1) Introduction to Liberty Alliance
2) Liberty Identity Federation Framework & SAML 2.0
3) Liberty Identity Web Services Framework
4) Liberty Conformance Program
5) Liberty Identity Service Interface Specifications
Introduction to Liberty Alliance
What is the Liberty Alliance?

The Liberty Alliance is the only global body working to define and drive open technology standards, privacy and business guidelines for federated identity management.
Value Proposition

- Need to be connected anytime, anyplace - without compromising security or control of personal information

- Liberty Alliance provides the technology, knowledge and certifications to build identity into the foundation of mobile and Web-based communications

- 150+ diverse member organizations:
  - Government organizations
  - End-user companies
  - System integrators
  - Software and hardware vendors
Activities

- Technical Specifications and Implementation Guidelines
- Interoperability and conformance testing
- Business guidelines and case studies
- Policy guidelines
- Developer resources and User Groups
- Adoption and evangelism
- Liaison and collaboration with other organizations
**Management Board**

- Defines Mission / Scope
- Drives Execution Timetable
- Allocates Budget

---

**Marketing**

- Determines market requirements for Alliance core specifications
- Drives Business Guidelines
- Drives positioning and outreach to media, analysts and external organizations

**Technology**

- Understands current standards; drives convergence
- Delivers, maintains & evolves core Liberty specifications

**Policy**

- Policy / regulatory environment input
- Helps design audits compliance approach
- Liaison to govt. and external policy organizations

**Services**

- Defines & manages process for creating new service specifications
- Includes gathering of market requirements as well as specification development

**Conformance**

- Defines and manages process for validating vendor interoperability
- Manages Conformance Testing Program
A Network Identity is a user’s overall global set of attributes and identifiers on the network.
Why Federation?

- Issue is not that there are multiple identities — rather the lack of connectivity -> Identity Archipelago
- These problems affect several types of Internet applications:
  - Consumer (portal providers, wireless operators, websites)
  - Intranet
  - Extranet (between trading partners, or between employees and benefit administration sites)
- Need to be able to connect together these identity islands
Why Federated?

**Centralized Model**
- Network identity and user information in single repository
- Centralized control
- Single point of failure
- Links similar systems

**Open Federated Model**
- Network identity and user information in various locations
- No centralized control
- No single point of failure
- Links similar and disparate systems
**Key Concepts**

- **Principal** – a person or “user”, a system entity whose identity can be authenticated
- **IdP** – Identity Provider – a service which authenticates and asserts a Principal’s identity
- **SP** – Service Provider
- **Federation** – The act of establishing a relationship between two entities, an association comprising any number of Service Providers and Identity Providers
- **Single Sign-On (SSO)** – the Principal’s ability to authenticate with one system entity (Identity Provider) and have that authentication honored by other system entities, often Service Providers
- **Circle of Trust** – a group of service providers and identity providers that have business relationships based on Liberty architecture and operational agreements and with whom users can transact business in a secure and apparently seamless environment.
Key Concepts cont'd

- **Pseudonyms** are arbitrary names assigned by the identity or service provider to identify a Principal to a given relying party so that the name has meaning only in the context of the relationship between the relying parties.

- **Anonymity** enables a service to request certain attributes without needing to know the user’s identity. For example, in order to provide personalized weather information to a user, a weather service provider can ask for a user’s zip code using anonymous service request without knowing the identity of that user.
<table>
<thead>
<tr>
<th>Liberty Identity Federation Framework (ID-FF) &amp; Security Assertion Markup Language (SAML) 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enables identity federation and management through features such as identity/account linkage, simplified sign on, and simple session management</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Liberty Identity Services Interface Specifications (ID-SIS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enables interoperable identity services such as personal identity profile service, contact book service, geo-location service, presence service and so on.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Liberty Identity Web Services Framework (ID-WSF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provides the framework for building interoperable identity services, permission based attribute sharing, identity service description and discovery, and the associated security profiles</td>
</tr>
</tbody>
</table>

| Liberty specifications build on existing standards (SAML, SOAP, WS-Security, XML, etc.) |
Liberty Identity Federation Framework & SAML 2.0
Enables identity federation and management through features such as identity/account linkage, simplified sign on, and simple session management.

