Liberty ID-WSF Implementation Guide

DRAFT Version 1.0-12

Editor:

David Weitzel, Mitretek Systems

Contributors:

Conor Cahill, AOL Time Warner
John Kemp, Liberty Alliance
Yuzo Koga, NTT
Susan Landau, Sun
Andrew Lindsay-Stewart, Vodafone
Paul Madsen, Entrust
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 document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to use the document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative works of this Specification. Entities seeking permission to reproduce portions of this document for other uses must contact the Liberty Alliance to determine whether an appropriate license for such use is available.

Implementation of certain elements of this document may require licenses under third party intellectual property rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are not, and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights. **This Specification is provided "AS IS", and no participant in the Liberty Alliance makes any warranty of any kind, express or implied, including any implied warranties of merchantability, non-infringement of third party intellectual property rights, and fitness for a particular purpose.** 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.

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

info@projectliberty.org

57 # Revision History

Version	Date	Editor	Description
1.0-01	8-1-2003	David Weitzel	Basic document as anticipated at brainstorming session in Helsinki in May 2003
1.0-02	8-20	David Weitzel	Added input from July 2003 Helsinki Face to Face and private IOP. Fixed bugs 422 & 423
1.0-03	8-24	David Weitzel	Added in implementation information from the Liberty Security & Privacy Overview
1.0-04	9-12	David Weitzel	Added mobile issues. Added proposed edits. Added input from Santa Clara Face to Face and private interop. Edited incomplete sections. Fixed minor formatting.
1.0-05	11-12	David Weitzel	Fixed bug 508: acronyms, edits, and reference section
1.0-06	1-15-2004	David Weitzel	Added service example section based on AOL inputs
1.0-07	2-11-2004	David Weitzel Conor Cahill	Revised service example section
1.0-08	4-16-2004	David Weitzel Yuzo Koga Paul Madsen	Added sections based on NTT and Entrust inputs
1.0-09	4-23-2004	David Weitzel	Formatting & Update Copyright notice
1.0-10	7-10-2004	David Weitzel	Formatting and added hyperlinks for Liberty Alliance specs referenced in the text
1.0-11	12-14-2004	Darryl Champagne	Misc. Editorial

58

59

Contents

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-WSFSOAPBindings] 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.

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.

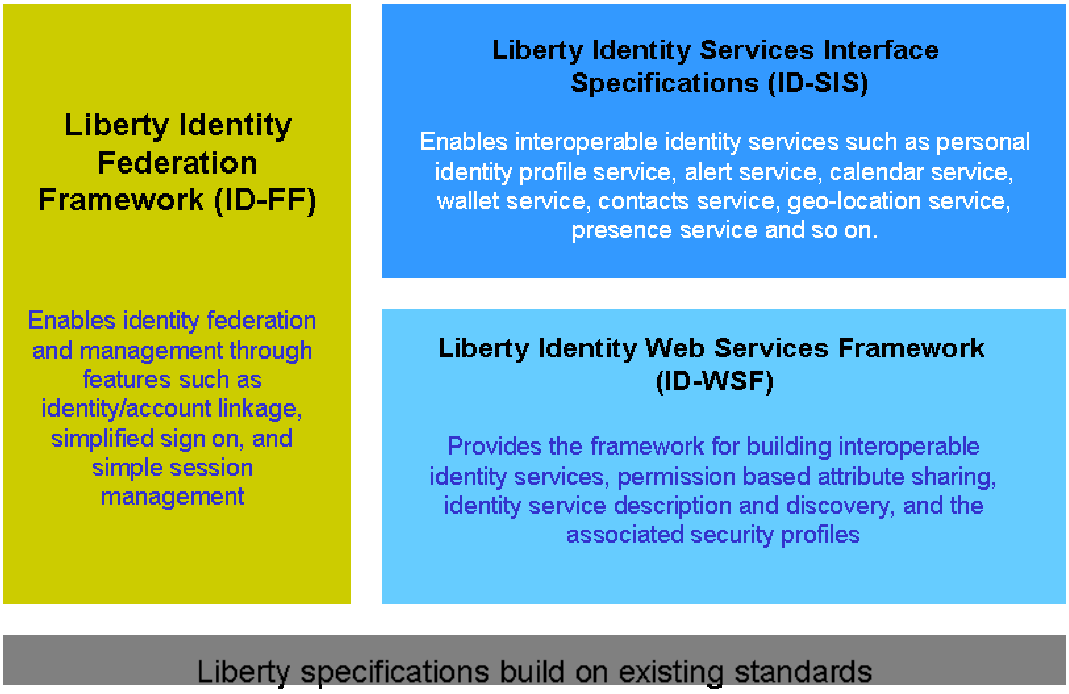


Figure 1: Liberty Architecture

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.

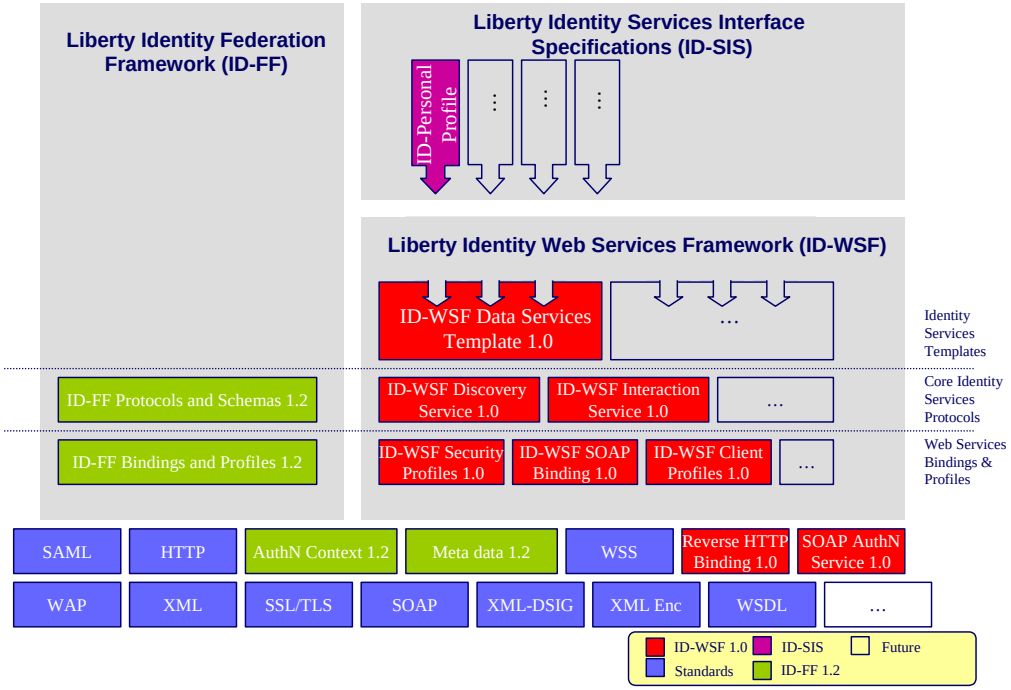


Figure 2: Liberty Modules

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.

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.

Key Environments

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

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.

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.

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.

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

246 e-government services, however, may be required to use the government's choice of Identity Provider(s). For this
247 reason, it is extremely important that governments choose e-authentication systems that appropriately protect both
248 privacy and confidentiality. For transparency's sake, systems that depend upon open standards provide a better choice
249 for government.

250

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.

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].

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-FFArchOverview] and [LibertyID-FFImplementationGuide].

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.

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

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

Interoperablility 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.

Data Services Template

An implementor should familiarize themselves with the specification check list provide in section 4 of the [LibertyID-WSFDataServiceTemplate] specification. Since identitiy 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.

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.

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.

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.

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.

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.

5.1.1.1. 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.

5.1.1.2. 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)

5.1.1.3. 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.

5.1.1.4. 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.

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.

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.

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.

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.

Liberty Service	Liberty Protocol	Recommendations
Discovery Service	QueryResponse	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.
Discovery Service	Modify	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. If there is a proxy resource offering and identity

		of the requester is not the identity of the provider of the proxy resource offering, the result set for that service type must contain only the proxy resource offering as well as all other resource offerings for which the requester is the provider. If the identity of the requester is the provider of the proxy resource offering, the result set must contain all resource offerings for the specified service type, including the proxy resource offering. Additionally, the directives for all instances of the requested service type must be aggregated when formulating the security tokens, as the proxying agent will need these tokens to fulfill the request.
Interaction Service	InteractionRequest	In the 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.
Interaction Service	InteractionResponse	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.
Metadata	Publication of metadata	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.

Metadata	Consumption of metadata	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.

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. Authentication Example Sessions

This document describes sample user experience and use-case of Liberty ID-WSF, which are simple and easy-to-understand. The user experience is described so that readers can intuitively understand what is Liberty ID-WSF, and what they can do with it, while the use-case is described with XML message traces so that implementers can refer for their implementation.

A more simplified version of the example is given in the Liberty *ID-WSF Overview* document.[1]

6.1 Overview

In the sample scenario, three websites appears, that are WhiteBroadBand.COM, BlueLiquor.COM, and YellowPizza.COM. Table 1 shows their roles in the scenario, and Figure 6.1 depicts overview of these three websites and their modules from the computational viewpoint.

Table 1 Three websites in the scenario

Abbr.	Website name	Explanation
IDP	WhiteBroadBand.COM	This is Identity Provide, and also hosts Discovery Service (DS).
SP1	BlueLiquor.COM	This is Service Provider that sells liquors on the Internet, and delivers them to customers. This website holds customer's attributes, (e.g. address information), and is able to share them with other websites based on Liberty ID-WSF and ID-SIS Personal Profile (i.e. it can behave as Attribute Provider).
SP2	YellowPizza.COM	This is Service Provider that sells pizzas on the Internet, and delivers them to customers. This website does not holds customer's attributes except for loginname and password, but is able to retrieve them from other websites based on Liberty ID-WSF and ID-SIS Personal Profile.

Liberty ID-WSF Sample User Experience and Use case

Sample Scenario

6.1.1.1. Assumptions

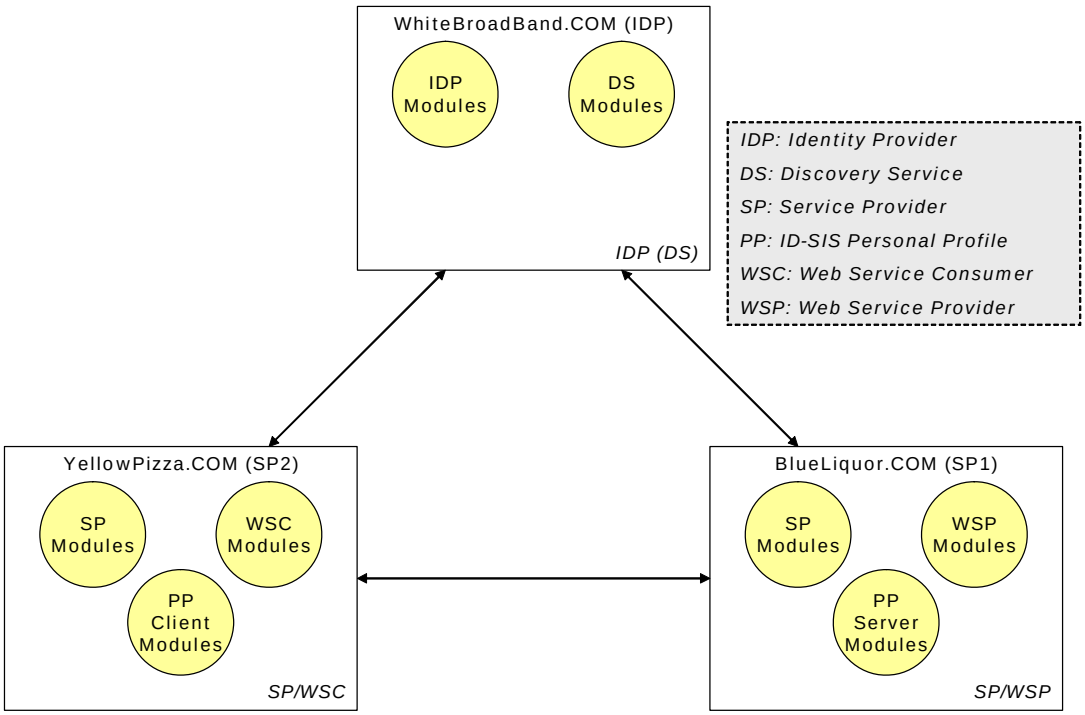
Joe Self (a Principal) has accounts at WhiteBroadBand.COM (IDP), BlueLiquor.COM (SP1), and YellowPizza.COM(SP2), and these are federated between them based on Liberty ID-FF. Joe Self's attributes are maintained at BlueLiquor.COM (SP1), and Blueliquor.COM can acts as Attribute Provider under the Liberty context.

6.1.1.2. Scenario

Joe Self orders liquors and pizzas on-line.
(01) He makes access to BlueLiquor.COM, and clicks a single sign-on link.
(02) He is redirected to WhiteBroadBand.COM, and authenticates with password
(03) He is redirected again to BlueLiquor.COM. BlueLiquor.COM gets SAML assertion from WhiteBroadBand.COM that states he has been authenticated, and responds to Joe Self with user-menu page.
(04) He orders some beers on-line, and they are delivered to the address where he has registered at BlueLiquor.COM.
(05) He requests BlueLiquor.COM to register its ResourceOffering to Discovery Service, so that his Personal Profile attribute at BlueLiquor.COM can be shared with other site.

[1] This example is provided by Liberty Alliance member NTT.

