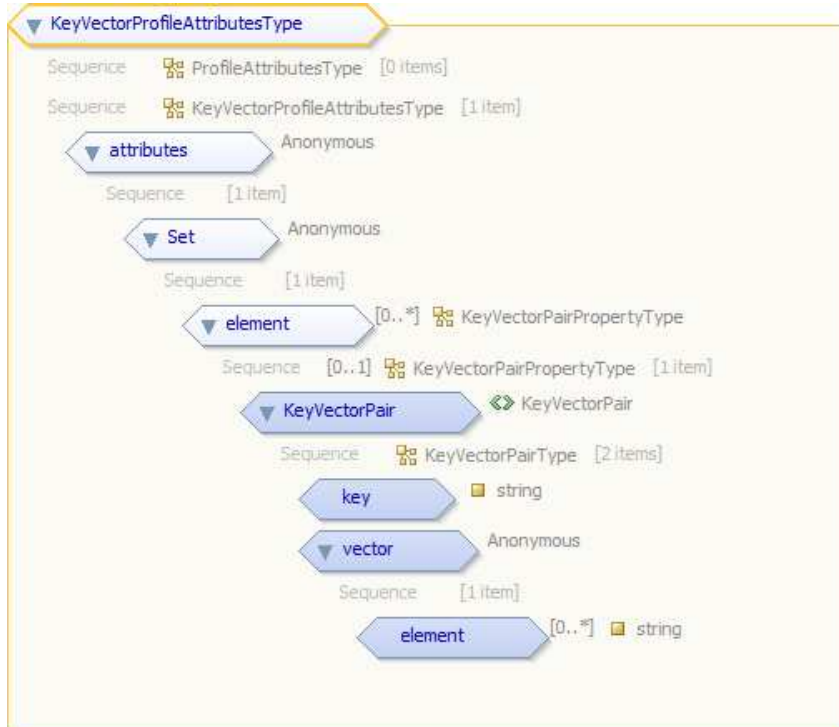


Figure 6: KeyVectorProfileAttributesType

5.1.2 createProfileResponse Type

The createProfileResponse is of type Profile. The Profile Type consists of the following elements:

- id: An identifier which uniquely identifies the newly created profile within the service instance.
- Origin: URI of the service instance.
- Identities: A list of identities assigned to this profile. As the profile was just created, this list is empty.
- ProfileAttributes: A sequence of KeyVectorProfileAttributesType objects representing the attributes of the profile.



Figure 7: createProfileResponse

5.2. deleteProfile Operation

The mandatory deleteProfile operation is used to delete profiles, its attributes as well as its associations to identities from a Profile Management Service instance.

Deleting a profile means to completely remove it from the whole service network. This includes to delete the profile from the correspondent Profile Management Instance, as well as the update of all identities currently member of it.

Notice: An administrator client that invokes the deleteProfile operation should ensure that the identities previously associated to the deleted profile are properly deactivated. It shall therefore invoke the updateIdentity operation of the respective Identity Management Service instances.

A request to perform the deleteProfile 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	Profile to be deleted exists.			
Post conditions	Not applicable			
Use	mandatory			
Receives	Name	Type	Use	Description
	request	deleteProfile Request	mandatory	It contains the profile to be deleted.
Returns	Type		Description	
	None		None	
Throws	Type		Cause	
	OA_InvalidParameter Value		Operation request contains an invalid parameter value.	
	OA_MissingParameter Value		Operation request does not include a parameter value.	
	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_ReferentialIntegrityException	Thrown if at least one profile related identity cannot be deleted.
	OA_ProfileNotFoundException	Thrown if the profile to be deleted does not exist.
	OA_PermissionDeniedException	The service requestor does not have permissions needed to perform the requested operation.

Table 3: Specification of the deleteProfile Operation

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



Figure 8: deleteProfile Operation

5.2.1 deleteProfileRequest Type

The deleteProfileRequest Type consists of the following elements:

- profile: The profile which shall be deleted. The object is expected to have its “id” attribute set in order to uniquely identify the profile within the service instance. Other attributes are ignored. To obtain a properly initialized profile, a client may first invoke the getProfiles operation.

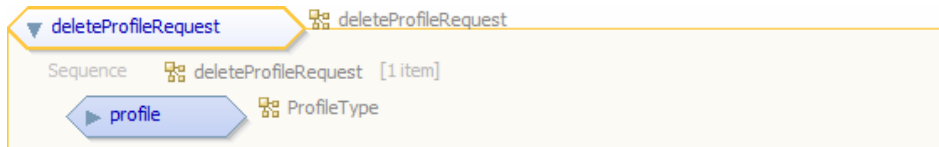


Figure 9: deleteProfileRequest

5.3. updateProfile Operation

The mandatory updateProfile operation is used to update profiles as well as their associated attributes. The existing attributes of the profile stored in a Profile Management Service instance are completely replaced by the attributes attached to the profile to be updated. Please note that the attribute "id" and "origin" are only needed to identify the profile but cannot be changed since they are internally assigned in a Profile Management Service instance..

A request to perform the updateProfile 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	Profile to be updated exists			
Post conditions	none			
Use	mandatory			
Receives	Name	Type	Use	Description
	request	updateProfile Request	mandatory	It contains the profile to be updated.
Returns	Type		Description	
	None		None	
Throws	Type		Cause	
	OA_InvalidParameter Value		Operation request contains an invalid parameter value.	
	OA_MissingParameter Value		Operation request does not include a parameter value.	
	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_ProfileNotFound Exception		Thrown if the profile to be updated does not exist.	
	OA_PermissionDenied Exception		The service requestor does not have permissions needed to perform the requested operation.	

Table 4: Specification of the updateProfile Operation

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

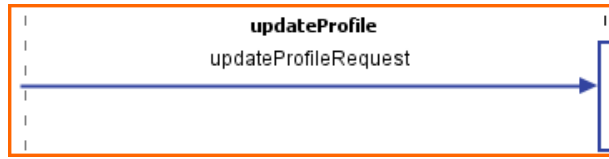


Figure 10 : updateProfile operation

5.3.1 updateProfileRequest Type

The updateProfileRequest Type consists of the following elements:

- profile: The profile which shall be modified. The object is expected to have its “id” attribute set in order to uniquely identify it within the service instance. If the “attributes” attribute of the profile is set, the service stores them as the current attributes or deletes the current attributes otherwise. Any other attributes being set (e.g. origin) are ignored. Please note that also unmodified attributes have to be set in order to prevent the deletion of those.

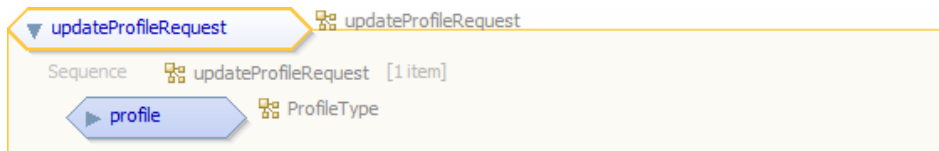


Figure 11 : updateProfileRequest

5.4. getProfiles Operation

The mandatory getProfiles operation is used to retrieve specified profiles from an Profile Management Service instance. The getProfiles operation return either all profiles maintained by a Profile Management Service instance or a subset of the available profiles that match a certain query.

A request to perform the getProfiles 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	getProfilesRequest	mandatory	It contains an optional query.
Returns	Type		Description	
	SequenceOfProfile		A list of profiles selected by the given query or all profiles maintained by the service instance if no query is specified.	
Throws	Type		Cause	
	OA_InvalidParameterValue		Operation request contains an invalid parameter value.	
	OA_MissingParameterValue		Operation request does not include a parameter value.	
	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_ProfileNotFoundException		No profiles match the query.	
	OA_PermissionDeniedException		The service requestor does not have permissions needed to perform the requested operation.	

