Service User Profile Management Technical Specification

SOAP binding for SUPM-1 interface
Candidate Version 1.0 - 11 Jan 2011

Open Mobile Alliance
OMA-TS-Service_User_Profile_Management_SOAP_Binding-V1_0-20110111-C
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 ................................................................................................................................................... 10
6. OPERATIONS/MESSAGES ...................................................................................................................................... 11
   6.1 SERVICE USER PROFILE MANAGEMENT .................................................................................................... 11
      6.1.1 Operation: CreateServiceUserProfileData ................................................................................................ 11
      6.1.2 Operation: ReadServiceUserProfileData .................................................................................................... 11
      6.1.3 Operation: UpdateServiceUserProfileData .................................................................................................. 12
      6.1.4 Operation: DeleteServiceUserProfileData ................................................................................................ 13
   6.2 STATUS AND FAULT REPORTING ....................................................................................................................... 14
7. SUPM SOAP DATA STRUCTURE ............................................................................................................................ 15
   7.1 TYPE: APPDATATYPE ....................................................................................................................................... 15
   7.2 TYPE: ATTRIBUTE .......................................................................................................................................... 15
8. EXAMPLES (INFORMATIVE) .................................................................................................................................. 16
   8.1 OPERATION: CREATESERVICEUSERPROFILEDATA .......................................................................................... 16
   8.2 OPERATION: READSERVICEUSERPROFILEDATA ............................................................................................ 17
   8.3 OPERATION: UPDATESERVICEUSERPROFILEDATA ........................................................................................ 21
   8.4 OPERATION: DELETESERVICEUSERPROFILEDATA ........................................................................................ 22
APPE NDI X A. CHANGE HISTORY (INFORMATIVE) ..................................................................................................... 24
   A.1 APPROVED VERSION HISTORY .................................................................................................................... 24
   A.2 DRAFT/CANDIDATE VERSION 1.0 HISTORY .................................................................................................. 24
APPE NDI X B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) ................................................................. 25
   B.1 SCR FOR SUPM.SOAP SERVER .................................................................................................................... 25
   B.2 SCR FOR SUPM.SOAP.ATTRIBUTESFORUSERPROFILE SERVER .................................................................. 25
   B.3 SCR FOR SUPM.SOAP.INDIVIDUALATTRIBUTESOFUSERPROFILE SERVER .................................................. 25
   B.4 SCR FOR SUPM.SOAP.DATAVIEWSOFUSERPROFILE SERVER ......................................................................... 26
APPE NDI X C. SUPM-1 SOAP SEQUENCE DIAGRAM (INFORMATIVE) ...................................................................... 27
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


[SUPM-AD] “OMA Service User Profile Management Architecture”, Open Mobile Alliance™, OMA-AD-Service_User_Profile_Management-V1_0, URL:http://www.openmobilealliance.org/


2.2 Informative References

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

<table>
<thead>
<tr>
<th>Term</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>See [OMADICT]</td>
</tr>
<tr>
<td>Authorized Principal</td>
<td>See [OMADICT]</td>
</tr>
<tr>
<td>Principal</td>
<td>See [OMADICT]</td>
</tr>
<tr>
<td>Resource</td>
<td>See [OMADICT]</td>
</tr>
<tr>
<td>Service</td>
<td>See [OMADICT]</td>
</tr>
<tr>
<td>Service Provider</td>
<td>See [OMADICT]</td>
</tr>
<tr>
<td>Service User Profile</td>
<td>See [SUPM-RD]</td>
</tr>
<tr>
<td>SUPM Data View</td>
<td>See [SUPM-AD]</td>
</tr>
<tr>
<td>User Profile</td>
<td>See [OMADICT]</td>
</tr>
<tr>
<td>User</td>
<td>See [OMADICT]</td>
</tr>
<tr>
<td>User’s Characteristic</td>
<td>See [SUPM-RD]</td>
</tr>
<tr>
<td>Description Information</td>
<td>See [SUPM-RD]</td>
</tr>
</tbody>
</table>

