Liberty ID-WSF 2.0 Errata v1.0

Version: 1.0

Editors:
Darryl Champagne, IEEE-ISTO
Rob Lockhart, IEEE-ISTO
Eric Tiffany, IEEE-ISTO

Contributors:
Frederick Hirsch, Nokia
Paul Madsen, NTT
Greg Whitehead, Hewlett Packard
Conor Cahill, Intel Corporation
Carolina Canales-Valenzuela, Ericsson

Abstract:

This document contains errata items pertaining to the Liberty ID-WSF 2.0 specification set.

Filename: liberty-idwsf-2.0-errata-v1.0.pdf

Notice

This document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to use the document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative works of this Specification. Entities seeking permission to reproduce portions of this document for other uses must contact the Liberty Alliance to determine whether an appropriate license for such use is available.

Implementation of certain elements of this document may require licenses under third party intellectual property rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are not and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights. **This Specification is provided "AS IS", and no participant in the Liberty Alliance makes any warranty of any kind, express or implied, including any implied warranties of merchantability, non-infringement of third party intellectual property rights, and fitness for a particular purpose.** Implementers of this Specification are advised to review the Liberty Alliance Project's website (http://www.projectliberty.org/) for information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance Management Board.

Liberty Alliance Project
Licensing Administrator
c/o IEEE-ISTO
445 Hoes Lane
Piscataway, NJ 08855-1331, USA
info@projectliberty.org

Contents

1. Introduction

The ID-WSF v2.0 protocols, as initially specified, contained certain material errors, collectively referred to as *errata*. This document describes the errata in the Liberty ID-WSF v2.0 specification set, and the approved corrections. *This normative document, in combination with the original ID-WSF 2.0 specifications, supersedes the original specification set.* The specifications targeted by this errata document are listed in References, below.

Readers of the Liberty ID-WSF v2.0 specification set should note the errata in this document and incorporate it into their reading of the specifications. To assist in this process, "red-line" versions of the affected specifications are available. Note that these "red-line" documents are only informative; in all cases the normative corrections in this document take precedence.

Additionally, implementers of the affected specifications should use the Liberty schemata and wsdl associated with the specifications listed below in place of those affected by the specified errata.

- liberty-idwsf-authn-svc-2.0-errata-v1.0.pdf
- liberty-idwsf-client-profiles-2.0-errata-v1.0.pdf
- liberty-idwsf-disco-svc-2.0-errata-v1.0.pdf
- liberty-idwsf-interaction-svc-2.0-errata-v1.0.pdf
- liberty-idwsf-people-service-1.0-errata-v1.0.pdf
- liberty-idwsf-security-mechanisms-core-2.0-diff-v1.0.pdf
- liberty-idwsf-security-mechanisms-saml-profile-2.0-errata-v1.0.pdf
- liberty-idwsf-soap-binding-2.0-errata-v1.0.pdf
- liberty-idwsf-2.0-scr-1.0-diff-v1.0.pdf

1.1. Template Errata Item

1.1.1. Summary

Each errata entry should have a Summary and a Resolution section. The Summary should contain a description of the issue and indicate which specifications are involved.

1.1.2. Resolution

The Resolution section should contain specific descriptions of the revisions to the text. These descriptions should include the section and line number(s) to be revised, and the revised text.

1. Multiple revisions may be indicated in an ordered list.
2. Where it improves clarity, an unordered list indicating the before and after state may be used:
    - From: The original text.
    - To: The changed text.

2. Errata

2.1. People Service Status element formatting

[LibertyPeopleService]

2.1.1. Summary

Language and formatting around the the Status element was inconsistent.

2.1.2. Resolution

Used consistent formatting and language for `<lu:Status>` at lines 1150, 1226, 1463, 1467, 1550, 1621, 1626, 1755, 1844, 1961 and 1967.

2.2. People Service Occurence of AddKnownEntity message

[LibertyPeopleService]

2.2.1. Summary

In one location, the People Service refers to an "AddKnownEntity" message incorrectly. It should be "AddEntityRequest."

2.2.2. Resolution

Changed Line 710 to "AddEntityRequest."

2.3. References to InvocationId

[LibertySecMech20]

2.3.1. Summary

Vestigial reference to the InvocationIdentity header, along with an unneeded key usage URI, in [LibertySecMech20].

2.3.2. Resolution

Proposed resolution: Frederick's revised text for section 6.3.1:

1. The text at line 469-479 (section 6.3.1) should now read:

   A token container type is defined to provide a uniform means to convey tokens, and allows a Web Services Security token to be directly contained in the container, or to be referenced from the container. A reference may be an external reference to a token or a reference to another local token container.

   The token container type (`TokenType`) may be used to define elements in the ID-WSF namespace, and has also been used to define a `<Token>` element in the security mechanisms namespace. This `<sec:Token>` element may be used in a number of ID-WSF 2.0 schema definitions, such as:

   - The security context container type used in the Discovery Service to profile EPRs,
   - The mapping input and output types for the Identity Mapping Service, and
   - The `AddKnownEntityRequestType` for the People Service.

2. The text at line 501 is changed to:

   This specification defines the following URN values for the `usage` attribute (others may be defined elsewhere):

- `urn:liberty:security:tokenusage:2006-08:TargetIdentity`
- `urn:liberty:security:tokenusage:2006-08:SecurityToken`

These two URNs are used when the token is contained in an EPR to be used to create a SOAP header by the Discovery Service. The TargetIdentity usage indicates that the token should be used to create a `<sb:TargetIdentity>` header block. Any token with the SecurityToken usage in an EPR is placed in a `<wsse:Security>` header block.

