Liberty ID-WSF 1.0 Errata
Version: 1.0

Editors:
Darryl Champagne, IEEE-ISTO
Peter Thompson, IEEE-ISTO

Contributors:
John Kemp, Nokia Corporation
Jukka Kainulainen, Nokia Corporation
Jonathan Sergent, Sun Microsystems, Inc.

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

Filename: draft-liberty-idwsf-1.0-errata-v1.0.pdf
Notice

1. 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.

2. 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. Implementors 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.

Copyright © 2004 ActivCard; America Online, Inc.; American Express Travel Related Services; Axalto; Bank of America Corporation; Bell Canada; Cingular Wireless; Cisco Systems, Inc.; Communicator, Inc.; Deloitte & Touche LLP; Earthlink, Inc.; Electronic Data Systems, Inc.; Entrust, Inc.; Epok, Inc.; Ericsson; Fidelity Investments; France Telecom; Gemplus; General Motors; Hewlett-Packard Company; i2 Technologies, Inc.; Internet2; Intuit Inc.; MasterCard International; NEC Corporation; Netegrity, Inc.; NeuStar, Inc.; Nextel Communications; Nippon Telegraph and Telephone Corporation; Nokia Corporation; Novell, Inc.; NTT DoCoMo, Inc.; OneName Corporation; Openwave Systems Inc.; Phaos Technology; Ping Identity Corporation; PricewaterhouseCoopers LLP; RegistryPro, Inc.; RSA Security Inc.; Sabre Holdings Corporation; SAP AG; SchlumbergerSema; Sigaba; SK Telecom; Sony Corporation; Sun Microsystems, Inc.; Symlabs, Inc.; Trustgenix; United Airlines; VeriSign, Inc.; Visa International; Vodafone Group Plc; Wave Systems. All rights reserved.
1. Introduction

This document lists errata in the Liberty ID-WSF specification set. The specification set targeted by this errata document is listed in section 2 below. This is not an authoritative document, nor a final version, but a precursor for changes that will likely be included in a future revision of the targeted specifications.

The ID-WSF protocols, as initially specified, contained certain material errors, collectively referred to as errata. Readers of the Liberty ID-WSF specification set should note the errata in this document, and incorporate it into their reading of the specifications.

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

- http://www.projectliberty.org/specs/liberty-idwsf-utility-1.0-errata-v1.0.xsd
- http://www.projectliberty.org/specs/liberty-metadata-1.0-errata-v1.0.xsd
- http://www.projectliberty.org/specs/liberty-paos-1.0-errata-v1.0.xsd
2. Abbreviations

The following abbreviations are used in this document:

- **SE** - Substantive Errata designator
- **EE** - Editorial Errata designator
- **CR** - Change Request number (included for internal reference only).
3. Target Specifications

• [1] Liberty Reverse HTTP Binding for SOAP Specification
  http://www.projectliberty.org/specs/liberty-paos-v1.0.pdf


• [3] Liberty ID-WSF Discovery Service Specification
  http://www.projectliberty.org/specs/liberty-idwsf-disco-svc-v1.0.pdf


• [5] Liberty ID-WSF SOAP Binding Specification

• [6] Liberty ID-WSF Data Services Template Specification

• [7] Liberty ID-WSF Utility XML Schema
  http://www.projectliberty.org/specs/liberty-idwsf-utility-v1.0.xsd

• [8] Liberty Metadata Description and Discovery Specification
4. Editorial Errata

4.1. [EE1] Incorrect statement about PAOS MEP being different than the regular SOAP MEPs

4.1.1. Summary

We state on line 148 of [1] that this binding does not introduce any new message exchange pattern for SOAP. However, it is also erroneously stated on lines 266-268 of [1], that we do introduce a new message exchange pattern.

4.1.2. Resolution

Delete lines 266-268 of [1]

4.2. [EE2] Incorrect Example of PAOS usage

4.2.1. Summary

There were a number of inaccuracies in the example of PAOS usage given in the specification. The example has been updated to reflect the correct namespace for the Liberty ID-SIS Personal Profile Service; to use namespaces in the XPATH query; to remove the unnecessary ResourceID from the query, and to correct the usage of the personal profile service.