456 (06) BlueLiquor.COM sends Discovery Update message to Discovery Service.
457



458

459 Figure 6.1 Three websites and system modules on the scenario

460 (07) He subsequently makes access to YellowPizza.COM. Since he has been authenticated by
461 WhiteBroadBand.COM, YellowPizza.COM can get SAML assertion from WhiteBroadBand.COM, and responds
462 to Joe Self with user-menu page.
463 (08) He orders pizza on-line.
464 (09) He is asked by YellowPizza.COM where they deliver it.
465 (10) He requests YellowPizza.COM to get his Personal Profile attributes from other site.
466 (11) YellowPizza.COM sends Discovery Lookup request to Discovery Service, and gets ResourceOffering of
467 BlueLiquor.COM.
468 (12) YellowPizza.COM sends Query message to BlueLiquor.COM, and gets his Personal Profile attribute from
469 them.
470 (13) YellowPizza.COM delivers ordered pizza to the address where they got from BlueLiquor.COM.

471 **Sequence flows and exchanged messages**

472 ### 6.1.1.3. Sample sequence flows

473 Figure 6.2 shows sequence flows between entities, that realizes the sample scenario described in section 6.1.1.2.
474 In this figure, each Liberty specific flow (i.e. Liberty specific message exchange between entities) is numbered
475 sequentially.

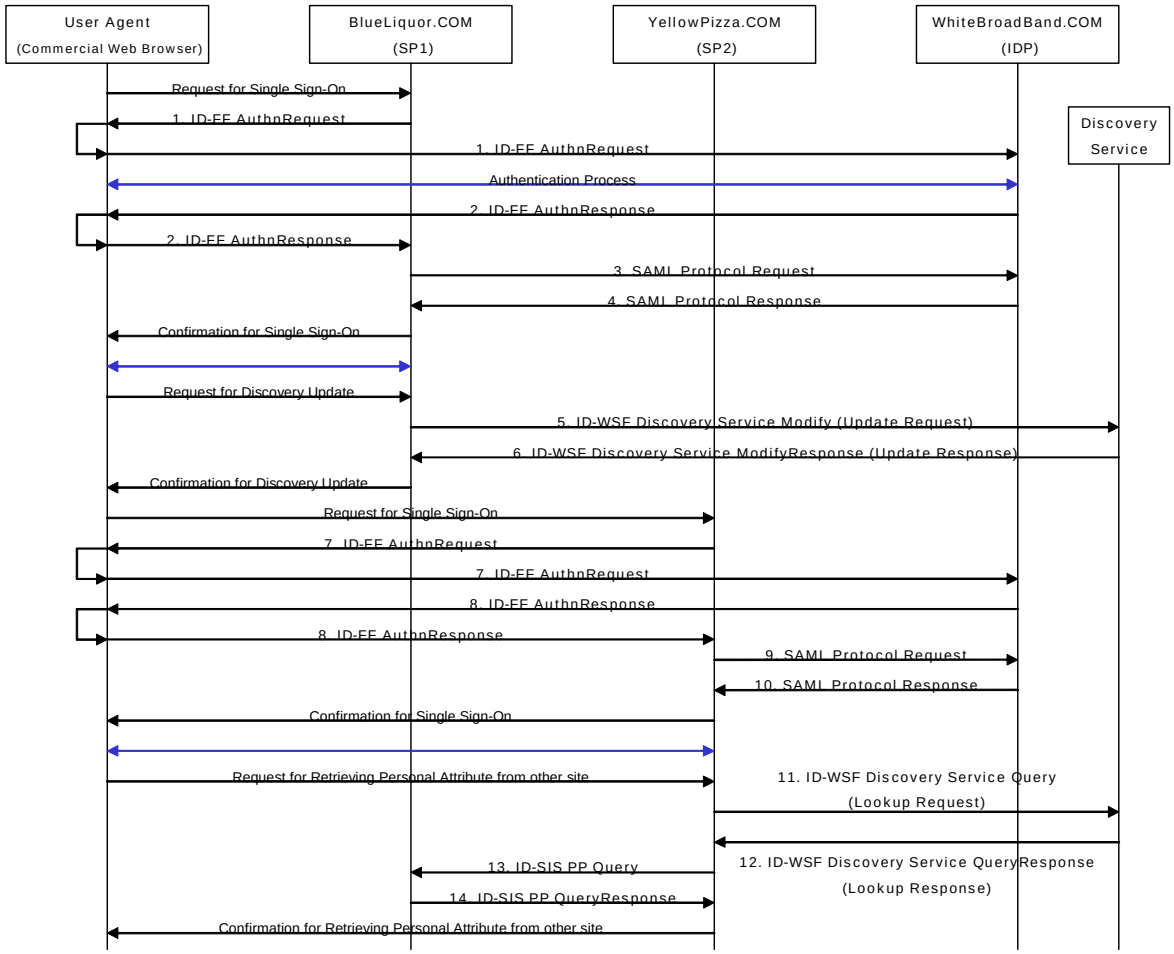


Figure 6.2 Sample Sequence Flow

6.1.1.4. Liberty Specific Messages Exchanged between Entities

In this section, each Liberty specific message in Figure 6.2, is explained with its sample XML trace.

6.1.1.4.1 1. ID-FF AuthnRequest

SP1 that has received single sign-on request from a Principal, and that confirms a session of the request has not been authenticated, subsequently sends ID-FF AuthnRequest to IDP using HTTP redirection. IDP that receives ID-FF AuthnRequest and that confirms the session of the request has not been authenticated, then authenticates a Principal (e.g. using loginname and password). Figure 6.3 shows an example of ID-FF AuthnRequest message. In this example, SP1 specifies to use the Browser/Artifact profile for single sign-on process.

```
https://whitebroadband.com:8443/idp/authn?RequestID=NTT2B3F4EEF8834E572B8A40E0A7A3AABBD&
MajorVersion=1&MinorVersion=2&consent=urn%3Aliberty%3Aconsent%3Aobtained&IssueInstant=2004-03-
10T05%3A57%3A08Z&ProviderID=https%3A%2F%2Fntt-a.liberty-
iop.org%3A8443%2Fsp1%2Fmetadata&NameIDPolicy=none&ForceAuthn=true&IsPassive=false&ProtocolProfil
e=http%3A%2F%2Fprojectliberty.org%2Fprofiles%2Fbrws-
art&RelayState=NTT77B9A190DF6F02C785E973386BC17C64
```

Figure 6.3 ID-FF AuthnRequest message sent from SP1 to IDP

6.1.1.4.2 2. ID-FF AuthnRequest

After authenticating a Principal, IDP sends ID-FF AuthnResponse to SP1 using HTTP redirection. Since SP1 specifies the Browser/Artifact profile in the AuthnRequest (sequence #1), an artifact is embedded in the AuthnResponse message. Figure 6.4 shows an example of ID-FF AuthnResponse message.

```
https://blueliquor.com:8443/sp1/asscon?SAMLart=AAPRT9itmuXxsqlPkKyrh3qQ6xW1gUtShydc%2FjJyrtzQ2UmMu%2BlCev3u
```

Figure 6.4 ID-FF AuthnResponse message sent from IDP to SP1

6.1.1.4.3 3. SAML Protocol Request

SP1 that has received ID-FF AuthnResponse, sends SAML Protocol Request message to IDP in order to get SAML assertion. In the message, an artifact that SP1 received with ID-FF AuthnResponse is embedded. Figure 6.5 shows an example of SAML Protocol Request message.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<samlp:Request IssueInstant="2004-03-10T05:57:16Z" MajorVersion="1" MinorVersion="0"
RequestID="NTTC9483587E959EE239CEFA5CF6B65C871"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
<ds:Reference URI="#NTTC9483587E959EE239CEFA5CF6B65C871">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<ds:DigestValue>3sNTt/sDyn2G5r6r4GLQAba/fT0=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
j5yODphPGGP0rhkJkXbYwN0ZfSChJ/4MZSie0jPpCNk4bzw+1WM7F2TuMc4AHAGTKqBqpmURqpW9
Qe77fNzuoQhBIl2z1KIoMYG/5c33Lxg2lz5Iy1hGzT0yj5Ns0EeU9o6wyJCX18z+pU4UV+TgDj4J
V+Jax2rGysYw7/uujwo=
</ds:SignatureValue>
</ds:Signature>
<samlp:AssertionArtifact>
AAPRT9itmuXxsqlPkKyrh3qQ6xW1gUtShydc/jJyrtzQ2UmMu+lCev3u
</samlp:AssertionArtifact>
</samlp:Request>
</soapenv:Body>
</soapenv:Envelope>
```

Figure 6.5 SAML Protocol Request message sent from SP1 to IDP

6.1.1.4.4 4. SAML Protocol Response

IDP that has received SAML Protocol Request, embeds SAML assertion that corresponds to specified artifact, and sends SAML Protocol Response to SP1. SP1 that receives the response, subsequently checks that SAML assertion, and consequently confirms that a Principal is authenticated by IDP.

Figure 6.6 shows an example of SAML Protocol Response message.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <samlp:Response InResponseTo="NTTC9483587E959EE239CEFA5CF6B65C871"
 IssueInstant="2004-03-10T05:57:20Z" MajorVersion="1" MinorVersion="0"
ResponseID="NTTA6D451D50F0D7303FAF2C3F38668DC76"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
   <samlp:Status>
    <samlp:StatusCode Value="samlp:Success"/>
   </samlp:Status>
   <lib:Assertion AssertionID="NTT7C39BA4B9CC702CD8D00E7BB3D195669"
InResponseTo="NTT2B3F4EEF8834E572B8A40E0A7A3AABBD"
IssueInstant="2004-03-10T05:57:15Z"
Issuer="https://whitebroadband.com:8443/idp/metadata"
MajorVersion="1" MinorVersion="2" xmlns:lib="urn:liberty:iff:2003-08">
    <saml:Conditions NotBefore="2004-03-10T05:57:15Z"
NotOnOrAfter="2004-03-11T15:00:00Z"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"/>
    <lib:AuthenticationStatement AuthenticationInstant="2004-03-10T05:57:15Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
     <lib:Subject>
      <saml:NameIdentifier Format="urn:liberty:iff:nameid:federated"
NameQualifier="https://blueliquor.com:8443/sp1/metadata"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
cd8cf101468a7744f07bb57c8bc49e41
</saml:NameIdentifier>
      <saml:SubjectConfirmation xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
       <saml:ConfirmationMethod>
urn:oasis:names:tc:SAML:1.0:cm:artifact
</saml:ConfirmationMethod>
      </saml:SubjectConfirmation>
      <lib:IDPProvidedNameIdentifier Format="urn:liberty:iff:nameid:federated"
NameQualifier="https://blueliquor.com:8443/sp1/metadata">
cd8cf101468a7744f07bb57c8bc49e41
</lib:IDPProvidedNameIdentifier>
     </lib:Subject>
    </lib:AuthenticationStatement>
    <saml:AttributeStatement xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
     <saml:Attribute AttributeName="DiscoveryResourceOffering"
AttributeNamespace="urn:liberty:disco:2003-08">
      <saml:AttributeValue>
       <disco:ResourceOffering xmlns:disco="urn:liberty:disco:2003-08">
        <disco:ResourceID>
https://whitebroadband.com:8443/idp/metadata/37e66f7afc918eb5c27b7b15fca55a01
</disco:ResourceID>
        <disco:ServiceInstance>
         <disco:ServiceType>urn:liberty:disco:2003-08</disco:ServiceType>
         <disco:ProviderID>https://whitebroadband.com:8443/idp/metadata</disco:ProviderID>
         <disco:Description>
          <disco:SecurityMechID>urn:liberty:security:2003-08:TLS:X509</disco:SecurityMechID>
          <disco:SecurityMechID>urn:liberty:security:2003-08:TLS:null</disco:SecurityMechID>
          <disco:Endpoint>https://whitebroadband.com:8443/idp/services/disco</disco:Endpoint>
         </disco:Description>
        </disco:ServiceInstance>
       </disco:ResourceOffering>
```

```
      </saml:AttributeValue>
     </saml:Attribute>
     <lib:Subject>
      <saml:NameIdentifier Format="urn:liberty:iff:nameid:federated"
NameQualifier="https://blueliquor.com:8443/sp1/metadata">
cd8cf101468a7744f07bb57c8bc49e41
</saml:NameIdentifier>
    </lib:Subject>
   </saml:AttributeStatement>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#NTT7C39BA4B9CC702CD8D00E7BB3D195669">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>+GpMzKtRAGzyP3qKe8W8vbU9ZMk=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
C99cr3ObEaRjSQL61uu7ObMjH0sK/5k+x0y9tysQa25q75eaGUguspAN4YQcG7oFR/yyunIC0Hsf
5boJSwj+1Re8yvOlianr1b4mEa4XcV8vATsCCGjjhSr1FjLxUCnSDzJsz0cWVW89EA0QStI58NmY
XeWv2xkk5p2a5Ut4940=
</ds:SignatureValue>
</ds:Signature>
</lib:Assertion>
  </samlp:Response>
</soapenv:Body>
</soapenv:Envelope>
```

Figure 6.6 SAML Protocol Response message sent from IDP to SP1

6.1.1.4.5 5. ID-WSF Discovery Service Modify (Discovery Update Request)

SP1 maintains Principal's attributes (e.g. address information) and is able to acts as Attribute Provider. By being requested by a Principal, SP1 registers its ResourceOffering to Discovery Service (DS). This process can be done by sending ID-WSF Discovery Service Modify message.

