Liberty ID-WSF Implementation Guide

DRAFT Version 1.0 - 08

Editor:
David Weitzel, Mitretek Systems

Contributors:
Conor Cahill, AOL Time Warner
John Kemp, Liberty Alliance
Susan Landau, Sun
Andrew Lindsay-Stewart, Vodafone
Tom Wason, Liberty Alliance

Abstract:
This Liberty Web Services Framework (WSF) Implementation Guideline (IG) conveys insights to developers implementing the Liberty WSF architecture. It is not an overview, but rather strives to give examples, lessons learned, and best practices for implementing the Liberty WSF specifications. It should be used in conjunction with the normative specifications of the Liberty WSF document suite by those who have a solid working understanding of web services technologies and protocols.
Notice

This specification 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 Specification 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.
Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Editor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0-01</td>
<td>8-1-2003</td>
<td>David Weitzel</td>
<td>Basic document as anticipated at brainstorming session in Helsinki in May 2003</td>
</tr>
<tr>
<td>1.0-02</td>
<td>8-20</td>
<td>David Weitzel</td>
<td>Added input from July 2003 Helsinki Face to Face and private IOP. Fixed bugs 422 &amp; 423</td>
</tr>
<tr>
<td>1.0-03</td>
<td>8-24</td>
<td>David Weitzel</td>
<td>Added in implementation information from the Liberty Security &amp; Privacy Overview</td>
</tr>
<tr>
<td>1.0-04</td>
<td>9-12</td>
<td>David Weitzel</td>
<td>Added mobile issues. Added proposed edits. Added input from Santa Clara Face to Face and private interop. Edited incomplete sections. Fixed minor formatting.</td>
</tr>
<tr>
<td>1.0-05</td>
<td>11-12</td>
<td>David Weitzel</td>
<td>Fixed bug 508: acronyms, edits, and reference section</td>
</tr>
<tr>
<td>1.0-06</td>
<td>1-15-2004</td>
<td>David Weitzel</td>
<td>Added service example section based on AOL inputs</td>
</tr>
<tr>
<td>1.0-07</td>
<td>2-11-2004</td>
<td>David Weitzel</td>
<td>Revised service example section</td>
</tr>
<tr>
<td>1.0-08</td>
<td>4-18-2004</td>
<td>Darryl Champagne</td>
<td>Update copyright notice</td>
</tr>
</tbody>
</table>
# Contents

1. Introduction .......................................................................................................................... 5

2. Goals & Scope ......................................................................................................................... 6

   2.1. Overview of WSFs in the LA Context ............................................................................. 6

   2.2. Assumed Knowledge of the Liberty Specifications ....................................................... 7

   2.3. Assumed Knowledge of Internet Technology ............................................................... 7

3. Process .................................................................................................................................. 8

   3.1. First Implementors .......................................................................................................... 8

   3.2. Key Environments .......................................................................................................... 8

      3.2.1. Enterprise ............................................................................................................... 8

      3.2.2. E-Commerce .......................................................................................................... 8

      3.2.3. Mobile .................................................................................................................. 8

      3.2.4. E-Government ...................................................................................................... 9

4. Structure ............................................................................................................................... 10

   4.1. Elements of WSF ........................................................................................................... 10

   4.2. Relation to ID-FF .......................................................................................................... 10

   4.3. Relation to Liberty Services Specifications: ID-PP and ID-EP Services ..................... 10

5. Implementation Lessons Learned .......................................................................................... 11

   5.1. Discovery ....................................................................................................................... 11

   5.2. Interaction Service ........................................................................................................ 11

   5.3. Data Services Template ............................................................................................... 11

   5.4. Security Mechanisms .................................................................................................. 11

   5.5. Key Environments ....................................................................................................... 11

      5.5.1. Enterprise ............................................................................................................. 12

      5.5.2. E-Commerce ......................................................................................................... 12

      5.5.3. Mobile ................................................................................................................ 12

      5.5.4. E-Government ..................................................................................................... 13

   5.6. Special Issues ................................................................................................................ 13

      5.6.1. Underlying Protocols ........................................................................................... 13

      5.6.2. Privacy and Security ............................................................................................ 14

      5.6.3. Development Environments ................................................................................ 16

6 Example Sessions ..................................................................................................................... 17

   6.1 Device boot up ................................................................................................................ 17

   6.2 Device initiates authentication ....................................................................................... 17

   6.3 Auth Server responds with auth mechanism choice ....................................................... 18

   6.4 Device submits credentials to Auth Server ..................................................................... 18

   6.5 Auth Server returns Security Token & Discovery Info ................................................ 19

   6.6 Device Requests Service Info from Discovery Service ............................................... 20

   6.7 Discovery Service returns Service Info ......................................................................... 21

   6.8 Device Requests data from Radio Service .................................................................... 23

   6.9 Radio Service returns Info ............................................................................................ 24

   6.10 Device Requests additional info from Radio ............................................................... 25

   6.11 Radio Service returns info .......................................................................................... 25

   6.12 Device Requests Photo Service Info from Discovery Service .................................. 26

   6.13 Discovery Service returns Photo Service info ............................................................. 26

   6.14 Device requests info from Photo Service ................................................................. 28

   6.15 Photo service returns info .......................................................................................... 29

   6.16 Device Renews Security Token .................................................................................. 29

   6.17 The Authentication Server returns new token ............................................................. 31

7. References ............................................................................................................................ 33

   7.1. Normative ..................................................................................................................... 33

   7.2. Informative ................................................................................................................... 33

© Copyright 2002-2004 Liberty Alliance Project. All Rights Reserved
1. Introduction

Liberty Alliance provides several documents in addition to the specifications. These documents are defined as “non-normative”, meaning that they are not requirements, but are supportive documents serving to explain various facets and applications of the specifications. The mode may be more conversational than normative documents. These documents are classified as “Other Supporting Documents” and are subject to the Liberty copyright constraints.

A Liberty Alliance implementation guidelines document is a complement to the normative specification documents; it provides guidelines on how the specifications should actually be implemented. Implementation guidelines provide clarification on the specifications as well as wisdom learned—often the hard way—by developers. The audience is application developers.

An implementation guidelines is a dynamic document that may change frequently as experience teaches effective means for implementing the specifications. It provides a narrative discussion of important issues and their resolution. The implementation guidelines may, at times, provide input to future versions of the specifications. It will make specific references to specific sections of the specifications, but is not a complete index to the specifications.

An implementation guidelines provides representative examples of implementations, or parts of implementations, that exercise specific functionality. For example, it demonstrates how specific protocols are executed, how security is maintained in specific scenarios and so forth. An implementation guidelines provides explanations of effective architectures, methods for optimizing performance, scaling notes, and warnings. It may illustrate with block and flow diagrams, sample messages and code fragments.