4.2.2. Resolution

Replace lines 176-252 of [1] with:

```
1. User-Agent requests a page...
GET /index HTTP/1.1
Host: horoscope.example.com
Accept: text/html; application/vnd.paos+xml
PAOS: ver="urn:liberty:paos:2003-08";

2. Server responds by asking for a date of birth...
HTTP 200
Content-Type: application/vnd.paos+xml
Content-Length: 1234

    <soap:Header>
        <paos:Request xmlns:paos="urn:liberty:paos:2003-08" responseConsumerURL="/soap"
            service="urn:liberty:id-sis-pp:2003-08"
            mustUnderstand="1"
            actor="http://schemas.xmlsoap.org/soap/actor/next"/>
        <sb:Correlation xmlns:sb="urn:liberty:sb:2003-08" messageID="6c34f8b9c2d"/>
    </soap:Header>
    <soap:Body>

```
3. Service at user agent responds to the SOAP request with a SOAP response inside an HTTP request...

```
POST /soap HTTP/1.1
Host: horoscope.example.com
Accept: text/html; application/vnd.paos+xml
Content-Type: application/vnd.paos+xml
Content-Length: 2345

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <sb:Correlation xmlns:sb="urn:liberty:sb:2003-08" messageID="4d3eae2e3f5g" refToMessageID="6c3a4f8b9c2d"/>
  </soap:Header>
  <soap:Body>
      <Data>
        <Birthday>--05-09</Birthday>
      </Data>
    </pp:QueryResponse>
  </soap:Body>
</soap:Envelope>
```

4. Finally the server responds with a page containing a personalized horoscope...

```
HTTP 200
Content-Type: text/html
Content-Length: 1234

<html>
  <head>
    <title>Your Horoscope from horoscope.example.com</title>
  </head>
  <body>
    <p>Dear Virgo,
      In May 2004 you will have to sit through many boring meetings.
      But this ordeal will be worth it and you will make new friends.</p>
  </body>
</html>
```

4.3. [EE3] Interaction Service Status Codes

4.3.1. Summary
184 Line 668 of the specification is incomplete, giving an incomplete description of the use of status codes within the interaction service.

186 **4.3.2. Resolution**

187  1. Replace lines 667, 668 of [2] with:
188     If processing is successful, the recipient MUST respond with a message containing an <InteractionResponse>
189     with a <Status> element holding a code attribute of is:success.
190     Other values for the code attribute are specified below, and MAY be returned in fault responses.

191  2. Replace lines 670-672 of [2] with:
192     The IS Service responds with an ID-WSF message that either contains an InteractionResponse element, or a SOAP fault. Either of these responses will contain a Status element and, upon success, the InteractionResponse will contain values for all the parameters in the query of the corresponding <InteractionRequest>. The code attribute of the Status element can take one of the following QNames:

196 **4.4. [EE4] Inconsistent text regarding the signing of Inquiry elements**

197 **4.4.1. Summary**

198 Clarification of text regarding the presence, and signing of Inquiry elements in InteractionStatements.

199 **4.4.2. Resolution**

200  1. Replace lines 692,693 of [2] with:
201     Contains one or more signed Inquiry element.

202  2. Replace lines 702-705 of [2] with:
203     This is a copy of the element submitted in the request, but with the value attributes of each InquiryElement set (or left blank) by the Principal. The <Inquiry> in an <InteractionStatement> MUST include all InquiryElements of InquiryElementType specified in the request; but other elements, such as <Help>, <Hint> and <Item>, MAY be omitted.

207  3. Replace lines 707 and 708 of [2] with:
208     Contains a signature that covers the Inquiry elements (and thus all child elements). The signature must be constructed by use of the private key associated with the content of the <ds:KeyInfo> of the <InteractionRequest>.

211  4. Replace line 725 of [2] with:
212     <element ref="Inquiry" maxOccurs="unbounded"/>
4.5. [EE5] Multiple Correlation Headers

4.5.1. Summary

It was previously stated on line 613 of [5] that the outgoing message "MUST include a <Correlation> header block", which does not prohibit a sender from adding multiple of these header blocks.

4.5.2. Resolution

Change lines 613, 614 to read:

The outgoing message MUST include exactly one <Correlation> header block (see <xref linkend="soap-msg-corr">) in the <S:Header> child element of the <S:Envelope> element.

4.6. [EE6] Incorrect examples

4.6.1. Summary

Several of the examples displayed in [5] are incorrect for various reasons pertaining both to other errata items, and other issues.