Figure 6.7 shows ID-WSF Discovery Service Modify message sent from SP1 to DS.

```
<soapenv:Envelope xmlns:soapenv=”http://schemas.xmlsoap.org/soap/envelope/”
xmlns:xsd=”http://www.w3.org/2001/XMLSchema”
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Header>
  <sb:Correlation id="NTT999DB7BE847E1693D8B90896D7BB481B"
messageID="uuid:f11b9e67-b855-0709-5e7e-f65f8b9ff9b1"
timestamp="2004-03-10T05:58:25Z"
soapenv:actor=”http://schemas.xmlsoap.org/soap/actor/next”
soapenv:mustUnderstand="1" xmlns:sb="urn:liberty:sb:2003-08"/>
  <sb:Provider providerID=”https://blueliquor.com:8443/sp1/metadata”
soapenv:mustUnderstand="0" xmlns:sb="urn:liberty:sb:2003-08"/>
  <wsse:Security soapenv:actor=”http://schemas.xmlsoap.org/soap/actor/next”
soapenv:mustUnderstand="1"
xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
<wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
soap-message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
 wsu:Id="X509Token" xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
MIICBDCCAW2gAwIBAgIBUTANBgkqhkiG9w0BAQQFADA1MQswCQYDVQQGEwJVUzEUMBIGA1UEChML
TGliZXJ0eSBJT1AxEDAOBgNVBAMTB1Rlc3QgQ0EwHhcNMDMxMjA1MTQ0MjI1WhcNMDQxMjA0MTQ0
MjI1WjBMMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPTGliZXJ0eSBJT1AgamtzMSMwIQYDVQQDExpu
dHQtYS1zaWduLmxpYmVydHktaW9wLm9yZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAq9Sl
+JvcHKNjjtE/v70TKMMXo+Ft05RBy/XUruHZsuH0b206MYG+CCBoPyJ6Jv+6+ZsCgIEjfJ6lqRZR
ZmPdGv92zcBHHO1/k1/GG7FPNFH+mrUm/66nRDysv6JFMzW7+Ct7132IUUiTEvj45CXHEcyy8JUf
fd1J1+FOrVweAEUCAwEAAaMNMAswCQYDVR0TBAIwADANBgkqhkiG9w0BAQQFAAOBgQBwqsW22HMT
LTcxn3jiifP+yBjKRaYpikrRzffeJ8XtLUrHCkm7ZOX/OeqidHAARB4lTxmITCB3LbHmViAk4G66
K4Yb9Y0FFVJCFyaYHnY6W6oLDkTv5IMqDL//vV6QF9boO2gvTpap4WL5+6meNmCyWKoeO4CuwX3q
ys5yrA8opg==
</wsse:BinarySecurityToken>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#NTT999DB7BE847E1693D8B90896D7BB481B">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>q2lVlJG2WV8mnpPeCTdY5SHj8FQ=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#NTTA68A82625412949E477FFB33ACF48560">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>C1fHwpAa/XR29boFKsmZuYbxoTk=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
V3vmszOBI3l7ZqP27PjikWfzqDVDew3DHPDuXJ98bkedG1GzPHjsvtpNvxD0SylhtijWSC6eemR2
JEJvQfEmGO5ScsjZURJcdyS6thbDWfsNhBhPv3nZtEX0zMkfvx1nNU3wd3QfsAGMHuxXhl7U8jAt
4/8A3nHupJ1dkefFqxg=
</ds:SignatureValue>
<ds:KeyInfo>
<wsse:SecurityTokenReference Usage="sec:MessageAuthentication"
xmlns:sec="urn:liberty:sec:2003-08">
<wsse:Reference URI="#X509Token"/>
</wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</soapenv:Header>
```

```
<soapenv:Body>
  <disco:Modify id="NTTA68A82625412949E477FFB33ACF48560" xmlns:disco="urn:liberty:disco:2003-08">
<disco:ResourceID>
https://whitebroadband.com:8443/idp/metadata/37e66f7afc918eb5c27b7b15fca55a01
</disco:ResourceID>
   <disco:InsertEntry>
    <disco:ResourceOffering>
     <disco:ResourceID>uuid:e427014e-1fde-cc03-85dd-690333bf695a</disco:ResourceID>
     <disco:ServiceInstance>
      <disco:ServiceType>urn:liberty:id-sis-pp:2003-08</disco:ServiceType>
      <disco:ProviderID>https://blueliquor.com:8443/sp1/metadata</disco:ProviderID>
      <disco:Description>
       <disco:SecurityMechID>urn:liberty:security:2003-08:TLS:X509</disco:SecurityMechID>
       <disco:SecurityMechID>urn:liberty:security:2003-08:TLS:null</disco:SecurityMechID>
       <disco:Endpoint>https://blueliquor.com:8443/sp1/services/idpp</disco:Endpoint>
      </disco:Description>
     </disco:ServiceInstance>
     <disco:Options>
      <disco:Option>urn:liberty:id-sis-pp:home</disco:Option>
      <disco:Option>urn:liberty:id-sis-pp:personal</disco:Option>
      <disco:Option>urn:liberty:id-sis-pp:cn</disco:Option>
      <disco:Option>urn:liberty:id-sis-pp:informalName</disco:Option>
      <disco:Option>urn:liberty:id-sis-pp:demographics</disco:Option>
     </disco:Options>
     <disco:Abstract>identity service for demostration</disco:Abstract>
    </disco:ResourceOffering>
   </disco:InsertEntry>
  </disco:Modify>
 </soapenv:Body>
</soapenv:Envelope>
```

Figure 6.7 Modify message sent from SP1 to DS

6.1.1.4.6 6. ID-WSF Discovery Service ModifyResponse (Discovery Update Response)

DS that has received ID-WSF Discovery Service Modify message, registers specified ResourceOffering, and responds to SP1 with ID-WSF Discovery Service Modify Response message.

Figure 6.8 shows ID-WSF Discovery Service ModifyResponse message.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Header>
  <sb:Provider id="NTT52DBE45FADF6384EFC295BDDA45C1CE0"
providerID="https://whitebroadband.com:8443/idp/metadata"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0" xmlns:sb="urn:liberty:sb:2003-08"/>
  <sb:Correlation id="NTT1E236AD1E4C6A098E03ABBB75DC43AE2"
messageID="uuid:1fa7c4d0-8e26-7819-b236-eb92eb6b4fc6"
refToMessageID="uuid:f11b9e67-b855-0709-5e7e-f65f8b9ff9b1"
timestamp="2004-03-10T05:58:26Z"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1" xmlns:sb="urn:liberty:sb:2003-08"/>
  <wsse:Security soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1" xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
<wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-
message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
wsu:Id="X509Token"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
MIICBDCCAW2gAwIBAgIBUTANBgkqhkiG9w0BAQQFADA1MQswCQYDVQQGEwJVUzEUMBIGA1UEChML
TGliZXJ0eSBJT1AxEDAOBgNVBAMTB1Rlc3QgQ0EwHhcNMDMxMjA1MTQ0MjI1WhcNMDQxMjA0MTQ0
MjI1WjBMMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPTGliZXJ0eSBJT1AgamtzMSMwIQYDVQQDExpu
dHQtYS1zaWduLmxpYmVydHktaW9wLm9yZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAq9Sl
+JvcHKNjjtE/v70TKMMXo+Ft05RBy/XUruHZsuH0b206MYG+CCBoPyJ6Jv+6+ZsCgIEjfJ6lqRZR
ZmPdGv92zcBHHO1/k1/GG7FPNFH+mrUm/66nRDysv6JFMzW7+Ct7132IUUiTEvj45CXHEcyy8JUf
fd1J1+FOrVweAEUCAwEAAaMNMAswCQYDVR0TBAIwADANBgkqhkiG9w0BAQQFAAOBgQBwqsW22HMT
LTcxn3jiifP+yBjKRaYpikrRzffeJ8XtLUrHCkm7ZOX/OeqidHAARB4lTxmITCB3LbHmViAk4G66
K4Yb9Y0FFVJCFyaYHnY6W6oLDkTv5IMqDL//vV6QF9boO2gvTpap4WL5+6meNmCyWKoeO4CuwX3q
ys5yrA8opg==
</wsse:BinarySecurityToken>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#NTT52DBE45FADF6384EFC295BDDA45C1CE0">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>hCYVH1FtuBCfIB1TxcDEBSXZNvw=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#NTT1E236AD1E4C6A098E03ABBB75DC43AE2">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>QW1gqgzPycEFNwOIP3cIPLXv+Pk=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#NTT18821653A7C16BEFF877DDC9A7D09B33">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>GSZWYAtwKaTR1FP+a8bxa7sGI6w=</ds:DigestValue>
```

```
<ds:SignatureValue>
JMlQLXFqtBVZyfo7TRGfqkFiPfNveU1X0yf+WwJqnHP1ADS3s7nBdW1SzKEwufZi4k+JNAy8E6Fk
Lh+nH0XDn9Pmw56DzAYNBK8JAZ2B7tGhntJwJHKLBZ3XgRXuH6A+wC7uvjTbu2ZQ9kcSY4EuWpxt
4tJYqQTqFYyRcVEAFLM=
</ds:SignatureValue>
<ds:KeyInfo>
<wsse:SecurityTokenReference Usage="sec:MessageAuthentication"
xmlns:sec="urn:liberty:sec:2003-08">
<wsse:Reference URI="#X509Token"/>
</wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</soapenv:Header>
 <soapenv:Body>
  <disco:ModifyResponse id="NTT18821653A7C16BEFF877DDC9A7D09B33"
newEntryIDs="uuid:1c1ccaeb-0c36-229b-d510-7ae33406ada4"
xmlns:disco="urn:liberty:disco:2003-08">
   <disco:Status code="disco:OK"/>
  </disco:ModifyResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

Figure 6.8 ID-WSF Discovery Service ModifyResponse message sent from DS to SP1

6.1.1.4.7 7. ID-FF AuthnRequest

SP2 that has received single sign-on request from a Principal, and that confirms a session of the request has not been authenticated, subsequently sends ID-FF AuthnRequest to IDP using HTTP redirection. IDP that receives ID-FF AuthnRequest and that confirms the session of the request has not been authenticated, then authenticates a Principal (e.g. using loginname and password). Figure 6.9shows an example of ID-FF AuthnRequest message. In this example, SP2 also specifies to use the Browser/Artifact profile for single sign-on process.

```
https://whitebroadband.com:8443/idp/authn?RequestID=NTTEC6D3DDAE91E0379423F1AD3B178C752&M
ajorVersion=1&MinorVersion=2&consent=urn%3Aliberty%3Aconsent%3Aobtained&IssueInstant=2004-03-
10T05%3A58%3A44Z&ProviderID=https%3A%2F%2Fntt-a.liberty-
iop.org%3A8443%2Fsp2%2Fmetadata&NameIDPolicy=none&ForceAuthn=true&IsPassive=false&ProtocolProfil
e=http%3A%2F%2Fprojectliberty.org%2Fprofiles%2Fbrws-
art&RelayState=NTTD4C48FF08D6098698A7EB5CE08BA9BB0
```

Figure 6.9 ID-FF AuthnRequest message sent from SP2 to IDP

6.1.1.4.8 8. ID-FF AuthnResponse

After confirming that a requested message's session has been authenticated, IDP sends ID-FF AuthnResponse to SP2 using HTTP redirection. Since SP2 specifies the Browser/Artifact profile in the AuthnRequest (sequence #1), an artifact is embedded in the AuthnResponse message. Figure 6.10 shows an example of ID-FF AuthnResponse message.

```
https://yellowpizza.com:8443/sp2/asscon?SAMLart=AAPRT9itmuXxsqlPkKyrh3qQ6xW1gc%2BR4UjUyHK
Nba6xUwkCIPVUUr34
```