Liberty Identity Services Interface Specifications (ID-SIS)
Enables interoperable identity services such as personal identity profile service, contact book service, geo-location service, presence service and so on.

Liberty Identity Web Services Framework (ID-WSF)
Provides the framework for building interoperable identity services, permission based attribute sharing, identity service description and discovery, and the associated security profiles.

Liberty specifications build on existing standards (SAML, SOAP, WS-Security, XML, etc.)
Liberty ID-FF

- Privacy-oriented identity federation and SSO
- Allows for authentication actions to be 'reused’ across different sites
- Defines a method of exchanging name identifiers that allows two providers to speak about a “subject” in a common language – the *federated name identifier* – whilst allowing that identifier to be hidden from third parties (opaque identifier)
- Extends the SAML authentication statement, adding the concepts of *session*, and *authentication context*
- Creates an authentication Request/Response protocol
- Additional protocols to provide global single logout, "de-federation", name identifier registration and mapping
- Specifies various profiles for requesting and sending SAML assertions in a web SSO environment, with intermediaries present
- SAML Overview and History
- What is Identity Federation?
- SAML 2.0 Components
- SAML 2.0 Specification Conformance
- SAML 2.0 & Privacy
- SAML 2.0 & Federated Identity Lifecycle
Security Assertion Markup Language

- Developed over the past ~three years in OASIS at the Security Services TC (SSTC)
- Defines assertions which carry statements
- Provides a protocol for disseminating assertions among authorities and relying parties
- A set of profiles to provide Web-based simplified sign-on
- SAML 2.0 ratified as OASIS standard in March 2005
SAML Business Benefits

- Platform and vendor neutrality
- Support for new devices
- Consistent online user experience
- Unified approach to identity federation
- Improved control over identity data helps meet regulatory compliance requirements
- Privacy protection and user consent mechanisms
- Reduced deployment and administrative costs
Liberty ID-FF and SAML

- Liberty defined ID-FF 1.1 *on-top* of and extending SAML 1.0
- SAML 1.1 copied some enhancements from ID-FF 1.1
- Liberty ID-FF 1.2 was based on SAML 1.1
- Somewhat complicated!
- To resolve this parallel development/duplication, Liberty submitted ID-FF 1.2 to the SSTC as input to SAML 2.0
SAML Family Tree

SAML 1.0
Completed: May 2002
OASIS Standard: November 2002

Liberty 1.1
Completed: Jan 2003

SAML 1.1
Completed: May 2003
OASIS Standard: September 2003

Liberty ID-FF 1.2
Completed: Oct 2003

Shibboleth OpenSAML 1.0
Completed: June 2003

Oct-2003: SSTC receives Digital ID World "Balancing Innovation & Reality" award

Shibboleth OpenSAML 1.1
Completed: August 2003

SAML 2.0
Completed: January 2005
OASIS Standard: March, 2005
Liberty & SAML co-evolution

- ID-FF 1.0
- SAML 1.0
  - SOAP
- ID-WSF 1.0
  - ID-FF 1.2
  - SAML 1.1
  - WS Security
  - SOAP
- ID-WSF 2.0
- Other Standards
  - WS Security
  - SAML 2.0
  - SOAP
What is identity federation?

- Agreement between an identity provider and one or more service providers concerning the data using which users will be described
  - By their e-mail address?
  - By their office number and employee Id?
  - By their role or membership in certain groups?
  - By a unique (privacy preserving) identifier known only to the IdP and SP?
- Agreement creation may be accomplished in different ways
  - Business agreements between IdP and SP’s
  - In some cases may require bulk update or synchronization of parts of the user store at both ends
Well known name or attribute

- Well known names include:
  - Email Address
  - X.509 Subject Name
  - Windows Domain Qualified Name
  - Kerberos Principal Name
  - Attribute (e.g., employee number)