4.6.2. Resolution

1. Replace Example 3 (lines 405 - 447) with:

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/
xmlns:sb="urn:liberty:sb:2003-08"
xmlns:pp="urn:liberty:id-sis-pp:2003-08">
  <S:Header>
    <!-- other header blocks, eg wsse:security, may go here -->
    <Correlation S:mustUnderstand="1"
      id="B6432...466"
      actor="http://schemas.../next"
      messageID="uuid:efefefef-aaaa-ffff-cc cc-eeeeffffbbbb"
      timestamp="2112-03-15T11:12:12Z"/>
    <Provider providerID="example.com"
      affiliationID="affiliation.example.com"
      S:mustUnderstand="1"
      id="A9kendan...542"
      actor="http://schemas.../next"/>
    <!-- other header blocks, eg wsse:security, may go here -->
  </S:Header>
  <S:Body>
    <S:Fault>
      <faultcode>S:server</faultcode>
      <faultstring>Server Error</faultstring>
      <!-- <S:faultactor> should be absent -->
      <detail namespace="urn:liberty:sb:2003-08">
        <Status code="sb:"
          ref="Foo"
          comment="Bar" />
```
2. Replace Example 6 (lines 761-794) with:

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/
xmlns:sb="urn:liberty:sb:2003-08"
xmlns:idpp="urn:liberty:id-sis-pp:2003-08">
  <S:Header>
    <!-- other header blocks, eg wsse:security, may go here -->
    <Correlation S:mustUnderstand="1"
      id="A13454...245"
      actor="http://schemas.../next"
      messageID="uuid:efefefef-aaaa-ffff-cccc-eeeeffffbbbb"
      timestamp="2112-03-15T11:12:12Z"/>
    <Provider providerID="example.com"
      affiliationID="affiliation.example.com"
      S:mustUnderstand="1"
      id="A9kendan...542"
      actor="http://schemas.../next"/>
    <!-- other header blocks, eg wsse:security, may go here -->
  </S:Header>
  <S:Body>
    <idpp:Modify>
      <!-- this is an ID-PP Modify message -->
    </idpp:Modify>
  </S:Body>
</S:Envelope>
```

3. Replace Example 7 (lines 807-842) with:

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/
xmlns:sb="urn:liberty:sb:2003-08"
xmlns:idpp="urn:liberty:id-sis-pp:2003-08">
  <S:Header>
    <!-- other header blocks, eg wsse:security, may go here -->
    <Correlation S:mustUnderstand="1"
      id="B893483..83736"
      actor="http://schemas.../next"
      messageID="uuid:aaaaeeeefefe-feef-eefababababab"
      refToMessageID="uuid:efefefef-aaaa-ffff-cccc-eeeeffffbbbb"
      timestamp="2112-03-15T11:12:13Z"/>
  </S:Header>
  <S:Body>
    <!-- this is an ID-PP Modify message -->
  </S:Body>
</S:Envelope>
```
4. Replace Example 9 (lines 1083 - 1112) with:

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:idpp="urn:liberty:id-sis-pp:2003-08">
    <S:Header>
        <!-- other header blocks, eg wsse:security, may go here -->
        <Consent id="A124395732495743"
                 uri="urn:liberty:consent:obtained"
                 timestamp="2112-03-15T11:12:10Z"/>
        <Correlation S:mustUnderstand="1"
                     id="B89348383736"
                     actor="http://schemas.../next"
                     messageID="uuid:eeeecccc-efef-efef-bdddbbddd"
                     timestamp="2112-03-15T11:12:13Z"/>
        <!-- other header blocks, eg wsse:security, may go here -->
    </S:Header>

    <S:Body>
        <idpp:ModifyResponse>
            <!-- this is an ID-PP ModifyResponse message -->
        </idpp:ModifyResponse>
    </S:Body>
</S:Envelope>
```

5. Replace Example 10 (lines 1165 -1206) with:

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:sb="urn:liberty:sb:2003-08"
            xmlns:pp="urn:liberty:id-sis-pp:2003-08">
    <S:Header>
        <!-- other header blocks, eg wsse:security, may go here -->
        <Consent id="A12395732495743"
                 uri="urn:liberty:consent:obtained"
                 timestamp="2112-03-15T11:12:10Z"/>
        <Correlation S:mustUnderstand="1"
                     id="B89348383736"
                     actor="http://schemas.../next"
                     messageID="uuid:eeeecccc-efef-efef-bdddbbddd"
                     timestamp="2112-03-15T11:12:13Z"/>
        <!-- other header blocks, eg wsse:security, may go here -->
    </S:Header>