Figure 6.10 ID-FF AuthnResponse message sent from IDP to SP2

6.1.1.4.9 9. SAML Protocol Request

SP2 that has received ID-FF AuthnResponse, sends SAML Protocol Request message to IDP in order to get SAML assertion. In the message, an artifact that SP2 received with ID-FF AuthnResponse is embedded. Figure 6.11 shows an example of SAML Protocol Request message.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<samlp:Request IssueInstant="2004-03-10T05:58:46Z" MajorVersion="1" MinorVersion="0"
RequestID="NTTB7CCE49363C5007F8CCC6277B217ED71"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
<ds:Reference URI="#NTTB7CCE49363C5007F8CCC6277B217ED71">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<ds:DigestValue>pw3BfCHpGQCf3w5DHelZsZLzbqY=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
Q9Kn95nnNU71taHA4X8HYY7kE02inEOW0yWRSpC2IZQPVt6zS30G+OPy5UO2lELsLwtNMqwyHBT9
OBY4k7HZNVCewVNwcGBslodKaOvV5neTSsiOgjZzvw+acrRha7qADCh0P5JAB3d0dRsy7f+odZlS
vll6/b7m6cAQA6rvLI4=
</ds:SignatureValue>
</ds:Signature>
<samlp:AssertionArtifact>
AAPRT9itmuXxsqlPkKyrh3qQ6xW1gc+R4UjUyHKNba6xUwkCIPVUUr34
</samlp:AssertionArtifact>
</samlp:Request>
</soapenv:Body>
</soapenv:Envelope>
```

Figure 6.11 SAML Protocol Request message sent from SP2 to IDP

6.1.1.4.10 10. SAML Protocol Response

IDP that has received SAML Protocol Request, embeds SAML assertion that corresponds to specified artifact, and sends SAML Protocol Response to SP2. SP2 that receives the response, subsequently checks that SAML assertion, and consequently confirms that a Principal is authenticated by IDP.

571

572

573 Figure 6.12 shows an example of SAML Protocol Response message.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <samlp:Response InResponseTo="NTTB7CCE49363C5007F8CCC6277B217ED71"
 IssueInstant="2004-03-10T05:58:48Z" MajorVersion="1" MinorVersion="0"
ResponseID="NTT53BF476E93C913C1CBEEB8A402C29EC7"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
   <samlp:Status>
    <samlp:StatusCode Value="samlp:Success"/>
   </samlp:Status>
   <lib:Assertion AssertionID="NTT3E0343B5B13442509112CDB32A81D461"
InResponseTo="NTTEC6D3DDAE91E0379423F1AD3B178C752"
IssueInstant="2004-03-10T05:58:46Z"
Issuer=https://whitebroadband.com:8443/idp/metadata
MajorVersion="1" MinorVersion="2" xmlns:lib="urn:liberty:iff:2003-08">
    <saml:Conditions NotBefore="2004-03-10T05:58:46Z"
NotOnOrAfter="2004-03-11T15:00:00Z"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"/>
    <lib:AuthenticationStatement AuthenticationInstant="2004-03-10T05:58:46Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
     <lib:Subject>
      <saml:NameIdentifier Format="urn:liberty:iff:nameid:federated"
NameQualifier=https://yellowpizza.com:8443/sp2/metadata
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
da275058804ee420d957623280d2f5f5
</saml:NameIdentifier>
      <saml:SubjectConfirmation xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
       <saml:ConfirmationMethod>
urn:oasis:names:tc:SAML:1.0:cm:artifact
</saml:ConfirmationMethod>
      </saml:SubjectConfirmation>
      <lib:IDPProvidedNameIdentifier Format="urn:liberty:iff:nameid:federated"
NameQualifier="https://yellowpizza.com:8443/sp2/metadata">
da275058804ee420d957623280d2f5f5
</lib:IDPProvidedNameIdentifier>
     </lib:Subject>
    </lib:AuthenticationStatement>
    <saml:AttributeStatement xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
     <saml:Attribute AttributeName="DiscoveryResourceOffering"
AttributeNamespace="urn:liberty:disco:2003-08">
      <saml:AttributeValue>
       <disco:ResourceOffering xmlns:disco="urn:liberty:disco:2003-08">
        <disco:ResourceID>
https://whitebroadband.com:8443/idp/metadata/37e66f7afc918eb5c27b7b15fca55a01
</disco:ResourceID>
        <disco:ServiceInstance>
         <disco:ServiceType>urn:liberty:disco:2003-08</disco:ServiceType>
         <disco:ProviderID>https://whitebroadband.com:8443/idp/metadata</disco:ProviderID>
         <disco:Description>
          <disco:SecurityMechID>urn:liberty:security:2003-08:TLS:X509</disco:SecurityMechID>
          <disco:SecurityMechID>urn:liberty:security:2003-08:TLS:null</disco:SecurityMechID>
          <disco:Endpoint>https://whitebroadband.com:8443/idp/services/disco</disco:Endpoint>
         </disco:Description>
        </disco:ServiceInstance>
       </disco:ResourceOffering>
      </saml:AttributeValue>
     </saml:Attribute>
```

```
      <lib:Subject>
       <saml:NameIdentifie r Format="urn:liberty:iff:nameid:federated"
    NameQualifier="https://yellowpizza.com:8443/sp2/metadata">
    da275058804ee420d957623280d2f5f5
    </saml:NameIdentifier>
      </lib:Subject>
     </saml:AttributeStatement>
     <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <ds:Reference URI="#NTT3E0343B5B13442509112CDB32A81D461">
    <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>UEWe8B8foW1ICZ68GoUl+Uz44Ws=</ds:DigestValue>
    </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
    oAb+mpZOVArfaMIA4+T6y23mW5A1OTHc4Ggqsnt41H8yCwiweRBa5WDNnpXkezRdu/s2n/iheZM7
    2uI2Z5mYjxkAV9FLGFISanLbHq5KoyVYgl0tiQaI/TCVysfZ9pYAuYVoB5Tu6EsUPDZ+C2zVnS9N
    gxwk64H+S4rnAvGrxcU=
    </ds:SignatureValue>
    </ds:Signature>
    </lib:Assertion>
      </samlp:Response>
    </soapenv:Body>
    </soapenv:Envelope>
```

Figure 6.12 SAML Protocol Response message sent from IDP to SP2

6.1.1.4.11 11. ID-WSF Discovery Service Query (Discovery Lookup Request)

SP2 does not maintain Principal's attributes. Therefore, by being requested by a Principal, SP2 tries to retrieve Principal's attributes from other websites. This process is realized by sending ID-WSF Query message to DS, and SP2 uses ResourceOffering of DS for sending the message, that it has received from IDP with ID-FF AuthnResponse (i.e. ResourceOffering of DS is embedded in the ID-FF AuthnResponse that is exchanged with sequence #10), and queries ResourceOfferings of other websites (i.e. Attribute Providers).

Figure 6.13 shows an example of ID-WSF Discovery Service Query message.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<sb:Correlation id="NTTAEB9DE0EB0B1A89B00797A14C6EE85F6"
messageID="uuid:debbffd3-4ea8-973e-5463-e5ecc2d95dde"
timestamp="2004-03-10T05:59:01Z"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1" xmlns:sb="urn:liberty:sb:2003-08"/>
<sb:Provider providerID="https://yellowpizza.com:8443/sp2/metadata"
soapenv:mustUnderstand="0" xmlns:sb="urn:liberty:sb:2003-08"/>
<wsse:Security soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1"
xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
```

```
<wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
wsu:Id="X509Token" xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
MIICBDCCAW2gAwIBAgIBUTANBgkqhkiG9w0BAQQFADA1MQswCQYDVQQGEwJVUzEUMBIGA1UEChML
TGliZXJ0eSBJT1AxEDAOBgNVBAMTB1Rlc3QgQ0EwHhcNMDMxMjA1MTQ0MjI1WhcNMDQxMjA0MTQ0
MjI1WjBMMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPTGliZXJ0eSBJT1AgamtzMSMwIQYDVQQDExpu
dHQtYS1zaWduLmxpYmVydHktaW9wLm9yZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAq9Sl
+JvcHKNjjtE/v70TKMMXo+Ft05RBy/XUruHZsuH0b206MYG+CCBoPyJ6Jv+6+ZsCgIEjfJ6lqRZR
ZmPdGv92zcBHHO1/k1/GG7FPNFH+mrUm/66nRDysv6JFMzW7+Ct7132IUUiTEvj45CXHEcyy8JUf
fd1J1+FOrVweAEUCAwEAAaMNMAswCQYDVR0TBAIwADANBgkqhkiG9w0BAQQFAAOBgQBwqsW22HMT
LTcxn3jiifP+yBjKRaYpikrRzffeJ8XtLUrHCkm7ZOX/OeqidHAARB4lTxmITCB3LbHmViAk4G66
K4Yb9Y0FFVJCFyaYHnY6W6oLDkTv5IMqDL//vV6QF9boO2gvTpap4WL5+6meNmCyWKoeO4CuwX3q
ys5yrA8opg==
</wsse:BinarySecurityToken>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#NTTAEB9DE0EB0B1A89B00797A14C6EE85F6">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>/vqZkvIo2MkbAntQ3j0+I0QsZ4k=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#NTT43EBDA48A7965082DA284C13DE33EFDE">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>js4Cmrbteuy9Epti94O9+xFj7yk=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
Rh9MenehPh/9zIB/8wNg4tCKaLIs5ayiRbfKrepXpD9qbsIOVjZ0/2R1ChiX/WaDANtVtdfj/sD3
utjTLLRNiXKF45RWKQtzZT3eRG2elAfm7a9ZnWgFBm0Q+/kSPmPHzo3aCx9K8yVUPmdg/S8BWjh5
VLvz9U99JDJKF4FEx3o=
</ds:SignatureValue>
<ds:KeyInfo>
<wsse:SecurityTokenReference Usage="sec:MessageAuthentication"
xmlns:sec="urn:liberty:sec:2003-08">
<wsse:Reference URI="#X509Token"/>
</wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</soapenv:Header>
<soapenv:Body>
<disco:Query id="NTT43EBDA48A7965082DA284C13DE33EFDE"
xmlns:disco="urn:liberty:disco:2003-08">
<disco:ResourceID>
https://whitebroadband.com:8443/idp/metadata/37e66f7afc918eb5c27b7b15fca55a01
</disco:ResourceID>
<disco:RequestedServiceType>
<disco:ServiceType>urn:liberty:id-sis-pp:2003-08</disco:ServiceType>
<disco:Options>
<disco:Option>urn:liberty:id-sis-pp:home</disco:Option>
<disco:Option>urn:liberty:id-sis-pp:informalName</disco:Option>
</disco:Options>
</disco:RequestedServiceType>
</disco:Query>
```

591

592 Figure 6.13 ID-WSF Discovery Service Query message sent from SP2 to DS

593 6.1.1.4.12 12. ID-WSF Discovery Service QueryResponse (Discovery Lookup Response)

594 DS that has received ID-WSF Discovery Service Query message, responds to SP2 with ID-WSF Discovery
595 Service QueryResponse in which ResourceOfferings that mach with specified ResourceID and ServiceType are
596 embedded. Figure 6.14 shows an example of ID-WSF Discovery Service QueryResponse message.

597 In the example in

598 Figure 6.13, SP2 specifies some Option keywords. These Option keywords are defined in ID-SIS Personal
599 Profile specification, and are used to specify particular attributes of Personal Profile and query them if they are
600 available to share. In the example in Figure 6.14, SP2 gets SP1's ResourceOffering.

601

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Header>
  <sb:Provider id="NTTE820FAA18F168B2AD7E627689489D4ED"
providerID="https://whitebroadband.com:8443/idp/metadata"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0" xmlns:sb="urn:liberty:sb:2003-08"/>
  <sb:Correlation id="NTTE8AE0B45F78061FE570DBFBB06EC62DB"
messageID="uuid:9ea67cd2-f414-c756-cb06-917d7f84dfe5"
refToMessageID="uuid:debbffd3-4ea8-973e-5463-e5ecc2d95dde"
timestamp="2004-03-10T05:59:02Z"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1" xmlns:sb="urn:liberty:sb:2003-08"/>
  <wsse:Security soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1" xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
<wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
soap-message-security-1.0#Base64Binary"
ValueType=”http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3”
wsu:Id="X509Token"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
MIICBDCCAW2gAwIBAgIBUTANBgkqhkiG9w0BAQQFADA1MQswCQYDVQQGEwJVUzEUMBIGA1UEChML
TGliZXJ0eSBJT1AxEDAOBgNVBAMTB1Rlc3QgQ0EwHhcNMDMxMjA1MTQ0MjI1WhcNMDQxMjA0MTQ0
MjI1WjBMMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPTGliZXJ0eSBJT1AgamtzMSMwIQYDVQQDExpu
dHQtYS1zaWduLmxpYmVydHktaW9wLm9yZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAq9Sl
+JvcHKNjjtE/v70TKMMXo+Ft05RBy/XUruHZsuH0b206MYG+CCBoPyJ6Jv+6+ZsCgIEjfJ6lqRZR
ZmPdGv92zcBHHO1/k1/GG7FPNFH+mrUm/66nRDysv6JFMzW7+Ct7132IUUiTEvj45CXHEcyy8JUf
fd1J1+FOrVweAEUCAwEAAaMNMAswCQYDVR0TBAIwADANBgkqhkiG9w0BAQQFAAOBgQBwqsW22HMT
LTcxn3jiifP+yBjKRaYpikrRzffeJ8XtLUrHCkm7ZOX/OeqidHAARB4lTxmITCB3LbHmViAk4G66
K4Yb9Y0FFVJCFyaYHnY6W6oLDkTv5IMqDL//vV6QF9boO2gvTpap4WL5+6meNmCyWKoeO4CuwX3q
ys5yrA8opg==
</wsse:BinarySecurityToken>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#NTTE820FAA18F168B2AD7E627689489D4ED">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>8PAUP7vDPOYT2JHoRBCkyrF8jTU=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#NTTE8AE0B45F78061FE570DBFBB06EC62DB">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>mZBMURT5gUco82RQsEEJHo3lC4U=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#NTTE6CF51BEB0320300DF0F4070CD04D1B6">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
```