2.4. Identity Mapping Issues

[LibertySecMech20], [LibertyAuthn]

2.4.1. Summary

Several problems identified in Identity Mapping Service (IMS) in [LibertyAuthn] and [LibertySecMech20]

1. IMS
   - Section 7.4.2.1 MappingOutput
     - It is not clearly stated whether or not MappingOutputs are included when that particular mapping was unsuccessful (especially in the case of a multiple MappingInput request.
   - Section 7.7 - Example
     - The response contains just an encrypted ID while we typically use SAML assertions for the identity token (and the encrypted ID would be in the Subject of such an assertion). We do talk about using an assertion for the identity token in section 7.5
     - I would suggest a more complex example showing multiple (even just 2) inputs using the same token via a reference with different destination providers.
2. Sec Mech
   - Section 6.6.2 Token policy - this seems under specified... examples include:
     - issueTo - no semantics about what this means... Is this just what should be in the field, or does this mean that the issuer should figure out how to send the issued token to that party?
     - type - since it's a URI we should define the URI that should be used for SAML tokens. "By Default" should be removed and instead use something along the lines "If not specified,..."
     - wantDSEPR - should this be binary and only WSF2.0 EPR? or should this be more "wantDSInfo" and have values to indicate the particular DS info you want (2.0 DS EPR, 3.0 DS EPR, 1.0 DS RO). Same comment re: "default" wording.

2.4.2. Resolution

Proposed resolution adjusts text in several places.

1. Alter text at line 911-923 (section 6.6.1) of [LibertySecMech20] to read as follows:

   Different mechanisms may be used to convey an identity including the following:

   - A SAML 2.0 assertion element (`saml2:Assertion`) as profiled in the Security Mechanisms SAML profile [LibertySecMech20SAML]. This is a `saml2:Assertion`, and not a `saml2:EncryptedAssertion`, `saml2:NameID`, or `saml2:EncryptedID`.

- An opaque value, for example a `saml2:EncryptedAssertion`, `saml2:NameID`, or `saml2:EncryptedID`, WSS Binary Security Token, or non-SAML values.

Any identity token SHOULD be able to convey information needed for discovery. This is typically an endpoint reference.

An identity token must have an attribute of type IDType that may be used as a target of a ds:Reference, e.g., an xml:id or wsu:Id attribute.

Normative details using SAML 2 assertions are given in the Security Mechanisms SAML profile [Liberty-SecMech20SAML].

A WSS SecurityTokenReference element may also be used to reference an identity token.

2. The description of the `<TokenPolicy>` at line 929-946 in [LibertySecMech20] is changed as follows:

- **`validUntil`** [Optional]

  Indicates the duration for which the requestor would like the token to be valid. The responder MAY disregard the value in favor of its own policies.

- **`issueTo`** [Optional]

  Identifies the party to whom the identity token should be issued, if not otherwise apparent from the request or policy content. Note that this is usually *not* the party requesting the token, but generally a WSP the requester wishes to access.

  For example, a `samlp:NameIDPolicy` element may be included in the `TokenPolicy` element, and in some cases the value of the associated `SPNameQualifier` attribute will already indicate the party to whom the token is being issued, making use of `issueTo` unnecessary.

- **`type`** [Optional]

  Specifies the type of identity token to be returned upon an identity token request. If no type is specified then the type of token returned is Opaque and need not necessarily be understood by the requestor.

  The value of the type attribute is a URI. The following are defined in this document:

  - SecMech-SAML-2.0-Assertion:
    - This MUST be a SAML 2.0 assertion (`saml2:Assertion`) as profiled in the Security Mechanisms SAML Profile. This is a `saml2:Assertion`, and not a `saml:EncryptedAssertion`, `saml:NameID`, or `saml:EncryptedID`, which are all considered Opaque types.
    - A `samlp2:NameIDPolicy` element SHOULD be included in the TokenPolicy element.
    - URI value: *urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion*
  - Opaque:
    - The format is not specified and may be any format chosen by the IdP including, but not limited to, SAML assertions, Encrypted Assertions, NameIDs, Encrypted NameIDs, WSS Binary Security Tokens or other forms.
    - URI value: *urn:liberty:security:2006-08::IdentityTokenType:Opaque*

- **`wantDSEPR`** [Optional]

  Specifies whether the requestor would like the token to include a WSF 2.0 Endpoint Reference for the Discovery Service in a token returned by that Discovery Service. The default value is 'true'.

- **`Any Attribute`** [Zero or More]

  Any attribute can be used to describe other characteristics of the desired identity token. The wildcard is necessary because of the arbitrary nature of identity tokens.

- **Any Element** [Zero or More]

  Any element can be used to describe other characteristics of the desired identity token. The wildcard is necessary because of the arbitrary nature of identity tokens.

3. The text at line 1263-1265 (section 7.4.2.2) of [LibertyAuthn] is changed as follows:

An `<IdentityMappingResponse>` consists of a status element and zero or more `<MappingOutput>` elements, one for each successfully processed token request. Unsuccessfully processed `<MappingInput>` elements do not result in a corresponding `<MappingOutput>` element. If multiple `<MappingInput>` elements were included in a request, then each output element MUST contain a `reqRef` attribute matching it to the corresponding input element.

4. The example at line 1330 (section 7.7) of [LibertyAuthn] is changed to the following:

The following example shows a request for a SAML identity token. The policy and input token indicate a request to map from an identifier scoped to one SP into an identifier scoped to another. In this case, the input token is a bare identifier (probably extracted from another SAML token).

```
<sa:IdentityMappingRequest>
   <sa:MappingInput>
      <sec:TokenPolicy type="urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion">
         <samlp2:NameIDPolicy Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
            SPNameQualifier="https://spb.example.com"/>
      </sec:TokenPolicy>
      <sec:Token>
         <saml2:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
            NameQualifier="https://idp.example.com" SPNameQualifier="https://spa.example.com">
            DBC63923-C718-4249-83CE-1E53D80D8A4A
         </saml2:NameID>
      </sec:Token>
   </sa:MappingInput>
</sa:IdentityMappingRequest>
```