This document is non-normative. However, it provides implementers and deployers guidance in the form of policy/security and technical notes. Further details of the Liberty ID-FF architecture are given in several normative technical documents associated with this implementation guide, specifically [LibertyID-WSFDataServiceTemplate], [LibertyID-WSFInteractionService], [LibertyID-WSFDiscoveryService], [LibertyID-WSFSecurityMechanisms], and [LibertyID-WSFSoapBinding] as well as the non-normative [LibertyID-WSFOverview]. Note: The more global term Principal is used for user in Liberty’s technical documents. Definitions for Liberty-specific terms can be found in the [LibertyGloss]. Also, many abbreviations are used in this document without immediate definition because the authors believe these abbreviations are widely known, for example, HTTP and SSL. However, the definitions of these abbreviations can also be found in [LibertyGloss]. Note: Phrases and numbers in brackets [ ] refer to other documents; details of these references can be found in Section 6 (at the end of this document). As this document is non-normative it does not use terminology "MUST", "MAY", "SHOULD" in a manner consistent with RFC-2119 (see [RFC2119]).

An implementation guidelines document should be considered a complement to the Liberty Alliance specifications and provides guidelines for how the Liberty specifications should be implemented. It provides additional clarifications on some issues in the specifications, as well as errata on the specifications. This document should be viewed as a continuing work in progress meant to assist serious implementers. If a reader is looking for basic overview information, deployment guidance, static conformance information, or certification specifications they must look elsewhere in the Liberty document set.
2. Goals & Scope

This Liberty Alliance Web Services Framework Implementation Guide (WSF-IG) only covers the Liberty Alliance specifications in the Web Services Framework (WSF) arena. Other Implementation Guides exist or are contemplated for other elements of the Liberty Alliance specifications.

As a non-introductory non-normative document, this section of the WSF-IG will lay out:

- Overview of WSFs
- Assumed knowledge of Liberty architecture
- Assumed knowledge of web services and Internet technology

These items are described below.

2.1. Overview of WSFs in the LA Context

The goal of the WSF-IG is to help developers understand the implementation details of the Liberty Alliance WSF architecture and to share best practices and lessons learned by earlier implementers of the framework. The major architectural components identified in Figure 1 should be familiar to those who have a working knowledge of the Liberty Alliance specifications. To reiterate, this WSF-IG is meant to concentrate on the WSF components of the Liberty specifications and will only touch on other parts of the Liberty Architecture as they relate to the WSF components.

Figure 1: Liberty Architecture
2.2. Assumed Knowledge of the Liberty Specifications

The normative technical specifications of the Liberty Alliance are identified in a series of documents explaining the format and syntax of each of the components. They are identified in Figure 2 along with significant Internet technologies that are utilized in the Liberty specifications. This WSF-IG document is meant to help developers implement the features and functions of the normative specifications of the Liberty WSF components. To become familiar with other components of the Liberty specifications one must look to the appropriate normative and non-normative documents for those components. An overview of the Liberty Alliance technical architecture is also available in the Liberty ID-WSF Overview document. A good starting place for other architectural components should be the appropriate Overview document related to the component of interest.

Figure 2: Liberty Modules

2.3. Assumed Knowledge of Internet Technology

There are a number of evolving Internet resources which are utilized by the Liberty Alliance specifications including such protocols such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP) and Security Assertion Markup Language (SAML). Additionally, an implementor of Liberty Alliance specifications should be familiar with basic Internet architectures, basic Public Key Infrastructure (PKI) digital signature concepts, basic Internet security, Secure Socket Layer/Transport Layer Security (SSL/TLS), Hypertext Transfer Protocol (HTTP) HTTP Secure (HTTPS), Uniform Resource Identifiers (URIs), Domain Name System (DNS), Web Services Description Language (WSDL), the Liberty Alliance WSF components they wish to implement as well as the Liberty components that must be utilized to implement or interface to the Liberty WSF component being implemented.
3. Process

This implementation guide concentrates on several areas. They are first implementers and key environments.

3.1. First Implementors

An implementation guidelines document should be considered a complement to the Liberty Alliance specifications and provides guidelines for how the Liberty specifications should be implemented. It provides additional clarifications on some issues in the specifications, as well as errata on the specifications.

3.2. Key Environments

There are several key service environments that this implementation guide will concentrate on. They are enterprise, e-commerce, mobile, and e-government.

3.2.1. Enterprise

Rather than jump into a heterogeneous authentication architecture outside the enterprise, many Liberty members have found that significant hurdles exist in merely rationalizing intra-enterprise authentication and web services. Developers should take heed both from development and marketing perspective of this fact. First they should stand up Liberty architectures in a simulated intra-enterprise environment both because it is somewhat simpler and second it simulates the first deployments of many organizational users of the Liberty specifications. However, due to the complexity of many modern enterprises, little comfort should be taken. Because many enterprises span multiple architectures, component systems, legacy authentication schemes, and world-wide footprints, the deployment of intra-enterprise authentication using Liberty components is far from easy.

3.2.2. E-Commerce

Most users of the Liberty specifications anticipate utilizing the power of the specifications in full-blown inter-enterprise deployments. In this environment very few simplifying assumptions can be made. Thus, step-wise and component-wise deployment strategies are recommended. Fortunately, the development of the Liberty specifications facilitates this approach. One can utilize the power of the Liberty ID-FF framework without having to delve into much of the Liberty WSF realm. One can develop Liberty WSF compliant software without having to deploy specific services on top of it. Of course, many user organizations anticipate full development and broad deployment of the full suite of Liberty specifications.

3.2.3. Mobile

The mobile environment presents both unique opportunities and unique challenges in the authentication environment. The widespread worldwide deployment of mobile devices is a ripe opportunity for the coordination of authentication architectures. The continued convergence of phone and personal digital assistant technology calls for devices that can utilize the full power of both the mobile telephony and wireless data environments. However, due to limitations on power, memory size, display size, and bandwidth mobile environments must live within certain constraints. Additionally, some legacy architectural decisions present current constraints on deployment of architectures and capabilities anticipated by the Liberty specifications.
3.2.4. E-Government

Governments play a special role in e-authentication both as a user and as a holder of identity information. We anticipate a number of different Service Providers will also serve as Identity Provider and in the business environment, consumers will have choice as to which Identity Providers they use. Due to the unique role of government, users of e-government services, however, may be required to use the government's choice of Identity Provider(s). For this reason, it is extremely important that governments choose e-authentication systems that appropriately protect both privacy and confidentiality. For transparency's sake, systems that depend upon open standards provide a better choice for government.
4. Structure

The Liberty WSF architecture can be viewed as a suite of capabilities to enable intra- and inter-enterprise web services to operate in a heterogeneous authentication environment. In short, in a Liberty enabled environment one should be able to interoperate with multiple principals, service providers and identity providers in a fashion where real-time and near real-time decisions can be made about what trust can be given to formerly unknown providers.