602
603

```
<ds:SignatureValue>
aFly651gBlQd5kNeOIE12Sku/fARaG+pf8j2emc2qt9F1BmJrTFF9SeRdnkCXAGa7zxbdytUbKBI
mzN7OAmRdgzUN/AUtqTD/fqPulm3KYGzn8otTsX61JV/73ZIEgP2+vC9/Tsa8VD8mTAceXeiEzRb
ZNEyWnfWhyhLPf5TgX0=
</ds:SignatureValue>
<ds:KeyInfo>
<wsse:SecurityTokenReference Usage="sec:MessageAuthentication"
xmlns:sec="urn:liberty:sec:2003-08">
<wsse:Reference URI="#X509Token"/>
</wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</soapenv:Header>
 <soapenv:Body>
  <disco:QueryResponse id="NTTE6CF51BEB0320300DF0F4070CD04D1B6"
xmlns:disco="urn:liberty:disco:2003-08">
   <disco:Status code="disco:OK"/>
   <disco:ResourceOffering entryID="uuid:1c1ccaeb-0c36-229b-d510-7ae33406ada4">
    <disco:ResourceID>uuid:e427014e-1fde-cc03-85dd-690333bf695a</disco:ResourceID>
    <disco:ServiceInstance>
     <disco:ServiceType>urn:liberty:id-sis-pp:2003-08</disco:ServiceType>
     <disco:ProviderID>https://blueliquor.com:8443/sp1/metadata</disco:ProviderID>
     <disco:Description>
      <disco:SecurityMechID>urn:liberty:security:2003-08:TLS:X509</disco:SecurityMechID>
      <disco:SecurityMechID>urn:liberty:security:2003-08:TLS:null</disco:SecurityMechID>
      <disco:Endpoint>https://blueliquor.com:8443/sp1/services/idpp</disco:Endpoint>
     </disco:Description>
    </disco:ServiceInstance>
    <disco:Options>
     <disco:Option>urn:liberty:id-sis-pp:home</disco:Option>
     <disco:Option>urn:liberty:id-sis-pp:personal</disco:Option>
     <disco:Option>urn:liberty:id-sis-pp:cn</disco:Option>
     <disco:Option>urn:liberty:id-sis-pp:informalName</disco:Option>
     <disco:Option>urn:liberty:id-sis-pp:demographics</disco:Option>
    </disco:Options>
    <disco:Abstract>identity service for demostration</disco:Abstract>
   </disco:ResourceOffering>
  </disco:QueryResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

Figure 6.14 ID-WSF Discovery Service QueryResponse message sent from DS to SP2

6.1.1.4.13 13. ID-SIS Personal Profile Query

SP2 that has received SP1's ResourceOffering with sequence #12, sends ID-SIS Personal Profile Query message to SP1 so as to get necessary attributes of a Principal. This message is defined in the ID-WSF Data Service Template specification.

Figure 6.15 shows an example of ID-SIS Personal Profile message.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<sb:Correlation id="NTT4128D3FA79CC812662B92C8E962A2AD5"
messageID="uuid:8419e396-01fd-a411-fb7f-46721c7a0bbb"
timestamp="2004-03-10T05:59:03Z"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1" xmlns:sb="urn:liberty:sb:2003-08"/>
<sb:Provider id="NTTD98E695B9B665694504972D1DF00A2B2"
providerID="https://yellowpizza.com:8443/sp2/metadata"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0" xmlns:sb="urn:liberty:sb:2003-08"/>
<wsse:Security soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1"
xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
<wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-
message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
wsu:Id="X509Token" xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
MIICBDCCAW2gAwIBAgIBUTANBgkqhkiG9w0BAQQFADA1MQswCQYDVQQGEwJVUzEUMBIGA1UEChMLTG
liZXJ0eSBJT1AxEDAOBgNVBAMTB1Rlc3QgQ0EwHhcNMDMxMjA1MTQ0MjI1WhcNMDQxMjA0MTQ0
MjI1WjBMMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPTGliZXJ0eSBJT1AgamtzMSMwIQYDVQQDExpu
dHQtYS1zaWduLmxpYmVydHktaW9wLm9yZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAq9Sl
+JvcHKNjjtE/v70TKMMXo+Ft05RBy/XUruHZsuH0b206MYG+CCBoPyJ6Jv+6+ZsCgIEjfJ6lqRZR
ZmPdGv92zcBHHO1/k1/GG7FPNFH+mrUm/66nRDysv6JFMzW7+Ct7132IUUiTEvj45CXHEcyy8JUf
fd1J1+FOrVweAEUCAwEAAaMNMAswCQYDVR0TBAIwADANBgkqhkiG9w0BAQQFAAOBgQBwqsW22HMT
LTcxn3jiifP+yBjKRaYpikrRzffeJ8XtLUrHCkm7ZOX/OeqidHAARB4lTxmITCB3LbHmViAk4G66
K4Yb9Y0FFVJCFyaYHnY6W6oLDkTv5IMqDL//vV6QF9boO2gvTpap4WL5+6meNmCyWKoeO4CuwX3q
ys5yrA8opg==
</wsse:BinarySecurityToken>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#NTT4128D3FA79CC812662B92C8E962A2AD5">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>M9rSK/PxICulsYEhUIlGVu4JE0s=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#NTTD98E695B9B665694504972D1DF00A2B2">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>VjPKcTmqRbKhxN2s24YIiuSTCBg=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#NTT279922C20F1473B04D14F21F5B929890">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>h6IAX/jtHLE/Swqwxw2k+q6l7YM=</ds:DigestValue>
</ds:Reference>
```

613
614

```
      NXz1NSu0BWGFkXpN42ka6Ub4G7ZE0AhZrsC0MgLijitfwHRPM/zeWfxKGR+msjDbhzIkZs/+icnv
      JDl0Mc4ktqGLRRQ02JianF+SXElr9kxTtwVb/mXnLrXbIEDaQZ3q3KtF14Q4XExreBVjczDDJbrw
      bVZg2rvo7bmtPIy7DeQ=
      </ds:SignatureValue>
      <ds:KeyInfo>
      <wsse:SecurityTokenReference Usage="sec:MessageAuthentication"
      xmlns:sec="urn:liberty:sec:2003-08">
      <wsse:Reference URI="#X509Token"/>
      </wsse:SecurityTokenReference>
      </ds:KeyInfo>
      </ds:Signature>
      </wsse:Security>
      </soapenv:Header>
      <soapenv:Body>
      <pp:Query id="NTT279922C20F1473B04D14F21F5B929890" xmlns:pp="urn:liberty:id-sis-
pp:2003-08">
      <pp:ResourceID>uuid:e427014e-1fde-cc03-85dd-690333bf695a</pp:ResourceID>
      <pp:QueryItem includeCommonAttributes="0">
      <pp:Select>/pp:PP/pp:InformalName</pp:Select>
      </pp:QueryItem>
      <pp:QueryItem includeCommonAttributes="0">
      <pp:Select>/pp:PP/pp:AddressCard/pp:Address/pp:PostalAddress</pp:Select>
      </pp:QueryItem>
      </pp:Query>
      </soapenv:Body>
      </soapenv:Envelope>
```

615
616

617 Figure 6.15 ID-SIS Personal Profile Query message sent from SP2 to SP1

618 #### 6.1.1.4.14 14. ID-SIS Personal Profile QueryResponse

619 SP1 that has received ID-SIS Personal Profile Query message with sequence #13, responds to SP2 with ID-SIS
620 Personal Profile QueryResponse message in which Principal's attributes are embedded. In the example in,
621 InformalName and PostalAddress are requested. Therefore, these two kinds of attribute values are embedded in the
622 QueryResponse.
623 Figure 6.16 shows an example of ID-SIS Personal Profile QueryResponse message.
624

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<sb:Correlation id="NTT8EF885B720C0B633948923E992DD86CF"
messageID="uuid:2e56b5e3-52c9-8876-102e-23c8f1b2a40c"
refToMessageID="uuid:8419e396-01fd-a411-fb7f-46721c7a0bbb"
timestamp="2004-03-10T05:59:06Z"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1" xmlns:sb="urn:liberty:sb:2003-08"/>
<sb:Provider id="NTTD65ZK695B9B635354504972D1DF00N85A"
providerID="https://blueliquor.com:8443/sp1/metadata"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
```

```
soapenv:mustUnderstand="0" xmlns:sb="urn:liberty:sb:2003-08"/>
<wsse:Security soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1"
xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
<wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-
message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
wsu:Id="X509Token" xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
MIICBDCCAW2gAwIBAgIBUTANBgkqhkiG9w0BAQQFADA1MQswCQYDVQQGEwJVUzEUMBIGA1UEChML
TGliZXJ0eSBJT1AxEDAOBgNVBAMTB1Rlc3QgQ0EwHhcNMDMxMjA1MTQ0MjI1WhcNMDQxMjA0MTQ0
MjI1WjBMMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPTGliZXJ0eSBJT1AgamtzMSMwIQYDVQQDExpu
dHQtYS1zaWduLmxpYmVydHktaW9wLm9yZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAq9Sl
+JvcHKNjjtE/v70TKMMXo+Ft05RBy/XUruHZsuH0b206MYG+CCBoPyJ6Jv+6+ZsCgIEjfJ6lqRZR
ZmPdGv92zcBHHO1/k1/GG7FPNFH+mrUm/66nRDysv6JFMzW7+Ct7132IUUiTEvj45CXHEcyy8JUf
fd1J1+FOrVweAEUCAwEAAaMNMAswCQYDVR0TBAIwADANBgkqhkiG9w0BAQQFAAOBgQBwqsW22HMT
LTcxn3jiifP+yBjKRaYpikrRzffeJ8XtLUrHCkm7ZOX/OeqidHAARB4lTxmITCB3LbHmViAk4G66
K4Yb9Y0FFVJCFyaYHnY6W6oLDkTv5IMqDL//vV6QF9boO2gvTpap4WL5+6meNmCyWKoeO4CuwX3q
ys5yrA8opg==
</wsse:BinarySecurityToken>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#NTT8EF885B720C0B633948923E992DD86CF">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>d5vebdZmHHJtct1YnmSeG9kDcpk=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
FnGXfjUGKs1iOnVRSuqvtKhwOJRhVOiDtDVGjh5IzmXNi2AxU8MKDxhFw3sGkDEm6uufwy6xstf0
6bGyJd2VT1Wpr38fASBfnXNb1tCdFQF9ssjtdjpsENTkxaa8v5ZVkFaik4TlaxGf53ui0xxQ6KDJ
onIernnDeAT8vDa3U5g=
</ds:SignatureValue>
<ds:KeyInfo>
<wsse:SecurityTokenReference Usage="sec:MessageAuthentication"
xmlns:sec="urn:liberty:sec:2003-08">
<wsse:Reference URI="#X509Token"/>
</wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</soapenv:Header>
 <soapenv:Body>
<pp:QueryResponse timeStamp="2004-03-10T05:59:06Z" xmlns:pp="urn:liberty:id-sis-pp:2003-08">
<pp:Status code="pp:OK"/>
<pp:Data>
<pp:InformalName>Yuzo KOGA</pp:InformalName>
</pp:Data>
<pp:Data>
<pp:PostalAddress>TOKYO</pp:PostalAddress>
</pp:Data>
</pp:QueryResponse>
```

Figure 6.16 ID-SIS Personal Profile QueryResponse message sent from SP1 to SP2

7. Anonymous B2B Example Sessions

This document describes how Liberty ID-FF & ID-WSF can be applied in the particular scenario of anonymous Principal B2B interactions.[2]

7.1 Overview

Liberty ID-Federation Framework (ID-FF) and ID-Web Services Framework (ID-WSF) define general frameworks for federated identity. As such, they offer a variety of options and mechanisms to enable information sharing (authentication status and attributes) between providers. In many real-world scenarios, only a fraction of these options will be relevant and so, the full complexity of the specifications can be profiled down to this subset.

This document demonstrates the application of ID-FF and ID-WSF to a particular scenario: an employee of an enterprise needing to access the resources/services of a business partner in order to perform their duties. As the employee will not be offered any customizations or individualized access, the business partner does not need to know the specific identity of the employee, rather merely that they have the appropriate entitlements, as captured in a role assigned to them by their employer. This captures a frequent reality in B2B transactions. Ultimately, a company needs to know that a partner will stand behind the actions of their employees in any dealings between the companies; in many cases the identity (either real or a pseudonym) of the individual is irrelevant.

7.2 Scenario

Geoff Smith is an employee of Acme Widgets, a leading manufacturer of widgets for the thingymajig industry. Geoff's role within Acme is a Junior Purchasing Agent, this role means that Acme authorizes him to place parts orders with Acme's suppliers up to a value of $1,000 at a time. Geoff occasionally deals with Acme's supplier Bolts-R-Us, placing orders for bolts through Bolts-R-Us's ordering interface. In the past, Geoff has had to maintain an account at Bolts-R-Us. In order to place an order, he would need to sign-in using a username and password used only at Bolts-R-Us. Such a system has many issues:

- the sporadic nature of Geoff's dealings there meant he often forgot both the account name and/or the password, causing delay for Geoff and support costs for Bolts-R-Us.
- the fast turnaround in Junior Purchasing Agents has meant that Bolts-R-Us has often had to create new accounts for Acme's new hires, an expensive process when the information needs to be verified by Acme.
- because he might apply for employment at Bolts-R-Us in the future, Geoff would prefer that his purchasing activity not be traceable to him (maybe he always bought the cheap stuff?)

Fortunately, both Acme and Bolts-R-Us have recently implemented support for Liberty's specifications into their identity infrastructure (even though neither did so motivated by the thought of interacting with the other). Liberty's technologies will allow Geoff to maintain his identity information at Acme which will, in order to enable appropriate access at Bolts-R-Us for Geoff, share with the supplier the relevant information regarding him.

Liberty's technology will address the issues listed above as follows:

- Geoff will not be required to establish an account at Bolts-R-Us. He will be able to access the appropriate resources there based on an authentication he performed to his own company, i.e. signing into Acme's intranet in the morning.
- As Bolts-R-Us will not need to maintain accounts for Acme's individual Purchasing Agents, they will be unaffected as Acme's employees come and go.
- Geoff's actions at Bolts-R-us will be untraceable because his identity will be unknown and untraceable to them.

The next sections describes the User Experience and the sequence of operations

[2] This example is provided by Liberty Member Entrust.

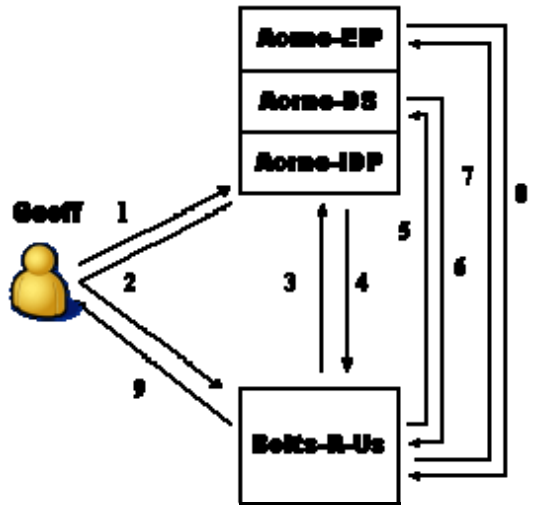
7.3 User Experience

1. Geoff goes to Acme's intranet portal
2. Geoff logs in using an X.509 certificate issued to him by Acme
3. Geoff sees a customized Acme interface, including a link 'Order at Bolts-R-Us'
4. As he knows Acme is running low on #45 bolts, Geoff clicks on 'Order at Bolts-R-Us' link
5. Geoff sees Bolts-R-Us's ordering interface
6. Geoff orders 20,000 #45 bolts at a unit cost of $0.10.
7. Geoff see's an alert that his order has failed because the amount exceeds his purchaing amount authorization
8. Geoff changes the order to 10,000 #45 bolts.
9. Geoff sees an acknowledgement that the order has gone through.

7.4 Message Flow

The figure below illustrates the message flow.

Figure 1. Message Flow



The steps are as follows:

1. Geoff authenticates to Acme-IDP. Geoff clicks on 'Order at Bolts-R-Us' button, browser is sent to Bolts-R-Us with artifact
2. Bolts-R-Us requests SAML assertion corresponding to artifact
3. Acme-IDP returns SAML assertion for Geoff containing anonymous one-time identifier for Geoff and bootstrap information for Geoff's DS.
4. SP queries Acme-DS for Geoff's EP service.
5. Acme-DS returns ResourceOffering for EP service, contains all necessary tokens.
6. Bolts-R-Us queries Acme-EP for Geoff's EmployeeType.
7. Acme-EP returns Geoff's EmployeeType.
8. Based on returned roles, Bolts-R-Us can make authorization decisions with respect to what resources Geoff can access.

The following sections present in more detail the different messages that flow between Acme and Bolts-R-Us.

7.4.1 Step 1

Geoff authenticates to Acme's company intranet using an Acme account and password. He is presented with an interface customized to his 'Junior Purchasing Agent' job responsibilities.

In addition to the usual News, Employee Resources, and Classified sections, Geoff's page contains a list of links to suppliers with which he often deals. In the past, clicking on these links would take Geoff to a login page of the particular supplier where he would authenticate using an account and password specific to that supplier.

Geoff knows that Acme is running dangerously low on #45 bolts and he knows that Bolts-R-Us is the preferred provider for these bolts. Amongst the other suppliers, he sees a 'Bolts-R-Us Order Page' link that he clicks on.

7.4.2 Step 2

Message 2 is a message sent from Acme-IDP to Bolts-R-Us, unsolicited because, in this scenario, it is not sent in response to a previous AuthnRequest from Bolts-R-Us. When Geoff clicks on the 'Order at Bolts-R-Us' button on his customized Acme intranet home page, his browser is initially sent to a transfer service URL at Acme. It is the transfer service that creates the Liberty artifact that will be sent to Bolts-R-Us. After creating the artifact, Acme-IDP sends it as as a query parameter to the appropriate Bolts-R-Us assertion consumer service URL (this obtained from previously exchanged Bolts-R-Us metadata.)