Table 5: Specification of the getProfiles Operation

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

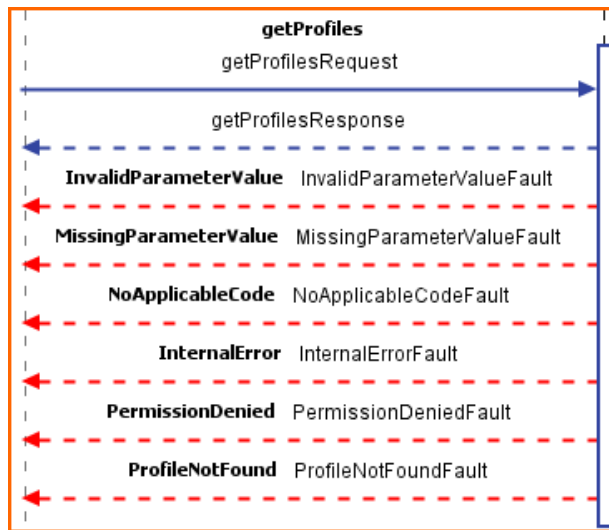


Figure 12: getProfiles operation

5.4.1 getProfilesRequest Type

The getProfilesRequest Type consists of the following elements:

- oaquery:** An optional query parameter, specifying a subset of profiles to be returned. If this parameter is not set, all available profiles are returned. The name of the currently supported query languages are "SQL PROFILE QUERY" and "SQL VECTOR ATTRIBUTES QUERY". The queries may consist of a simple SQL WHERE clause to select either profiles by their ID (SQL SUBJECT QUERY example: WHERE Profile_id = #) or vector attributes by their key and value(s) (SQL VECTOR ATTRIBUTES QUERY example: WHERE AttributeKey = "ROLE" AND AttributeValue IN ("ADMIN", "ROOT", "ADMINISTRATOR")).

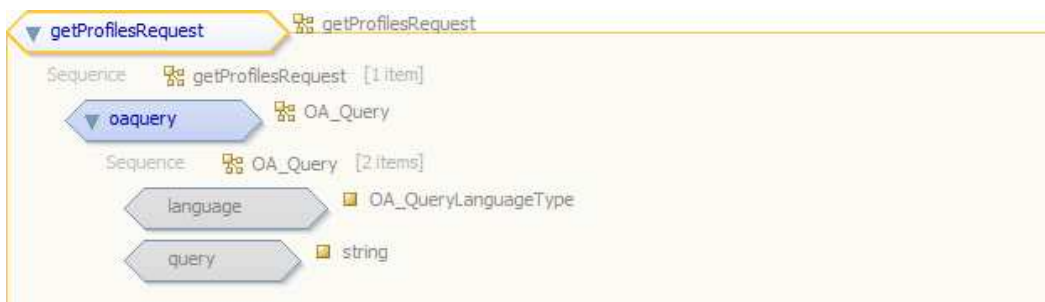


Figure 13: getProfilesRequest

5.4.2 getProfilesResponse Type

The getProfilesResponse Type consists of the following elements:

- profiles: This object consists of a sub-element “Sequence” containing the profiles as selected through the query of the request parameter (see above).

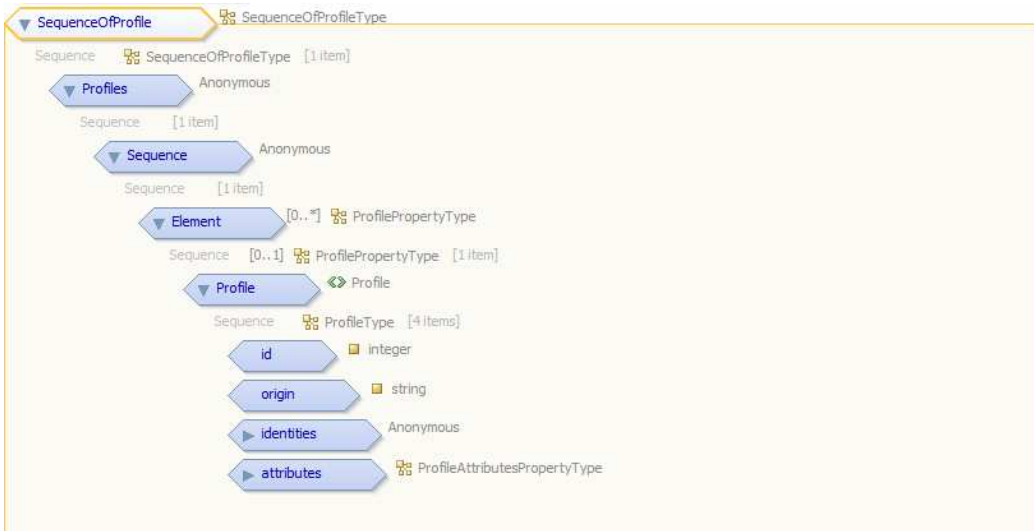


Figure 14: getProfilesResponse

5.5. addIdentityToProfile Operation

The mandatory addIdentityToProfile operation is used to add an existing identity to an existing profile in an Profile Management Service instance. The identity is assumed to be stored in any Identity Management and Authentication Service instance.

A request to perform the addIdentityToProfile 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	Profile has to exist. Identity has to exist.			
Post conditions	Not applicable			
Use	mandatory			
Receives	Name	Type	Use	Description
	request	addIdentityToProfile Request	mandatory	It contains the profile and the identity that shall be associated with this identity.
Returns	Type		Description	
	None		None	
Throws	Type		Cause	
	OA_InvalidParameterValue		Operation request contains an invalid parameter value.	
	OA_MissingParameterValue		Operation request does not include a parameter value.	
	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_ProfileNotFound Exception		Profile does not exist.	
	OA_IdentityNotFound Exception		Identity to be added to the profile does not exist.	
	OA_PermissionDenied Exception		The service requestor does not have permissions needed to perform the requested	

		operation.
--	--	------------

Table 6: Specification of the addIdentityToProfile Operation

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

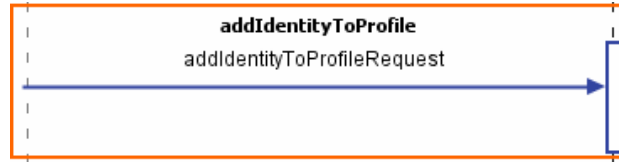


Figure 15: addIdentityToProfile operation

5.5.1 addIdentityToProfile Operation

The addIdentityToProfile Type consists of the following elements:

- **identity:** The identity which shall be added to the profile. The identity object is expected to have its “id” and “origin” attributes set properly. Thereby the origin is the URI to the Identity Management and Authentication Service defining the identity and the ID uniquely identifies the identity within that service instance. Any other attributes are ignored.
- **profile:** The profile to which the identity shall be assigned. The profile is expected to have its “id” attribute set properly, in order to uniquely identify it within the service instance. Any other attributes are ignored.

To obtain properly initialised objects one may first invoke the “getProfiles” (resp. “getIdentities”) operations of the respective services.

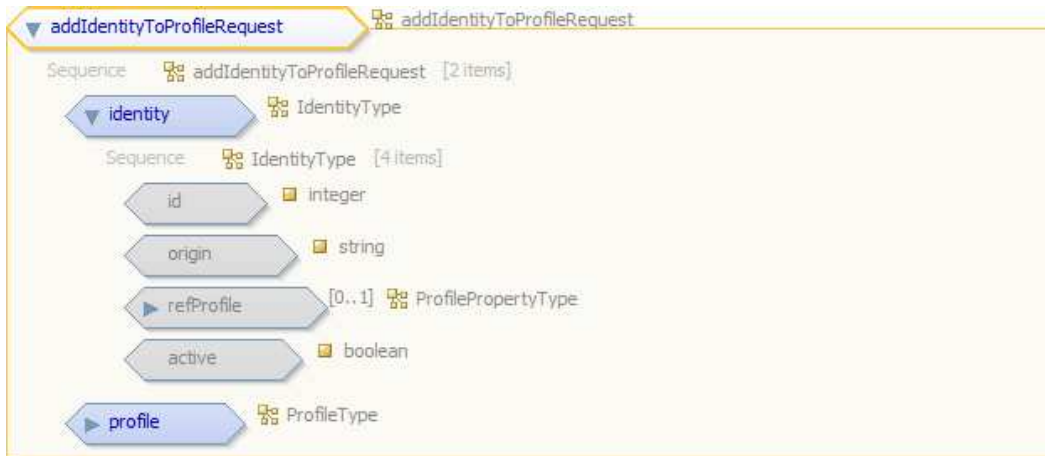


Figure 16: addIdentityToProfileRequest

5.6. removeIdentityFromProfile Operation

The mandatory removeIdentityFromProfile operation is used to remove an identity from a profile in an Profile Management Service instance. The identity is assumed to be stored in any Identity Management and Authentication Service instance.