4.1. Elements of WSF

The web services model is rapidly gaining acceptance in the Internet community as a scalable and adaptable model for implementing services and systems that need to interoperate among multiple systems providers utilizing multiple components. To meet this emerging Internet development model the Liberty Alliance has adopted with use of a web services framework for implementing the core architecture of the Liberty Alliance specifications.

Specifically, the components of the Liberty ID-WSF framework are outlined in the [LibertyID-WSFOverview].

4.2. Relation to ID-FF

The Liberty ID-WSF framework works in conjunction with the structure of the Authentication techniques developed in the ID-FF framework. It is generally anticipated that most deployments of Liberty ID-WSF technologies will be done in conjunction with the use of ID-FF capabilities. Implementors of ID-FF should have a strong grounding in the techniques and capabilities of the ID-FF framework. They are well served to have a strong working knowledge of the companion [LibertyID-FFOverview] and [LibertyID-FFImplementationGuide].

4.3. Relation to Liberty Services Specifications: ID-PP and ID-EP Services

The Liberty ID-WSF framework forms a foundation of structures that can be used to implement identity service specifications. The Liberty ID-PP and ID-EP services are the first two specifications that have been created in this fashion. However, many more identity services can be envisioned to utilize the Liberty ID-WSF framework. For developers, examination of the Liberty ID-PP and ID-EP specifications can assist in learning how the ID-WSF framework can be put to use. For developers these insights may be helpful from both a development and testing perspective.
5. Implementation Lessons Learned

Many of the best implementation insights are those gained by developers who have already succeeded in implementing a specification. To that end, the early developers of systems invoking the ID-WSF specifications have begun to share their development insights and lessons learned.

5.1. Discovery

An implementor should be familiar with the Conceptual Model and Terminology section of the normative [LibertyID-WSFDiscoveryService]. The model gives a solid introduction to understanding what the normative portion of the specification describes. The end of that document also contains the XSD, WSD, and example XSL stylesheets.

5.2. Interaction Service

An implementor should be familiar with the Interaction Service cases identified in the [LibertyID-WSFInteractionService]. Similarly, the end of that document also contains the XSD, WSD, and example XSL stylesheets.

Interoperability note: If a Service Provider, SP, does not send the UserInteraction header then it probably can not redirect. So, the SP should warn that it is OK to redirect but this SP can not do the redirection. Also, if the SP does send the UserInteraction header with redirect, then it should have the user available.

5.3. Data Services Template

An implementor should familiarize themselves with the specification check list provide in section 4 of the [LibertyID-WSFDataServiceTemplate] specification. Since identity service specifications such at ID-PP and ID-EP utilize the DST specification extensively, an implementor can aid their understanding of the uses of the [LibertyID-WSFDataServiceTemplate] specification by looking at the normative and non-normative documents of the ID-PP and ID-EP services.

5.4. Security Mechanisms

The Liberty ID-WSF Security Mechanisms document contains several non-normative sections which help an implementor understand the purpose of the security mechanisms.

A quality policy engine is critical. There is an important role of the Policy Decision Point and Policy Enforcement Point in enforcing good security practice. While Liberty will not make any specific recommendation, an implementor should evaluate the various offerings closely.

5.5. Key Environments

The developers of systems utilizing Liberty ID-WSF specifications in the key environments identified in the previous section have similarly shared their insights and lessons learned.
5.5.1. Enterprise

Many early deployments of the Liberty specifications are occurring in enterprise environments. The deployments anticipate the ability to integrate many formerly unconnected authentication and attribute systems into a seamless enterprise instantiation of standards-based authentication web services.

5.5.2. E-Commerce

Most commercial deployments of the Liberty ID-WSF framework will be in the general e-commerce web services environment. Such a deployment must anticipate the seemingly limitless uses that the deployment may be called upon to support. Rigorous development lab testing, boundary case testing, stress testing and interoperability testing should be utilized.

One particular issue that has been raised is the possible security impact of too short a cache life thus not being able to detect a replay attack. Another is the judicious use of fault logging.

5.5.3. Mobile

Privacy should be of increased concern in the mobile environment and typically should allow for affirmative end user action before using a service offering.

5.5.3.1. Roaming:

The current specifications do not yet provide a robust solution to share an identity's data when roaming across circles of trust. However, when the functionality becomes available mobile operators should be able to leverage the established trust that they have with their existing voice roaming agreements.

5.5.3.2. LUAD-WSP:

- Dual Identity Services:
As the LUAD-WSP may not always be reachable, there is a strong likelihood that there will be a dual network-based identity service registered with the DS. Therefore, there should be a mechanism for the client to synchronize its service information with that of the network-based service such that the end-user only has to update one service and the data propagated to other dual identity service. Possible options might include:
  - An existing protocol e.g SyncML, or …
  - Add a synchronization method to the DST specification.

As identity services can be extensible, “limited” storage devices may only store a subset of an identity service. Therefore, the synchronization mechanism should also be able to cope with this “limited” identity service.

- Security/Privacy:
Since a LUAD-WSP needs to advertise the presence of a service, there is a higher risk the privacy of an end-user may be compromised by a rogue service provider. PAOS-enabled clients should therefore:
  - Allow for affirmative end-user action before advertising the service to a service provider; and
  - For the service residing on the client, enable privacy/permission preferences under the control of the end-user

Due to the limited bandwidth of current mobile networks, when using PAOS with message level security, the SOAP messages should not include the certificates but URL references to them.

Key management issue from privacy point of view ... (see Client Profiles document)

- Discovery:
The identity service should not be listed in the discovery service as the client cannot act in the role of a standard WSP being without an IP address or having reachable, associated metadata. (see Client Profiles document)

5.5.3.3. LUAD-WSC:

There may be use cases were Group System Mobile (GSM) authentication information may need to be exchanged using the SOAP Authentication protocol. Currently, the Simple Authentication and Security Layer (SASL) registry does not hold such a mechanism and therefore it would need to be added. Procedures for registering SASL mechanisms are given in RFC2222. Schedules for specifying the mechanism would be tied to Internet Engineering Task Force (IETF) timelines. Alternatively, GSS API can be used.

5.5.3.4. Interaction Service:

As mobile operators have (1) a number of established, reliable channels of communication with end-users such as Secure Messaging System (SMS) or Wireless Access Protocol (WAP) push, (2) the trust relationship with both the service providers and end-users, and (3) would like to provide a consistent user experience, it is recommended that an operator host an interaction service registered as an end-user service. Deployment of an interaction service should specify the possible communication channel interfaces with the network. For mobile operators, these might include SMS, WAP push, or Interactive Voice Response (IVR).

A key benefit of the Liberty technology for end-users is the ease-of-use when using Liberty-enabled services particularly in the case of mobile devices, having limited display and input capabilities. To reinforce this ease-of-use, it is recommended that mobile operators promote, where possible, a consistent user experience when interacting with end-users across service categories. For example, in the service category of secure, mobile transactions, Mobile electronic Transactions (MeT) Ltd. have developed specifications establishing a framework, ensuring a consistent user experience independent of device, service and network experience.