    <S:Body>
        <idpp:Modify> <!-- This is an ID-PP "Modify" message bound -->
            <!-- into the <Body> of a SOAP message. -->
        </idpp:Modify>
    </S:Body>
</S:Envelope>
```
4.7. [EE7] Propagation of modificationTime common attribute in the Data Services Template

4.7.1. Summary

The propagation of the value for the modificationTime attribute in [6] was not stated clearly enough to describe the intention.

4.7.2. Resolution

1. Replace the sentence starting on line 182 of [6] with:

So the time of the modification MUST be propagated up all the way to the root element, when container elements have the modificationTime attribute. If the root element has the modificationTime attribute, it states the time of the latest modification. Note that a data service may have the modificationTime attribute used only in leaf elements or not even for those as it is optional.
2. Add new text after the end of the line 1042 of [6]:

When the modificationTime attribute is used by a data service, the WSP MUST keep it up to date to indicate the time of the latest modification of an element and update it, whenever a modification is done either using the <Modify> request or some other way. When the modificationTime attribute is used in container elements, the time of a modification MUST be propagated to all ancestor elements of the modified element all the way up to the root element.

4.8. [EE8] The existence of ResourceID element in Query message of the Data Services Template

4.8.1. Summary

The <ResourceID> element is optional in [6]. There was unnecessary requirements for cases, when it can be left out.

4.8.2. Resolution

Replace lines 563-566 of [6] with:

The <ResourceID> element would have had the value urn:liberty:isf:implied-resource (see [LibertyDisco]). In this case the <ResourceID> MAY be left out.

4.9. [EE9] The TimeOut Status Code in the Data Services Template

4.9.1. Summary

The TimeOut status code defined in [6] should be returned, when the WSP can finish the processing as fast as it wants. The WSC has no control how long the WSP tries to process before giving up and returning the TimeOut status code. This needs to be stated more clearly.

4.9.2. Resolution

1. Replace the line 706 of [6] with:

The TimeOut status code SHOULD be used to indicate this, when the data is not returned to the WSC due to a WSP internal time out. The WSP defines how long it tries to process before giving up and returning the TimeOut status code.

2. Replace the line 1084 of [6] with:

The TimeOut status code SHOULD be used to indicate that the requested modification was not made due to a WSP internal time out. The WSP defines how long it tries to process before giving up and returning the TimeOut status code.
4.10. [EE10] Examples in the Data Services Template

4.10.1. Summary

The examples in [6] have two bugs.

4.10.2. Resolution

1. The time in line 816 of [6] is in wrong format, it should be 12:10:12.

2. The XPath expression in line 1150 is incorrect. Replace the line 1150 of [6] with:
   

4.11. [EE11] Keeping common attributes up to date in services based on the Data Services Template

4.11.1. Summary

The semantics of the common attributes are stated and those imply that they should be kept up to date, but there are no related processing rules stating this explicitly.

4.11.2. Resolution

Add new text after the line 1029 of [6]:

When any of the ACC, modifier, ACCTime or modificationTime attributes is used in a resource, the WSP hosting the data service MUST keep their values up to date. When data is modified, the modifier MUST contain the ProviderID of the modifier or have no value, and the modificationTime MUST define the time of the modification or have no value. The ACC MUST define the attribute collection context of the current value of a data element or have no value and the ACCTime MUST define the time, when the current value of the ACC was defined or have no value.

4.12. [EE12] Use of ID-* Fault message in DST

4.12.1. Summary

A WSP is expected to send an ID-* Fault message in case the processing of a request fails before starting to really to process the messages specified in [6].

4.12.2. Resolution


4.13. [EE13] CredentialRef

4.13.1. Summary

In [3], the CredentialRef element is incorrectly referred to as CredentialIDRef in several places.

4.13.2. Resolution

Replace all occurrences of "CredentialIDRef" in [3] with "CredentialRef".

Liberty Alliance Project

15

4.14.1. Summary

In [7], the <Status> element is missing a maxOccurs="unbounded".

4.14.2. Resolution

1. Update the StatusType XML type in [7] to read as follows:

```
<xs:complexType name="StatusType">
  <xs:annotation>
    <xs:documentation> A type that may be used for status codes. </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="Status" minOccurs="0" maxOccurs="unbounded"/>
    <xs:attribute name="code" type="xs:QName" use="required"/>
    <xs:attribute name="ref" type="xs:NCName" use="optional"/>
    <xs:attribute name="comment" type="xs:string" use="optional"/>
  </xs:sequence>
</xs:complexType>
```

2. XML Schema included references to [7] present in [2,3,5,6] should be updated to include this schema change.

4.15. [EE16] Reference to Discovery Service "SecMecID" is Incorrect

4.15.1. Summary

In [1] and [2], reference is made to the SecMecID element of the Liberty ID-WSF Discovery Service Specification. Such an element does not exist.

4.15.2. Resolution

1. line 129 of [1] should read:

```
The URI values of <SecurityMechID> elements of the <ResourceOffering> element can be given as PAOS Extensions (if the security mechanism(s) can be used for all PAOS exposed services), or as Option(s) for a particular Service.
```

2. Line 197 of [2] should read:

```
```
4.16. [EE17] SOAP Fault Language is Too Narrowly Defined

4.16.1. Summary

In [5], it was intended that SOAP Faults could be used in reference not only to header blocks, but also SOAP body elements. Additionally, it was intended that SOAP Fault codes could be defined by other specifications.

4.16.2. Resolution

1. Lines 347-351 of [5] should read:

   The <S:Fault> element’s <detail> child element SHOULD also contain an indicator of which header block or ID-* message body element, in the message being processed, is being referred to by the fault. The indicator is constructed by including in the <detail> element, after the <Status> element, an element of the same type as the one referred to by the fault. This element SHOULD have only one attribute, the id attribute whose value MUST be the value of the id attribute of the element referred to by the fault.

2. Lines 341-342 of [5] should read:

   MUST contain a code attribute set to the value as specified when the issuance of a ID-* Fault message is indicated.
5. Substantive Errata

5.1. [SE1] Discovery Service Extensibility

5.1.1. Summary

The Discovery Service modify response in [3] is not extensible. The modify request is extensible via the directive mechanism, but some new directives may require additional information to be provided in the response.

5.1.2. Resolution

1. Replace lines 774-777 of [3] with the following:

   The response contains the following elements and attributes:

   • Status: Contains status code; see processing rules.

   • newEntryIDs: If the status is OK, and InsertEntry was present in the Modify request, the newEntryIDs attribute MUST contain the list of entry IDs assigned to the new entries. The list MUST be in the same order that the InsertEntry elements were in.

   • Extension: Contains future extensions in other namespaces.

2. Insert the following line between lines 781 and 782 of [3], and between lines 1043 and 1044 of [3]:

   <xs:element ref="Extension" minOccurs="0" maxOccurs="1"/>

5.2. [SE2] Discovery Service directives

5.2.1. Summary

The desired behavior is not clear when both the AuthenticateSessionContext and AuthorizeRequester directives are specified.

5.2.2. Resolution

1. Insert the following between lines 756 and 757 of [3]:

   However, if both AuthorizeRequester and AuthenticateSessionContext are present, the discovery service provider SHOULD NOT generate both a ResourceAccessStatement and a SessionContextStatement; instead, it SHOULD generate a SAML assertion containing a ResourceAccessStatement, and the ResourceAccessStatement SHOULD contain a SessionContext element.
5.3. [SE3] Discovery Service Options matching

5.3.1. Summary

The algorithm for matching options when processing queries does not work as intended.

5.3.2. Resolution

1. Replaces lines 619-626 of [3] with the following:

The discovery service provider returns entries based on the requester’s criteria, the policies of the discovery resource, and the contents of the discovery resource. For each RequestedServiceType, the following matching rules MUST be followed in determining the subset of result that will be returned to the requester:

• If no Options element is present in the ResourceOffering, all entries in the discovery resource with the specified service type match.

• A ResourceOffering in the discovery resource matches the query if each Option element value in the RequestedServiceType is also present in the ResourceOffering. Note that this means that if there are no Option elements in the RequestedServiceType, the resource matches.

5.4. [SE4] Discovery Service ResourceID should be optional

5.4.1. Summary

The ResourceID element should be optional, and the default should be the implied ResourceID.

5.4.2. Resolution

1. Add the following paragraph between lines 121 and 122 of [3]:

The ResourceID element is optional (minOccurs="0"). If the ResourceID element is not present, the message should be processed the same way as if the ResourceID element was present and its value was "urn:liberty:isf:implied-resource". Note that in version 1.0 of this specification ResourceID was not optional and so older implementations may not process messages which have omitted ResourceID instead of including it with this value.

2. Change lines 141 and 929 to read:

<xs:choice minOccurs="0" maxOccurs="1">

5.5. [SE5] Security Mechanisms SessionContext missing attribute

5.5.1. Summary

The SessionContext element of [4] is missing a session index attribute.

5.5.2. Resolution

1. Insert the following after line 607 of [4]:

<xs:attribute name="SessionIndex" type="xs:string" use="optional"/>
2. Insert the following after line 623 of [4]

The SessionIndex attribute is used as an aid when in managing multiple sessions with a Principal. This attribute represents the same thing as the attribute of the same name in the \texttt{lib:AuthenticationStatement} [LibertyProtSchema]


5.6.1. Summary

The privacy protections accorded to the name identifier are burdensome when the producer and consumer are the same system entity.

5.6.2. Resolution

1. Replace lines 615-618 of [4] with the following text:

The \texttt{<SessionSubject>} element could be available to parties which are not privy to the name identifier of the subject. When this is the case the \texttt{<saml:SessionSubject>} element MUST be constructed in such a way as to protect the privacy of the identifier it carries by adhering to the rules defined in Encrypted Name Identifiers [Section 5.3.1]. An example of when the contents would not require these privacy protections is when the name identifier in the SessionSubject is produced and consumed by the same entity and the party with which it passes through has knowledge of the name.


5.7.1. Summary

Lines 467-470 and 489-490 of [4] (and the various examples) state that the TTP issues an assertion with a AuthenticationStatement including a SubjectConfirmation element which the sender is subsequently obliged to confirm. This is overspecified for the AuthenticationStatement since its purpose in the assertion is to inform the relying party how the sender authenticated to the assertion issuer. The text should state that the sender is obliged to confirm itself for any derived SubjectStatements which indicate subject confirmation obligations.

5.7.2. Resolution

1. Replace lines 467-470 with the following text:

These mechanisms utilize the Web Services Security SAML Profile [wss-saml] as the means by which the message sender authenticates to the recipient. In general these mechanisms assume that a TTP issues an assertion which includes an \texttt{<saml:AuthenticationStatement>} and other statements derived from \texttt{<saml:SubjectStatement>}. The \texttt{<saml:AuthenticationStatement>} describes the authentication event at the authority. For each of the \texttt{<saml:SubjectStatement>} bound to the assertion the authority dictates the subject confirmation obligations the subject must affirm to a relying party in order for the statement to be considered trustworthy.

2. Replace lines 489-490 of [4] with the following text:

The sender MUST adhere to its subject confirmation obligation in accordance with the semantics of the confirmation method described by each \texttt{<saml:SubjectStatement>} bound to the \texttt{<saml:Assertion>}.
3. Replace lines 1072-1094 of [4] with the following text:

```xml
<saml:AuthenticationStatement
  AuthenticationMethod="urn:ietf:rfc:2246"
  AuthenticationInstant="2003-09-09T16:57:30.000Z">
  <saml:Subject>
    <saml:NameIdentifier format="urn:liberty:iff:nameid:entityID">
      http://serviceprovider.com/
    </saml:NameIdentifier>
  </saml:Subject>
</saml:AuthenticationStatement>
```

4. Replace lines 1214-1236 of [4] with the following text:

```xml
<saml:AuthenticationStatement
  AuthenticationMethod="urn:ietf:rfc:2246"
  AuthenticationInstant="2003-09-09T16:57:30.000Z">
  <saml:Subject>
    <saml:NameIdentifier format="urn:liberty:iff:nameid:entityID">
      http://serviceprovider.com/
    </saml:NameIdentifier>
  </saml:Subject>
</saml:AuthenticationStatement>
```


5.8.1. Summary
We state on line 803 of [4] that the confirmation data includes the `saml:AssertionID` of the assertion which the proxy presented to the requester for authorization purposes. However, the actual schema element is of type `saml:AssertionIDReference`. The processing rules and data requirements to build a proxy chain were confusing and/or undefined.

5.8.2. Resolution

1. Replace lines 782-787 of [4] starting with sentence beginning with the word "OPTIONALLY" with the following text:

```
OPTIONALLY, the assertion issuer MAY include information which assists in building a chain of transited proxies. It is RECOMMENDED that the `<saml:Advice>` element be decorated with a `<saml:AssertionIDReference>` which is a reference to the assertion bearing it. Also as the chain builds the assertion should be augmented with `<ProxyTransitedStatement>`. The issuer should include a `<ProxyTransitedStatement> for each proxy which has participated in the progression of assertion issuance. See Proxy Chaining [Section 8.3.2] for a recommendation on constructing the proxy chain.
```
2. Replace lines 799-807 of [4] with the following text:

In some operational settings it may be necessary to carry the chain of proxies traversed. The following algorithm describes how an assertion issuing authority could formulate the proxy chain.

It is presumed that when a system entity interacts with the assertion issuing authority that the system entity will include a claim which contains a `<ProxyTransitedStatement>` bearing `<ProxyInfoConfirmationData>`.

This claim SHOULD be in the form of a SAML assertion carried as a security token within the security header of the request to the assertion issuing authority.

D1 The confirmation data includes the `<saml:AssertionIDReference>` of the assertion which the initial requester presented to the system entity needing subsequent proxy capability. It is presumed that the assertion issuing authority decorates assertions with `<saml:AssertionIDReference>` within the `<saml:Advice>` element for assertions which it deems to be proxiable. Thus, the assertion issuer could use this information to locate the requester’s assertion and add it to the list of proxies transited. Given the assertion the proxy chain can be formulated and a new chain created by appending the most recently transited proxy to the chain. The result will be an `<saml:Assertion>` comprised of `<ProxyTransitedStatement>` elements for each of the proxies transited. It is recommended that this assertion be carried within an `<saml:Advice>` element of the assertion issued to the proxy. The following example depicts an assertion containing advice regarding transited proxies.

