



Service User Profile Management Technical Specification

SOAP binding for SUPM-1 interface

Approved Version 1.0 - 29 May 2012

Open Mobile Alliance

OMA-TS-Service_User_Profile_Management_SOAP_Binding-V1_0-
20120529-A

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2012 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

| | | |
|-------------|-------------------------------------------------------------------------|----|
| 1. | SCOPE..... | 4 |
| 2. | REFERENCES..... | 5 |
| 2.1 | NORMATIVE REFERENCES..... | 5 |
| 2.2 | INFORMATIVE REFERENCES..... | 5 |
| 3. | TERMINOLOGY AND CONVENTIONS..... | 6 |
| 3.1 | CONVENTIONS..... | 6 |
| 3.2 | DEFINITIONS..... | 6 |
| 3.3 | ABBREVIATIONS..... | 6 |
| 4. | INTRODUCTION..... | 7 |
| 4.1 | VERSION 1.0..... | 7 |
| 5. | SUPM SOAP BINDING PRINCIPLES..... | 8 |
| 5.1 | SCOPE OF REUSE OF THE LIBERTY ALLIANCE ID-WSF FRAMEWORK..... | 8 |
| 5.1.1 | Instantiation of the Liberty Alliance ID-WSF Data Service Template..... | 8 |
| 5.2 | NAMESPACES..... | 11 |
| 6. | OPERATIONS/MESSAGES..... | 12 |
| 6.1 | SERVICE USER PROFILE MANAGEMENT..... | 12 |
| 6.1.1 | Operation: CreateServiceUserProfileData..... | 12 |
| 6.1.2 | Operation: ReadServiceUserProfileData..... | 13 |
| 6.1.3 | Operation: UpdateServiceUserProfileData..... | 14 |
| 6.1.4 | Operation: DeleteServiceUserProfileData..... | 14 |
| 6.2 | STATUS AND FAULT REPORTING..... | 15 |
| 7. | SUPM SOAP DATA STRUCTURE..... | 16 |
| 7.1 | TYPE: APPDATATYPE..... | 16 |
| 7.2 | TYPE: ATTRIBUTE..... | 16 |
| 8. | EXAMPLES (INFORMATIVE)..... | 17 |
| 8.1 | OPERATION: CREATESERVICEUSERPROFILEDATA..... | 17 |
| 8.2 | OPERATION: READSERVICEUSERPROFILEDATA..... | 18 |
| 8.3 | OPERATION: UPDATESERVICEUSERPROFILEDATA..... | 22 |
| 8.4 | OPERATION: DELETESERVICEUSERPROFILEDATA..... | 24 |
| APPENDIX A. | CHANGE HISTORY (INFORMATIVE)..... | 26 |
| A.1 | APPROVED VERSION HISTORY..... | 26 |
| APPENDIX B. | STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)..... | 27 |
| B.1 | SCR FOR SUPM.SOAP SERVER..... | 27 |
| B.2 | SCR FOR SUPM.SOAP.ATTRIBUTESFORUSERPROFILE SERVER..... | 27 |
| B.3 | SCR FOR SUPM.SOAP.INDIVIDUALATTRIBUTESOFUSERPROFILE SERVER..... | 27 |
| B.4 | SCR FOR SUPM.SOAP.DATAVIEWSOFUSERPROFILE SERVER..... | 28 |
| APPENDIX C. | SUPM-1 SOAP SEQUENCE DIAGRAM (INFORMATIVE)..... | 29 |

1. Scope

The scope of this document is to specify a SOAP protocol binding for the set of operations defined in [SUPM-TS].

SOAP/XML can be used to access and manage (create, read, update, delete) the data related to Service User Profile. OMA enablers or other resources/services can read and update those data using SOAP/XML as described in this document.

2. References

2.1 Normative References

- [LibertySOAPBinding] "Liberty ID-WSF SOAP Binding Specification", Liberty Alliance, version 2.0-errata-v1.0, URL: <http://www.projectliberty.org/liberty/content/download/3483/23075/file/liberty-idwsf-soap-binding-2.0-errata-v1.0.pdf>
- [LibertyDST] "Liberty ID-WSF Data Services Template", Liberty Alliance, version 2.1, URL: <http://www.projectliberty.org/liberty/content/download/879/6213/file/liberty-idwsf-dst-v2.1.pdf>
- [RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:<http://www.ietf.org/rfc/rfc2119.txt>
- [RFC4234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL:<http://www.ietf.org/rfc/rfc4234.txt>
- [SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:<http://www.openmobilealliance.org/>
- [SUPM-AD] "OMA Service User Profile Management Architecture", Open Mobile Alliance™, OMA-AD-Service_User_Profile_Management-V1_0, URL:<http://www.openmobilealliance.org/>
- [SUPM-RD] "OMA Service User Profile Management Requirements", Open Mobile Alliance™, OMA-RD-Service_User_Profile_Management-V1_0, URL:<http://www.openmobilealliance.org/>
- [SUPM-TS] "Service User Profile Management Technical Specification", Open Mobile Alliance™, OMA-TS-Service_User_Profile_Management-V1_0, URL:<http://www.openmobilealliance.org/>
- [XML] "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation 6-October-2000. T. Bray, et al, 6-October-2000. URL: <http://www.w3.org/TR/REC-xml>