5.5.4. E-Government

Government authentication systems have all of the complexities of enterprise and general ecommerce authentication systems with the added responsibilities that a government has in protecting core citizen identity and attributes from unauthorized access or use. National government authentication systems should strive for interoperability with regional and provincial systems so that citizens can have the ability to reuse identification credentials. Often as the repository of basic identity information, government authentication and attribute sharing systems should utilize greater security than general e-commerce authentication and attribute sharing systems.

5.6. Special Issues

Not all implementation issues fall neatly within the categories identified above. Some issues exist with items not within the scope of the Liberty specifications such as the underlying Internet based protocols. Some issues deal with the Liberty enabled tools utilized by the Liberty ID-WSF specifications. Yet other issues arise from the use of certain development environments and tools. Each of these is dealt with below.

5.6.1. Underlying Protocols

The Liberty architecture has utilized standards based protocols where possible. Some of these protocols are under active development and revision. This circumstance has created challenges for implementers of the Liberty architecture. Likewise, resolution of conflicts among similarly named protocol components has created certain challenges.

A number of implementors have been challenged by maintaining proper major and minor version numbers depending on whether certain assertions are “pure SAML” (version 1.1) or Liberty adapted SAML assertions (version 1.2). This is especially troublesome where a response to an Liberty adapted SAML assertion utilizes a “pure SAML” assertion.
At least one development team encountered interoperability issues by not sufficiently canonicalizing their XML schemas.

### 5.6.2. Privacy and Security

Implementors should be familiar with the ID-WSF Security and Privacy Overview.

Liberty specifications require that all communications from Principals to Liberty-enabled sites be integrity protected and confidentiality must be ensured. Liberty-enabled sites must use SSL 3.0 or TLS 1.0 for conducting communications with Principals. The security of the SSL or TLS session depends upon the chosen ciphersuite; Liberty specifications recommend the use of at least a 112-bit symmetric key. Use of TLS should be preferred and non-use can lead to operational security issues.

If there are no intermediaries in the message path, then transport layer protection mechanisms (SSL/TLS) suffice to ensure the integrity and confidentiality of the message exchange. If there are intermediaries in the message path, then the content of `<S:Body>` must be encrypted using the confidentiality mechanisms in [WSScore]. Information supplied by a TA may contain private information and thus the TA and ultimate recipient must use the mechanisms of Encrypted Name Identifier and Encrypted URI.

If there are no intermediaries in the message path, then peer authentication can use SSL/TLS mutual authentication as outlined in section 6.2 of [LibertyID-WSFSecurityMechanisms]. In the presence of active intermediaries, Web Services Security SOAP Message Security, X.509 token profile sender authentication or Web Services Security SOAP Message Security, SAML token profile sender authentication must be used.

Trusted Authorities (TA) may issue assertions that will be subsequently used in conjunction with accessing a resource at an identity service. TAs must enforce any access control policies pertaining to the resource and the assertion must be by the TA.

Before authorization data can be consumed, the sender must authenticate itself to the recipient and the recipient must authenticate the sender, including checking the sender’s certificate is still valid (e.g., has not been revoked). The recipient must locate the security token and verify that it is properly structured, that the signature is valid, etc.

Generally when there is risk to a principal of release of personal or financial information, stronger security mechanisms should be preferred where practicable.

<table>
<thead>
<tr>
<th>Liberty Service</th>
<th>Liberty Protocol</th>
<th>Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery Service</td>
<td>QueryResponse</td>
<td>Responders should construct a response to be as qualified as possible. The Discovery Service provider should provide security tokens if it knows that these tokens will be necessary and it is able to provide them based on the security token included in the request. The ResourceID must be sent encrypted using a key encrypted with the public key of the resource provider. This encrypted key must exhibit nonce-like capabilities.</td>
</tr>
<tr>
<td>Discovery Service</td>
<td>Modify</td>
<td>Access control policy for the resource offering may be placed in the any element of the ResourceOffering attribute. If the AuthorizeRequester directive is specified for a resource, then the discovery service provider should include a SAML assertion containing a Resource Access Statement in any future QueryResponse for the resource. If the AuthenticateSessionContext directive is specified for a resource, then the discovery service provider should include a SAML assertion in the Session Context Statement in any future QueryResponse.</td>
</tr>
<tr>
<td>Interaction Service</td>
<td>InteractionRequest</td>
<td>InteractionRequest, if the attribute ds:KeyInfo is present, the attribute signed must also be present. If the response is be signed (that is, the “signed” attribute is present), the InteractionRequest should contain only a single query. The Inquiry element Id component lays out the importance of its nonce like properties. If the InteractionResponse contains a signed InteractionStatement, the recipient must verify the signature and also that the id attribute of the signed inquiry matches the id of the corresponding request inquiry. The response must be discarded if the signature cannot be verified.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Interaction Service</td>
<td>InteractionResponse</td>
<td>If the InteractionRequest requests signing, then the recipient should attempt to obtain an signed InteractionStatement from the Principal. If the value of the signed attribute is “strict,” then the InteractionResponse must include either an InteractionStatement or a status element with its code attribute set to is: notSigned. The Interaction Service should authenticate the Principal and save the proof of authentication. To prove that the information provided was provided by the Principal, the Interaction Service could have the Principal sign the response with the private key for which the requester (the WSC) has the corresponding public key.</td>
</tr>
<tr>
<td>Metadata</td>
<td>Publication of metadata</td>
<td>Metadata should always be transported securely, e.g., via SSL/TLS. Entities should publish their metadata document location via a “well-known location” or through DNS. DNS signatures and TLS Server authentication are recommended, and the use of Metadata ds:signature is strongly recommended. Express document expiration at the EntityDescriptor level only and not on the child nodes.</td>
</tr>
<tr>
<td>Metadata</td>
<td>Consumption of metadata</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>Relying parties should process the SSL/TLS certificate presented by the server using normal validation processes. The relying party should validate the various signatures including those from the zone in which metadata location URI was resolved (as described in DNSSEC) and from the metadata document itself (especially important in the case of local caching of the document). Consumers of metadata documents should observe the validUntil and cacheDuration of documents, and must use the most restrictive of these if they conflict.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5.6.3. Development Environments

A number of web services development environments contain support documentation that may assist an implementor in the proper utilization of the various web services related protocols used within the Liberty guidelines.
6 Example Sessions

This section walks through the complete messages passed from and to a client invoking a service. At each step, the complete SOAP message is included, headers and all. Note that the security tokens passed will not be verifiable (the signatures are fake) as these are only example messages.

In this example, a digital media adaptor device is used to present the user with both radio and photo services in their entertainment center. The steps taken here are but one example of performing the tasks – there are several other ways to accomplish the same task that might be more appropriate in different circumstances. This is just one example.