- User entry at the IdP and SP(s) are keyed off the name or attribute
  - Privacy preservation is not an issue here
  - Names may be encrypted to protect against intermediaries
- Common use-case in many SAML 1.X deployments
Anonymous user with attributes or roles

- User is never explicitly identified by a persistent identifier
  - A transient identifier is used as the “name” of the user
  - One or more roles or attributes describe the user
    - EmploymentLevel: Manager
    - AccessRights: Platinum
    - MemberOf: BellRingers
  - Access at Service Provider is given against roles or attributes
  - No need to maintain user entry at SP
    - Privacy Preserving as user identity at IDP remains unknown
  - Main use case in Shibboleth and some SAML 1.X deployments
User identified by pseudonym

- User is identified by a persistent randomized string private to IdP and SP pairs
  - Unique handle per service provider
- Privacy-preserving since no information about user is available at SP
- Complicates SP collusion
- Requires IdP and SP to synchronize portions of their user stores
- Affiliations: important sub-case where a single persistent randomized string is shared between a set of Service Providers
- Main use case in ID-FF 1.X specifications and deployments
In March 2005, the OASIS Security Services Technical Committee ratified SAML v2.0 as an OASIS standard.

Going forward, SAML 2.0 is the preferred federation standard for Liberty.

No further development work is planned for ID-FF (although Liberty continues to support through errata and Conformance testing).

Much of the functionality in SAML 2.0 can be traced to its origins in ID-FF.

As SAML 2.0 is the 'future' we focus on SAML 2.0 in this tutorial (mentioning differences as appropriate).
SAML 2.0 Actors and Mechanisms

**Actors**
- Identity Provider
- Session Authority
- Attribute Provider
- Service Provider
- Session Participant
- Attribute Consumer

**Indirect & Direct channels**
- Federation
- SSO
- ID Mgmt
- Session Mgmt
- Attribute Exchange

**Metadata**

**Trust Relationship**
The “core” SAML specification defines the structure and content of **Assertions** – which carry statements about a Principal as asserted by an IDP. An XML Schema defines the structure of the Assertions. Assertions are either requested or just “pushed” out to the Service Provider.

How and which assertions are requested is defined by the SAML **Protocols**. An XML Schema defines the structure of the protocol messages.

How the SAML Protocol messages are actually communicated over lower-level messaging protocols (such as HTTP or SOAP) are defined by **Bindings**.

SAML Protocols and Bindings, together with the structure of Assertions, can be combined together to create a **Profile**. In general Profiles can be thought of a satisfying a particular use case, for example the Web Browser SSO profile.
Assertions

- Assertions are declarations of fact, according to the issuer.
- SAML assertions are compounds of one or more of three kinds of “statement” about a “subject”:
  - Authentication
  - Attribute
  - Authorization decision
- You can extend SAML to make your own kinds of assertions and statements.
- Assertions can be digitally signed, using XML Signature.
• Carry information about the issuer and the subject
• “Conditions” under which assertion is valid
  ▪ SAML clients must reject assertions containing unsupported conditions
  ▪ Special kind of condition: assertion validity period
• Additional “advice”
  ▪ To explain how or why the assertion was made – for example, authorization assertion might be based on a supplied attribute assertion
• Can be used as tokens (via OASIS WS-Security SAML Token Profile) to secure SOAP messages, binding statements made in the assertion to the content of a SOAP message, via XML Signature of that message
Example Assertion

```xml
<saml:Assertion
    Version="2.0"  ID="buGxcG4gIL" IssueInstant="2002-06-19T17:05:37.795Z">
    <saml:Issuer Format="entity">IDP.com</saml:Issuer>
    <ds:Signature/>
    <saml:Conditions NotBefore="2002-06-19T17:00:37.795Z"
        NotOnOrAfter="2002-06-19T17:10:37.795Z"/>
    <saml:Subject>
        <saml:NameID NameQualifier=http://www.acompny.com Format="persistent">jsdyfhs</saml:NameID>
        <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>
    </saml:Subject>
    <saml:AuthnStatement
        AuthnInstant="2002-06-19T17:05:17.706Z">
        <AuthnContext>
            <AuthnContextClassRef>Password</AuthnContextClassRef>
        </AuthnContext>
    </saml:AuthnStatement>
</saml:Assertion>
```
Protocols

• **Authentication Request Protocol**: Defines a `<AuthnRequest>` message that causes a `<Response>` to be returned containing one of more assertions pertaining to a Principal. Typically the `<AuthnRequest>` is issued by a Service Provider with the Identity Provider returning the `<Response>` message.