A request to perform the removeIdentityFromProfile 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	Association between given profile and identity has to exist.			
Post conditions	Association between given profile and identity has been removed.			
Use	mandatory			
Receives	Name	Type	Use	Description
	request	removeIdentityFromProfileRequest	mandatory	It contains the profile and the identity that shall be removed from the profile.
Returns	Type		Description	
	None		None	
Throws	Type		Cause	
	OA_InvalidParameterValue		Operation request contains an invalid parameter value.	
	OA_MissingParameterValue		Operation request does not include a parameter value.	
	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_ProfileNotFoundException		Group does not exist.	
	OA_IdentityNotFoundException		Identity to be added to the group does not exist.	

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

Table 7: Specification of the removeIdentityFromProfile Operation

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

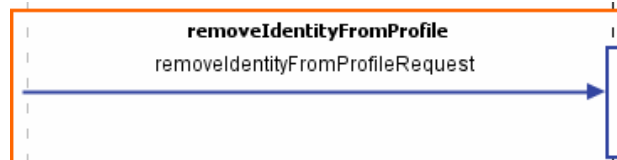


Figure 17: removeIdentityFromProfile operation

5.6.1 removeIdentityFromProfile Type

The removeIdentityFromProfileRequest Type consists of the following elements:

- identity: The identity which shall be removed from the profile. The identity object is expected to have its “id” and “origin” attributes set properly. Thereby the origin is the URI to the Identity Management and Authentication Service defining the identity. The ID uniquely identifies the identity within that service instance. Any other attributes et are ignored.
- profile: The profile from which the identity shall be removed. The profile is expected to have its “id” attribute set properly, in order to uniquely identify it within the service instance.

To obtain properly initialised objects one may first invoke the “getProfiles” (resp. “getIdentities”) operations of the respective services.

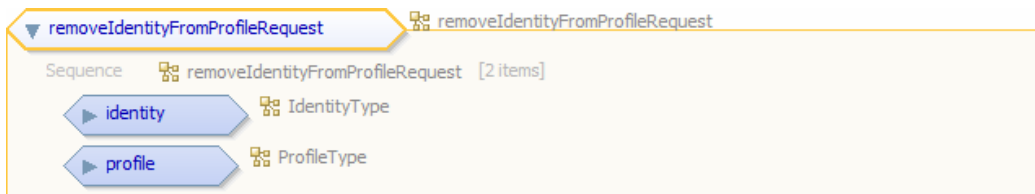


Figure 18: removeIdentityFromProfileRequest

6. References

- 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-Authorisation** Specification of the Policy Management and Authorisation 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>

7. Appendix A: XML Schema and WSDL Documents

7.1. 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 Profile Management Service.

The namespaces

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

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

<http://www.enviromatics.net/WS/ProfileManagementService/requests/2.0>

<http://www.enviromatics.net/WS/ProfileManagementService/types/2.0>

<http://www.enviromatics.net/WS/ProfileManagementService/exceptions/2.0>

are used.

7.1.1 profilemanagement_exceptions.xsd

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

xmlns:pms_exc="http://www.enviromatics.net/WS/ProfileManagementService/exceptions/2.0"
      xmlns:oab_exc="http://eu-orchestra.org/OA/OABasicService/exceptions/1.0"

targetNamespace="http://www.enviromatics.net/WS/ProfileManagementService/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.x
sd" />

  <element name="ReferentialIntegrityException"
type="pms_exc:ReferentialIntegrityExceptionType"
substitutionGroup="oab_exc:OA_AbstractException" />
  <complexType name="ReferentialIntegrityExceptionType">
    <complexContent>
      <extension base="oab_exc:OA_AbstractException">
        <sequence />
      </extension>
    </complexContent>
  </complexType>
  <complexType name="ReferentialIntegrityExceptionPropertyType">
    <sequence minOccurs="0">
      <element ref="pms_exc:ReferentialIntegrityException" />
    </sequence>
  </complexType>

  <element name="NoSuchMemberException"
type="pms_exc:NoSuchMemberExceptionType"
substitutionGroup="oab_exc:OA_AbstractException" />
  <complexType name="NoSuchMemberExceptionType">
    <complexContent>
      <extension base="oab_exc:OA_AbstractException">
        <sequence />
      </extension>
    </complexContent>
  </complexType>
  <complexType name="NoSuchMemberExceptionPropertyType">
    <sequence minOccurs="0">
      <element ref="pms_exc:NoSuchMemberException" />
    </sequence>
  </complexType>

  <element name="ProfileNotFoundException"
type="pms_exc:ProfileNotFoundExceptionType"
substitutionGroup="oab_exc:OA_AbstractException" />
  <complexType name="ProfileNotFoundExceptionType">
    <complexContent>
      <extension base="oab_exc:OA_AbstractException">
        <sequence />
      </extension>
    </complexContent>
  </complexType>
  <complexType name="ProfileNotFoundExceptionPropertyType">
    <sequence minOccurs="0">
      <element ref="pms_exc:ProfileNotFoundException" />
    </sequence>
  </complexType>
</schema>

```

7.1.2 profilemanagement_requests.xsd

```

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

xmlns:pms_types="http://www.enviomatics.net/WS/ProfileManagementService/types/2.0"

xmlns:authen_types="http://www.enviomatics.net/WS/IdentityManagementAndAuthenticationService/types/2.0"

xmlns:pms_requests="http://www.enviomatics.net/WS/ProfileManagementService/requests/2.0"
        xmlns:oas="http://eu-orchestra.org/OA/OABasicService/types/1.0"

targetNamespace="http://www.enviomatics.net/WS/ProfileManagementService/requests/2.0"
        elementFormDefault="qualified"
        version="1.0">
    <import
namespace="http://www.enviomatics.net/WS/ProfileManagementService/types/2.0"

schemaLocation="http://www.enviomatics.net/WS/ProfileManagementService/types/2.0/profilemanagement_types.xsd" />
    <import
namespace="http://www.enviomatics.net/WS/IdentityManagementAndAuthenticationService/types/2.0"

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

schemaLocation="http://www.enviomatics.net/WS/OrchestraArchive/oa_basic.xsd" />

    <element name="createProfileRequest"
type="pms_requests:createProfileRequest" />
    <complexType name="createProfileRequest">
        <sequence>
            <element name="newProfile" type="pms_types:ProfileType" />
        </sequence>
    </complexType>

    <element name="deleteProfileRequest"
type="pms_requests:deleteProfileRequest" />
    <complexType name="deleteProfileRequest">
        <sequence>
            <element name="profile" type="pms_types:ProfileType" />
        </sequence>
    </complexType>

    <element name="updateProfileRequest"
type="pms_requests:updateProfileRequest" />
    <complexType name="updateProfileRequest">
        <sequence>
            <element name="profile" type="pms_types:ProfileType" />

```

```
        </sequence>
    </complexType>

    <element name="getProfilesRequest"
type="pms_requests:getProfilesRequest" />
    <complexType name="getProfilesRequest">
        <sequence>
            <element name="oaquery" type="oas:OA_Query" />
        </sequence>
    </complexType>

    <element name="addIdentityToProfileRequest"
type="pms_requests:addIdentityToProfileRequest" />
    <complexType name="addIdentityToProfileRequest">
        <sequence>
            <element name="identity" type="authen_types:IdentityType" />
            <element name="profile" type="pms_types:ProfileType" />
        </sequence>
    </complexType>