In this example, the device has previously been associated with a user account so the user does not need to perform any authentication/registration process.

6.1 Device boot up

The user turns on the device which brings up the main screen for the user. There are several areas on this screen that require user specific content (such as the “now playing” area for radio, or a “what’s new” area for data in their photo service.

6.2 Device initiates authentication

Needing user content the device initiates a device authentication with the authentication server. This request is submitted to "https://auth.ws.aol.com" (the bootstrap entry point for the authentication service).

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/
    xmlns:aol=" http://schemas.corp.aol.com/
    xmlns:lib="urn:liberty:wsf:soap-bind:1.0"
    xmlns:auth="urn:liberty:wsf:soap-auth:1.0"
>
    <S:Header>
        <lib:Correlation S:mustUnderstand="1"
            messageID="uuid:0023923-28329023-238239023"
            timestamp="2003-06-06T12:10:10Z" />
    </S:Header>

    <S:Body>
        <auth:SASLRequest advisoryAuthnID="123456789012:10023923"
            mechanism="CRAM-MD5"/>
    </S:Body>
</S:Envelope>
```

6.3 Auth Server responds with auth mechanism choice

The authentication server responds, choosing to use CRAM-MD5 as the authentication method and providing the challenge data.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/
    xmlns:aol=" http://schemas.corp.aol.com/
    xmlns:sa="urn:liberty:wsf:soap-auth:1.0"
    xmlns:sb="urn:liberty:wsf:soap-bind:1.0"
    xmlns:sa="urn:liberty:wsf:soap-auth:1.0" >
    <S:Header>
        <sa:SASLResponse serverMechanism="CRAM-MD5">
            <Status code="continue" />
            <Data>
                1896.697170952@postoffice.example.net
            </Data>
        </sa:SASLResponse>
    </S:Body>
</S:Envelope>
```

Notes:
- The “refToMessageID” field is set to the message ID in the Auth Request from the client.
- The value inside of the `<SASLResponse>` is the CRAM-MD5 challenge value for the client. In this case, it is a value directly out of Error! Reference source not found. to make it easier to see how the specification is incorporated into this protocol.
- The code of “continue” indicates that this is a continuing authentication operation.

6.4 Device submits credentials to Auth Server

The device prepares the MD5 digest using the provided challenge as well as the shared secret (in this case “tanstaafstanstaaf”) and sends a response to the Authentication Server. Please review Error! Reference source not found. for details on how the digest is generated (this particular value was actually lifted from the document).

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/
    xmlns:aol=" http://schemas.corp.aol.com/soap/sif-2004-02"
    xmlns:sa="urn:liberty:wsf:soap-auth:1.0"
    xmlns:sa="urn:liberty:wsf:soap-auth:1.0" >
    <S:Header>
        <sa:SASLRequest authzID="123456789012:10023923"
            mechanism="CRAM-MD5">
            dGltIGI5MTNhNjAyYzdlZGE3YTQ5NWI0ZTZlNzMzNGQzODkw
        </sa:SASLRequest>
    </S:Body>
</S:Envelope>
```
Notes:

- The authzID was included in this request even though it was included on the original request. This is mostly for clarity since the server must be able to reconnect the authentication request with a previous response (since it needs to correlate this response to the challenge data).

- The refToMessageID ties this request to the previous response so that the authentication server can correlate this message to the challenge it sent in the previous message.

### 6.5 Auth Server returns Security Token & Discovery Info

The server processes the request and returns the security token to the caller along with the bootstrap information for accessing the discovery service.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aol="http://schemas.corp.aol.com/soap/sif-2004-02"
xmlns:disco="urn:liberty:disco:2003-08"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:sa="urn:liberty:wsf:soap-auth:1.0"  >
  <S:Header>
    <sb:Correlation S:mustUnderstand="1"
messageID="uuid:00287-23928392-193482390"
refToMessageID="uuid:0023923-28329023-238239026"
timestamp="2003-06-06T12:10:11Z"  />
  </S:Header>
  <S:Body>
    <sa:SASLResponse>
      <sa:Status code="success" />
      <disco:ResourceOffering>
        <disco:ServiceInstance>
          <disco:ServiceType>urn:liberty:disco:2003-08</disco:ServiceType>
          <disco:ProviderID>http://discovery.aol.com</disco:ProviderID>
          <disco:Description CredentialRef="e06e5a28-bc80-4ba6-9ecb-712949db686e">
            <disco:Endpoint>https://discovery.ws.aol.com</disco:Endpoint>
          </disco:Description>
        </disco:ServiceInstance>
        <sa:Credentials>
          <saml:Assertion MajorVersion="1" MinorVersion="1"
          AssertionID="e06e5a28-bc80-4ba6-9ecb-712949db686e"
          Issuer="http://idp.aol.com"
          IssueInstant="2003-06-06T12:10:11Z"
          InResponseTo="uuid:0023923-23829023-238239026">
            <saml:Conditions NotOnOrAfter="2003-06-07T12:10:10Z" >
              <saml:Audience RestrictionCondition>
                <saml:Audience>http://discovery.aol.com</saml:Audience>
              </saml:AudienceRestrictionCondition>
            </saml:Conditions>
          </saml:Assertion>
        </sa:Credentials>
      </disco:ResourceOffering>
    </sa:SASLResponse>
  </S:Body>
</S:Envelope>
```
6.6 Device Requests Service Info from Discovery Service

The device now submits a request to the Discovery Service (at the entry point returned in the previous message “https://discovery.ws.aol.com”) – Note that this address could change on a user by user, call by call basis, so the client MUST retrieve the correct value from the message returned during the authentication process) for information about the radio service.

<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aol="http://schemas.corp.aol.com/soap/sif-2004-02"
xmlns:disco="urn:liberty:disco:2003-08"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:sa="urn:liberty:wsf:soap-auth:1.0" xmlns:sb="urn:liberty:wsf:soap-bind:1.0">
  <S:Header>
    <s:Security>
      <wsse:SecurityToken Service="http://discovery.aol.com"/>
    </s:Security>
    <s:Correlation S:mustUnderstand="1" messageID="uuid:0023923-28329328-2378404578"
timestamp="2003-06-06T12:10:12Z"/>
  </S:Header>
  <S:Body>
                                 SessionIndex="1">
      <lib:AuthnContext>
        <lib:AuthnContextClassRef></lib:AuthnContextClassRef>
      </lib:AuthnContext>
    </lib:AuthenticationStatement>
  </S:Body>
</S:Envelope>

Notes:

- There are 2 key pieces of information in this message: the discovery service resource offering and the authentication assertion to be used at that service.
618  http://schemas.corp.aol.com/authctx/classes/DeviceProtectedTransport
619  </lib:AuthnContextClassRef>
620  </lib:AuthnContext>
621  <saml:Subject>
622    <saml:NameIdentifier>
623      <saml:NameQualifier>http://aol.com</saml:NameQualifier>
625      AOLScreenname
626    </saml:NameIdentifier>
627    <saml:SubjectConfirmation>
628      <saml:ConfirmationMethod>
629        urn:oasis:names:tc:SAML:1.0:cm:Bearer
630      </saml:ConfirmationMethod>
631    </saml:SubjectConfirmation>
632  </saml:Subject>
633  </saml:AuthenticationStatement>
634  </saml:AttributeStatement>
635  <saml:Subject>
636    <saml:NameIdentifier>
637      <saml:NameQualifier>http://aol.com</saml:NameQualifier>
639      AOLScreenname
640    </saml:NameIdentifier>
641  </saml:Subject>
642  <saml:Attribute AttributeName="devUPC"
643    AttributeNamespace="http://schemas.corp.aol.com">
644    <saml:AttributeValue>123456789012</saml:AttributeValue>
645  </saml:Attribute>
646  </saml:AttributeStatement>
647  </ds:Signature>
648  Signature data goes here
649  </ds:Signature>
650  </saml:Assertion>
651  </wsse:Security>
652  </S:Header>
653  <S:Body>
654    <disco:Query>
655      <ResourceID> urn:liberty:isf:implied-resource</ResourceID>
656      <RequestedServiceType>
657        <ServiceType>urn:aol-com:services:radio</ServiceType>
658      </RequestedServiceType>
659    </disco:Query>
660  </S:Body>
661  </S:Envelope>

Notes:

• The Assertion returned from the authentication process is included in the <ws:Security> header in the
  message.

• There is no “refToMessageID” in the <Correlation> header because this message is the first message in
  the communication with the Discovery Service.

6.7 Discovery Service returns Service Info

The Discover Service processes the request and responds to the client with the radio server resource offering, the
necessary credentials for the radio server, and a session context for subsequent calls to the discovery service.

<?xml version="1.0" encoding="utf-8" ?>
<S:Envelopem
667  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
668  xmlns:aol="http://schemas.corp.aol.com/soap/isf-2004-02"
669  xmlns:disco="urn:liberty:disco:2003-08"
670  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
671  xmlns:sb="urn:liberty:wsf:soap-bind:1.0" >
672  <S:Header>
673    <sb:Correlation S:mustUnderstand="1"
674      messageID="uuid:00287-23234564-098098798"
675      refToMessageID="uuid:0023923-28329328-23789404578"
<sb:ServiceInstanceUpdate mustUnderstand="1">
  <sec:SecurityMechID>
    urn:liberty:security:2003-08:TLS:Bearer
  </sec:SecurityMechID>
  <Credential NotOnOrAfter="2003-06-06T09:30Z">
    <wsse:BinarySecurityToken wsu:Id="..."
      ValueType="anyPrefix:ServiceSessionContext">
      A233adfojwe8ldhweoidfdlsjdw (Base 64 Encoded Data)
    </wsse:BinarySecurityToken>
  </Credential>
</sb:ServiceInstanceUpdate>
</S:Header>
</S:Body>
<disco:QueryResponse>
  <Status code="OK" />
  <disco:ResourceOffering EntryID="1">
    <disco:ServiceInstance>
      <disco:ServiceType>urn:aol-com:services:radio</disco:ServiceType>
      <disco:ProviderID>http://radio.ws.aol.com/</disco:ProviderID>
      <disco:Description CredentialRef="9f3d54a0-4899-8a3d-9328-328ad3e4ef90">
        <SecurityMechID>
        </SecurityMechID>
        <Endpoint>https://radio.ws.aol.com/</Endpoint>
      </disco:Description>
    </disco:ServiceInstance>
  </disco:ResourceOffering>
  <disco:Credentials>
    <saml:Assertion MajorVersion="1" MinorVersion="1"
      AssertionID="9f3d54a0-4899-8a3d-9328-328ad3e4ef90"
      Issuer="http://idp.aol.com"
      IssueInstant="2003-06-06T12:10:11Z"
      InResponseTo="uuid:0023923-28329023-238239026">
      <saml:Conditions NotOnOrAfter="2003-06-07T12:10:10Z" >
        <saml:AudienceRestrictionCondition>
          <saml:Audience>http://radio.ws.aol.com</saml:Audience>
        </saml:AudienceRestrictionCondition>
      </saml:Conditions>
      <lib:AuthenticationStatement
        AuthenticationInstant="2003-06-06:12:10:11Z"
        SessionIndex="1" >
        <lib:AuthnContext>
          <lib:AuthnContextClassRef>
            http://schemas.corp.aol.com/authctx/classes/DeviceProtectedTransport
          </lib:AuthnContextClassRef>
        </lib:AuthnContext>
      </lib:AuthenticationStatement>
    </saml:Assertion>
  </disco:Credentials>
</disco:QueryResponse>
6.8 Device Requests data from Radio Service

The device, having the contact information and credentials for the Radio service, submit a service request to the Radio server (to the Endpoint identified in the Resource Offering: “https://radio.ws.aol.com”).

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:aol="http://schemas.corp.aol.com/soap/sif-2004-02"
    xmlns:disco="urn:liberty:disco:2003-08"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:sb="urn:liberty:wsf:soap-bind:1.0" >
    <S:Header>
        <sb:Correlation S:mustUnderstand="1"
            messageID="uuid:9897923-82398723-092739723"
            timestamp="2003-06-06T12:10:16Z" />
        <wsse:Security>
            <saml:Assertion MajorVersion="1" MinorVersion="1"
                AssertionID="9f3d54a0-4899-8a3d-9328-328ad3e4ef90"
                Issuer="http://idp.aol.com"
                IssueInstant="2003-06-06T12:10:11Z"
                InResponseTo="uuid:0023923-28329023-238239026" >
                <saml:Conditions NotOnOrAfter="2003-06-07T12:10:10Z" >
                    <saml:AudienceRestrictionCondition>
                        <saml:Audience>http://radio.ws.aol.com</saml:Audience>
                    </saml:AudienceRestrictionCondition>
                </saml:Conditions>
                <lib:AuthenticationStatement
                    AuthenticationInstant="2003-06-06:12:10:11Z"
                    SessionIndex="1" >
                    <lib:AuthnContext>
                        <lib:AuthnContextClassRef>
                            http://schemas.corp.aol.com/authctx/classes/DeviceProtectedTransport
                        </lib:AuthnContextClassRef>
                    </lib:AuthnContext>
                </lib:AuthenticationStatement>
            </saml:Assertion>
        </wsse:Security>
    </S:Header>
</S:Envelope>
```
Notes:

- The authentication assertion returned with the Discovery Service response is included in the request to the Radio Service to identify the user.

6.9 Radio Service returns Info

The Radio Service processes the request and returns the list of stations to the client.

Notes:

- The Radio Service returned a session context for the client for use on subsequent requests.
- The NotOnOrAfter attribute on the credential was set to the same expiration time as the assertion which initiated the session.
- The Radio Service told the client to submit subsequent requests to a new server ("https://Radio15.ws.aol.com/").
6.10 Device Requests additional info from Radio

The Device now needs the detailed station info for one of the stations returned in the previous. This time, because of the `<ServiceSessionContext>` returned in the previous call, the request is submitted to: “https://Radio15.ws.aol.com” and the Assertion is not needed on the request.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope
 xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:aol="http://schemas.corp.aol.com/soap/sif-2004-02"
 xmlns:wsse="urn:liberty:wsf:soap-bind:1.0">
  <S:Header>
    <sb:Correlation S:mustUnderstand="1"
      messageID="uuid:23409723497-20972347-23407234" refToMessageID="uuid:23452-7345097234-0974234097"
      timestamp="2003-06-06T12:10:16Z" />
    <wsse:Security>
      <wsse:BinarySecurityToken wsu:Id="..."
        ValueType="anyPrefix:ServiceSessionContext">A233asdjwe8lwejisd8sadj2weqj9ajd2h2h2h72zdxb2esad</wsse:BinarySecurityToken>
    </wsse:Security>
  </S:Header>
  <S:Body>
    // Get Station Detail command
  </S:Body>
</S:Envelope>
```