2.2 Informative References

- [OMADICT] "Dictionary for OMA Specifications", Version 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_8, URL:<http://www.openmobilealliance.org/>

3. Terminology and Conventions

3.1 Conventions

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

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

3.2 Definitions

| | |
|----------------------------------------------------------|---------------|
| Application | See [OMADICT] |
| Authorized Principal | See [OMADICT] |
| Principal | See [OMADICT] |
| Resource | See [OMADICT] |
| Service | See [OMADICT] |
| Service Provider | See [OMADICT] |
| Service User Profile | See [SUPM-RD] |
| SUPM Data View | See [SUPM-AD] |
| User Profile | See [OMADICT] |
| User | See [OMADICT] |
| User’s Characteristic Description Information | See [SUPM-RD] |

3.3 Abbreviations

| | |
|-------------|---------------------------------|
| AD | Architecture Document |
| HTTP | HyperText Transfer Protocol |
| OMA | Open Mobile Alliance |
| RD | Requirement Document |
| REST | REpresentational State Transfer |
| SUPM | Service User Profile Management |
| TS | Technical Specification |
| XML | Extensible Mark-up Language |

4. Introduction

The SUPM enabler allows an authorised principal to manipulate Services User Profile Management data, i.e. any element or group of element belonging to a managed set of information related to a User that may be used to create personalized and contextualized services. The set of information may include both static and dynamic information.

For a SOAP/XML interface, the corresponding requests and responses are defined in this specification.

4.1 Version 1.0

This document covers all requirements [SUPM-RD] of SUPM V 1.0.

The SUPM enabler 1.0 supports requests to read/update/create/delete Service User Profile data.

5. SUPM SOAP Binding principles

This section describes the general principles followed for the creation of this SUPM SOAP Binding and especially how it leverages the Liberty Alliance ID-WSF specifications.

5.1 Scope of reuse of the Liberty Alliance ID-WSF framework

This binding leverages two Liberty Alliance specifications of the ID-WSF framework specification set: Liberty ID-WSF SOAP Binding Specification [LibertySOAPBinding] and the Liberty ID-WSF Data Service Template [LibertyDST].

These two specifications are used to cover the SUPM-1 interface features as specified in [SUPM-TS] and are each used for a specific purpose:

- **SOAP Headers:** reuse of the header blocks defined in [LibertySOAPBinding] in order to convey the generic invocation parameters defined in [SUPM-TS], such as "Originator" and "UserId".
- **SOAP Body:** instantiation of the CRUD template defined in [LibertyDST] in order to specify the SUPM-1 create/read/update/delete operations.

This binding has no other MANDATORY dependency on other ID-WSF specifications.

This binding makes no assumption on how the identity of the user is retrieved by a data consumer and thus does not introduce any dependency with specific Identity frameworks (e.g. Liberty ID-FF, SAML or OpenID).

5.1.1 Instantiation of the Liberty Alliance ID-WSF Data Service Template

The SUPM SOAP Binding corresponds to an instantiation of the Liberty ID-WSF Data Service Template [LibertyDST]. This [LibertyDST] specification defines a template that can be instantiated by services in order to provide protocols for the creation, read (or query), update (or modification), and deletion (a.k.a. "CRUD") of data attributes, exposed by a data service, related to a Principal. Some guidelines, common XML attributes and data types are also defined for data services.

In concrete terms, this document:

- Enumerates methods defined in [LibertyDST] that are supported and states the required level of support. The default set of methods is <Create>, <Query>, <Modify>, and <Delete>.
- Fills place holders defined in [LibertyDST] that are specific to the service instantiation (SUPM in our context).

The common rules defined in [LibertyDST] that MUST be followed for the SUPM SOAP Binding are specified in the following chapters of [LibertyDST]: introductory part of §3 followed by § 3.1, §3.2, §3.3, §3.4, §3.5 (chapters §3.6, §3.7, §3.8 are refined in this specification – *see the two chapters below* – and chapters §3.9, §3.10 are out-of-scope of this specification).