    <element name="removeIdentityFromProfileRequest"
type="pms_requests:removeIdentityFromProfileRequest" />
    <complexType name="removeIdentityFromProfileRequest">
        <sequence>
            <element name="identity" type="authen_types:IdentityType" />
            <element name="profile" type="pms_types:ProfileType" />
        </sequence>
    </complexType>

</schema>
```

7.1.3 profilemanagement_types.xsd

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

xmlns:pms_types="http://www.enviomatics.net/WS/ProfileManagementService/types/2.0"

xmlns:ia_types="http://www.enviomatics.net/WS/IdentityManagementAndAuthenticationService/types/2.0"

targetNamespace="http://www.enviomatics.net/WS/ProfileManagementService/types/2.0"
    elementFormDefault="qualified"
    version="1.0">
    <import
namespace="http://www.enviomatics.net/WS/IdentityManagementAndAuthenticationService/types/2.0"

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

schemaLocation="http://www.enviomatics.net/WS/OrchestraArchive/oa_basic.xsd" />

    <element name="KeyValuePair" type="pms_types:KeyValuePairType" />
    <complexType name="KeyValuePairType">
        <sequence>
            <element name="key" type="string" />
            <element name="value" type="string" />
        </sequence>
    </complexType>
    <complexType name="KeyValuePairPropertyType">
        <sequence minOccurs="0">
            <element ref="pms_types:KeyValuePair" />
        </sequence>
    </complexType>

    <element name="KeyVectorPair" type="pms_types:KeyVectorPairType" />
    <complexType name="KeyVectorPairType">
        <sequence>
            <element name="key" type="string" />
            <element name="vector">
                <complexType>
                    <sequence>
                        <element name="element" type="string" minOccurs="0"
maxOccurs="unbounded" />
                    </sequence>
                </complexType>
            </element>
        </sequence>
    </complexType>
    <complexType name="KeyVectorPairPropertyType">
        <sequence minOccurs="0">
            <element ref="pms_types:KeyVectorPair" />
        </sequence>
    </complexType>
</schema>
```

```

    <element name="ProfileAttributes"
type="pms_types:ProfileAttributesType" />
    <complexType name="ProfileAttributesType">
        <sequence />
    </complexType>
    <complexType name="ProfileAttributesPropertyType">
        <sequence minOccurs="0">
            <element ref="pms_types:ProfileAttributes" />
        </sequence>
    </complexType>

    <element name="Profile" type="pms_types:ProfileType" />
    <complexType name="ProfileType">
        <sequence>
            <element name="id" type="integer" />
            <element name="origin" type="string" />
            <element name="identities">
                <complexType>
                    <sequence>
                        <element name="Sequence">
                            <complexType>
                                <sequence>
                                    <element name="element"
type="ia_types:IdentityPropertyType" minOccurs="0" maxOccurs="unbounded" />
                                </sequence>
                            </complexType>
                        </element>
                    </sequence>
                </complexType>
            </element>
            <element name="attributes"
type="pms_types:ProfileAttributesPropertyType" />
        </sequence>
    </complexType>
    <complexType name="ProfilePropertyType">
        <sequence minOccurs="0">
            <element ref="pms_types:Profile" />
        </sequence>
    </complexType>

    <element name="KeyVectorProfileAttributes"
type="pms_types:KeyVectorProfileAttributesType"
substitutionGroup="pms_types:ProfileAttributes" />
    <complexType name="KeyVectorProfileAttributesType">
        <complexContent>
            <extension base="pms_types:ProfileAttributesType">
                <sequence>
                    <element name="attributes">
                        <complexType>
                            <sequence>
                                <element name="Set">
                                    <complexType>
                                        <sequence>
                                            <element name="element"
type="pms_types:KeyVectorPairPropertyType" minOccurs="0"
maxOccurs="unbounded" />
                                        </sequence>
                                    </complexType>
                                </element>
                            </sequence>
                        </complexType>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

```

```

        </sequence>
      </complexType>
    </element>
  </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="KeyVectorProfileAttributesPropertyType">
  <sequence minOccurs="0">
    <element ref="pms_types:KeyVectorProfileAttributes" />
  </sequence>
</complexType>

  <element name="KeyValueProfileAttributes"
type="pms_types:KeyValueProfileAttributesType"
substitutionGroup="pms_types:ProfileAttributes" />
  <complexType name="KeyValueProfileAttributesType">
    <complexContent>
      <extension base="pms_types:ProfileAttributesType">
        <sequence>
          <element name="attributes">
            <complexType>
              <sequence>
                <element name="Set">
                  <complexType>
                    <sequence>
                      <element name="element"
type="pms_types:KeyValuePairPropertyType" minOccurs="0"
maxOccurs="unbounded" />
                    </sequence>
                  </complexType>
                </element>
              </sequence>
            </complexType>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
<complexType name="KeyValueProfileAttributesPropertyType">
  <sequence minOccurs="0">
    <element ref="pms_types:KeyValueProfileAttributes" />
  </sequence>
</complexType>

  <element name="SequenceOfProfile"
type="pms_types:SequenceOfProfileType" />
  <complexType name="SequenceOfProfileType">
    <sequence>
      <element name="Profiles">
        <complexType>
          <sequence>
            <element name="Sequence">
              <complexType>
                <sequence>
                  <element name="Element"
type="pms_types:ProfilePropertyType" minOccurs="0" maxOccurs="unbounded" />
                </sequence>
              </complexType>
            </element>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>

```

```
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
<complexType name="SequenceOfProfilePropertyType">
  <sequence minOccurs="0">
    <element ref="pms_types:SequenceOfProfile" />
  </sequence>
</complexType>
</schema>
```

7.2. WSDL Document

The following WSDL Version 1.1 document is the formal specification of the Profile Management 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:pms="http://www.enviromatics.net/WS/ProfileManagementService"

  xmlns:pms_exc="http://www.enviromatics.net/WS/ProfileManagementService/exceptions/2.0"

  xmlns:pms_types="http://www.enviromatics.net/WS/ProfileManagementService/types/2.0"

  xmlns:ia_exc="http://www.enviromatics.net/WS/IdentityManagementAndAuthenticationService/exceptions/2.0"

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

  xmlns:pms_requests="http://www.enviromatics.net/WS/ProfileManagementService/requests/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"
    name="ProfileManagementServiceInstance"

  targetNamespace="http://www.enviromatics.net/WS/ProfileManagementService">
  <wSDL:types>
    <xs:schema
      targetNamespace="http://www.enviromatics.net/WS/ProfileManagementService"
      xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
      attributeFormDefault="qualified">
      <xs:import
        namespace="http://www.enviromatics.net/WS/ProfileManagementService/requests/2.0"

        schemaLocation="http://www.enviromatics.net/WS/ProfileManagementService/requests/2.0/profilemanagement_requests.xsd"/>
```