The following is a possible response to the request above. The returned token is a signed SAML assertion with an encrypted name identifier. The requester can establish the expiration from the response, giving it guidance as to when the token might need renewal.

```
<sa:IdentityMappingResponse>
   <sa:Status code="OK"/>
   <sa:MappingOutput>
      <sec:Token>
         <saml2:Assertion Version="2.0" IssueInstant="2006-03-19T07:35:00Z"
            ID="e9ab6ff0-4ee0-4ce2-868f-18873bdc87de">
            <saml2:Issuer>https://idp.example.com</saml2:Issuer>
            <ds:Signature>...</ds:Signature>
            <saml2:Subject>
               <saml2:EncryptedID>
                  <xenc:EncryptedData>U2XTCNvRX7Bl1NK182nmY00TEk==</xenc:EncryptedData>
               </saml2:EncryptedID>
            </saml2:Subject>
            <saml2:Conditions NotOnOrAfter="2006-03-19T08:35:00Z">
               <saml2:AudienceRestriction>
                  <saml2:Audience>https://spb.example.com</saml2:Audience>
               </saml2:AudienceRestriction>
            </saml2:Conditions>
         </saml2:Assertion>
      </sec:Token>
   </sa:MappingOutput>
</sa:IdentityMappingResponse>
```

Example 1.

2.5. Namespace prefix definition missing

[LibertySecMech20]

2.5.1. Summary

The `saml2p` namespace prefix definition was omitted from Table 1 in [LibertySecMech20].

2.5.2. Resolution

The following row is added to Table 1 which starts at line 165 of [LibertySecMech20]:

samlp2:	`urn:oasis:names:tc:SAML:2.0:protocol` The prefix `samlp2:` stands for the SAML v2 protocol namespace. It is defined in [SAMLCore2].

2.6. Format of Identity Token ("Type")

[LibertySecMech20]

2.6.1. Summary

The `type` attribute of the `<TokenPolicyType>` element in [LibertySecMech20] refers to the type of token being requested. The text is unclear, and needs to explicitly state that there are two possible types of requested tokens (mapped as two possible URIs to be conveyed inside the `type` attribute): SAML assertion or other (opaque). The default value refers to opaque tokens

2.6.2. Resolution

Included in resolution in 2.4.2.

2.7. Disco Should Refer to <ds:SvcMDRegisterResponse> Instead of <ds:ModifyResponse>

[LibertyDisco]

2.7.1. Summary

Update Section 3.12 to Refer to `<ds:SvcMDRegisterResponse>` instead of `<ds:ModifyResponse>`..

2.7.2. Resolution

Proposed resolution entails the following changes to [LibertyDisco]:

1. Change occurrences of `ModifyResponse` in lines 2135, 2144, 2150, 2164 and 4140 to `SvcMDRegisterResponse`

2. Change the paragraph at lines 2146-2148 to read:

   The Discovery Service instance SHOULD ONLY include the `<Keys>` element in `<SvcMDRegisterResponse>` messages if has no `<ProviderID>` and the `<SvcMDRegister>` message included Service Metadata that relies upon signed security tokens for one or more of its security mechanisms.

2.8. Client Profiles: References to ServiceInstanceEPR

[LibertyClientProfiles]

2.8.1. Summary

The Client Profiles spec refers to ServiceInstanceEPR, a deprecated construct

2.8.2. Resolution

Replaced occurrences of disco:ServiceInstanceEPR at Lines 78 and 84 with wsa:EndpointReference

2.9. Client Profiles refers to 'providerID'

[LibertyClientProfiles]

2.9.1. Summary

The Client Profiles spec refers to 'providerID', the case of the initial letter should be upper.

2.9.2. Resolution

Replaced multiple occurrences 'providerID' with 'ProviderID'.

2.10. SCR ID-WSF2.0, Line 198

[LibertyIDWSF20SCR]

2.10.1. Summary

In the list of security mechanisms that SP WSCs are required to support, at line 198 urn:liberty:security:2006-08:ClientTLS:SAMLV2 was meant to be urn:liberty:security:2006-08:ClientTLS:peerSAMLV2 to match the requirement on SP WSPs at line 179.

2.10.2. Resolution

Update SCR ID-WSF2.0, line 198 to be urn:liberty:security:2006-08:ClientTLS:peerSAMLV2

2.11. Value of soapAction (HTTP Header) attributes in the WSDL

[LibertyAuthn], [LibertyPeopleService], [LibertyInteract], [LibertyDisco]

2.11.1. Summary

The value of the SOAPAction (HTTP Header) attributes in the WSDL may not match the wsaw:Action attributes.

2.11.2. Resolution

Delete the soapAction attribute lines from the WSDL and main body text in the following normative specifications and the associated informative WSDL files:

1. [LibertyAuthn]: lines 2150, 2220, and 2289
   - liberty-idwsf-authn-svc-v2.0.wsdl: line 53
   - liberty-idwsf-sso-svc-v2.0.wsdl: line 54
   - liberty-idwsf-idmapping-svc-v2.0.wsdl: line 53
2. [LibertyDisco]: lines 3869, 4595, 4601-4602, 4609-4610, 4616-4617, 4623, 4629, 4635, and 4641
   - liberty-idwsf-disco-svc-v2.0.wsdl: lines 165, 171-172, 179-180, 186-187, 193, 199, 205, and 211
3. [LibertyInteract]: line 777
   - liberty-idwsf-interaction-svc-v2.0.wsdl: line 64
4. [LibertyPeopleService]: lines 3231, 3237, 3242, 3247, 3252, 3257, 3262, 3267, 3272, 3277, 3282, 3287, and 3292
   - liberty-idwsf-people-service-v1.0.wsdl: lines 275, 281, 286, 291, 296, 301, 306, 311, 316, 321, 326, 331, and 336

2.12. Update Text about ReplyTo

[LibertySOAPBinding]

2.12.1. Summary

Line 590 states that "If this header block is not present, then no reply will be sent." This is wrong according to WS-A and should be deleted.

The sender processing rules, at line 750, say that the ReplyTo header MUST be included. This is not needed according to WS-A and should also be deleted. Note, the sender processing rules properly handle the case when the ReplyTo header is not present.