```
HTTP/1.0 302 Found
Location:
http://acs.boltsrus.com?SAMLart=AAM1uXw6+f+jyA/4XuFHqPl7QDvc/LIQL9+t7YQtG1Gwk9bph0Adl+o+
<other HTTP 1.0 or 1.1 components>
```

Step 2 Notes

1. Message 2 is sent by Acme to the Bolts-R-Us Assertion Consumer Service at 'http://acs.boltsrus.com' - this URL previously specified by Bolts-R-Us to Acme.
2. The SAML artifact is passed as a URL query parameter, i.e. that which follows the '?' in the above URL. Sending an artifact in this manner rather than the actual authentication assertion addresses the limitations for URL length.

7.4.3 Step 3

Message 3 is a SOAP message sent from Bolts-R-Us to Acme-IDP in which Bolts-R-Us presents the artifact it just received in Message 2 and requests that it be exchanged for the corresponding Authentication assertion for Geoff.

```
POST /soap HTTP/1.0
Host: idp.acme.com
Content-length: ...
Content-type: text/xml
<s:Envelope
   xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:sb="urn:liberty:sb:2003-08"
   xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
   <s:Header>
      <sb:Correlation
         s:mustUnderstand="true"
         messageId="NK44V79NdfPaE5jCwlk_"
         timestamp="2003-06-06T12:06:12Z"/>
   </s:Header>
   <s:Body>
      <samlp:Request IssueInstant="2002-12-12T10:08:56Z"
         MajorVersion="1" MinorVersion="1"
         RequestID="e4d71c43-c89a-426b-853e-a2b0c14a5ed8"
         id="b6dc3636-f2ad-42d1-9427-220f2cf70ec1">
         <samlp:AssertionArtifact>
         AAM1uXw6+f+jyA/4XuFHqPl7QDvc/LIQL9+t7YQtG1Gwk9bph0Adl+o+
         </samlp:AssertionArtifact>
      </samlp:Request>
   </s:Body>
```

```
</s:Envelope>
```

Step 3 Notes

1. Message 3 is sent by Bolts-R-Us to Acme at *idp.acme.com* - this URL previously specified by Acme.
2. The *messageId* attribute on the *Correlation* element has the value 'NK44V79NdfPaE5jCwlk_'. This will allow Bolts-R-Us to correlate Acme's response with this request.
3. The *AssertionArtifact* element carries the string 'AAM1uXw6+f+jyA/4XuFHqPl7QDvc/LIQL9+t7YQtG1Gwk9bph0Adl+o+' - this the value of the artifact sent in Message 2.

7.5.4 Step 4

Message 4 is a SOAP response message sent from Acme-IDP to Bolts-R-Us in which the SAML authentication assertion is returned to Bolts-R-Rs.

```
HTTP/1.0 200 OK
Content-length: ...
Content-type: text/xml

<s:Envelope
   xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:sb="urn:liberty:sb:2003-08"
   xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
   xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
   xmlns:lib="urn:liberty:iff:2003-08">
   <s:Header>
      <sb:Correlation
         s:mustUnderstand="true"
         refToMessageId="NK44V79NdfPaE5jCwlk_"
         messageId="uuid:0048345-47329874-45873278"
         timestamp="2003-06-06T12:07:12Z"/>
   </s:Header>
   <s:Body>
      <samlp:Response
         InResponseTo="e4d71c43-c89a-426b-853e-a2b0c14a5ed8"
         IssueInstant="2 002-10-31T21:42:13Z" MajorVersion="1" MinorVersion="1"
         Recipient="http://www.boltsrus.com"
         ResponseID="LANWfL2xLybnc+BCwgY+p1/vIVAj">
         <samlp:Status>
            <samlp:StatusCode>
               Value="qns:Success">
            </samlp:StatusCode>
         </samlp:Status>
         <lib:Assertion AssertionID="SqMC8Hs2vJ7Z+t4UiLSmhKOSUO0U"
            InResponseTo="e4d71c43-c89a-426b-853e-a2b0c14a5ed8"
            IssueInstant="2003-06-06T12:07:12Z" Issuer="http://idp.acme.com"
            MajorVersion="1" MinorVersion="2">
            <saml:Conditions
               NotBefore="2003-06-06T12:07:12Z"
               NotOnOrAfter="2003-06-06T12:10:12Z">
               <saml:AudienceRestrictionCondition>
                  <saml:Audience>http://www.boltsrus.com</saml:Audience>
               </saml:AudienceRestrictionCondition>
            </saml:Conditions>
            <lib:AuthenticationStatement
               AuthenticationInstant="2003-06-06T12:07:12Z"
               AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
```

```
            <lib:Subject xsi:type="lib:SubjectType">
              <saml:NameIdentifier Format="urn:liberty:iff:nameid:one-time">
              S2T4R5E7A8K1I8S9O2V9E0R
              </saml:NameIdentifier>
              <saml:SubjectConfirmation>
                <saml:Confirmation Method>
                urn:oasis:names:tc:SAML:1.0:cm:artifact-01
                </saml:ConfirmationMethod>
              </saml:SubjectConfirmation>
              <lib:IDPProvidedNameIdentifier
                NameQualifier="http://idp.acme.com"
                Format="urn:liberty:iff:nameid:one-time">
              S2T4R5E7A8K1I8S9O2V9E0R
              </lib:IDPProvidedNameIdentifier>
            </lib:Subject>
            <lib:AuthnContext>
              <lib:AuthnContextClassRef>
            http://www.projectliberty.org/schemas/authctx/classes/PasswordProtectedTransport
              </lib:AuthnContextClassRef>
            </lib:AuthnContext>
          </lib:AuthenticationStatement>
          <saml:AttributeStatement>
            <saml:Subject>
              <saml:NameIdentifier Format="urn:liberty:iff:nameid:one-time">
              S2T4R5E7A8K1I8S9O2V9E0R
              </saml:NameIdentifier>
            </saml:Subject>
            <saml:Attribute AttributeName="DiscoveryResourceOffering"
              AttributeNamespace="urn:liberty:disco:2003-08">
              <saml:AttributeValue>
                <lib:ResourceOffering>
                  <disco:ResourceID>http://disco.acme.com/d0CQF8elJTDLmzEo</disco:ResourceID>
                  <disco:ServiceInstance>
                    <disco:ServiceType>urn:liberty:disco:2003-08</disco:ServiceType>
                    <disco:ProviderID>http://disco.acme.com</disco:ProviderID>
                    <disco:Description>
                      <SecurityMechID>
                      urn:liberty:security:2003-08:TLS:X509
                      </SecurityMechID>
                      <disco:Endpoint>https://disco.acme.com</disco:Endpoint>
                    </disco:Description>
                  </ServiceInstance>
                  <Abstract>Acme Discovery service</Abstract>
                </lib:ResourceOffering>
              </saml:AttributeValue>
            </saml:Attribute>
          </saml:AttributeStatement>
          <ds:Signature>
          Acme digital signature
          </ds:Signature>
        </lib:Assertion>
      </samlp:Response>
    </s:Body>
</s:Envelope>
```

Step 4 Notes

1. Message 4 is sent by Acme to Bolts-R-Us in response to Message 3.
2. The *refToMessageId* on the *Correlation* element has the value 'NK44V79NdfPaE5jCwlk_'. This matches the *messageId* of Message 3.

3. The SAML *Status* element indicates that Message 4 is a successful response.
4. The *Format* attribute on the *AuthenticationStatement/Subject/NameIdentifier* element indicates that the identifier being returned for Geoff (namely 'S2T4R5E7A8K1I8S9O2V9E0R') is 'one-time', i.e. it does not correspond to any previously used identifier for either Geoff or another Acme employee.
5. The *IDPProvidedNameIdentifier* element contains the same string of 'S2T4R5E7A8K1I8S9O2V9E0R' indicating that this is the string that Acme (the IDP) has chosen to represent Geoff. If this were not a 'one-time' interaction, Bolts-R-Us could specify its own preferred value as an *SPProvidedNameIdentifier* element.
6. The *AuthnContext* element indicates that that Geoff originally authenticated to Acme using a password over SSL.
7. As well as the assertion for Geoff, Acme-IDP returns to Bolts-R-Us a *ResourceOffering* for the relevant DiscoveryService as an *AttributeStatement*. The *ResourceID* for this ResourceOffering has a value of 'http://disco.acme.com/d0CQF8elJTDLmzEo' - this string wll be used by Bolts-R-Us on subsequent calls to Acme's Discovery Service to refer to Geoff (anonymously).
8. The *SecurityMechID* element contains the value 'urn:liberty:security:2003-08:TLS:X509' - indicating that subsequent queries to the Discovery Service must be protected with both SSL and an X.509 based message-layer signature.
9. The *EndPoint* element within the *ResourceOffering* contains the string 'disco.acme.com' - this is the Acme URL to which Bolts-R-Us will send subsequent discovery queries.

7.5.5 Step 5

Message 5 is a request from Bolts-R-Us to Acme-DS in which Bolts-R-Us queries for the location of Geoff's EP Service.

```
POST /soap HTTP/1.0
Host: disco.acme.com
Content-length: ...
Content-type: text/xml

<s:Envelope
   xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:disco="urn:liberty:disco:2003-08"
   xmlns:sb="urn:liberty:sb:2003-08">
   <s:Header>
      <sb:Correlation
         id="K8H6F53gh89HGY"
         s:mustUnderstand="1"
         messageID="K8H6F53gh89HGY"
            timestamp="2003-06-06T12:08:12Z"/>
      <ws:Security>
         <ds:Signature>
            Bolts-R-Us signature as specified by Acme.
            Needs detail
         </ds:Signaure>
      </ws:Security>
   </s:Header>
   <s:Body>
      <disco:Query>
         <disco:ResourceID>http://disco.acme.com/d0CQF8elJTDLmzEo</disco:ResourceID>
         <disco:RequestedServiceType>
            <disco:ServiceType>urn:liberty:id-sis-ep:2003-08</disco:ServiceType>
         </discoRequestedServiceType>
      </disco:Query>
     </s:Body>
</s:Envelope>
```

Step 5 Notes

1. Message 5 is sent by Bolts-R-Us to Acme at *disco.acme.com* - this the URL specified in the *Endpoint* element of Message 4's *ResourceOffering*.
2. The *messageId* attribute on the *Correlation* element has the value 'K8H6F53gh89HGY'. This will allow Bolts-R-Us to correlate Acme's response with this request.
3. The *ResourceID* element in the *Query* element contains the identifier 'http://disco.acme.com/d0CQF8elJTDLmzEo' previously provided to Bolts-R-Us by Acme in Message 4.
4. The *RequestedServiceType* indicates to Acme's Discovery Service that Bolts-R-Us is interested in the location of Geoff's EP Service.

7.5.6 Step 6

Message 6 is the response to Message 5 in which Acme's Discovery Service returns to Bolts-R-Us the relevant *ResourceOffering* for the EP Service.