Other rules defined in [LibertyDST] that are operation-specific are referenced in the chapter **Error! Reference source not found.** of this specification.

5.1.1.1 Identification of the resource (on which the operation applies)

In [LibertyDST], the resource (i.e., the Service User Profile in our context) on which the operation applies is always related to a specific user.

In the context of this specification, the <TargetIdentity> SOAP header defined in [LibertySOAPBinding] MUST be used to identify the user for whom the Service User Profile is created/read/updated/deleted.

5.1.1.2 Selection

The [LibertyDST] specification defines "selection" mechanisms to specify more in details what is manipulated within the identified resource.

The generic selection principles and associated processing rules expressed in chapters §3.7 and §3.8 of the [LibertyDST] specification MUST be followed in this specification with the following refinements:

- As all SUPM operations defined in this binding are defined to access to attributes (or data elements as defined in [SUPM-TS]) of the Service User Profile, the value of the XML "objectType" attribute defined in [LibertyDST] MUST be set to "Attribute" when provided.
- When provided, the <Select> element defined in [LibertyDST] MUST either be empty (default value if omitted) or set to an attribute name. If empty, the scope of the operation encompasses all attributes of the Service User Profile. If set to the name of an attribute, the scope of the operation is restricted to that element only.
- The "predefined" XML attribute defined in [LibertyDST] has been designed to specify pre-defined selections of data elements. It well corresponds to the notion of view introduced in [SUPM-TS]. Therefore, this mechanism MUST be used to reference a view. When provided, the value of the "predefined" XML attribute MUST be set to the name of a view. The scope of the operation then encompasses all attributes of that view.

5.1.1.3 ID-WSF DST reused features

The following table based on the check-list provided in [LibertyDST] states which features are supported in this SUPM binding specification:

| | | |
|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Specify service type. | N/A. Integration with a Liberty ID-WSF Discovery Service is out-of-scope of this specification. |
| 2 | Provide definition service schema including the methods as elements based on DST types. | SUPM Soap Binding XSD provided based on the DST Reference Model. XML Elements and Types not supported in this SUPM SOAP binding specification have been removed. |
| 3 | Enumerate object types. | See §5.1.1.2 in this SUPM SOAP binding specification. |
| 4 | Describe AppDataType and its contents. The description can come in form of XML schema, or the description can simply describe the contents of the string that is to appear in elements derived from AppDataType, i.e., <NewData>, <Data>, and <ItemData>. The data description may make allowance for different object types. | See §7 in this SUPM SOAP binding specification and provided XSD. <ItemData> is not defined as <ResultQuery> is not supported (see 12.b). |
| 5 | Describe SelectType and how it applies to various types of objects. | See §5.1.1.2 in this SUPM SOAP binding specification. The SelectType has not been redefined in the XSD (string). |
| 6 | Describe TestOpType, considering how to test all object types supported by the specification | N/A as Testing feature is not supported (see 12.a). |
| 7 | Describe SortType. | N/A as Sorting feature is not supported (see 12.c). |
| 8 | Enumerate Methods and state the required level of support. The default set of methods is <Create>, <Query>, <Modify>, and <Delete>. All methods MUST be supported, but each method MAY be disabled administratively or by configuration in a deployment (e.g. to provide read only or write only service). | See § Error! Reference source not found. in this SUPM SOAP binding specification. |
| 9 | The discovery option keywords (see [LibertyDisco]) can either be listed with semantics here, or via a reference to the correct | N/A. Integration with a Liberty ID-WSF Discovery Service is out-of-scope of this specification. |