2.12.2. Resolution

1. Removed ReplyTo example at line 465:

```
<wsa:ReplyTo>
   <wsa:Address>...</wsa:Address>
</wsa:ReplyTo>
```

2. Removed ReplyTo reference from example at line 507:

```
<!-- reference params from FaultTo/ReplyTo EndpointReference -->
```

3. Updated text at line 590 to read:

"If the <wsa:ReplyTo> header block is not present, the value defaults to http://www.w3.org/2005/03/addressing/role/anonymous; so, when constructing a message, the header block can be omitted if this is the value that would be used. This typically allows the <wsa:ReplyTo> header block to be omitted during synchronous request-response message exchanges over HTTP. Please refer to [WSAv1.0] for default processing rules in the absence of the <wsa:ReplyTo> header block."

4. Updated text at line 596:

- From: "If not present, faults are sent to the address specified in the <wsa:ReplyTo> header block (if present)."
- To: "If not present, faults are sent to the reply address."

5. Updated text at line 750:

- From: "sending a request message, the outgoing message MUST include exactly one <wsa:ReplyTo> header block and at most one <wsa:FaultTo> header block (if the <wsa:FaultTo> header block is not included, faults will be delivered to the <wsa:ReplyTo> endpoint)."
- To: "sending a request message, the outgoing message MUST include at most one <wsa:ReplyTo> header block and at most one <wsa:FaultTo> header block (if the <wsa:FaultTo> header block is not included, faults will be delivered to the reply endpoint)."

6. Removed ReplyTo example at 927:

```
<wsa:ReplyTo>
   <wsa:Address>http://www.w3.org/2005/03/addressing/role/anonymous</wsa:Address>

</wsa:ReplyTo>
```

2.13. Update Note about wsa:To header block usage

[LibertySOAPBinding]

2.13.1. Summary

Note regarding absence of <wsa:To> header block is slightly misleading. This is a non-normative note which is misaligned with [WSAv1.0], and could therefore lead to confusion.

2.13.2. Resolution

1. Replace the sentence beginning at line 578 with 'This..." and continuing on 579, with

"This typically allows the <wsa:To> header block to be omitted during synchronous request-response message exchanges over HTTP. Please refer to [WSAv1.0] for default processing rules in the absence of the <wsa:To> header block."

2.14. wsa:Action for Faults, Header faults

[LibertySOAPBinding]

2.14.1. Summary

Treatment of wsa:Action and wsa:Fault header for Fault messages in [LibertySOAPBinding] seems to contradict the behavior defined in WS-Addressing [WSAv1.0].

2.14.2. Resolution

In [LibertySOAPBinding] after line 422 add the following:

Note

When reporting SOAP processing errors, the WS-Addressing action http://www.w3.org/2005/08/addressing/soap/fault SHOULD be used. When reporting WS-Addressing processing errors, the WS-Addressing action http://www.w3.org/2005/08/addressing/fault SHOULD be used. When reporting other processing errors, if no specific WS-Addressing action is defined, then http://www.w3.org/2005/08/addressing/soap/fault SHOULD be used.

2.15. People Service vague on processing of circularity in collections

[LibertyPeopleService]

2.15.1. Summary

There is a need for clarification of the PS provider's responsibilities for dealing with circularity in group structure in the [LibertyPeopleService] specification.

2.15.2. Resolution

Added after Line 1151

- MUST NOT allow the creation of circular collections. A circular collection is one which includes at any layer of the structure below it a reference to the same collection such that a dereference of the collection would result in an infinite loop. If the PS provider receives an `<AddToCollectionRequest>` message that would result in a circular collection, it MUST respond with *Failed* as the code attribute of the top level `<lu:Status>` element, and the code attribute of the second level `<lu:Status>` element MAY be set with the following status code:
    - CircularCollection

2.16. People Service vague on Count and Offset Processing

[LibertyPeopleService]

2.16.1. Summary

There is a need for clarification of the PS provider's responsibilities for processing the offset and count attributes if specified on a request message.

2.16.2. Resolution

Added before Line 1460.

- If the Count attribute is included in a `<ListMembersRequest>` message, the PS provider SHOULD NOT respond with more objects than specified. A PS provider MAY return a smaller number of objects than specified by the Count attribute.

- If the Offset attribute is included in a `<ListMembersRequest>` message and the PS provider returns any objects, it MUST respond with a list of `<Object>` elements that starts with the `<Object>` element whose position in the complete list is specified by the value of the Offset attribute.

2.17. People Service vague of Count and Offset attributes on subscription processing

[LibertyPeopleService]

2.17.1. Summary

There is a need for clarification of the PS provider's responsibilities for processing messages that had both a Subscription element and Count/Offset attributes.

2.17.2. Resolution

Added after Line 402

- SubscribeToChildrenOnly

Added following before Line 1449

- The WSC SHOULD NOT include a `<Subscription>` element in a `<ListMembersRequest>` message if the offset attribute has any value other than '0'.

- The PS provider MUST, if the `<ListMembersRequest>` contains a Subscription and the Offset attribute has any value other than '0', and it is otherwise capable of returning results, return those results as indicated by the Count and Offset attributes, but still reject the Subscription by responding with "OKButNoSubscription" as the code attribute of the top level `<lu:Status>` element. In this case the code attribute of the second level `<lu:Status>` element MAY be set with the following status code:

  - NoSubscribeWithOffset

2.18. People Service Occurrence Rule too Restrictive for <Tag> element

[LibertyPeopleService]

2.18.1. Summary

The [LibertyPeopleService] schema is overly restrictive in specifying that there be, at most, a single `<Tag>` element for a given `<Object>` element.