```
<saml:Assertion
   xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
   MajorVersion="1" MinorVersion="0"
   AssertionID="2sxJu9g/vvLG9saAN9bKp/8q0HKU="
   Issuer="authority.example.com"
   IssueInstant="2003-09-09T16:58:33.173Z">

  <saml:Advice>
    <saml:AssertionIDReference>refers to this assertion</saml:AssertionIDReference>
    <saml:Assertion>
      <!-- This statement reflects the first proxy transited -->
      <ProxyTransitedStatement>
        <Issuer>authority.example.com</Issuer>
        <IssueInstant>2003-09-09T16:58:30.173Z</IssueInstant>
      </saml:AssertionIDReference>
      <saml:Subject>
        <saml:NameIdentifier format="urn:liberty:iff:nameid:entityID">
          http://first.example.com/
        </saml:NameIdentifier>
        <!-- refers to assertion first.example.com confirmed to authority.example.com to get an assertion which named first.example.com the ProxySubject. -->
      </saml:Subject>
    </ProxyTransitedStatement>
    <ProxyTransitedStatement>
      <Issuer>authority.example.com</Issuer>
      <IssueInstant>2003-09-09T16:58:32.173Z</IssueInstant>
      <saml:AssertionIDReference>
        <!-- refers to assertion second.example.com confirmed to authority.example.com to get an assertion which named second.example.com the ProxySubject. -->
      </saml:AssertionIDReference>
      <saml:Subject>
        <saml:NameIdentifier format="urn:liberty:iff:nameid:entityID">
          http://second.example.com/
        </saml:NameIdentifier>
      </saml:Subject>
    </ProxyTransitedStatement>
  </saml:Advice>
</saml:Assertion>
```

<!-- The AuthenticationStatement carries information which describes the identity of the entity this assertion was issued to (the Subject) and the method the Subject
5.9. [SE9] Optionality of Provider Header Block

5.9.1. Summary

Most senders of ID-WSF SOAP messages will be Liberty providers, but some may not, and thus cannot claim to be a Liberty provider, with the addition of a provider header block to messages they send. The Provider header is currently mandated by the specification, but should be optional.
5.9.2. Resolution

1. Change lines 63, 64 of [5] to read:
   Participants in ID-* interactions may declare themselves by their Provider ID, as well as their Affiliation ID if appropriate. This specification defines the <Provider> header block for this purpose.

2. Change line 360 of [5] to read:
   having a <Correlation> header block, and possibly a <Provider> header block in its <S:Header> element, and,

3. Change lines 633, 634 of [5] to read:
   The message SHOULD include exactly one, and MUST include no more than one <Provider> header block (see <xref linkend="provider"/> in the <S:Header> child element of the <S:Envelope> element. If this header block is present, the sender is claiming to be a Liberty provider with the specified providerID.

4. Change line 713, 714 to read:
   Verify that any declared providerID or affiliationID, are valid. The receiver SHOULD perform this verification and validation against metadata obtained via methods specified by <xref linkend="LibertyMetadata"/>

5.10. [SE10] SOAP Fault Schema Incorrect

5.10.1. Summary

A fragment from the SOAP 1.1 Envelope Schema <xref linkend="SOAPv1.1-Schema"/> in [5] was displayed incorrectly, with sub-elements qualified in the SOAP namespace. Similarly, several examples, and references to elements from that schema are displayed incorrectly as namespace-qualified.

5.10.2. Resolution

1. Replace Figure 3 (lines 282 - 300) of [5] with:
   <xs:complexType name="Fault" final="extension">
     <xs:annotation>
       <xs:documentation>
         Fault reporting structure
       </xs:documentation>
     </xs:annotation>
     <xs:sequence>
       <xs:element name="faultcode" type="xs:QName"/>
       <xs:element name="faultstring" type="xs:string"/>
       <xs:element name="faultactor" type="xs:anyURI" minOccurs="0"/>
       <xs:element name="detail" type="tns:detail" minOccurs="0"/>
     </xs:sequence>
   </xs:complexType>
   <xs:complexType name="detail">
     <xs:sequence>
       <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
     </xs:sequence>
   </xs:complexType>

Liberty Alliance Project
2. Replace all instances of `<S:faultcode>`, `<S:faultstring>`, `<S:faultactor>`, `<S:detail>` with `<faultcode>`, `<faultstring>`, `<faultactor>`, `<detail>` respectively.

5.11. [SE11] SOAP Fault Usage, Codes

5.11.1. Summary

There were some inconsistencies and typos in the status codes defined for SOAP Fault responses, and the processing steps for using faults.

5.11.2. Resolution

1. Replace all occurrences of `sb:BogusActor` with `sb:InvalidActor`.
2. Replace all occurrences of `sb:BogusMustUnstnd` with `sb:InvalidMustUnderstand`.
3. Replace all occurrences of `sb:invalidRefToMsgID` with `sb:InvalidRefToMsgID`.
4. Replace all occurrences of `sb:IDStarMsgNotUnstd` with `sb:IDStarMsgNotUnderstood`.
5. Replace all occurrences of `sb:ProcCtxURINotUnstd` with `sb:ProcCtxURINotUnderstood`.
6. Replace line 334 of [5] with:

   SHOULD contain a `<faultcode>` element whose value SHOULD be either "S:server" or "S:client".


5.12.1. Summary