| | | |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | chapter in the specification. | |
| 10 | Element uniqueness. State here how elements with the same name are distinguished from each other. | SUPM data elements are uniquely identified through their names. |
| 11 | Extension support. State whether extension is supported and if so, describe this support. A reference to the specification chapter defining this can be given. | Extension feature SHALL NOT be supported. |
| 12 | Statement of optionality of query features (and their manifestation on discovery option keywords, see above): a. Support testing b. Support <ResultQuery> c. Support sorting d. Support pagination of results e. Support static sets in pagination f. Support multiple <Query> elements g. Support multiple <QueryItem> elements h. Support multiple <TestItem> elements i. Support changedSince (and which formats) in <ResultQuery> and <QueryItem> j. Support includeCommonAttributes | Level of support of query features in this binding specification is the following: a. Testing SHALL NOT be supported. b. <ResultQuery> SHALL NOT be supported. c. Sorting SHALL NOT be supported. d. Pagination SHALL NOT be supported. e. N/A as Pagination is not supported (see 12.d). f. Multiple <Query> elements SHALL NOT be supported. g. Multiple <QueryItem> elements MUST be supported. h. N/A as Testing is not supported (see 12.a). i. changedSince feature SHALL NOT be supported. j. includeCommonAttributes feature SHALL NOT be supported. |
| 13 | Statement of optionality of create features (and their manifestation on discovery option keywords, see above): a. Support multiple <Create> elements | Level of support of create features in this binding specification is the following: a. Multiple <Create> elements SHALL NOT be supported. |
| 14 | Statement of optionality of delete features (and their manifestation on discovery option keywords, see above): a. Support multiple <Delete> elements | Level of support of delete features in this binding specification is the following: a. Multiple <Delete> elements SHALL NOT be supported. |
| 15 | Statement of optionality of modify features (and their manifestation on discovery option keywords, see above): a. Support multiple <Modify> elements b. Support multiple <ModifyItem> elements c. Support partial success. If multiple <ModifyItem> elements are supported, is partial success supported or are only atomic modifications allowed? | Level of support of modify features in this binding specification is the following: a. Multiple <Modify> elements SHALL NOT be supported. b. Multiple <ModifyItem> elements MUST be supported. c. Partial success MUST be supported. No atomic modify or transactional semantic needs to be supported. d. notChangedSince feature SHALL NOT be supported. |

| | | |
|--|----------------------------|--|
| | d. Support notChangedSince | |
|--|----------------------------|--|

5.2 Namespaces

The namespace used for the service MUST be the following: "**urn:oma:xml:supm:soap:1**".

This namespace is abbreviated as "**supm:**" in this document. If the namespace has been omitted at any place in this document, "**supm:**" is to be considered to be the default namespace.

6. Operations/messages

This section describes the SUPM operations supported by the SUPM SOAP Binding.

All operations are request/response - for every initiated message, there is a response back to the data consumer (initiator of the request).

6.1 Service User profile management

6.1.1 Operation: CreateServiceUserProfileData

A data consumer can create a new Service User Profile for a specific user using the <Create> operation defined in [LibertyDST].

The <Create> element MUST be constructed according to the following rules:

- A <Create> element MUST contain only one <CreateItem> element.
- The "objectType" XML attribute of a <CreateItem> MUST be omitted.
- The <Select> sub-element of a <CreateItem> MUST be omitted or set to empty.
- The "predefined" XML attribute of a <CreateItem> MUST be omitted.
- The <CreateItem> element MUST specify the initial content inside the <NewData> element. This <NewData> element MUST be either empty or contain one or more <attribute> elements (see §7).

It is REQUIRED to use the Update operation to create individual attributes of a Service User Profile.

The <CreateResponse> element provides the result of this operation.

6.1.1.1 Input message: <Create>

| Abstract part name | SOAP / ID-WSF part name | Optional | Description |
|----------------------|---------------------------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Originator | <Sender> header | Yes | The data consumer which requests the creation of a Service User Profile for a specific user. This <sender> header defined in [LibertySOAPBinding] is optional as the data consumer may be identified implicitly based on mutual authentication mechanisms |
| UserId | <TargetIdentity> header | No | Identification of the user for whom the profile is created. |
| AttributeValue Pairs | <NewData> element inside <CreateItem> | No | List of attribute names and attribute values provided as initial data for the newly created Service User Profile. |

6.1.1.2 Output message: <CreateResponse>

| Abstract part name | SOAP / ID-WSF part name | Optional | Description |
|--------------------|-------------------------|----------|------------------------------------------------------------------------------------------------------|
| ResultCode | <Status code="..."> | No | Success indication or error code (with potential second level status code to refine the error code). |
| ErrorMessage | <Status comment="..."> | Yes | Optional error diagnostic description. |

6.1.2 Operation: ReadServiceUserProfileData

A data consumer can read service user profile data using the <Query> operation defined in [LibertyDST].

The <Query> element MUST be constructed according to the following rules:

- A <Query> element MAY contain multiple <QueryItem> elements to request multiple attributes or views in a single Read operation.
- The selection mechanisms defined in §5.1.1.2 of this document MUST be used to either request all attributes of the user's Service User Profile or all attributes of a specific view or a particular attribute.

The <QueryResponse> element provides the result of this operation. As stated in [LibertyDST], the requested data (<attribute> element(s) in our context) is encapsulated inside <Data> elements. One <Data> element contains data requested by one <QueryItem> element. If there were multiple <QueryItem> elements in the <Query>, the <Data> elements are linked to their corresponding <QueryItem> elements using the itemIDRef XML attributes.