```

        <xs:import
namespace="http://www.enviomatics.net/WS/PolicyManagementAndAuthorisationService/exceptions/2.0"

schemaLocation="http://www.enviomatics.net/WS/PolicyManagementAndAuthorisationService/exceptions/2.0/policy_author_exceptions.xsd" />
        <xs:import namespace="http://eu-orchestra.org/OA/OABasicService/exceptions/1.0"

schemaLocation="http://www.enviomatics.net/WS/OrchestraArchive/oa_basic_exceptions.xsd" />
        <xs:import
namespace="http://www.enviomatics.net/WS/ProfileManagementService/exceptions/2.0"

schemaLocation="http://www.enviomatics.net/WS/ProfileManagementService/exceptions/2.0/profilemanagement_exceptions.xsd" />
        <xs:import
namespace="http://www.enviomatics.net/WS/IdentityManagementAndAuthenticationService/exceptions/2.0"

schemaLocation="http://www.enviomatics.net/WS/IdentityManagementAndAuthenticationService/exceptions/2.0/identity_authen_exceptions.xsd" />
        <xs:import
namespace="http://www.enviomatics.net/WS/ProfileManagementService/types/2.0"

schemaLocation="http://www.enviomatics.net/WS/ProfileManagementService/types/2.0/profilemanagement_types.xsd" />
        <xs:import namespace="http://eu-orchestra.org/OA/OABasicService/types/1.0"

schemaLocation="http://www.enviomatics.net/WS/OrchestraArchive/oa_basic.xsd" />
        <xs:import namespace="http://eu-orchestra.org/basicTypes/1.0"

schemaLocation="http://www.enviomatics.net/WS/OrchestraArchive/basic_types.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="createProfileRequest">
        <wsdl:part name="request"
element="pms_requests:createProfileRequest" />
    </wsdl:message>
    <wsdl:message name="createProfileResponse">
        <wsdl:part name="response" element="pms_types:Profile" />
    </wsdl:message>
    <wsdl:message name="deleteProfileRequest">
        <wsdl:part name="request"
element="pms_requests:deleteProfileRequest" />
    </wsdl:message>
    <wsdl:message name="updateProfileRequest">

```

```

        <wsdl:part name="request"
element="pms_requests:updateProfileRequest" />
    </wsdl:message>
    <wsdl:message name="getProfilesRequest">
        <wsdl:part name="request" element="pms_requests:getProfilesRequest" />
    </wsdl:message>
    <wsdl:message name="getProfilesResponse">
        <wsdl:part name="response" element="pms_types:SequenceOfProfile" />
    </wsdl:message>
    <wsdl:message name="addIdentityToProfileRequest">
        <wsdl:part name="request"
element="pms_requests:addIdentityToProfileRequest" />
    </wsdl:message>
    <wsdl:message name="removeIdentityFromProfileRequest">
        <wsdl:part name="request"
element="pms_requests:removeIdentityFromProfileRequest" />
    </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="ReferentialIntegrityFault">
        <wsdl:part name="OReferentialIntegrityException"
element="pms_exc:ReferentialIntegrityException" />
    </wsdl:message>
    <wsdl:message name="ProfileNotFoundFault">
        <wsdl:part name="ProfileNotFoundException"
element="pms_exc:ProfileNotFoundException" />
    </wsdl:message>
    <wsdl:message name="IdentityNotFoundFault">
        <wsdl:part name="IdentityNotFoundException"
element="ia_exc:IdentityNotFoundException" />
    </wsdl:message>
    <wsdl:message name="NoSuchMemberFault">
        <wsdl:part name="NoSuchMemberException"
element="pms_exc:NoSuchMemberException" />
    </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="ProfileManagementService">
  <wsdl:operation name="getCapabilities">
    <wsdl:input name="capabilitiesRequest"
message="pms:getCapabilitiesRequest" />
    <wsdl:output name="capabilitiesResponse"
message="pms:getCapabilitiesResponse" />
    <wsdl:fault name="InvalidParameterValue"
message="pms:InvalidParameterValueFault" />
    <wsdl:fault name="MissingParameterValue"
message="pms:MissingParameterValueFault" />
    <wsdl:fault name="NoApplicableCode"
message="pms:NoApplicableCodeFault" />
    <wsdl:fault name="InternalError"
message="pms:InternalErrorFault" />
    <wsdl:fault name="VersionNegotiationFailed"
message="pms:VersionNegotiationFailed" />
    <wsdl:fault name="UnsupportedCapSchema"
message="pms:UnsupportedCapSchema" />
  </wsdl:operation>
  <wsdl:operation name="removeIdentityFromProfile">
    <wsdl:input name="removeIdentityFromProfileRequestRequest"
message="pms:removeIdentityFromProfileRequest" />
    <wsdl:fault name="InvalidParameterValue"
message="pms:InvalidParameterValueFault" />
    <wsdl:fault name="MissingParameterValue"
message="pms:MissingParameterValueFault" />
    <wsdl:fault name="NoApplicableCode"
message="pms:NoApplicableCodeFault" />
    <wsdl:fault name="InternalError"
message="pms:InternalErrorFault" />
    <wsdl:fault name="PermissionDenied"
message="pms:PermissionDeniedFault" />
    <wsdl:fault name="IdentityNotFound"
message="pms:IdentityNotFoundFault" />
    <wsdl:fault name="ProfileNotFound"
message="pms:ProfileNotFoundFault" />
  </wsdl:operation>
  <wsdl:operation name="addIdentityToProfile">
    <documentation>Parameter Profile: The profile to which the identity
shall be assigned. The profile is expected to have its "id" attribute set
properly, in order to uniquely identify it within the service instance. Any
other attributes are ignored</documentation>
    <wsdl:input name="addIdentityToProfileRequestRequest"
message="pms:addIdentityToProfileRequest" />
    <wsdl:fault name="InvalidParameterValue"
message="pms:InvalidParameterValueFault" />
    <wsdl:fault name="MissingParameterValue"
message="pms:MissingParameterValueFault" />
    <wsdl:fault name="NoApplicableCode"
message="pms:NoApplicableCodeFault" />
    <wsdl:fault name="InternalError"
message="pms:InternalErrorFault" />
    <wsdl:fault name="PermissionDenied"
message="pms:PermissionDeniedFault" />
    <wsdl:fault name="IdentityNotFound"
message="pms:IdentityNotFoundFault" />
    <wsdl:fault name="ProfileNotFound"
message="pms:ProfileNotFoundFault" />
  </wsdl:operation>

```

```

    <wsdl:operation name="getProfiles">
      <wsdl:input name="getProfilesRequestRequest"
message="pms:getProfilesRequestRequest"/>
      <wsdl:output name="getProfilesRequestResponse"
message="pms:getProfilesResponse"/>
      <wsdl:fault name="InvalidParameterValue"
message="pms:InvalidParameterValueFault"/>
      <wsdl:fault name="MissingParameterValue"
message="pms:MissingParameterValueFault"/>
      <wsdl:fault name="NoApplicableCode"
message="pms:NoApplicableCodeFault"/>
      <wsdl:fault name="InternalError"
message="pms:InternalErrorFault"/>
      <wsdl:fault name="PermissionDenied"
message="pms:PermissionDeniedFault"/>
      <wsdl:fault name="ProfileNotFound"
message="pms:ProfileNotFoundFault"/>
    </wsdl:operation>
    <wsdl:operation name="updateProfile">
      <wsdl:documentation>
        Edit attributes of an existing Profile

        Input:
        User identification
        User attributes (e.g. user name, address, email, phone, fax,
preferred language)

        Output:
        none
      </wsdl:documentation>
      <wsdl:input name="updateProfileRequestRequest"
message="pms:updateProfileRequestRequest"/>
      <wsdl:fault name="InvalidParameterValue"
message="pms:InvalidParameterValueFault"/>
      <wsdl:fault name="MissingParameterValue"
message="pms:MissingParameterValueFault"/>
      <wsdl:fault name="NoApplicableCode"
message="pms:NoApplicableCodeFault"/>
      <wsdl:fault name="InternalError"
message="pms:InternalErrorFault"/>
      <wsdl:fault name="PermissionDenied"
message="pms:PermissionDeniedFault"/>
      <wsdl:fault name="ProfileNotFound"
message="pms:ProfileNotFoundFault"/>
    </wsdl:operation>
    <wsdl:operation name="deleteProfile">
      <wsdl:input name="deleteProfileRequestRequest"
message="pms:deleteProfileRequestRequest"/>
      <wsdl:fault name="InvalidParameterValue"
message="pms:InvalidParameterValueFault"/>
      <wsdl:fault name="MissingParameterValue"
message="pms:MissingParameterValueFault"/>
      <wsdl:fault name="NoApplicableCode"
message="pms:NoApplicableCodeFault"/>
      <wsdl:fault name="InternalError"
message="pms:InternalErrorFault"/>
      <wsdl:fault name="PermissionDenied"
message="pms:PermissionDeniedFault"/>
      <wsdl:fault name="ReferentialIntegrity"
message="pms:ReferentialIntegrityFault"/>

```

```

        <wsdl:fault name="ProfileNotFound"
message="pms:ProfileNotFoundFault" />
    </wsdl:operation>
    <wsdl:operation name="createProfile">
        <wsdl:documentation>
            Create new OSN Profile

            Input:
            User identification
            User attributes (e.g. user name, address, email, phone, fax,
preferred language).