The processing context facet allowing request simulation was incorrectly combined with the facets available for indicating that a request should be conducted as if the Principal were either online, or offline.

5.12.2. Resolution

1. Replace lines 970-973 of [5] with:

   Three processing context facet URIs are defined below for use with the `<ProcessingContext>` header block:

   `urn:liberty:sb:2003-08:ProcessingContext:PrincipalOffline`

   Conduct the processing of the ID-* message as if the Principal is offline.

   `urn:liberty:sb:2003-08:ProcessingContext:PrincipalOnline`

   Conduct the processing of the ID-* message as if the Principal is online.


   Simulate the processing of the ID-* message.

2. Replace lines 993 - 998 of [5] with:

   In summary, the overall intended side-effect of using the above-defined processing context facets is for the receiver to evaluate applicable policy, and return a putative indication of success or failure to the sender. This provides WSCs the capability to make an ID-WSF or ID-SIS service request and ascertain whether it will be successful or not - without the service request actually being carried out. Additionally, it facilitates carrying out any user interaction that may be indicated by the current combination of service request context and applicable policy. This will, for example, facilitate some WSCs to fashion more "user friendly" experiences.
3. Insert after line 1009 of [5]:
A sender MAY indicate that it believes either that the Principal is currently "online" or "offline" when it sends a message by specifying one of the two processing context facet URIs:

- urn:liberty:sb:2003-08:ProcessingContext:PrincipalOnline

4. Replace lines 1010-1013 of [5] with:
The sender will typically receive a response from the receiver indicating success or failure or will receive a SOAP fault indicating a processing error with the SOAP-bound ID-* message. Note that in the case of a "successful" request simulation, the service will not return any result data other than an indication of success or failure to the sender.

5. Insert after line 1022 of [5]:
If the receiver is asked to simulate processing of the request (by the inclusion of the urn:liberty:sb:2003-08:ProcessingContext:Simulate facet URI), and they are both able and willing to honor that processing context, then the receiver MUST evaluate the conveyed ID-* message, but MUST NOT actually perform the operation. That is, the receiver MUST NOT make actual changes to underlying ID-* service datastore, and it MUST NOT return any data as a result of evaluating the ID-* message.

If the sender includes a <UserInteraction> header block, in addition to the <ProcessingContext> header block, then both participants MUST initiate the indicated user interaction (see [LibertyInteract] appropriately, and incorporate information supplied by the user as a part of the interaction into appropriate data and/or policy stores, even if the urn:liberty:sb:2003-08:ProcessingContext:Simulate URI is specified in a <ProcessingContext> header.

6. Insert after line 1036 of [5]:
Note:
The receiver MAY reference multiple <ProcessingContext> headers in the <detail> of the fault response (in accordance with the rules specified in <Link to ID-* Fault Messages section>).


5.13.1. Summary
Various metadata changes were required to accommodate the other errata. These included revising the schema element IDFFSOAPAuthnProtocolProfile with AuthnServiceURL and created additional processing rules for use of this resource, and failure conditions, Provided clarifying text around when the use of multiple provider descriptors is appropriate, and clarifications in the use and meaning of affiliationOwnerID attribute on AffiliationDescriptor.

5.13.2. Resolution
Delete lines 401, 434, Remove RelationshipTerminationNotificationProtocolProfile
Lines 405, 436, insert NameIdentifierMappingEncryptionProfile
Delete lines 496, 507, 508, Remove IntroductionNotificationProtocolProfile
Line 496, Insert: AuthnServiceURL [Optional, 0-1] of type anyURI describes the SOAP Endpoint supporting the ID-FF authentication by the identity provider as defined in [LibertyInteract] and supports the relevant profile(s) cited in SingleSignOnProtocolProfile. IF the IDP supports SOAP-based IDFF authentication, indicated by the
associated SingleSignOnProtocolProfile, and there is no AuthnServiceURL provided, then the IDP supports this profile at the URL identified by SingleSignOnServiceURL.

Line 507, Insert:

```
<xs:element name="AuthnServiceURL" type="xs:anyURI" minOccurs="0"/>
```

Line 520, Insert:

Entities describing a single providerID, but wish to publish different metadata for two implementations protocol support (e.g. IDFF 1.1 services vs. IDFF 1.2 services) may use the plurality of IDPDescriptor and SPDescriptor to convey this.

Delete lines 591, 592, 617: affiliationID

Expand line 593-594 to:

```
affiliationOwnerID [Required] The providerID of the owner or parent operator of the affiliation, from which, additional metadata may be derived. This DOES NOT indicate affiliation membership of entity described as affiliationOwnerID. Thus if a member is both the owner and a member of the affiliation, they must indicate both in the instance (e.g. the entities providerID appears in both affiliationOwnerID AND AffiliateMember.)
```
References