6.1.2.1 Input message: <Query>

| Abstract part name | SOAP / ID-WSF part name | Optional | Description |
|--------------------|------------------------------------------------------------------------------------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Originator | <Sender> header | Yes | The data consumer who wants to read the Service User Profile of a specific user. This <sender> header defined in [LibertySOAPBinding] is optional as the data consumer may be identified implicitly based on mutual authentication mechanisms |
| UserId | <TargetIdentity> header | No | Identification of the user for whom the profile is requested |
| AttributeNames | <Select> element inside the <QueryItem> element(s) or "predefined" XML attribute of <QueryItem> element(s) | No | A data consumer can request the whole profile, a view or a specific attribute per <QueryItem> through the selection mechanisms defined in §5.1.1.2 of this document. In order to request multiple views or multiple attributes, or combination of views and attributes, a data consumer SHOULD provide several <QueryItem> elements inside the <Query> operation. |

6.1.2.2 Output message: <QueryResponse>

| Abstract part name | SOAP / ID-WSF part name | Optional | Description |
|--------------------|-------------------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ResultEntries | <Data> element(s) | Yes | Returns the requested attribute(s) (as defined in §7). The response can contain multiple <Data> elements if multiple <QueryItem> were provided part of the <Query> operation. NOTE: Check id dataview is returned as well |
| ResultCode | <Status code="..."> | No | Success indication or error code (with potential second level status code to refine the error code). |
| ErrorMessage | <Status | Yes | Optional error diagnostic description. |

| | | | |
|--|----------------|--|--|
| | comment="..."> | | |
|--|----------------|--|--|

6.1.3 Operation: UpdateServiceUserProfileData

A data consumer can update service user profile data using the **<Modify>** element defined in [LibertyDST].

The **<Modify>** element MUST be constructed according to the following rules:

- A **<Modify>** element MAY contain multiple **<ModifyItem>** elements to request multiple updates on attributes through a single Update operation. **<attribute>** element(s) to be updated are provided in the **<NewData>** element of a **<ModifyItem>**. If the **<NewData>** element is empty, all selected attributes are deleted (set to "undefined" as specified in [SUPM-TS]).
- The "overrideAllowed" attribute of **<ModifyItem>** elements MUST be set to "true" or "1".
-

The **<ModifyResponse>** element provides the result of this operation.

6.1.3.1 Input message: **<Modify>**

| Abstract part name | SOAP / ID-WSF part name | Optional | Description |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Originator | <Sender> header | Yes | The data consumer which requests the update of the Service User Profile for a specific user. This <sender> header defined in [LibertySOAPBinding] is optional as the data consumer may be identified implicitly based on mutual authentication mechanisms |
| UserId | <TargetIdentity> header | No | Identification of the user for whom the profile is updated. |
| AttributeValue Pairs | <NewData> element inside <ModifyItem> element(s) qualified through the <Select> element. | No | A data consumer can update the whole Service User Profile or several attributes (a specific attribute per <ModifyItem>). |

6.1.3.2 Output message: **<ModifyResponse>**

| Abstract part name | SOAP / ID-WSF part name | Optional | Description |
|--------------------|-------------------------------------|----------|------------------------------------------------------------------------------------------------------|
| ResultCode | <Status code="..."> | No | Success indication or error code (with potential second level status code to refine the error code). |
| ErrorMessage | <Status comment="..."> | Yes | Optional error diagnostic description. |

6.1.4 Operation: DeleteServiceUserProfileData

A data consumer can delete a Service User Profile for a specific user using the **<Delete>** element defined in [LibertyDST].

The **<Delete>** element MUST be constructed according to the following rules:

- A **<Delete>** element MUST contain only one **<DeleteItem>** element.

- The "objectType" XML attribute of a <DeleteItem> MUST be omitted.
- The <Select> sub-element of a <DeleteItem> MUST be omitted or set to empty.
- The "predefined" XML attribute of a <DeleteItem> MUST be omitted.

It is REQUIRED to use the Update operation to delete individual attributes or views of a Service User Profile.

The <DeleteResponse> element provides the result of this operation.

6.1.4.1 Input message: <Delete>

| Abstract part name | SOAP / ID-WSF part name | Optional | Description |
|--------------------|-------------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Originator | <Sender> | Yes | The data consumer which requests the deletion of the Service User Profile for a specific user. This <sender> header defined in [LibertySOAPBinding] is optional as the data consumer may be identified implicitly based on mutual authentication |
| UserId | <TargetIdentity> | No | Identification of the user for whom the profile is deleted. |

6.1.4.2 Output message: <DeleteResponse>

| Abstract part name | SOAP / ID-WSF part name | Optional | Description |
|--------------------|-------------------------|----------|------------------------------------------------------------------------------------------------------|
| ResultCode | <Status code="..."> | No | Success indication or error code (with potential second level status code to refine the error code). |
| ErrorMessage | <Status comment="..."> | Yes | Optional error diagnostic description. |

6.2 Status and Fault reporting

The processing rules defined in chapter §3.2 "Status and Fault Reporting" of the [LibertyDST] specification MUST be followed.