Notes:
- Because the `<wsse:BinarySecurityToken>` was included, the assertion is not necessary.
- The “refToMessageID” attribute is set to the message id of the previous response message from the radio server.

6.11 Radio Service returns info

The Radio Service processes the request and returns the detailed station info.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope
 xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:aol="http://schemas.corp.aol.com/soap/sif-2004-02"
 xmlns:urn:liberty:wsf:soap-bind:1.0">
  <S:Header>
    <sb:Correlation S:mustUnderstand="1"
      messageID="uuid:23568989-07123493294-23723" refToMessageID="uuid:23409723497-20972347-23407234"
      timestamp="2003-06-06T12:10:16Z" />
  </S:Header>
  <S:Body>
    // Station Details
  </S:Body>
</S:Envelope>
```

Notes:
- The Radio Server did not return another `<ServiceSessionContext>` to the caller. This means the existing context is still valid and should be used on the next request.
6.12 Device Requests Photo Service Info from Discovery Service

The user selects the photo tab on the display and the device now needs to contact the photo service. So the device submits a discovery request to lookup the photo service contact information.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aol="http://schemas.corp.aol.com/soap/sif-2004-02"
xmlns:disco="urn:liberty:disco:2003-08"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:sb="urn:liberty:wsf:soap-bind:1.0"/>
<S:Header>
<sb:Correlation S:mustUnderstand="1"
messageID="uuid:09213802-230987987-238797234"
refToMessageID="uuid:00287-23234564-098098798"
timestamp="2003-06-06T18:29:18Z"/>
<wsse:Security>
<wsse:BinarySecurityToken wsu:Id="...
ValueType="anyPrefix:ServiceSessionContext">
A233adfjwe81ghweoiidfdjdwe (Base 64 Encoded Data)
<S:Body>
<disco:Query>
<disco:RequestedServiceType>
<disco:ServiceType>urn:aol-com:services:photo</disco:ServiceType>
</disco:RequestedServiceType>
</disco:Query>
</S:Body></S:Envelope>
```

Notes:
- The request included the session context returned from the Discovery Service in step 0 and does not include a Liberty assertion in the header.
- Since this is essentially a continuation of the conversation with the DS, we include the message ID of the last response from the DS in this request.

6.13 Discovery Service returns Photo Service info

The Discover Service processes the request and responds to the client with the radio server resource offering, the necessary credentials for the radio server, and a session context for subsequent calls to the discovery service.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aol="http://schemas.corp.aol.com/soap/sif-2004-02"
xmlns:disco="urn:liberty:disco:2003-08"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:sa="urn:liberty:wsf:soap-auth:1.0"/>
<S:Header>
<sb:Correlation S:mustUnderstand="1"
messageID="uuid:33489-8972323-89798237912"
refToMessageID="uuid:09213802-230987987-238797234"
timestamp="2003-06-06T18:29:18Z"/>
</S:Header>
<S:Body>
```

© Copyright 2002-2004 Liberty Alliance Project. All Rights Reserved
<disco:QueryResponse>
  <Status code="OK" />
  <disco:ResourceOffering EntryID="1">
    <disco:ServiceInstance>
      <disco:ServiceType>urn:aol-com:services:photo</disco:ServiceType>
      <disco:ProviderID>http://photo.ws.aol.com/</disco:ProviderID>
      <disco:Description CredentialRef="9fd3eda-b34a-9008-a334-3234dea90f5">
        <SecurityMechID>
        </SecurityMechID>
        <Endpoint>https://photo.ws.aol.com/</Endpoint>
      </disco:Description>
    </disco:ServiceInstance>
    <disco:Credentials>
      <saml:Assertion MajorVersion="1" MinorVersion="1"
        AssertionID="9fd3eda-b34a-9008-a334-3234dea90f5"
        Issuer="http://idp.aol.com"
        IssueInstant="2003-06-06T18:29:18Z"
        InResponseTo="uuid:0023923-28329023-238239026">
        <saml:Conditions NotOnOrAfter="2003-06-07T12:10:10Z"/>
        <saml:AudienceRestrictionCondition>
          <saml:Audience>http://photo.ws.aol.com</saml:Audience>
        </saml:AudienceRestrictionCondition>
      </saml:Assertion>
    </disco:Credentials>
  </disco:ResourceOffering>
</disco:QueryResponse>
6.1.14 Device requests info from Photo Service

The device requests a list of folders from the photo service.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope>
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:aol="http://schemas.corp.aol.com/soap/sif-2004-02"
  xmlns:disco="urn:liberty:disco:2003-08"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:sb="urn:liberty:wsf:soap-bind:1.0" >
  <S:Header>
    <sb:Correlation S:mustUnderstand="1"
        messageID="uuid:958312848-29348938-232342121"
        timestamp="2003-06-06T18:29:18Z" />
    <wsse:Security>
      <saml:Assertion MajorVersion="1" MinorVersion="1"
          IssueInstant="2003-06-06T18:29:18Z"
          InResponseTo="uuid:0023923-28329023-238239026">
        <saml:Conditions NotOnOrAfter="2003-06-07T12:10:10Z" />
        <saml:AudienceRestrictionCondition>
          <saml:Audience>http://photo.ws.aol.com</saml:Audience>
        </saml:AudienceRestrictionCondition>
      </saml:Assertion>
    </wsse:Security>
  </S:Header>
      SessionIndex="1" >
    <lib:AuthnContext>
      <lib:AuthnContextClassRef>
        http://schemas.corp.aol.com/authctx/classes/DeviceProtectedTransport
      </lib:AuthnContextClassRef>
    </lib:AuthnContext>
    <saml:Subject>
      <saml:NameIdentifier>
        <saml:NameQualifier>http://aol.com</saml:NameQualifier>
        AOLScreenname
      </saml:NameIdentifier>
      <saml:SubjectConfirmation>
        <saml:ConfirmationMethod>
          urn:oasis:names:tc:SAML:1.0:cm:Bearer
        </saml:ConfirmationMethod>
      </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Attribute AttributeName="devUPC"
        AttributeNamespace="http://schemas.corp.aol.com" >
      123456789012
    </saml:Attribute>
  </lib:AuthenticationStatement>
</S:Envelope>
```
Signature data goes here
</ds:Signature>
</saml:Assertion>
</wsse:Security>
</S:Header>
</S:Body>
// Photo Service Request
</S:Body>
</S:Envelope>

Notes:
- As this is the first request to the Photo Service, there is no “refToMessageID” included.
- The Assertion returned with the Discovery Service response is included in this message.