2.18.2. Resolution

1. Change line 242 in [LibertyPeopleService] from:

   ```
   <xs:element name="Tag" type="TagType" minOccurs="0"/>
   ```

   to read:

   ```
   <xs:element name="Tag" type="TagType" minOccurs="0" maxOccurs="unbounded"/>
   ```

2. Corresponding change made in People Service schema.

2.19. People Service vague on processing of Tag elements

[LibertyPeopleService]

2.19.1. Summary

There is a need for clarification of the PS provider's responsibilities for processing the `<Tag>` element.

2.19.2. Resolution

1. Add after Line 306:

   If they understand the tag space for a `<Tag>` element, WSCs and PS providers MAY process as appropriate. WSCs and PS providers MAY ignore `<Tag>` elements.

2.20. People Service Vague on ObjectID Management

[LibertyPeopleService]

2.20.1. Summary

There is a need for clarification of the WSC and PS provider's responsibilities for dealing with to the creation of object identifiers.

2.20.2. Resolution

Added the following to Section 2.1.4.

- The PS provider controls the creation of object identifiers. When a WSC requests the creation of an object, the WSC MUST NOT provide an ObjectID in such a request message. If the request is successful, the PS provider MUST return an object identifier for the new object in an `ObjectID` element in its response message - the WSC MUST use this returned identifier in subsequent operations on that object.

Added the following after line 671.

- MUST include an `ObjectID` in the returned `Object`.

2.21. BugZ 816: Remove Two References to ProxyTransitedPath

[LibertySecMech20]

2.21.1. Summary

Remove two references to ProxyTransitedPath in [LibertySecMech20]:

- [LibertySecMech20] lines 1041 and 1047 - ProxyTransitedPath
- Note: The reference on [LibertySecMech20] line 1063 is fine.

Clarify provider chaining text to indicate that both proxying and generation of new requests is supported.

2.21.2. Resolution

1. Changed first paragraph of Section 7.3 (Provider Chaining) in [LibertySecMech20] (starts at line 998)

   from: "Provider chaining refers to scenarios in which a service provider (WSP), upon receiving a request from a sender, itself passes the request onto another service provider until the destination service provider is reached. This mechanism allows proxying to be performed, where each provider proxies the request to the next party."

   to: "Provider chaining refers to scenarios in which a service provider (WSP), upon receiving a request from a sender, sends a request to the next service provider. This may be done by forwarding the request it received, acting as a proxy, or by generating a new request. This may be done until the destination service provider is reached."

2. Changed at line 1041 of [LibertySecMech20]

   from: "The DS may have included <ProxyTransitedPath> in this token contained in the bootstrap EPR,"

   to: "The DS may have included the <TransitedProviderPath> element in the security token contained in the bootstrap EPR,"

3. Changed "ProxyTransitedPath" to "TransitedProviderPath" at line 1047 of [LibertySecMech20].

2.22. People Service Uniqueness Requirements of ObjectID element [LibertyPeopleService]

2.22.1. Summary

The People Service specification is unnecessarily constraining on the uniqueness requirements of the ObjectID.

2.22.2. Resolution

Changed Line 229 to:

"The value of the `<ObjectID>` element uniquely identifies the `<Object>` within the set of all `<Object>` elements that are accessible to a particular consumer of the People Service for the targeted identity."

Changed Line 267 to:

"The `<ObjectID>` element is defined so that WSCs can unambiguously refer to the parent `<Object>` elements."

Changed Line 279 to:

"Unique identifiers for different WSCs (e.g., pairwise identifiers), reuse of identifiers across different services, and encrypted identifiers are potential mechanisms for addressing this concern."

2.23. People Service Vague on Count and Structure Attribute [LibertyPeopleService]

2.23.1. Summary

There is a need for clarification of the PS provider's responsibilities for processing the Count and Structured attributes on the ListMembersRequest message.

2.23.2. Resolution

Changed Line 1455 to:

- If the `Structured` attribute is set as *entities* the PS provider MUST return all the direct child and descendant entity `<Object>` elements of the specified `Object` (subject to the restriction defined by the `Count` attribute if present). Any collection `<Object>` elements MUST be removed and only entity `<Object>` elements returned.

2.24. People Service Vague on Token Notification processing

[LibertyPeopleService]

2.24.1. Summary

There is a need for clarification of the PS provider's processing for a Notification message by which a Token is returned in response to AddEntityRequest message.

2.24.2. Resolution

1. Added after Line 471

   There MUST be no `<Aggregation>` element present in a subscription.

2. Added after Line 473

   Such a `<Subscription>` MUST be considered to have expired at such time as the PS provider has delivered a `<Token>` for the invited user to the WSC and the WSC has acknowledged its receipt.

3. Added after Line 469

   For a `<Subscription>` sent within an `<AddEntityRequest, >` message, the `starts` attribute SHOULD be omitted. If present, the PS provider MAY ignore the attribute.

4. Added after Line 482

   For a `<Subscription>` sent within an `<AddEntityRequest, >` message, the `includeData` attribute SHOULD be omitted.

2.25. People Service vague on relationship between Subscription support and SCR

[LibertyPeopleService]

2.25.1. Summary

There is a need for clarification of the relationship between Subscription Support and Static Conformance Requirements.

2.25.2. Resolution

1. Added after Line 1976 in People Service:

   Notwithstanding this, supporting the invitation model is optional from a conformance point of view (i.e., as defined by the [LibertyIDWSF20SCR]) and so any normative requirements expressed within this specification should be understood in this context.

2. Changed Lines 218, 253, and 328 in SCR:

   ... support management of groups & users, as described in sections 3.9-3.20 of [LibertyPeopleService].

2.26. People Service vague on CreatePSObject in AddEntityRequest

[LibertyPeopleService]

2.26.1. Summary

There is a need for clarification of PS provider's responsibilities for dealing with the CreatePSObject in an AddEntityRequest.

2.26.2. Resolution

Changed list items at lines 682 and 853

SHOULD, if the `<AddEntityRequest>` message contained a `<CreatePSObject>` element, attempt to create or verify the existence of an object for the inviting user in the PS of the invited user (when made possible by a federated identifier being established for the invited user).