3.3 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>Architecture Document</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>OMA</td>
<td>Open Mobile Alliance</td>
</tr>
<tr>
<td>RD</td>
<td>Requirement Document</td>
</tr>
<tr>
<td>REST</td>
<td>REpresentational State Transfer</td>
</tr>
<tr>
<td>SUPM</td>
<td>Service User Profile Management</td>
</tr>
<tr>
<td>TS</td>
<td>Technical Specification</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Mark-up Language</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th></th>
<th>Specify service type.</th>
<th>Provide definition service schema including the methods as elements based on DST types.</th>
<th>Enumerate object types.</th>
<th>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., &lt;NewData&gt;, &lt;Data&gt;, and &lt;ItemData&gt;. The data description may make allowance for different object types.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N/A. Integration with a Liberty ID-WSF Discovery Service is out-of-scope of this specification.</td>
<td>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.</td>
<td>See §5.1.1.2 in this SUPM SOAP binding specification.</td>
<td>See §7 in this SUPM SOAP binding specification and provided XSD. &lt;ItemData&gt; is not defined as &lt;ResultQuery&gt; is not supported (see 12.b).</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>See §5.1.1.2 in this SUPM SOAP binding specification.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Describe SelectType and how it applies to various types of objects.</td>
<td>Describe TestOpType, considering how to test all object types supported by the specification</td>
<td>N/A as Testing feature is not supported (see 12.a).</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>See §5.1.1.2 in this SUPM SOAP binding specification. The SelectType has not been redefined in the XSD (string).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>N/A as Sorting feature is not supported (see 12.c).</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Enumerate Methods and state the required level of support. The default set of methods is &lt;Create&gt;, &lt;Query&gt;, &lt;Modify&gt;, and &lt;Delete&gt;. 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).</td>
<td></td>
<td>See §Error! Reference source not found. in this SUPM SOAP binding specification.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>The discovery option keywords (see [LibertyDisco]) can either be listed with semantics here, or via a reference to the correct chapter in the specification.</td>
<td></td>
<td>N/A. Integration with a Liberty ID-WSF Discovery Service is out-of-scope of this specification.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>SUPM data elements are uniquely identified through their names.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Extension support. State whether extension is supported and if so, describe this support. A</td>
<td>Extension feature SHALL NOT be supported.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Statement of optionality of query features (and their manifestation on discovery option keywords, see above):</td>
<td>Level of support of query features in this binding specification is the following:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 12 | 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` | 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. |
|   | Statement of optionality of create features (and their manifestation on discovery option keywords, see above): | Level of support of create features in this binding specification is the following: |
| 13 | a. Support multiple `<Create>` elements | a. Multiple `<Create>` elements SHALL NOT be supported. |
|   | Statement of optionality of delete features (and their manifestation on discovery option keywords, see above): | Level of support of delete features in this binding specification is the following: |
| 14 | a. Support multiple `<Delete>` elements | a. Multiple `<Delete>` elements SHALL NOT be supported. |
|   | Statement of optionality of modify features (and their manifestation on discovery option keywords, see above): | Level of support of modify features in this binding specification is the following: |
| 15 | 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?  
d. Support `notChangedSince` | 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. |

### 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>

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

6.1.1.2 Output message: <CreateResponse>

<table>
<thead>
<tr>
<th>Abstract part name</th>
<th>SOAP / ID-WSF part name</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultCode</td>
<td>&lt;Status code=&quot;…&quot;&gt;</td>
<td>No</td>
<td>Success indication or error code (with potential second level status code to refine the error code).</td>
</tr>
<tr>
<td>ErrorMessage</td>
<td>&lt;Status comment=&quot;…&quot;&gt;</td>
<td>Yes</td>
<td>Optional error diagnostic description.</td>
</tr>
</tbody>
</table>