7. SUPM SOAP Data Structure

7.1 Type: AppDataType

The <Data> and <NewData> XML elements are used to convey SUPM data elements inherit from the XML "AppDataType" defined as following:

| Element | Type | Optional | Description |
|-----------|-----------------------------|----------|--------------------------------|
| attribute | Attribute [0..unbounded] | Yes | Contains a list of attributes. |

Note: to check if the element is attribute or attributeList

7.2 Type: Attribute

SUPM data elements are conveyed as <attribute> elements of type "Attribute" defined as following:

| Element | Type | Optional | Description |
|----------------|-------------|----------|----------------------------------------------------------------------|
| attributeName | xsd:string | No | Name of the attribute |
| attributeValue | xsd:anyType | Yes | The value(s) of the attribute. Complex types SHALL NOT be supported. |

8. Examples

(Informative)

In the following examples :

- If the <Query> contains multiple <QueryItem> elements, an itemID XML attribute must be add to each <QueryItem> element in order to distinguish sub-queries and corresponding answers.
- The <ModifyItem> element MUST also have the itemID XML attribute, when multiple <ModifyItem> elements are included in one <Modify> element.

8.1 Operation: CreateServiceUserProfileData

The following is an example of a Create request.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">
  <S:Header>

  <sb:Sender providerID="http://dataconsumer1.com"/>
  <sb:TargetIdentity>mailto: bob@example.com</sb:TargetIdentity>

  </S:Header>
  <S:Body>
  <supm:Create>
  <supm:CreateItem>
  <supm:NewData>
  <supm:attribute>
  <supm:attributeName>Title</supm:attributeName>
  <supm:attributeValue xsi:type="xs:string">Mr</supm:attributeValue>
  </supm:attribute>
  <supm:attribute>
  <supm:attributeName>PreferredLang</supm:attributeName>
  <supm:attributeValue xsi:type="xs:string">FR</supm:attributeValue>
  </supm:attribute>
  </supm:NewData>
  </supm:CreateItem>
  </supm:Create>
  </S:Body>
```

```
</S:Envelope>
```

The following is an example of a Create response.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:lu="urn:liberty:util:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">
<S:Header>

  <sb:Sender providerID="http://supmenablerA.com"/>

</S:Header>
<S:Body>
  <supm:CreateResponse timeStamp="2010-06-17T04:49:18Z">
    <lu>Status code="OK"/>
  </supm:CreateResponse>
</S:Body>
</S:Envelope>
```

8.2 Operation: ReadServiceUserProfileData

The following is an example of a Read request. This example asks for an attribute (sub-query identified by itemId="sup1") and for a view (sub-query identified by "itemId="sup2"). The view name is "CABData" (defined by "predefined="CABData").

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:lu="urn:liberty:util:2006-08"
  xmlns:dst="urn:liberty:dst:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">
<S:Header>
```

```

<sb:Sender providerID="http://dataconsumer1.com"/>
<sb:TargetIdentity>mailto: bob@example.com</sb:TargetIdentity>

</S:Header>
<S:Body>
  <supm:Query>
    <supm:QueryItem lu:itemID="sup1">
      <supm:Select>Title</supm:Select>
    </supm:QueryItem>
    <supm:QueryItem lu:itemID="sup2" dst:predefined="CABData"/>
  </supm:Query>
</S:Body>
</S:Envelope>

```

The following is an example of a Read response.

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:lu="urn:liberty:util:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">
  <S:Header>
    <sb:Sender providerID="http://supmenablerA.com"/>
  </S:Header>
  <S:Body>
    <supm:QueryResponse timeStamp="2010-06-17T04:49:18Z">
      <lu:Status code="OK"/>
      <supm:Data lu:itemIDRef="sup1">
        <supm:attribute>
          <supm:attributeName>Title</supm:attributeName>
          <supm:attributeValue xsi:type="xs:string">Mr</supm:attributeValue>
        </supm:attribute>
      </supm:Data>
      <supm:Data lu:itemIDRef="sup2">
        <supm:attribute>

```

```

        <supm:attributeName>Name1</supm:attributeName>
        <supm:attributeValue xsi:type="xs:string">Value1</supm:attributeValue></supm:attribute>
    </supm:attribute>
        <supm:attributeName>Name2</supm:attributeName>
        <supm:attributeValue xsi:type="xs:string">Value2</supm:attributeValue></supm:attribute>
    </supm>Data>
</supm:QueryResponse>
</S:Body>
</S:Envelope>