It may be the case that the inviting user is already in the PS of the invited user as a result of a prior invitation sequence initiated 'from the other side'. The PS of the inviting user MUST ensure that no duplicate object be added.

MAY, in order to determine whether an object for the inviting user already exists, query the members of the PS of the invited user using the `<ListMembersRequest>` message and ask the invited user to assist in determining whether the inviting user is already in the list. Other mechanisms (e.g., using a `<TestMembershipRequest>`) for making this determination MAY alternatively be used.

SHOULD, if there is no existing object for the inviting user, request that an object be created with either the `<AddEntityRequest>` or `<AddKnownEntityRequest>` messages.

SHOULD, if sending a `<AddKnownEntityRequest>` message for the addition, include a `<sec:Token>` element carrying a token for the inviting user - this `<sec:Token>` obtained from the Identity Mapping Service of the inviting user.

2.27. People Service vague AddEntityRequest Processing Rules

[LibertyPeopleService]

2.27.1. Summary

There is a need for clarification of PS provider's responsibilities for creating objects upon receiving an AddEntityRequest.

2.27.2. Resolution

Added at line 540:

"The `Object` element created by an `<AddEntityRequest>` message becomes a direct child of the root node."

Changed Line 671 to:

In responding to a successful *role="sgmltag"><AddEntityRequest>* message, the PS provider:

Added at Line 672:

- MUST create a new `Object` element as a direct child of the root node.

Changed Line 1284 to:

the WSC is indicating that it desires only the direct child collection and entity objects of the targeted object.

2.28. Adapt examples to TokenPolicy changes

[LibertyPeopleService]

2.28.1. Summary

Examples of IdentityMappingRequest messages in the People Service have outdated TokenPolicy structure

2.28.2. Resolution

1. Changed XML in Lines 2249-2259 to

```
<soap:Envelope>
<soap:Header>
 <ws:Security>
  <saml:Assertion ID="assertionid">
   credentials for Bob at IDPb
  </saml:Assertion>
 </ws:Security>
</soap:Header>
<soap:Body>
<ims:IdentityMappingRequest>
 <ims:MappingInput>
  <sec:TokenPolicy type="urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion">
   <samlp:NameIDPolicy SPNameQualifier="https://psa.com"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">
  </sec:TokenPolicy>
  <sec:Token ref="#assertionid"/>
 </ims:MappingInput>
</ims:IdentityMappingRequest>
</soap:Body>
</soap:Envelope>
```

2. Changed Line 1908 to

```
<sec:TokenPolicy type="urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion">
```

3. Changed Line 1909 to

```
<samlp2:NameIDPolicy Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
SPNameQualifier="https://psa.com"/>
```

4. Changed Line 2251 to

```
<sec:TokenPolicy type="urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion">
<samlp2:NameIDPolicy Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
SPNameQualifier="https://psa.com"/> </sec:TokenPolicy>
```

5. Changed Lines 2342 and 2352 to

```
<sec:TokenPolicy type="urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion">
```

2.29. Namespace on XPath expression in Filter element

[LibertyPeopleService]

2.29.1. Summary

In the People Service, the Filter element should carry an XPath expression specifying the matching criteria on a QueryObjectsRequest. The examples of such an expression are not namespace qualified.

2.29.2. Resolution

Added namespace qualifiers to Xpath expressions on Lines 1678, 1683, and 1750.

2.30. People Service vague on XPath Filter application

[LibertyPeopleService]

2.30.1. Summary

The People Service does not clearly define the document against which a Filter's XPath expression should be matched.

2.30.2. Resolution

Added after Line 1675

The XPath expression MUST be evaluated against the set of Objects that would be returned to a hypothetical ListMembersRequest that had targeted the root node and in which the Structured attribute was set to "tree"

2.31. People Service vague on processing of Subscriptions for QueryObjectsRequest

[LibertyPeopleService]

2.31.1. Summary

There is a need for clarification of PS provider's responsibilities for processing of subscription requests.

2.31.2. Resolution

Added after Line 1756

- A PS provider MAY choose to reject Subscriptions within certain `<QueryObjectsRequest>` messages (as might be desirable should the computational expense of determining when to send Notifications be prohibitive).

  The PS provider MAY, if the `<QueryObjectsRequest>` contained a Subscription it does not wish to accept, and it is otherwise capable of returning results, return those results but still reject the Subscription by responding with "OKButNoSubscription" as the code attribute of the top level `<lu:Status>` element. In this case, the code attribute of the second level `<lu:Status>` element MAY be set with the following status code:

  - PolicyDoesNotAllow

  An "OKButNoSubscription" Status code SHOULD NOT be interpreted as reflecting a PS provider's overall ability to support subscriptions.

2.32. People Service vague on processing of Subscription for a ListMembersRequest

[LibertyPeopleService]

2.32.1. Summary

There is a need for clarification of the set of objects against which the implicit trigger of 'changed' should be assessed.

2.32.2. Resolution

Added after Line 1258

For a subscription within a `<ListMembersRequest>` message, the PS provider MUST assess 'changes' against whatever was returned in the original `<ListMembersResponse>`. A `<Notify>` MUST be sent if, were the WSC to resend the same request, the results would be different than originally sent.

2.33. People Service vague on communication of known identifier

[LibertyPeopleService]

2.33.1. Summary

There is a need for clarification on how a user-supplied known identifier for an invited user should be communicated to the PS Provider.

2.33.2. Resolution

Changed Line 829 to

When the token is not a identity token (as is the likely case when the known identifier is provided by the inviting user), the WSC SHOULD use a SAML `<saml:NameID>` element within the `<Token>` element. If the WSC knows the format of the known identifier, it SHOULD use the appropriate value for the Format attribute on the `<saml:NameID>` element.

2.34. People Service vague on Subscription processing for ListMembersRequest with structured attribute not equal to 'children'.

[LibertyPeopleService]

2.34.1. Summary

There is a need for clarification of PS provider's responsibilities for processing a ListMembersRequest on which the structured attribute had a value other than 'children' was insufficiently defined.