```
HTTP/1.0 200 OK
Content-length: ...
Content-type: text/xml

<s:Envelope
   xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:disco="urn:liberty:disco:2003-08"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:sb="urn:liberty:sb:2003-08"
   xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
   xmlns:lib="urn:liberty:iff:2003-08"
   xmlns:ws="http://schemas.xmlsoap.org/ws/2003/06/secext"
   xmlns:sec="urn:liberty:sec:2003-08">
   <s:Header>
      <sb:Correlation
         s:mustUnderstand="true"
         refToMessageId="K8H6F53gh89HGY"
         messageId="uuid:008678-98538765-27589543"
         timestamp="2003-06-06T12:09:12Z">
   </s:Header>
   <s:Body>
      <disco:QueryResponse>
         <Status code="OK"/>
         <disco:ResourceOffering entryID="1">
            <disco:ResourceID>http://ep.acme.com/zsjsdkjfsdf</disco:ResourceID>
            <disco:ServiceInstance>
               <disco:ServiceType>urn:liberty:id-sis-ep:2003-08</disco:ServiceType>
               <disco:ProviderID>http://www.acme.com/</disco:ProviderID>
               <disco:Description>
                  <disco:SecurityMechID>urn:liberty:security:2003-08:TLS:X509</disco:SecurityMechID>
                  <disco:CredentialRef>SqMkfghjs2v+jskhdfHU</disco:CredentialRef>
                  <disco:Endpoint>https://ep.acme.com:443/soap</disco:Endpoint>
               </disco:Description>
            </disco:ServiceInstance>
            <disco:Abstract>Anonymous User's Employee Profile</disco:Abstract>
         </disco:ResourceOffering>
      </disco:QueryResponse>
   </s:Body>
</s:Envelope>
```

Step 6 Notes

1. Message 6 is sent by Acme to Bolts-R-Us in response to Message 5. It contains a *ResourceOffering* for Geoff's EP Service.

783 2. Acme used the *ResourceOffering* element in Message 4 to specify where Geoff's Discovery Service was
784 located, here it uses the same element structure (but not values) to specify where Geoff's EP Service is
785 located.
786 3. The location of Geoff's EP Service is provided in the *Endpoint* element of the returned *ResourceOffering*
787 element - namely the URL 'https://ep.acme.com:443/soap'.
788 4. The *refToMessageId* on the *Correlation* element has the value 'K8H6F53gh89HGY'. This matches the
789 *messageId* of Message 5.
790 5. The *ResourceID* element contains the string 'http://ep.acme.com/zsjsdkjfsdf' - this will be used by Bolts-
791 R-us on subsequent queries of the EP Service to refer to Geoff. In a more distributed scenario in which
792 the DS and EIS were not co-located, then the DS would need to ensure that the Service provider (Bolts-
793 R-Us in this scenario) would be unable to directly read the *ResourceID* - it would do so by encrypting the
794 value for the EIS. The Service provider would be able to forward this encrypted value onto the EIS in
795 subsequent queries but would be unable to use this identifier in a privacy-inappropriate manner.
796 6. The *SecurityMechID* element indicates the Security Mechanisms that Bolts-R-Us is expected to use in
797 subsequent interactions with the EP Service - namely TLS and an X.509 based signature.

798 ### 7.5.7 Step 7

799 Message 7 is a request from Bolts-R-Us to Acme's EP Service for the EmployeeType of Geoff.

```
POST /soap HTTP/1.0
Host: ep.acme.com
Content-length: ...
Content-type: text/xml

<s:Envelope
   xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:sb="urn:liberty:sb:2003-08"
   xmlns:lib="urn:liberty:iff:2003-08"
   xmlns:ws="http://schemas.xmlsoap.org/ws/2003/06/secext"
   xmlns:sec="urn:liberty:sec:2003-08"
   xmlns:ep="urn:liberty:id-sis-ep:2003-08">

   <s:Header>
      <sb:Correlation
         s:mustUnderstand="1"
         messageID="LJY756FGt96GBHF"
            timestamp="2003-06-06T12:11:12Z" />
      <ws:Security>
         <ds:Signature>
            Bolts-R-Us signature as specified by Acme.
            Needs detail
         </ds:Signature>
      </ws:Security>
      </ws:Security>
   </s:Header>
   <s:Body>
      <ep:Query>
         <ep:ResourceID>http://ep.acme.com/zsjsdkjfsdf</ep:ResourceID>
         <ep:QueryItem itemID="type">
            <ep:Select>/ep:EP/ep:EmployeeType</ep:Select>
         </ep:QueryItem>
      </ep:Query>
   </s:Body>
</s:Envelope>
```

800
801

Step 7 Notes

1. Message 7 is sent by Bolts-R-Us to Acme at *ep.acme.com* - this the URL specified in the *Endpoint* element of Message 6's *ResourceOffering* for the EP Service.
2. The *messageId* attribute on the *Correlation* element has the value 'LJY756FGt96GBHF'. This will allow Bolts-R-Us to correlate Acme's response with this request.
3. The *ResourceID* element in the *Query* element contains the identifier 'http://ep.acme.com/zsjsdkjfsdf' previously provided to Bolts-R-Us by Acme in Message 6.
4. The *QueryItem* element contains the string '/ep:EP/ep:EmployeeType' to indicate that Bolts-R-Us is specifically interested in Geoff's EmployeeType rather than the other data elements in the EP schema.

7.5.8 Step 8

Message 8 is the response to Message 7 in which Acme-EP returns the EmployeeType of Geoff to Bolts-R-Us.

```
HTTP/1.0 200 OK
Content-length: ...
Content-type: text/xml

<s:Envelope
   xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:sb="urn:liberty:sb:2003-08"
   xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
   xmlns:lib="urn:liberty:iff:2003-08"
   xmlns:ws="http://schemas.xmlsoap.org/ws/2003/06/secext"
   xmlns:sec="urn:liberty:sec:2003-08"
   xmlns:ep="urn:liberty:id-sis-ep:2003-08">
   <s:Header>
      <sb:Correlation
         s:mustUnderstand="1"
         refToMessageID="LJY756FGt96GBHF"
         messageID="uuid:0032945-28686728-25695608"
         timestamp="2003-06-06T12:12:12Z" />
   </s:Header>
   <s:Body>
      <ep:QueryResponse>
         <ep:Status code="ep:OK"/>
         <ep:Data itemIDRef="type">
            <ep:EmployeeType>
            JuniorPurchasingAgent
            </ep:EmployeeType>
         </ep:Data>
      </ep:QueryResponse>
   </s:Body>
</s:Envelope>
```

Step 8 Notes

1. Message 8 is sent by Acme to Bolts-R-Us in response to Message 7. It contains a *QueryResponse* containing Geoff's EmployeeType.
2. The *refToMessageId* on the *Correlation* element has the value 'LJY756FGt96GBHF'. This matches the *messageId* of Message 7.
3. The *EmployeeType* element carries Geoff's role, namely that he is a 'JuniorPurchasingAgent'. Acme and Bolts-R-us would have had to have previously agreed on what this attribute represents and Bolts-R-Us would have defined appropriate authorizations for this role.
4. 7.5.9. Step 9
5. With its knowledge of Geoff's role at Acme of Junior Purchasing Agent, Bolts-R-Us can provide a customized experience for him (i.e. ensure that he isn't presented with the ability to place orders on big-ticket items) and make appropriate authorization decisions for those orders he does place.

6. Its important to note that Bolts-R-Us would be unable to provide to Geoff any sort of 'Past Activity' information that was specific to him - this because the identifier Acme provided for Geoff was one-time and so prevented this sort of correlation. The best Bolts-R-Us could do would be create a list of products that 'Other Junior Purchasing Agents have ordered in the past'.

7.5 Optimizations

As illustrated, a number of message pairs are exchanged between Acme and Bolts-R-Us before Bolts-R-Us obtains the necessary attribute information for Geoff, namely his 'EmployeeType'. This general flow can be optimized as described below:

```
<s:Envelope>
  <s:Body>
    <samlp:Response>
      <lib:Assertion>
        <lib:AuthenticationStatement>

        </lib:AuthenticationStatement>
        <saml:AttributeStatement>
          <saml:Subject>
            <saml:NameIdentifier Format="urn:liberty:iff:nameid:one-time">
            S2T4R5E7A8K1I8S9O2V9E0R
            </saml:NameIdentifier>
          </saml:Subject>
          <saml:Attribute
            AttributeName="EmployeeType"
            AttributeNamespace="http://ep.acme.com">
            <saml:AttributeValue>JuniorPurchasingAgent</saml:AttributeValue>
          </saml:Attribute>
        </saml:AttributeStatement>
      </lib:Assertion>
    </samlp:Response>
  </s:Body>
</s:Envelope>
```

1. If Acme knew that Bolts-R-Us required Geoff's EmployeeType, then it could include this information in the original assertion it sent to Bolts-R-Us (Message 4 above). Message 4 would then appear (omitting previous details)
2. While this model significantly decreases the traffic between Acme and Bolts-R-Us, it assumes that Acme can anticipate all the attributes for Geoff that Bolts-R-Us might eventually need. This may or may not be realistic. For instance, in addition to EmployeeType, Bolts-R-Us might want to know if Geoff had a fixed spending limit
3. A potential compromise between the two extremes is to have Acme return a ResourceOffering for its EP service (rather than its Discovery Service) in the original assertion it creates for Bolts-R-Us (Message 4). This model would remove a request/response pair (Messages 5 & 6) and yet still allow Bolts-R-Us to subsequently query Acme's EP service for other attributes if necessary.

7.6 Summary

From Geoff's point of view, Liberty provides the following advantages over the previous model:

1. He no longer has to maintain an identity at Bolts-R-Us - meaning no account name and password to remember. The value of this grows significantly if Geoff deals with many other Acme suppliers.
2. He is given a customized interface at Bolts-R-Us based on the authentication he performed at Acme. Throughout the day, his interactions with other Liberty-enabled suppliers will be the same.

From Acme's point of view, Liberty provides the following advantages over the previous model:

1. Acme's employees can concentrate on their job responsibilities rather than remembering maintaining identity information at the business partners with which they interact.

2. Acme can be confident that the actions of its employees at its business partners will be consistent with the entitlements associated with their role.
3. The privacy of Acme'e employees is protected, Acme not unnecessarily disclosing information these employees to tis business partners.
4. There is no need for Acme to provision new employees into its business partners in order to ensure that they are set up with the appropriate authorizations. As the new employees interact with the business partners, the Liberty infrastructure will ensure that these authorizations 'flow with them' as required. Importantly, there is also no need for Acme to deprovision its employees from its business partners when its employees leave - all Acme need do is remove that employee from its won systems to ensure that the ex-employee will not be able to inappropriately access business partners.
5. The infrastructure Acme puts in place to support Bolts-R-Us can be leveraged with all other Liberty enabled companies its employees interact with, the cost amortized across all.

From Bolts-R-Us's point of view, Liberty provides the following advantages over the previous model:

1. Bolts-R-Us no longer needs to bear the costs associated with supporting (e.g. password resets) the employees of its business partners.
2. The infrastructure Bolts-R-Us puts in place to support Acme can be leveraged with all other Liberty enabled companies its employees interact with, the cost amortized across all.

8. Device Authentication Example Sessions

This section walks through the complete messages passed from and to a client invoking a service.[3] 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.

8.1 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.

8.2 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 version="1.0" encoding="utf-8" ?>
<S:Envelope
   xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:aol=" http://schemas.corp.aol.com/"
   xmlns:sb="urn:liberty:wsf:soap-bind:1.0"
   xmlns:sa=”urn:liberty:wsf:soap-auth:1.0” >
   <S:Header>
      <sb:Correlation S:mustUnderstand="1"
            messageID=”uuid:0023923-28329023-238239023”
            timestamp="2003-06-06T12:10:10Z"  />
   </S:Header>
   <S:Body>
      <sa:SASLRequest advisoryAuthnID=”123456789012:10023923”
            mechanism=”CRAM-MD5”/>
   </S:Body>
</S:Envelope>
```

8.3 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.

[3] With minor editorial changes, this example is taken from the document “Digital Media Services – Draft Services Invocation Framework Specification”, Conor P. Cahill, America OnLine, Inc., February 9, 2004.

```
<?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:sb="urn:liberty:wsf:soap-bind:1.0"
   xmlns:sa=”urn:liberty:wsf:soap-auth:1.0”  >
   <S:Header>
      <sb:Correlation S:mustUnderstand="1"
            messagID=”uuid:00287-83782-238891-09981”
            refToMessageID =”uuid:0023923-28329023-238239023”
            timestamp="2003-06-06T12:10:10Z"  />
   </S:Header>
   <S:Body>
      <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.

8.4 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 “tanstaaftanstaaf”) and sends a response to the Authentication Server. Please review for details on how the digest is generated (this particular value was actually lifted from the document).