```

Read response in case of error for one of the two items

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:sb="urn:liberty:sb:2006-08"
    xmlns:lu="urn:liberty:util:2006-08"
    xmlns:supm="urn:oma:xml:supm:soap:1">
<S:Header>

    <sb:Sender providerID="http://supmenablerA.com"/>

</S:Header>
<S:Body>
    <supm:QueryResponse timeStamp="2010-06-17T04:49:18Z">
        <lu:Status code="Partial">
            <lu:Status code="UnexpectedError" ref="sup2"/>
        </lu:Status>
        <supm>Data lu:itemIDRef="sup1">
            <supm:attribute>
                <supm:attributeName>Title</supm:attributeName>
                <supm:attributeValue xsi:type="xs:string">Mr</supm:attributeValue>
            </supm:attribute>
        </supm>Data>
    </supm:QueryResponse>
</S:Body>
</S:Envelope>

```

Note that some implementations might also directly return (implementation choice)

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:lu="urn:liberty:util:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">
<S:Header>

  <sb:Sender providerID="http://supmenablerA.com"/>

</S:Header>
<S:Body>
  <supm:QueryResponse timeStamp="2010-06-17T04:49:18Z">
    <lu:Status code="Failed">
      <lu:Status code="UnexpectedError"/>
    </lu:Status>
  </supm:QueryResponse>
</S:Body>
</S:Envelope>
```

The following are 2 examples of a read request asking for a whole user profile

- 1st example

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">
<S:Header>

  <sb:Sender providerID="http://dataconsumer1.com"/>
  <sb:TargetIdentity>mailto: bob@example.com</sb:TargetIdentity>

</S:Header>
<S:Body>
```

```

<supm:Query>
  <supm:QueryItem/>
</supm:Query>
</S:Body>
</S:Envelope>

```

- 2nd example

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">
<S:Header>

  <sb:Sender providerID="http://dataconsumer1.com"/>
  <sb:TargetIdentity>mailto: bob@example.com</sb:TargetIdentity>

</S:Header>
<S:Body>
  <supm:Query>
    <supm:QueryItem >
      <supm:Select/>
    </supm:QueryItem>
  </supm:Query>
</S:Body>
</S:Envelope>

```

8.3 Operation: UpdateServiceUserProfileData

The following is an example of an Update request.

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">

```

```

<S:Header>

  <sb:Sender providerID="http://dataconsumer1.com"/>
  <sb:TargetIdentity>mailto: bob@example.com</sb:TargetIdentity>

</S:Header>
<S:Body>
  <supm:Modify>
    <supm:ModifyItem overrideAllowed="true">
      <supm:Select>Title</supm:Select>
      <supm:NewData>
        <supm:attribute>
          <supm:attributeName>Title</supm:attributeName>
          <supm:attributeValue xsi:type="xs:string">Mr</supm:attributeValue>
        </supm:attribute>
      </supm:NewData>
    </supm:ModifyItem>
  </supm:Modify>
</S:Body>
</S:Envelope>

```

The following is an example of an Update response.

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:lu="urn:liberty:util:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">
  <S:Header>

    <sb:Sender providerID="http://supmenablerA.com"/>

  </S:Header>
  <S:Body>
    <supm:ModifyResponse timeStamp="2010-06-17T04:49:18Z">
      <lu:Status code="OK"/>
    </supm:ModifyResponse>
  </S:Body>
</S:Envelope>

```

```

</supm:ModifyResponse>
</S:Body>
</S:Envelope>

```

8.4 Operation: DeleteServiceUserProfileData

The following is an example of a Delete request.

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">
<S:Header>

  <sb:Sender providerID="http://dataconsumer1.com"/>
  <sb:TargetIdentity>mailto: bob@example.com</sb:TargetIdentity>

</S:Header>
<S:Body>
  <supm:Delete>
    <supm:DeleteItem />
  </supm:Delete>
</S:Body>
</S:Envelope>

```

The following is an example of a Delete response.

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:lu="urn:liberty:util:2006-08"
  xmlns:supm="urn:oma:xml:supm:soap:1">
<S:Header>

  <sb:Sender providerID="http://supmenablerA.com"/>

```



```
</S:Header>
<S:Body>
  <supm:DeleteResponse>
    <lu:Status code="OK"/>
  </supm:DeleteResponse>
</S:Body>
</S:Envelope>
```

Appendix A. Change History

(Informative)

A.1 Approved Version History

| Reference | Date | Description |
|---------------------------------------------------------------------|-------------|----------------------------------------------------------------------------------------------------|
| OMA-TS-Service_User_Profile_Management_SOAP_Binding-V1_0-20120529-A | 29 May 2012 | Status changed to Approved by TP Ref TP Doc # OMA-TP-2012-0204-INP_SUPM_V1_0_for_Final_Approval |