### 6.15 Photo service returns info

The Photo Service returns the requested information

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope>
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:aol="http://schemas.corp.aol.com/soap/sif-2004-02"
  xmlns:sb="urn:liberty:wsf:soap-bind:1.0" >
  <S:Header>
    <sb:Correlation S:mustUnderstand="1" messageID="uuid:23452-7345097234-0974234097"
      refToMessageID="uuid:958312848-29348938-232342121"
      timestamp="2003-06-06T12:10:16Z" />
    <sb:ServiceInstanceUpdate mustUnderstand="1">
      <sec:SecurityMechID>
        urn:liberty:security:2003-08:TLS:Bearer
      </sec:SecurityMechID>
      <Credential NotOnOrAfter="2003-06-07T12:10:10Z">
        <wsse:BinarySecurityToken wsu:Id="…"
          ValueType="anyPrefix:ServiceSessionContext">
          A233asdfjwe81wefjinds3ad3j2weqwe9eajh2h5d9h72zac2eas23ad
        </wsse:BinarySecurityToken>
      </Credential>
    </sb:ServiceInstanceUpdate>
  </S:Header>
  <S:Body>
    // Station List data included here
  </S:Body>
</S:Envelope>
```

Notes:
- As the Radio Service did, the Photo Service returns a `<ServiceInstanceUpdate>` to the caller. However, in this response the Photo Service does not redirect the user to a different SOAP Endpoint.

### 6.16 Device Renews Security Token

It is now almost 24 hours since the original authentication by the device and the device, being a good client, has monitored the validity period on the security token it received and so knows that it needs to perform a renewal of the token. This request is submitted to the authentication server (the same place where the original authentication took place).

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope>
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:aol="http://schemas.corp.aol.com/soap/sif-2004-02"
  xmlns:disco="urn:liberty:disco:2003-08"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:sb="urn:liberty:wsf:soap-bind:1.0"
  xmlns:ss="urn:liberty:wsf:soap-auth:1.0" >
  <S:Header>
    // Authentication Service Request
  </S:Header>
  <S:Body>
    // Station List data included here
  </S:Body>
</S:Envelope>
```
<sb:Correlation S:mustUnderstand="1"
messageID="uuid:234235-993209787-099087238923"
timestamp="2003-06-07T12:00:00Z" />
</wsse:Security>
<saml:Assertion MajorVersion="1" MinorVersion="1"
AssertionID="e06e5a28-bc80-4ba6-9ecb-712949db686e"
Issuer="http://ipd.aol.com"
IssueInstant="2003-06-06T12:10:11Z"
InResponseTo="uuid:0023923-28329023-238239026">
<saml:Conditions NotOnOrAfter="2003-06-07T12:10:12" >
<saml:AudienceRestrictionCondition>
<saml:Audience>http://discovery.aol.com</saml:Audience>
</saml:AudienceRestrictionCondition>
</saml:Conditions>
</saml:Assertion>
</wsse:Security>
</S:Header>
</S:Body>
</S:Envelope>

Notes:

- The previously returned security token is presented back to the authentication service.
- The “renewal” attribute is all that is needed on this authentication request.
6.17 The Authentication Server returns new token

The server processes the request and returns the renewed security token to the caller.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope>
  <S:Header>
    <sb:Correlation S:mustUnderstand="1"
      messageID="uuid:87432-79234723-072347893"
      refToMessageID="uuid:234235-993209787-099087238923"
      timestamp="2003-06-07T12:00:00Z" />
  </S:Header>
  <S:Body>
    <sa:SASLResponse>
      <sa:Status code="success" />
      <sa:Credential>
        <saml:Assertion MajorVersion="1" MinorVersion="1"
          AssertionID="9fe4357-df43-b902-9123-da8082fe7"
          Issuer="http://idp.aol.com"
          IssueInstant="2003-06-07T12:00:00Z"
          InResponseTo="uuid:234235-993209787-099087238923">
          <saml:Conditions NotOnOrAfter="2003-06-08T12:00:00Z" >
            <saml:AudienceRestrictionCondition>
              <saml:Audience>http://discovery.aol.com</saml:Audience>
            </saml:AudienceRestrictionCondition>
          </saml:Conditions>
          <lib:AuthenticationStatement
            AuthenticationInstant="2003-06-06T12:10:11Z"
            SessionIndex="1" >
            <lib:AuthnContext>
              <lib:AuthnContextClassRef>
                http://schemas.corp.aol.com/authctx/classes/DeviceProtectedTransport
              </lib:AuthnContextClassRef>
            </lib:AuthnContext>
            <saml:Subject>
              <saml:NameIdentifier>
                <saml:NameQualifier>http://aol.com</saml:NameQualifier>
                AOLScreenname
              </saml:NameIdentifier>
              <saml:SubjectConfirmation>
                <saml:ConfirmationMethod>
                  urn:oasis:names:tc:SAML:1.0:cm:Bearer
                </saml:ConfirmationMethod>
              </saml:SubjectConfirmation>
            </saml:Subject>
          </lib:AuthenticationStatement>
          <saml:AttributeStatement>
            <saml:Subject>
              <saml:NameIdentifier>
                <saml:NameQualifier>http://aol.com</saml:NameQualifier>
                AOLScreenname
              </saml:NameIdentifier>
              <saml:Attribute AttributeName="devUPC"
                AttributeNamespace="http://schemas.corp.aol.com">
                <saml:AttributeValue>123456789012</saml:AttributeValue>
              </saml:Attribute>
            </saml:Subject>
          </saml:AttributeStatement>
        </saml:Assertion>
      </sa:Credential>
    </sa:SASLResponse>
  </S:Body>
</S:Envelope>
```
<s:Envelope>
  <s:Body>
    <sa:SASLResponse>
      
      </sa:SASLResponse>
    </s:Body>
  </s:Envelope>

Notes:
- The discovery service bootstrap information is not included since it was sent previously.
- The renewed token still has the same "AuthenticationInstant" since this is a renewal not a re-authentication.
7. References

7.1. Normative


7.2. Informative