2.34.2. Resolution

Added after Line 405

- SubscribeToChildrenOnly

Added after Line 1471

- The PS provider SHOULD, if the `<ListMembersRequest>` contains a Subscription and the Structured attribute has any value other than 'children', and it is otherwise capable of returning results, return those results but still reject the Subscription by responding with "OKButNoSubscription" as the code attribute of the top level `<lu:Status>` element. In this case the code attribute of the second level `<lu:Status>` element MAY be set with the following status code:

  - SubscribeToChildrenOnly

  An "OKButNoSubscription" Status code SHOULD NOT be interpreted as reflecting a PS provider's overall ability to support subscriptions, rather simply its unwillingness to accept the particular subscription requested.

2.35. TestMembershipRequest requires TargetObjectID

[LibertyPeopleService]

2.35.1. Summary

The schema for the TargetObjectID of a TestMembershipRequest makes its occurrence optional but the message requires it.

2.35.2. Resolution

Changed Line 1776 to:

`<TargetObjectID>` **[Optional]** The `<TargetObjectID>` element is used to convey the `ObjectID` of the target group `Object` for which the membership of a user is being tested.

Absence of the `<TargetObjectID>` indicates a request to test if the identity associated with the submitted token is associated to an Object (of type entity) in the PS.

2.36. People Service vague on processing of unknown Objects for ResolveIdentifierRequest

[LibertyPeopleService]

2.36.1. Summary

There is a need for clarification of PS provider's responsibilities for processing a ResolveIdentifierRequest message that includes ObjectIDs that are "unknown" or refer to collections.

2.36.2. Resolution

Added before Line 1954

- The PS provider MUST, if unable to find one or more (but not all) specified input objects, respond PartialSuccess as the code attribute of the top level `<lu:Status>` element, and SHOULD use second level `<lu:Status>` elements that containing a ref attribute equal to the associated `<ResolveInput>`'s reqID attribute. If unable to find any input objects, the PS provider MUST respond Failed as the code attribute of the top level `<lu:Status>` element. The second level `<lu:Status>` elements corresponding to such failed inputs MUST be set with the following status code:

  - CannotFindObject

Added after Line 1969

- The PS provider MUST, if one or more (but not all) specified input objects is of type "collection" , the PS provider MUST respond PartialSuccess as the code attribute of the top level `<lu:Status>` element, and SHOULD use second level `<lu:Status>` elements that containing a ref attribute equal to the associated `<ResolveInput>`'s reqID attribute. If all input objects are of type "collection," the PS provider MUST respond Failed as the code attribute of the top level `<lu:Status>` element. The second level `<lu:Status>` elements corresponding to such failed inputs MUST be set with the following status code:

  - ObjectIsCollection

2.37. ResolveIdentifierRequest with no TargetOBjectID

[LibertyPeopleService]

2.37.1. Summary

There is a need for clarification of PS provider's responsibilities for processing a ResolveIdentifierRequest message that does not include a TargetObjectID.

2.37.2. Resolution

Changed Line 1232 to:

If a WSC does not specify a TargetObjectID element in the ListMembersRequest message, the default targeted Object is the root node.

2.38. Optionality of Locale attribute on DisplayName

[LibertyPeopleService]

2.38.1. Summary

There is a need for clarification on the use of the 'IsDefault' attribute on multiple DisplayName elements.

2.38.2. Resolution

Changed Line 290 to

```
<xs:attribute name="Locale" type="xs:language" use="required"/>
```

Added after Line 295

The `<Locale>` attribute specifies the language in which the display name is expressed. If not present, providers SHOULD determine how to best display the name through other means.

The `<IsDefault>` attribute identifies which `<DisplayName>` element, if there are multiple, is default. There MUST NOT be more than one `<DisplayName>` element with `IsDefault` set as "true."

2.39. People Service vague on Aggregation support in Subscriptions

[LibertyPeopleService]

2.39.1. Summary

There is a need for clarification of support for the Aggregation mechanism in Subscriptions.

2.39.2. Resolution

Added after Line 471

There MUST be no <Aggregation> element present in a subscription.

2.40. People Service vague on expiration for Subscriptions to AddEntityRequests

[LibertyPeopleService]

2.40.1. Summary

There is a need for clarification of how Subscriptions created within AddEntityRequests should expire.

2.40.2. Resolution

Added after Line 473

Unless the value of the *expires* attribute specifies that expiration should occur earlier, for a <Subscription> sent within an <AddEntityRequest> message, expiration is considered to have occurred at such time as the PS provider has delivered a <Token> for the invited user to the WSC and the WSC has acknowledged its receipt.

2.41. People Service vague on use of starts attribute for Subscriptions to AddEntityRequest

[LibertyPeopleService]

2.41.1. Summary

There is a need for clarification of whether Subscriptions created within AddEntityRequests should have a starts attribute.

2.41.2. Resolution

Added after Line 469

For a <Subscription> sent within an <AddEntityRequest> message, the starts attribute SHOULD be omitted. If present, the PS provider MAY ignore the attribute.

2.42. People Service vague on includeData for a Subscription to AddEntityRequest

[LibertyPeopleService]

2.42.1. Summary

There is a need for clarification of whether Subscriptions created within AddEntityRequests should have an includeData attribute.

2.42.2. Resolution

Added after Line 482

For a <Subscription> sent within an <AddEntityRequest> message, the includeData attribute SHOULD be omitted.

2.43. Schema for ItemData in Notification

[LibertyPeopleService]

2.43.1. Summary

The text and processing rules for Subscriptions and Notifications allow for the possibility of a PS provider returning an empty ItemData but the schema stipulates a minOccurs="1."