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for SUPM.SOAP Server.

| Item | Function | Reference | Requirement |
|---------------------------|--------------------------------------------------|-----------|-------------|
| SUPM-SOAP-SUPPORT-S-001-O | Support for the SOAP binding of SUPM-1 interface | 6 | |
| SUPM-SOAP-SUPPORT-S-002-M | Support for the XML request & response format | 6 | |

B.2 SCR for SUPM.SOAP.AttributesForUserProfile Server

| Item | Function | Reference | Requirement |
|--------------------------|-----------------------------------------------------------------|-----------|-------------|
| SUPM-SOAP-ATTRIB-S-001-M | Management support for all attributes for a given user profile. | 6.1 | |
| SUPM-SOAP-ATTRIB-S-002-M | This operation creates a Service User Profile -Create | 6.1.1 | |
| SUPM-SOAP-ATTRIB-S-003-M | This operation returns all attributes - Query | 6.1.2 | |
| SUPM-SOAP-ATTRIB-S-004-M | This operation updates all the attributes-Modify | 6.1.3 | |
| SUPM-SOAP-ATTRIB-S-005-M | This operation deletes a Service User Profile - Delete | 6.1.4 | |

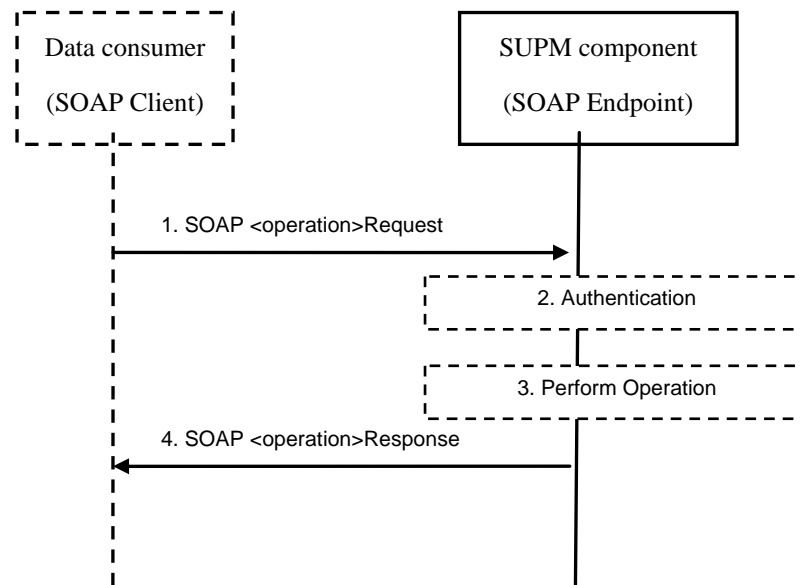
B.3 SCR for SUPM.SOAP.IndividualAttributesOfUserProfile Server

| Item | Function | Reference | Requirement |
|------------------------------|-----------------------------------------------------------------------------------------------|-----------|-------------|
| SUPM.SOAP-IND-ATTRIB-S-001-M | Support for management (create, update, retrieve) of individual attributes for a user profile | 6.1 | = |
| SUPM-SOAP-IND-ATTRIB-S-002-M | This operation returns the value of an user profile attribute - Query | 6.1.2 | |
| SUPM.SOAP-IND-ATTRIB-S-003-M | This operation creates or updates an attribute of an user profile-Modify | 6.1.3 | |
| | | | |

B.4 SCR for SUPM.SOAP.DataViewsOfUserProfile Server

| Item | Function | Reference | Requirement |
|----------------------|----------------------------------------------------------------------------------------------------|-----------|----------------------|
| SUPM.SOAP-DV-S-001-O | Support for operations (retrieve) for a Data View of an user profile | 6.1 | SUPM.SOAP-DV-S-002-O |
| SUPM.SOAP-DV-S-002-O | This operation returns the list of attributes and values of a Data View of an user profile - Query | 6.1.2 | |

Appendix C. SUPM-1 SOAP sequence diagram (Informative)



1. Data consumer issues a SUPM-1 SOAP <operation>Request.
2. SUPM component performs data consumer authentication (if mutual authentication mechanisms are enforced).
3. SUPM component performs the operation for the identified Service User Profile data. SUPM component determines based on configuration at deployment time, where the actual data specified in the request resides and reads (or creates, deletes, modifies) the data from the data source.
4. SUPM component returns the result of the operation to the data consumer using the corresponding SUPM-1 SOAP operation Response.