```
<?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"
   xmlns:sa=”urn:liberty:wsf:soap-auth:1.0”  >
   <S:Header>
      <sb:Correlation S:mustUnderstand="1"
            messageID=”uuid:0023923-28329023-238239026”
            refToMessageID=”uuid:00287-83782-238891-09981”
            timestamp="2003-06-06T12:10:11Z"  />
   </S:Header>
   <S:Body>
      <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.

964 ## 8.5 Auth Server returns Security Token & Discovery Info

965 The server processes the request and returns the security token to the caller along with the bootstrap information
966 for accessing the discovery 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:sb="urn:liberty:wsf:soap-bind:1.0"
    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:ResourceID>urn:liberty:isf:implied-resource</disco:ResourceID>
                <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:SecurityMechID>…</disco:SecurityMechID>
                        <disco:Endpoint>https://discovery.ws.aol.com</disco:Endpoint>
                    </disco:Description>
                </disco:ServiceInstance>
            </disco:ResourceOffering>
            <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:0023923-28329023-238239026">
                    <saml:Conditions      NotOnOrAfter="2003-06-07T12:10:10Z" >
                        <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>
                                <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                                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>
                                <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                                AOLScreenname
                            </saml:NameIdentifier>
                        </saml:Subject>
                        <saml:Attribute AttributeName="devUPC"
                                AttributeNamespace="http://schemas.corp.aol.com">
                            <saml:AttributeValue>123456789012</saml:AttributeValue>
                        </saml:Attribute>
                    </saml:AttributeStatement>
                    <ds:Signature>
                        Signature data goes here
```

```
                    </ds:Signature>
                </saml:Assertion>
            </sa:Credentials>
        </sa:SASLResponse>
    </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.

8.6 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:sb="urn:liberty:wsf:soap-bind:1.0"
  xmlns:sa="urn:liberty:wsf:soap-auth:1.0" >
  <S:Header>
    <sb:Correlation S:mustUnderstand="1"
        messageID="uuid:0023923-28329328-23789404578"
        timestamp="2003-06-06T12:10:12Z"  />
    <wsse:Security>
          <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:0023923-28329023-238239026">
            <saml:Conditions       NotOnOrAfter="2003-06-07T12:10:10Z" >
              <saml:AudienceRestrictionCondition>
                <saml:Audience>http://discovery.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>
              <saml:Subject>
                <saml:NameIdentifier>
                    <saml:NameQualifier>http://aol.com</saml:NameQualifier>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    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>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    AOLScreenname
                </saml:NameIdentifier>
              </saml:Subject>
              <saml:Attribute AttributeName="devUPC"
                    AttributeNamespace="http://schemas.corp.aol.com">
                <saml:AttributeValue>123456789012</saml:AttributeValue>
              </saml:Attribute>
            </saml:AttributeStatement>
            <ds:Signature>
              Signature data goes here
            </ds:Signature>
          </saml:Assertion>
    </wsse:Security>
  </S:Header>
  <S:Body>
    <disco:Query>
      <ResourceID> urn:liberty:isf:implied-resource</ResourceID>
      <RequestedServiceType>
        <ServiceType>urn:aol-com:services:radio</ServiceType>
      </RequestedServiceType>
    </disco:Query>
    </S:Body>
</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.

8.7 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: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:00287-23234564-098098798"
          refToMessageID="uuid:0023923-28329328-23789404578"
        timestamp="2003-06-06T12:10:12Z"  />
    <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">
          A233asdfjwe8ldghweoiidfdlsjdwe (Base 64 Encoded Data)
        </wsse:BinarySecurityToken>
      </Credential>
    </sb:ServiceInstanceUpdate>
    </S:Header>
    <S:Body>
      <disco:QueryResponse>
        <Status code="OK" />
        <disco:ResourceOffering EntryID="1">
          <disco:ResourceID>urn:liberty:isf:implied-resource<disco:ResourceID>
          <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>
                http://ws.aol.com/security/2003-11:TLS:bearer
              </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>
              <saml:Subject>
                <saml:NameIdentifier>
                    <saml:NameQualifier>http://aol.com</saml:NameQualifier>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    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>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    AOLScreenname
                  </saml:NameIdentifier>
                </saml:Subject>
                <saml:Attribute AttributeName="devUPC"
                    AttributeNamespace="http://schemas.corp.aol.com">
                  <saml:AttributeValue>123456789012</saml:AttributeValue>
                </saml:Attribute>
              </saml:AttributeStatement>
              <ds:Signature>
                Signature data goes here
              </ds:Signature>
            </saml:Assertion>
          </disco:Credentials>
        </disco:QueryResponse>
      </S:Body>
</S:Envelope>
```

8.8 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 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>
                <saml:Subject>
                  <saml:NameIdentifier>
                    <saml:NameQualifier>http://aol.com</saml:NameQualifier>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    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>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    AOLScreenname
                  </saml:NameIdentifier>
                </saml:Subject>
                <saml:Attribute AttributeName="devUPC"
                    AttributeNamespace="http://schemas.corp.aol.com">
                  <saml:AttributeValue>123456789012</saml:AttributeValue>
                </saml:Attribute>
              </saml:AttributeStatement>
              <ds:Signature>
                Signature data goes here
              </ds:Signature>
            </saml:Assertion>
        </wsse:Security>
    </S:Header>
    <S:Body>
        <GetStationList/>
    </S:Body>
</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

8.9 Radio Service returns Info

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

```
<?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:9897923-82398723-092739723"
            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">
            A233asdfjwe8lwefjisde8asddj2weqw9ejajdh2hqdh72zxcb2easad
         </wsse:BinarySecurityToken>
      </Credential>
            <Endpoint>https://Radio15.ws.aol.com/</Endpoint>
   </sb:ServiceInstanceUpdate>
   </S:Header>
   <S:Body>
      // Station List data included here
   </S:Body>
</S:Envelope>
```

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/").

8.10 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 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:23409723497-20972347-23407234"
            refToMessageID="uuid:23452-7345097234-0974234097"
            timestamp="2003-06-06T12:10:16Z"  />
      <wsse:Security>
      <wsse:BinarySecurityToken wsu:Id="…"
            ValueType="anyPrefix:ServiceSessionContext">
         A233asdfjwe8lwefjisde8asddj2weqw9ejajdh2hqdh72zxcb2easad
      </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.

8.11 Radio Service returns info

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

```
<?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: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.

8.12 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 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">
         A233asdfjwe8ldghweoiidfdlsjdwe (Base 64 Encoded Data)
      </wsse:BinarySecurityToken>
      </wsse:Security>
   </S:Header>
   <S:Body>
      <disco:Query>
         <disco:ResourceID> urn:liberty:isf:implied-resource</disco:ResourceID>
         <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.

8.13 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 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: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>
        <disco:QueryResponse>
          <Status code="OK" />
          <disco:ResourceOffering EntryID="1">
            <disco:ResourceID>urn:liberty:isf:implied-resource<disco:ResourceID>
            <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>
                  http://ws.aol.com/security/2003-11:TLS:bearer
                </SecurityMechID>
                <Endpoint>https://photo.ws.aol.com/</Endpoint>
              </disco:Description>
            </disco:ServiceInstance>
          </disco:ResourceOffering>
          <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: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>
                <saml:Subject>
                  <saml:NameIdentifier>
                    <saml:NameQualifier>http://aol.com</saml:NameQualifier>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    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>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    AOLScreenname
                  </saml:NameIdentifier>
                </saml:Subject>
                <saml:Attribute AttributeName="devUPC"
                   AttributeNamespace="http://schemas.corp.aol.com">
                  <saml:AttributeValue>123456789012</saml:AttributeValue>
                </saml:Attribute>
              </saml:AttributeStatement>
```

```
              <ds:Signature>
                Signature data goes here
              </ds:Signature>
            </saml:Assertion>
          </disco:Credentials>
        </disco:QueryResponse>
    </S:Body>
</S:Envelope>
```

Notes:
[reserved]

8.14 Device requests info from Photo Service

The device requests a list of folders from the photo 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: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"
            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: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>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    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>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    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">
                <saml:AttributeValue>123456789012</saml:AttributeValue>
              </saml:Attribute>
            </saml:AttributeStatement>
            <ds:Signature>
              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.

8.15 Photo service returns info

The Photo Service returns the requested information

```
<?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”>
            A233asdfjwe8lwefjisde8asddj2weqw9ejajdh2hqdh72zxcb2easad
         </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.

8.16 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 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:sa="urn:liberty:wsf:soap-auth:1.0" >
    <S:Header>
      <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://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://discovery.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.aol.com/authctx/classes/DeviceProtectedTransport
                    </lib:AuthnContextClassRef>
                </lib:AuthnContext>
                <saml:Subject>
                    <saml:NameIdentifier>
                           <saml:NameQualifier>http://aol.com</saml:NameQualifier>
                           <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                           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>
                           <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                           AOLScreenname
                    </saml:NameIdentifier>
                </saml:Subject>
                <saml:Attribute AttributeName="devUPC"
                           AttributeNamespace="http://schemas.corp.aol.com">
                    <saml:AttributeValue>123456789012</saml:AttributeValue>
                </saml:Attribute>
              </saml:AttributeStatement>
              <ds:Signature>
                Signature data goes here
              </ds:Signature>
            </saml:Assertion>
      </wsse:Security>
    </S:Header>
    <S:Body>
      <sa:SASLRequest advisoryAuthnID="123456789012:10023923"
          mechanism="CRAM-MD5"/>
    </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.

8.17 The Authentication Server returns new token

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

```
<?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:sa="urn:liberty:wsf:soap-auth:1.0"  >
    <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>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    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>
                    <saml:Format>urn:liberty:iff:nameid:federated</saml:Format>
                    AOLScreenname
                  </saml:NameIdentifier>
                </saml:Subject>
                <saml:Attribute AttributeName="devUPC"
                    AttributeNamespace="http://schemas.corp.aol.com">
                  <saml:AttributeValue>123456789012</saml:AttributeValue>
                </saml:Attribute>
              </saml:AttributeStatement>
              <ds:Signature>
                Signature data goes here
              </ds:Signature>
            </saml:Assertion>
          </sa:Credential>
        </sa:SASLResponse>
    </S:Body>
</S:Envelope>
```

1766 Notes:
1767 The discovery service bootstrap information is not included since it was sent previously.
1768 The renewed token still has the same "*AuthenticationInstant*" since this is a renewal not a re-authentication.
1769

9. References

Normative

[LibertyAuthnContext] Madsen, Paul , eds. (12 Nov 2003). "Liberty ID-FF Authentication Context Specification," Version 1.2, Liberty Alliance Project, http://www.projectliberty/specs

[LibertyBindProf] Cantor, Scott & Kemp, John , eds. (April 2004). "Liberty ID-FF Bindings and Profiles Specification," Version 1.2,Liberty Alliance Project http://www.projectliberty/specs

[LibertyIDPersonalProfile] Kellomaki, Sampo, ed., (12 Nov 2003), "Liberty ID-SIS Personal Profile Service Specification", 1.0, Liberty Alliance Project, http://www.projectliberty/specs

[LibertyID-WSFDataServiceTemplate] Kainulainen, Jukka, Ranganathan, Aravindan, eds., "ID-WSF Data Service Template," Version 1.0, Liberty Alliance Project.

[LibertyID-WSFDiscoveryService] Sergent, Jonathan, eds. (12 Nov 2003). "Liberty ID-WSF Discovery Service Specification," Version 1.0- 08,Liberty Alliance Project http://www.projectliberty/specs

[LibertyID-WSFInteraction Service] Aarts, Robert, eds. (2003). "Liberty ID-WSF Interaction Service Specification," 1.0 , Liberty Alliance Project http://www.projectliberty/specs

[LibertyID-WSFSecurityMechanisms] Ellison, Gary, ed. (12 Nov 2003), " Liberty ID-WSF Security Mechanisms", Liberty Alliance Project, http://www.projectliberty/specs.

[LibertyMetadata] P. Davis, ed. "Liberty Metadata Description and Discovery Specification" 1.0, Liberty Alliance Project,. http://www.projectliberty/specs

[LibertySOAPBinding] Hodges, Jeff, Aarts, Robert, eds. (12 Nov 2003). " Liberty ID-WSF: SOAP Binding ," Version 1.0, Liberty Alliance Project http://www.projectliberty/specs

[PAOS] Aarts, Robert, eds. (2003). "Liberty Reverse HTTP Binding," 1.0 , Liberty Alliance Project http://www.projectliberty/specs

[OASISGlossary] Hodges, Jeff, ed. Copyright 2003, OASIS Security Services TC: Glossary.

[SAMLGlossary] Maler, E. & Philpott, R., eds. Copyright 2003, OASIS, Glossary for the OASIS Security Assertion Markup Language (SAML) V1,1, 27 May 2003. http://www.oasis-open.org/committees/download.php/2284/sstc-saml-glossary-1.1-cs-01.pdf

[SASLCram] Nerneberg, L. (2003) "The CRAM-MD5 SASL Mechanism", draft-ietf-sasl-crammd5-00.txt.

Informative

[LibertyID-WSFOverview] Koga, Yuzo & Tourzan, Jonathan , eds. (12 November 2003). "Liberty ID-WSF Overview, Version 1.2, Liberty Alliance Project http://www.projectliberty/specs

[LibertyGloss] Wason, Tom., ed. (12 November 2003). "Liberty Architecture Glossary," Version 1.2. Liberty Alliance Project, <http://www.projectliberty.org/specs/>.