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>`

<table>
<thead>
<tr>
<th>Abstract part name</th>
<th>SOAP / ID-WSF part name</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Originator</td>
<td><code>&lt;Sender&gt;</code> header</td>
<td>Yes</td>
<td>The data consumer who wants to read the Service User Profile of a specific user. This <code>&lt;sender&gt;</code> header defined in [LibertySOAPBinding] is optional as the data consumer may be identified implicitly based on mutual authentication mechanisms.</td>
</tr>
<tr>
<td>UserId</td>
<td><code>&lt;TargetIdentity&gt;</code> header</td>
<td>No</td>
<td>Identification of the user for whom the profile is requested</td>
</tr>
<tr>
<td>AttributeNames</td>
<td><code>&lt;Select&gt;</code> element inside the <code>&lt;QueryItem&gt;</code> element(s) or &quot;predefined&quot; XML attribute of <code>&lt;QueryItem&gt;</code> element(s)</td>
<td>No</td>
<td>A data consumer can request the whole profile, a view or a specific attribute per <code>&lt;QueryItem&gt;</code> 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 <code>&lt;QueryItem&gt;</code> elements inside the <code>&lt;Query&gt;</code> operation.</td>
</tr>
</tbody>
</table>

### 6.1.2.2 Output message: `<QueryResponse>`

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

### 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>`

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

### 6.1.3.2 Output message: `<ModifyResponse>`

<table>
<thead>
<tr>
<th>Abstract part name</th>
<th>SOAP / ID-WSF part name</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultCode</td>
<td>&lt;Status code=&quot;…&quot;/&gt;</td>
<td>No</td>
<td>Success indication or error code (with potential second level status code to refine the error code).</td>
</tr>
<tr>
<td>ErrorMessage</td>
<td>&lt;Status comment=&quot;…&quot;/&gt;</td>
<td>Yes</td>
<td>Optional error diagnostic description.</td>
</tr>
</tbody>
</table>

### 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>

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

6.1.4.2 Output message: <DeleteResponse>

<table>
<thead>
<tr>
<th>Abstract part name</th>
<th>SOAP / ID-WSF part name</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultCode</td>
<td>&lt;Status code=&quot;...&quot;&gt;</td>
<td>No</td>
<td>Success indication or error code (with potential second level status code to refine the error code).</td>
</tr>
<tr>
<td>ErrorMessage</td>
<td>&lt;Status comment=&quot;...&quot;&gt;</td>
<td>Yes</td>
<td>Optional error diagnostic description.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attribute</td>
<td>Attribute [0..unbounded]</td>
<td>Yes</td>
<td>Contains a list of attributes.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attributeName</td>
<td>xsd:string</td>
<td>No</td>
<td>Name of the attribute</td>
</tr>
<tr>
<td>attributeValue</td>
<td>xsd:anyType</td>
<td>Yes</td>
<td>The value(s) of the attribute. Complex types SHALL NOT be supported.</td>
</tr>
</tbody>
</table>
8. Examples

In the following examples:

- If the `<Query>` contains multiple `<QueryItem>` elements, an `<itemID>` XML attribute must be added 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
<?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">
<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">
 <Header>
  <sb:Sender providerID="http://supmenablerA.com"/>
  <sb:TargetIdentity>mailto: bob@example.com</sb:TargetIdentity>
 </Header>
 <Body>
  <supm:Query>
   <supm:CreateResponse timeStamp="2010-06-17T04:49:18Z">
    <lu:Status code="OK"/>
   </supm:CreateResponse>
  </supm:Query>
 </Body>
</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").
The following is an example of a Read response.

```xml
<?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
<?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
<?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:Envelope>
```
The following are 2 examples of a read request asking for a whole user profile:

- 1st example

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

- 2nd example

```xml
<?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>
```
8.3 Operation: UpdateServiceUserProfileData

The following is an example of an Update request.

```xml
<?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
<?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>
```