• **Artifact Protocol**: Provides a mechanism to obtain a previously created assertion by providing a reference. In SAML terms the reference is called an “artifact”. Thus a SAML protocol can refer to an assertion by an artifact, and then when a Service Provider obtains the artifact it can use the artifact Protocol to obtain the actual assertion.

• **Single Logout Protocol**: Defines a request that allows near-simultaneous logout of all sessions associated by a Principal. The logout can be directly initiated by the Principal or due to a session timeout.
Protocols cont'd

• **Name Identifier Management Protocol**: Provides mechanisms to change the value or format of the name of a Principal. The issuer of the request can be either the Service Provider or the Identity Provider.

• **Name Identifier Mapping Protocol**: Allows a SP to query an Identity Provider for a 'mapped' identifier understandable at some other provider.

• **Assertion Query and Request Protocol**: Defines a set of queries by which existing SAML assertions may be obtained. The query can be on the basis of a reference, subject or the statement type.
• **Bindings:** Details exactly how the SAML protocol maps onto the transport protocols. For instance, the SAML specification provides a binding of how SAML request/responses are carried with SOAP exchange messages.

• The defined bindings are:

  • **SAML SOAP Binding:** Defines how SAML protocol messages are transported within SOAP 1.1 messages. In addition, it also defines how the SOAP messages are transported over HTTP.

  • **HTTP Redirect Binding:** Defines how SAML protocol messages can be transported using HTTP redirect messages (i.e. 302 status code responses)

  • **HTTP Artifact Binding:** Defines how a reference to a SAML request or response (i.e. an artifact) is transported by HTTP. Defines two mechanisms, either an HTML form control, or a query string in the URL.
• **HTTP POST Binding:** Defines how SAML protocol messages can be transported within the base64-encoded content of an HTML form control.

• **Reverse SOAP (PAOS) Binding:** Defines a multi-stage SOAP/HTTP message exchange that permits a HTTP client to be a SOAP responder. Used in the Enhanced Client and Proxy Profile and particularly designed to support WAP gateways.

• **SAML URI Binding:** Defines how a SAML assertion can be retrieved by URI resolution. The specific format of the message depends on the underlying transport protocol.
The HTTP Redirect binding is intended for cases in which the SAML requester and responder need to communicate using an HTTP user agent as an intermediary (either because they don't have a direct channel or because a user interaction is required).

Binding defines a mechanism by which SAML protocol messages can be transmitted within URL parameters.

Permissible URL length in browsers is unpredictably limited. Therefore, specialized encodings are needed to carry XML messages on a URL.

Messages are transmitted using the HTTP GET method.
<samlp:LogoutRequest
     xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
     xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
     ID="d2b7c388cec36fa7c39c28fd298644a8" IssueInstant="2004-01-21T19:00:49Z" Version="2.0">
  <Issuer>https://IdentityProvider.com/SAML</Issuer>
  <NameID Format="persistent">005a06e0-ad82-110d-a556004005b13a2b</NameID>
  <samlp:SessionIndex>1</samlp:SessionIndex>
</samlp:LogoutRequest>