Additionally, the ItemData schema does not allow for a Notification to include a Token, this necessary when a Subscription is used within an AddEntityRequest

2.43.2. Resolution

ItemDataType at Line 531 and at Line 2929 changed as follows

```
<xs:complexType name="ItemDataType">
     <xs:choice>
       <xs:element ref="Object" minOccurs="0" maxOccurs="unbounded"/>
       <xs:element ref="sec:Token" minOccurs="0"/>
     </xs:choice>
</xs:complexType>
```

2.44. MUST vs MAY for second level status codes

[LibertyPeopleService]

2.44.1. Summary

The People Service was overly constraining on the processing of second level status codes.

2.44.2. Resolution

1. Changed Line 421

   Failed — The value *Failed* means that the processing of the request message has failed. A second level status code MAY be used to indicate the reason for the failure.

2. Replaced multiple occurrences of:

   and the code attribute of the second level `<lu:Status>` element MUST be set with the following status code:

   with:

   A second level `<lu:Status>` MAY be inserted. If so, the code attribute of that second level `<lu:Status>` element MUST be set with the following status code:

2.45. People Service vague on dealing with existing Objects in AddToCollectionRequest

[LibertyPeopleService]

2.45.1. Summary

There is a need for clarification on how a PS provider should deal with an AddToCollectionRequest that asked for an Object to be added to a collection to which it already belonged.

2.45.2. Resolution

1. Added after Line 395

   DuplicateObject

2. Added after Line 1151

- MUST, if the `Object` specified by the value of a `<ObjectID>` element is already a member of the targeted collection, respond with *Failed* as the code attribute of the top level `<lu:Status>` element. A second level `<lu:Status>` MAY be inserted. If so, the code attribute of that second level `<lu:Status>` element MUST be set with the following status code:
  - DuplicateObject

3. Added after Line 1227

- MUST, if the `Object` specified by the value of a `<ObjectID>` element is not a member of the targeted collection, respond with *Failed* as the code attribute of the top level `<lu:Status>` element. A second level `<lu:Status>` MAY be inserted. If so, the code attribute of that second level *<lu:Status>* element MUST be set with the following status code:
  - CannotFindObject

2.46. Discovery Service EPR Profile notOnOrAfter attribute name

[LibertyDisco]

2.46.1. Summary

The `notOnOrAfter` attribute for the ID-WSF Endpoint Reference Profile (section 2.3.1.3) documents the name of the attribute as having an initial lower case 'n' in the normative text as well as in all of the examples throughout the document. However, the schema snippet in that section, the schema appendix, and the separate schema file all define the attribute with an upper case 'N'.

2.46.2. Resolution

The intent was for the attribute to have a lower case initial character and this follows the general Liberty naming paradigm for attributes, so the schema and schema snippet will be updated to reflect the attribute name with the initial lower case 'n'. The specific changes include:

1. Change the name of the attribute on line 404 of the specification to be `notOnOrAfter`.
2. Change the name of the attribute on line 4099 of the specification to be `notOnOrAfter`.
3. Change the name of the attribute on line 82 of the Discovery Service schema file to be `notOnOrAfter`.

2.47. Inconsistent Capitalization of `InteractIfNeeded` attribute

[LibertySOAPBinding]

2.47.1. Summary

The value for the interact attribute is shown as "InteractIfNeeded" on line 1721, but the schema examples on lines 1756 and 2265 show the default value as "interactIfNeeded" (different initial capitalization).

2.47.2. Resolution

Changed occurrences on referenced lines to have uppercase initial character.

References

[LibertyAuthn] Hodges, Jeff, Aarts, Robert, Madsen, Paul, Cantor, Scott, eds. " Liberty ID-WSF Authentication, Single Sign-On, and Identity Mapping Services Specification ," Version 2.0-errata-v1.0, Liberty Alliance Project (28 November, 2006). *http://www.projectliberty.org/specs*

[LibertyClientProfiles] Aarts, Robert, Kainulainen, Jukka, Kemp, John, eds. "Liberty ID-WSF Profiles for Liberty-Enabled User Agents and Devices," Version 2.0-errata-v1.0, Liberty Alliance Project (22 January, 2007). *http://www.projectliberty.org/specs*

[LibertyDisco] Cahill, Conor, Hodges, Jeff, eds. "Liberty ID-WSF Discovery Service Specification," Version 2.0-errata-v1.0, Liberty Alliance Project (29 November, 2006). *http://www.projectliberty.org/specs*

[LibertyInteract] Aarts, Robert, Madsen, Paul, eds. "Liberty ID-WSF Interaction Service Specification," Version 2.0-errata-v1.0, Liberty Alliance Project (21 April, 2007). *http://www.projectliberty.org/specs*

[LibertyPeopleService] Koga, Yuzo, Madsen, Paul, eds. "Liberty ID-WSF People Service Specification," Version 1.0-errata-v1.0, Liberty Alliance Project (06 March, 2007). *http://www.projectliberty.org/specs*

[LibertyIDWSF20SCR] Whitehead, Greg, eds. Version 1.0 errata v1.0, Liberty Alliance Project (21 April, 2007). *http://www.projectliberty.org/specs*

[LibertySecMech20] Hirsch, Frederick, eds. "Liberty ID-WSF Security Mechanisms Core," Version 2.0-errata-v1.0, Liberty Alliance Project (21 April, 2007). *http://www.projectliberty.org/specs*

[LibertySecMech20SAML] Hirsch, Frederick, eds. "ID-WSF 2.0 SecMech SAML Profile," Version 2.0-errata-v1.0, Liberty Alliance Project (08 November, 2006). *http://www.projectliberty.org/specs*

[LibertySOAPBinding] Hodges, Jeff, Kemp, John, Aarts, Robert, Whitehead, Greg, Madsen, Paul, eds. "Liberty ID-WSF SOAP Binding Specification," Version 2.0-errata-v1.0, Liberty Alliance Project (21 April, 2007). *http://www.projectliberty.org/specs*

[WSAv1.0] "Web Services Addressing (WS-Addressing) 1.0," Gudgin, Martin, Hadley, Marc, Rogers, Tony, eds. World Wide Web Consortium W3C Recommendation (9 May 2006). *http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/*