            Output:
            none
        </wsdl:documentation>
        <wsdl:input name="createProfileRequestRequest"
message="pms:createProfileRequest" />
        <wsdl:output name="createProfileRequestResponse"
message="pms:createProfileResponse" />
        <wsdl:fault name="InvalidParameterValue"
message="pms:InvalidParameterValueFault" />
        <wsdl:fault name="MissingParameterValue"
message="pms:MissingParameterValueFault" />
        <wsdl:fault name="NoApplicableCode"
message="pms:NoApplicableCodeFault" />
        <wsdl:fault name="InternalError"
message="pms:InternalErrorFault" />
        <wsdl:fault name="PermissionDenied"
message="pms:PermissionDeniedFault" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ProfileManagementService"
type="pms:ProfileManagementService">
    <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="removeIdentityFromProfile">
        <soap:operation soapAction="removeIdentityFromProfile"
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="NoApplicableCode">
            <soap:fault name="NoApplicableCode" use="literal" />
        </wsdl:fault>
        <wsdl:fault name="InternalError">
            <soap:fault name="InternalError" use="literal" />
        </wsdl:fault>
        <wsdl:fault name="PermissionDenied">
            <soap:fault name="PermissionDenied" use="literal" />
        </wsdl:fault>
        <wsdl:fault name="IdentityNotFound">
            <soap:fault name="IdentityNotFound" use="literal" />
        </wsdl:fault>
        <wsdl:fault name="ProfileNotFound">
            <soap:fault name="ProfileNotFound" use="literal" />
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="addIdentityToProfile">
        <soap:operation soapAction="addIdentityToProfile"
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="NoApplicableCode">
            <soap:fault name="NoApplicableCode" use="literal" />
        </wsdl:fault>
        <wsdl:fault name="InternalError">
            <soap:fault name="InternalError" use="literal" />
        </wsdl:fault>
        <wsdl:fault name="PermissionDenied">
            <soap:fault name="PermissionDenied" use="literal" />
        </wsdl:fault>
        <wsdl:fault name="IdentityNotFound">
            <soap:fault name="IdentityNotFound" use="literal" />
        </wsdl:fault>
        <wsdl:fault name="ProfileNotFound">
            <soap:fault name="ProfileNotFound" use="literal" />
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getProfiles">
        <soap:operation soapAction="getProfiles" 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="InternalServerError">
  <soap:fault name="InternalServerError" use="literal"/>
</wsdl:fault>
<wsdl:fault name="PermissionDenied">
  <soap:fault name="PermissionDenied" use="literal"/>
</wsdl:fault>
<wsdl:fault name="ProfileNotFound">
  <soap:fault name="ProfileNotFound" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="updateProfile">
  <wsdl:documentation>
    Edit attributes of an existing Profile

    Input:
    User identification
    User attributes (e.g. user name, address, email, phone, fax,
preferred language)

    Output:
    none
  </wsdl:documentation>
  <soap:operation soapAction="updateProfile" 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="NoApplicableCode">
    <soap:fault name="NoApplicableCode" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="InternalServerError">
    <soap:fault name="InternalServerError" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="PermissionDenied">
    <soap:fault name="PermissionDenied" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="ProfileNotFound">
    <soap:fault name="ProfileNotFound" use="literal"/>
  </wsdl:fault>
</wsdl:operation>

```

```

<wsdl:operation name="deleteProfile">
  <soap:operation soapAction="deleteProfile" 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="NoApplicableCode">
    <soap:fault name="NoApplicableCode" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="InternalError">
    <soap:fault name="InternalError" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="PermissionDenied">
    <soap:fault name="PermissionDenied" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="ReferentialIntegrity">
    <soap:fault name="ReferentialIntegrity" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="ProfileNotFound">
    <soap:fault name="ProfileNotFound" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="createProfile">
  <wsdl:documentation>
    Create new OSN Profile

    Input:
    User identification
    User attributes (e.g. user name, address, email, phone, fax,
preferred language).

    Output:
    none
  </wsdl:documentation>
  <soap:operation soapAction="createProfile" 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="PermissionDenied">
    <soap:fault name="PermissionDenied" use="literal"/>
  </wsdl:fault>
</wsdl:operation>

```

```

        </wsdl:fault>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="ProfileManagementService">
    <wsdl:port name="ProfileManagementService"
binding="pms:ProfileManagementService">
        <soap:address
location="http://localhost:8080/axis2/services/ProfileManagementService"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

7.3. 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/ProfileManagementService/mi/2.0
http://www.enviromatics.net/WS/ProfileManagementService/mi/2.0/profilemanagem
ent_mi.xsd
    http://eu-orchestra.org/OAS-MI/service/invocation/1.1
http://www.enviromatics.net/WS/OrchestraArchive/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:pms:oami="http://www.enviromatics.net/WS/ProfileManagementService/mi/2.0
"
    xmlns:inv="http://eu-orchestra.org/OAS-MI/service/invocation/1.1">
    <oami:serviceCommonCapabilities>
        <oami:OA_MI_Service_CommonCapabilities>
            <oami:id>Specification of the Profile Management Service
v1.0</oami:id>
            <oami:publicationDate>2009-01-13</oami:publicationDate>
            <oami:serviceDescription>The Profile Management Service is used
to create and maintain profiles. In general, profiles (of users, services,
etc.) represent entities that need to be
authenticated.</oami:serviceDescription>
            <oami:serviceDocumentation>http://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/ProfileManagementService</inv:u
ri>

                                </inv:OA_MI_AccessPoint>
                            </inv:accessPoints>
                        </inv:operation>
                    </inv:OA_MI_Service_InvocationBasic>
                </oami:serviceInvocationBasic>
            </oami:OA_MI_Service_CommonCapabilities>
        </oami:serviceCommonCapabilities>
    </oami:OA_MI_Service_Capabilities>
</oami:OA_MI_Service_Capabilities>

```

`<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/ProfileManagementService</inv:uri>

```

```

  </inv:OA_MI_AccessPoint>
</inv:accessPoints>

```

`<inv:description>`The mandatory `addIdentityToProfile` operation is used to add an existing identity to an existing profile in an Profile Management Service instance. The identity may be stored in any Identity Management Service instance.`</inv:description>`

```

<inv:name>addIdentityToProfile</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>addIdentityToProfileRequest</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/ProfileManagementService</inv:u
ri>
      </inv:OA_MI_AccessPoint>
    </inv:accessPoints>
    <inv:description>The mandatory createProfile
operation is used to add new profiles to a Profile Management Service
instance. New profiles may but must not be populated with attributes. The
operation returns the newly created profile including a unique id and the url
of the Profile Management Service instance.</inv:description>
    <inv:name>createProfile</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>createProfileRequest</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>profile</inv:name>
        <inv:optionality>>false</inv:optionality>

<inv:repeatability>>true</inv:repeatability>
      <inv:valueType>OA_Profile</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/ProfileManagementService</inv:u
ri>
      </inv:OA_MI_AccessPoint>
    </inv:accessPoints>
    <inv:description>The mandatory deleteProfile
operation is used to delete profiles, its attributes as well as its
associations to identities from a Profile Management Service
instance.</inv:description>
    <inv:name>deleteProfile</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>deleteProfileRequest</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/ProfileManagementService</inv:u
ri>
      </inv:OA_MI_AccessPoint>
    </inv:accessPoints>
    <inv:description>The mandatory getProfiles
operation is used to retrieve specified profiles from an Profile Management
Service instance. The getProfiles operation return either all profiles
maintained by a Profile Management Service instance or a subset of the
available profiles that match a certain query.</inv:description>
    <inv:name>getProfiles</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>getProfilesRequest</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>profiles</inv:name>
    <inv:optionality>>false</inv:optionality>

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

<inv:valueType>SequenceOfProfile</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/ProfileManagementService</inv:u
ri>
      </inv:OA_MI_AccessPoint>
    </inv:accessPoints>
    <inv:description>The mandatory
removeIdentityFromProfile operation is used to remove an identity from a