HTTP/1.1 302 Object Moved
Date: 21 Jan 2005 07:00:49 GMT
Location: https://ServiceProvider.com/SAML/SLO/Browser?SAMLRequest=fVFdS8MwFH0f7D%2BUvGdNsq62oSsIQyhMESc%2B%2BJYlmRbWpObeyvz3puv2IMjyFM7HPedyK1DdsZdb%2F%2B1KA5G1EWeZaogSQMw2MYBKodrIhjLKOUNU8FdeSsZkVr6T5M0GiHMjvWCKnqZXZ2OoPxF7kGnaGOuwxZ%2Fm4L9bY8NC
3D&RelayState=0043bfc1bc45110dae17004005b13a2b&SigAlg=http%3A%2F%2Fwww.w3.org%2F200%2F09%2Fxmldsig%23rsasha1&Signature=NOTAREALSIGNATUREBUTHEREALONEWOULDDOHERE
Content-Type: text/html; charset=iso-8859-1
• The SAML SOAP binding describes how SAML request/response message exchanges are mapped into SOAP message exchanges. SAML request-response protocol elements are enclosed within the SOAP message body.
• Like SAML, SOAP can be used over multiple underlying transports. The SAML SOAP Binding binding has protocol-independent aspects, but also calls out the use of SOAP over HTTP as REQUIRED.
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: nnnn

<SOAP-ENV:Body>
  <samlp:Response ID="_6c3a4f8b9c2d" Version="2.0" IssueInstant="2004-03-27T08:42:00Z">
    <ds:Signature> ... </ds:Signature>
    <Status>
      <StatusCode Value="..."/>
    </Status>
  </samlp:Response>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

Both Response and Assertion can be signed
Profiles

- **Profiles**: define how the SAML assertions, protocols and bindings are combined to address particular use case requirements.

  - **Web Browser SSO Profile**: Defines how a Web Browser supports SSO, when using `<AuthnRequest>` protocol messages in combination with HTTP Redirect, HTTP POST and HTTP Artifact bindings.

  - **Enhanced Client and Proxy (ECP) Profile**: Defines how `<AuthnRequest>` protocol messages are used when combined with the Reverse-SOAP binding (PAOS). Designed to support mobile devices front-ended by a WAP gateway.

  - **Identity Provider Discovery Profile**: Defines how a service provider can discover which identity providers a principal is using with the Web Server.

  - **Single Logout Profile**: A profile of the SAML Single Logout protocol is defined. Defines how SOAP, HTTP Redirect, HTTP POST and HTTP Artifact bindings may be used.
Profiles cont'd

• **Name Identifier Management Profile**: Defines how the Name Identifier Management protocol may be used with SOAP, HTTP Redirect, HTTP POST and HTTP Artifact bindings.

• **Artifact Resolution Profile**: Defines how the Artifact Resolution protocol uses a synchronous binding, for example the SOAP binding.

• **Assertion Query/Request Profile**: Defines how the SAML query protocols (used for obtaining SAML assertions) use a synchronous binding such as the SOAP binding.

• **Name Identifier Mapping Profile**: Defines how the Name Identifier Mapping protocol uses a synchronous binding such as the SOAP binding.
How they fit together

PROFILES
(How SAML protocols, bindings and/or assertions combine to support a defined use case)

BINDINGS
(how SAML Protocols map onto standard messaging or communication protocols)

PROTOCOL
(Request/Response pairs for obtaining Assertions and Federation Management)

ASSERTIONS
(Authentication, Attribute and Authorization Information)
SAML Specification Conformance

- Specification Conformance programs test for compliance with a standard.

- Each SAML profile (other than attribute profiles) describes details of selected SAML message flows and can also be viewed as indivisible functionality that could be implemented by a software component.

- For each profile, the relevant SAML V2.0 message flows are defined, and for each message flow the set of possible bindings. The combination of profile, message exchange and a selected binding is termed a SAML V2.0 feature.