8.4 Operation: DeleteServiceUserProfileData

The following is an example of a Delete request.

```xml
<?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:ModifyResponse timeStamp="2010-06-17T04:49:18Z">
            <lu:Status code="OK"/>
        </supm:ModifyResponse>
    </S:Body>
</S:Envelope>
```
The following is an example of a Delete response.

```xml
<?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

### A.1 Approved Version History

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/a</td>
<td>n/a</td>
<td>No prior version – or – No previous version within OMA</td>
</tr>
</tbody>
</table>

### A.2 Draft/Candidate Version 1.0 History

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Sections</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft Versions</td>
<td>13 Apr 2010</td>
<td>all</td>
<td>Baseline for the document as agreed in: OMA-ARC-SUPM-2010-0070R02-INP_draft_version_TSs_for_REST_and_SOAP_uses_on_SUPM_1.zip</td>
</tr>
<tr>
<td></td>
<td>1 Sep 2010</td>
<td>All</td>
<td>Agreed CRs included:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• OMA-ARC-SUPM-2010-0144R01-CR_Intro_and_principles_for_SOAP_TS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• OMA-ARC-SUPM-2010-0145R01-CR_Operations_and_messages_for_SOAP_TS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• OMA-ARC-SUPM-2010-0146R01-CR_Data_structure_for_SOAP_TS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• OMA-ARC-SUPM-2010-0147R01-CR_Examples_for_SOAP_TS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• OMA-ARC-SUPM-2010-0148R01-CR_SCR_tables_for_SOAP_TS</td>
</tr>
<tr>
<td></td>
<td>16 Nov 2010</td>
<td>All</td>
<td>OMA-ARC-SUPM-2010-0162R03-CR_CONR_Resolution_G001toG005_G009_G012toG014_G016_G018_to_G019</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-SUPM-2010-0166R02-CR_CONR_Resolution_G006G008_G010toG011_G015_G017_G020</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-SUPM-2010-0171R02-CR_CONR_Resolution_G017</td>
</tr>
<tr>
<td></td>
<td>19 Nov 2010</td>
<td>App B</td>
<td>OMA-ARC-SUPM-2010-0171R02-CR_CONR_Resolution_G017</td>
</tr>
<tr>
<td></td>
<td>01 Dec 2010</td>
<td>Chapter 8 and others</td>
<td>OMA-ARC-SUPM-2010-0190R03-CR_SOAP_TS_Examples_issues_CR183</td>
</tr>
<tr>
<td></td>
<td>06 Dec 2010</td>
<td>All</td>
<td>Editorial fixes: cover page, history table and styles</td>
</tr>
<tr>
<td>Candidate Version</td>
<td>11 Jan 2011</td>
<td>All</td>
<td>Status changed to Candidate by TP: OMA-TP-2010-0530-INP_SUPM_V1_0_ERP_for_Candidate_Approval</td>
</tr>
</tbody>
</table>
Appendix B. Static Conformance Requirements  (Normative)

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

### B.1 SCR for SUPM.SOAP Server.

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUPM-SOAP-SUPPORT-S-001-O</td>
<td>Support for the SOAP binding of SUPM-1 interface</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>SUPM-SOAP-SUPPORT-S-002-M</td>
<td>Support for the XML request &amp; response format</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

### B.2 SCR for SUPM.SOAP.AttributesForUserProfile Server

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

### B.3 SCR for SUPM.SOAP.IndividualAttributesOfUserProfile Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUPM.SOAP-IND-ATTRIB-S-001-M</td>
<td>Support for management (create, update, retrieve) of individual attributes for a user profile</td>
<td>6.1</td>
<td>=</td>
</tr>
<tr>
<td>SUPM.SOAP-IND-ATTRIB-S-002-M</td>
<td>This operation returns the value of an user profile attribute - Query</td>
<td>6.1.2</td>
<td></td>
</tr>
<tr>
<td>SUPM.SOAP-IND-ATTRIB-S-003-M</td>
<td>This operation creates or updates an attribute of an user profile- Modify</td>
<td>6.1.3</td>
<td></td>
</tr>
</tbody>
</table>
### B.4 SCR for SUPM.SOAP.DataViewsOfUserProfile Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUPM.SOAP-DV-S-001-O</td>
<td>Support for operations (retrieve) for a DataView of an user profile</td>
<td>6.1</td>
<td>SUPM.SOAP-DV-S-002-O</td>
</tr>
<tr>
<td>SUPM.SOAP-DV-S-002-O</td>
<td>This operation returns the list of attributes and values of a DataView of an user profile - Query</td>
<td>6.1.2</td>
<td></td>
</tr>
</tbody>
</table>
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.