```

profile in an Profile Management Service instance. The identity may be stored in any Identity Management Service instance.

```

</inv:description>
  <inv:name>removeIdentityFromProfile</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:OA_MI_OperationParameter>
  </inv:parameters>
</inv:operation>
<inv:operation>
  <inv:OA_MI_Operation>
    <inv:accessPoints>
      <inv:OA_MI_AccessPoint>
        <inv:uri>http://localhost:8080/axis2/services/ProfileManagementService</inv:uri>
      </inv:OA_MI_AccessPoint>
    </inv:accessPoints>
    <inv:description>The mandatory updateProfile
operation is used to update profiles as well as their associated attributes.
The existing attributes of the profile stored in a Profile Management Service
instance are completely replaced by the attributes attached to the profile to
be updated. Please note that the attribute "id" and "origin" are ignored
since they are internally assigned in a Profile Management Service instance
and cannot be changed.</inv:description>
    <inv:name>updateProfile</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:OA_MI_OperationParameter>
    </inv:parameters>
  </inv:OA_MI_Operation>
</inv:operation>
</inv:OA_MI_Service_InvocationBasic>
</oami:serviceInvocationBasic>
<oami:serviceName>Profile Management Service</oami:serviceName>
<oami:serviceSpecVersion>1.0</oami:serviceSpecVersion>
<oami:serviceType>
  <oabs:identifier>Profile Management Service</oabs:identifier>
</oami:serviceType>
</oami:OA_MI_Service_CommonCapabilities>
</oami:serviceCommonCapabilities>
<oami:serviceSpecificCapabilities>
  <pmsoami:MI_ProfileManagementServiceCapabilities>
    <pmsoami:dynamicAttributesSupported>true</pmsoami:dynamicAttributesSupported>
  </oami:serviceSpecificCapabilities>
</oami:serviceSpecificCapabilities>

```

```
<pmsoami:supportedProfileAttributeType>businessCategory</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>commonName</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>countryName</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>description</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>dnQualifier</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>facsimileTelephoneNumber</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>generationQualifier</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>givenName</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>houseIdentifier</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>initials</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>internationalISDNNumber</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>localityName</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>member</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>organisationName</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>organisationalUnitName</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>owner</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>physicalDeliveryOfficeName</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>preferredLanguage</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>postalAddress</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>postalCode</pmsoami:supportedProfileAttributeType>
```

```

<pmsoami:supportedProfileAttributeType>postOfficeBox</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>preferredDeliveryMethod</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>registeredAddress</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>roleOccupant</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>surname</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>stateOrProvinceName</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>street</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>telephoneNumber</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>telexNumber</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>title</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>uid</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>uniqueMember</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>userPassword</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>x121Address</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>x500UniqueIdentifier</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>labeledURI</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>mail</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>mobilePhoneNumber</pmsoami:supportedProfileAttributeType>

<pmsoami:supportedProfileAttributeType>roomNumber</pmsoami:supportedProfileAttributeType>
  </pmsoami:MI_ProfileManagementServiceCapabilities>
</oami:serviceSpecificCapabilities>
</oami:OA_MI_Service_Capabilities>

```

8. Appendix B: Default LDAP Schema

The default LDAP schema describes the attributes of a federated user profile that has to be supported by any instance of the SANY Profile Management Service. It specifies the minimal set of properties that may be taken into account when taking an authorisation decision based on user profile properties. An extension of this profile is explicitly allowed.

A SANY Policy Management and Authorisation Service (Policy Decision Point) must be able to rely on the presence of this federated user profile in order to take basic authorisation decisions in a cross-organisation environment (multiple authentication and authorisation domains) where trust may be granted based on organisational membership or related properties. Advanced access control restrictions in such a multi-domain environment that go beyond the possibilities of a federated profile require mutual agreement on a common security policy.

The LDAP Schema shall be implemented as identity and profile attributes. Since the Identity and Profile Management Service support arbitrary attributes which are not bound to the attributes defined in the default LDAP schema an administration client has to ensure that the appropriate attributes are provided when creating or updating a profile or an identity.

The properties of the default LDAP profile are described in the following table:

Key	Description
businessCategory	"The 'businessCategory' attribute type describes the kinds of business performed by an organization. Each kind is one value of this multi-valued attribute. [...] Examples: 'banking', 'transportation', and 'real estate'. [RFC 4519, 2.1. 'businessCategory']
commonName	"The 'cn' ('commonName' in X.500) attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name." [RFC 4519, "2.3 cn"]
countryName	"The 'c' ('countryName' in X.500) attribute type contains a two-letter ISO 3166 [ISO3166] country code." [RFC 4519, 2.2 c]
description	"The 'description' attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. [...] Examples: 'a color printer', 'Maintenance is done every Monday, at 1pm.', and 'distribution list for all technical staff'. [RFC 4519, "2.5 description"]
dnQualifier	"The 'dnQualifier' attribute type contains disambiguating information strings to add to the relative distinguished name of an entry. The information is intended for use when

	<p>merging data from multiple sources in order to prevent conflicts between entries that would otherwise have the same name. Each string is one value of this multi-valued attribute. It is recommended that a value of the 'dnQualifier' attribute be the same for all entries from a particular source." [RFC 4519, 2.8 dnQualifier]</p>
facsimileTelephoneNumber	<p>"The 'facsimileTelephoneNumber' attribute type contains telephone numbers (and, optionally, the parameters) for facsimile terminals. Each telephone number is one value of this multi-valued attribute. [?] Examples: ?+61 3 9896 7801? and ?+81 3 347 7418\$fineResolution?" [RFC 4519, 2.10]</p>
generationQualifier	<p>"The 'generationQualifier' attribute type contains name strings that are typically the suffix part of a person's name. Each string is one value of this multi-valued attribute. [...] Examples: 'III', '3rd', and 'Jr.'." [RFC 4519, 2.11]</p>
givenName	<p>"The 'givenName' attribute type contains name strings that are the part of a person's name that is not their surname. Each string is one value of this multi-valued attribute. [...] Examples: 'Andrew', 'Charles', and 'Joanne'. " [RFC 4519, "2.12 givenName"]</p>
houseIdentifier	<p>"The 'houseIdentifier' attribute type contains identifiers for a building within a location. Each identifier is one value of this multi-valued attribute. [...] Example: "20" to represent the house number 20." [RFC 4519, 2.13]</p>
initials	<p>"The initials attribute contains the initials of some or all of an individuals names, but not the surname(s). [...] Examples: 'K. A.' and 'K'. " [RFC 4519, 2.14]</p>
internationalISDNNumber	<p>"The 'internationalISDNNumber' attribute type contains Integrated Services Digital Network (ISDN) addresses, as defined in the International Telecommunication Union (ITU) Recommendation E.164 [E.164]. Each address is one value of this multi-valued attribute. [...] Example: '0198 333 333'. " [RFC 4519, 2.15]</p>
localityName	<p>"The 'l' ('localityName' in X.500) attribute type contains</p>

	<p>names of a locality or place, such as a city, county, or other geographic region. Each name is one value of this multi-valued attribute. [...]</p> <p>Examples: 'Geneva', 'Paris', and 'Edinburgh'." [RFC 4519, 2.17]</p>
member	<p>"The 'member' attribute type contains the distinguished names of objects that are on a list or in a group. Each name is one value of this multi-valued attribute. [...]</p> <p>Examples: 'cn=James Clarke,ou=Finance,o=Widget\, Inc.' and 'cn=John Xerri,ou=Finance,o=Widget\, Inc.' may be two members of the financial team (group) at Widget, Inc., in which case, both of these distinguished names would be present as individual values of the member attribute." [RFC 4519, 2.17]</p>
organisationName	<p>"The 'o' ('organizationName' in X.500) attribute type contains the names of an organization. Each name is one value of this multi-valued attribute. [...]</p> <p>Examples: '?Widget?', '?Widget, Inc.?', and '?Widget, Incorporated.?'" [RFC 4519, 2.19]</p>
organisationalUnitName	<p>"The 'ou' ('organizationalUnitName' in X.500) attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. [...]</p> <p>Examples: "Finance", "Human Resources", and "Research and Development"." [RFC 4519, 2.20]</p>
owner	<p>"The 'owner' attribute type contains the distinguished names of objects that have an ownership responsibility for the object that is owned. Each owner's name is one value of this multi-valued attribute. [...]</p> <p>Example: The mailing list object, whose DN is 'cn=All Employees, ou=Mailing List,o=Widget\, Inc.', is owned by the Human Resources Director. Therefore, the value of the 'owner' attribute within the mailing list object, would be the DN of the director (role): 'cn=Human Resources Director,ou=employee,o=Widget\, Inc.'" [RFC 4519, 2.21]</p>
physicalDeliveryOfficeName	<p>"The 'physicalDeliveryOfficeName' attribute type contains names that a Postal Service uses to identify a post office. [...]</p> <p>Examples: 'Bremerhaven, Main' and 'Bremerhaven, Bonnstrasse'."</p>

	[RFC 4519, 2.22]
preferredLanguage	This allows to set the users preferred language. [RFC 2798]
postalAddress	"The 'postalAddress' attribute type contains addresses used by a Postal Service to perform services for the object. Each address is one value of this multi-valued attribute. [...] Example: '15 Main St.\$Ottawa\$Canada'. [RFC 4519, 2.23]
postalCode	"The 'postalCode' attribute type contains codes used by a Postal Service to identify postal service zones. Each code is one value of this multi-valued attribute. [...] Example: '22180', to identify Vienna, VA, in the USA." [RFC 4519, 2.24]
postOfficeBox	"The 'postOfficeBox' attribute type contains postal box identifiers that a Postal Service uses when a customer arranges to receive mail at a box on the premises of the Postal Service. Each postal box identifier is a single value of this multi-valued attribute. [...] Example: 'Box 45'. [RFC 4519, 2.25]
preferredDeliveryMethod	"The 'preferredDeliveryMethod' attribute type contains an indication of the preferred method of getting a message to the object. [...] Example: If the mhs-delivery Delivery Method is preferred over telephone-delivery, which is preferred over all other methods, the value would be: 'mhs \$ telephone'. [RFC 4519, 2.26]
registeredAddress	"The 'registeredAddress' attribute type contains postal addresses suitable for reception of telegrams or expedited documents, where it is necessary to have the recipient accept delivery. Each address is one value of this multi-valued attribute. [...] Example: 'Receptionist\$Widget, Inc.\$15 Main St.\$Ottawa\$Canada'. [RFC 4519, 2.27]
roleOccupant	"The 'roleOccupant' attribute type contains the distinguished names of objects (normally people) that fulfil the responsibilities of a role object. Each distinguished name is one value of this multi-valued attribute. [...] Example: The role object, 'cn=Human Resources Director,ou=Position,o=Widget, Inc.', is fulfilled by two people whose object names are 'cn=Mary

	Smith,ou=employee,o=Widget\, Inc.' and 'cn=James Brown,ou=employee,o=Widget\, Inc.'. The 'roleOccupant' attribute will contain both of these distinguished names, since they are the occupants of this role." [RFC 4519, 2.28]
surname	The 'sn' ('surname' in X.500) attribute type contains name strings for the family names of a person. Each string is one value of this multi-valued attribute. [...] Example: 'Smith'." [RFC 4519, 2.32]
stateOrProvinceName	"The 'st' ('stateOrProvinceName' in X.500) attribute type contains the full names of states or provinces. Each name is one value of this multi-valued attribute. [...] Example: 'California'." [RFC 2256, 2.33]
street	"The 'street' ('streetAddress' in X.500) attribute type contains site information from a postal address (i.e., the street name, place, avenue, and the house number). Each street is one value of this multi-valued attribute. [...] Example: '15 Main St.'." [RFC 4519, 2.34]
telephoneNumber	"The 'telephoneNumber' attribute type contains telephone numbers that comply with the ITU Recommendation E.123 [E.123]. Each number is one value of this multi-valued attribute. [...] Example: '+1 234 567 8901'." [RFC 4519, 2.35]
telexNumber	"The 'telexNumber' attribute type contains sets of strings that are a telex number, country code, and answerback code of a telex terminal. Each set is one value of this multi-valued attribute. [...] Example: '12345\$023\$ABCDE'." [RFC 4519, 2.37]
title	"This attribute contains the title, such as "Vice President", of a person in their organizational context. The "personalTitle" attribute would be used for a person's title independent of their job function." [RFC 2256, "5.13 title"]
uid	"The 'uid' ('userid' in RFC 1274) attribute type contains computer system login names associated with the object. Each name is one value of this multi-valued attribute. [...] Examples: 's9709015', 'admin', and 'Administrator'."

	[RFC 4519, 2.39]
uniqueMember	<p>"The 'uniqueMember' attribute type contains the distinguished names of an object that is on a list or in a group, where the relative distinguished names of the object include a value that distinguishes between objects when a distinguished name has been reused. Each distinguished name is one value of this multi-valued attribute. [...]</p> <p>Example: If 'ou=1st Battalion,o=Defense,c=US' is a battalion that was disbanded, establishing a new battalion with the 'same' name would have a unique identifier value added, resulting in 'ou=1st Battalion, o=Defense,c=US#010101'B'."</p> <p>[RFC 4519, 2.40]</p>
userPassword	<p>"The 'userPassword' attribute contains octet strings that are known only to the user and the system to which the user has access. Each string is one value of this multi-valued attribute. The application SHOULD prepare textual strings used as passwords by transcoding them to Unicode, applying SASLprep [RFC4013], and encoding as UTF-8. The determination of whether a password is textual is a local client matter. [...]</p> <p>Passwords are stored using an Octet String syntax and are not encrypted. Transfer of cleartext passwords is strongly discouraged where the underlying transport service cannot guarantee confidentiality and may result in disclosure of the password to unauthorized parties.</p> <p>An example of a need for multiple values in the 'userPassword' attribute is an environment where every month the user is expected to use a different password generated by some automated system. During transitional periods, like the last and first day of the periods, it may be necessary to allow two passwords for the two consecutive periods to be valid in the system."</p> <p>[RFC 4519, 2.41]</p>
x121Address	<p>"The 'x121Address' attribute type contains data network addresses as defined by ITU Recommendation X.121 [X.121]. Each address is one value of this multi-valued attribute. [...]</p> <p>Example: '36111222333444555'."</p> <p>[RFC 4519, 2.42]</p>
x500UniqueIdentifier	<p>"The 'x500UniqueIdentifier' attribute type contains binary strings that are used to distinguish between objects when a</p>

	distinguished name has been reused. Each string is one value of this multi-valued attribute. In X.520 [X.520], this attribute type is called 'uniqueIdentifier'. This is a different attribute type from both the 'uid' and 'uniqueIdentifier' LDAP attribute types. The 'uniqueIdentifier' attribute type is defined in [RFC4524]." [RFC 4519, 2.43]
labeledURI	An example for a labeled URI would be: "labeledURI: http://www.siroe.com/users/bjensen My Home Page" [RFC 2798]
mail	RFC 2798
mobilePhoneNumber	This attribute represents the number of a mobile-phone. [RFC 2798]
roomNumber	Room number [RFC 2798]

Table 8: Default LDAP Schema

Some properties may be left blank when creating a user profile, but the following properties must be provided in order to enable the enforcement of a cross-network access policy:

- commonName
- givenName
- surname
- organisationName
- organisationalUnitName
- roleOccupant