- A number of different operation modes are identified. The conformance matrix describes the feature set that must be implemented by each different operational mode.
Conformance Modes

- SAML 2.0 defines a number of different 'modes' in which an implementation may operate.
  - IdP – Identity Provider
  - IdP Lite – Identity Provider Lite
  - SP – Service Provider
  - SP Lite – Service Provider Lite
  - ECP – Enhanced Client/Proxy
  - SAML Attribute Authority
  - SAML Authorization Decision Authority
  - SAML Authentication Authority
  - SAML Requester.
The conformance matrix describes the feature set that must be implemented by an implementation claiming any particular operational mode(s).
Privacy and User Consent

- Privacy
  - SAML 2.0 includes recommendations for privacy preservation if and when desired
  - Main idea is that Identity providers need not release any personal information about users to service providers

- User Consent
  - SSO protocol includes ability to query and record user-consent
  - Identity providers and service providers can choose to provide services based on whether user-consent was obtained and recorded
Federated Identity Life-Cycle

- These are the significant milestones in a federated lifecycle
- Following slides explore how SAML 2 supports them
SAML profiles require agreements between system entities regarding identifiers, binding support and endpoints, certificates and keys, etc.

A metadata specification describes this information in a standardized way.

Metadata specification defines an extensible format for SAML system entities, organized by roles that reflect SAML profiles.

Such roles include that of SSO Identity Provider, SSO Service Provider, Affiliation, Attribute Authority, Attribute Requester, and Policy Decision Point.

Origin and document verification can be ensured through use of signatures.
<EntityDescriptor
entityID="https://ServiceProvider.com/SAML">
<ds:Signature>...</ds:Signature>
<SPSSODescriptor AuthnRequestsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<KeyDescriptor use="signing">
<ds:KeyInfo>
<ds:KeyName>ServiceProvider.com SSO Key</ds:KeyName>
</ds:KeyInfo>
</KeyDescriptor>
</SPSSODescriptor>
<SingleLogoutService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://ServiceProvider.com/SAML/SLO/SOAP"/>
<SingleLogoutService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://ServiceProvider.com/SAML/SLO/Browser"
ResponseLocation="https://ServiceProvider.com/SAML/SLO/Response"/>

Two endpoints for receiving SLO messages distinguished by relevant binding
Two endpoints for consuming assertions distinguished by relevant binding
(Pseudonymous) Federation

- **User visits SP** and authenticates
  - Sees list of possible IdP federation choices, or
  - Is offered federation based on IdP Introduction
- **User sent to appropriate IDP**
- **At IdP, User authenticates**
- **Now that both SP and IDP 'know' who the User is, they federate him/her by agreeing on a persistent pseudonym**
  - User is sent back to SP, now federated
Federation

User

Login/Authenticate

Introduction cookie

Login/Authenticate

You have a cookie from IDP, federate accounts?

Yes, federate my accounts

Redirect to IDP with Authentication Request

AuthnRequest

IDP generates NameID & issues AuthnStatement

Redirect to SP

Here is artifact for <Assertion> @ IDP

SOAP

SOAP

SP

Note: here the user authenticates to the IDP first – this does not change the overall flow

The SP specified that the IDP could create a new identifier by specifying AllowCreate = "true" in the <NameIDPolicy>
Single Sign-on Profile

- User visits SP
  - Sees list of possible IdP choices, or
  - The IDP is inferred based on common domain cookie
- User sent to appropriate IDP
- At IdP, User authenticates
- IDP generates authentication statement using the identifier previously established at federation time
Single Sign-on

- Instead of the SP directly authenticating the user the SP queries the IdP and the IdP issues an authentication assertion.

1. User attempts to access SP
2. User authentication request (from SP)
3. Authentication Assertion issued
4. Authentication Assertion sent

Identity Provider

Service Provider
SAML 2.0 defines an XML Schema by which the IDP can assert the context of the SAML assertions it issues.

Assumption is that the SP cares about the details of how the principal authenticated (and related info).

Liberty also defined specific Authentication Context ‘classes’ – patterns against which an IdP can claim conformance.

Classes are designed to be representative of today’s (and future) authentication technologies, for instance:

- Password over SSL
- Smartcard
- Pre-paid Mobile Login
- Biometric
As well as allowing the IDP to describe how the user authenticated (i.e. What already happened), the SP is able to state its requirements to the IDP when it asks for an authentication statement (i.e. What must happen)

The SP specifies general authentication context requirements or refers to previously agreed classes using the `<RequestedAuthenticationContext>` element in the `<AuthnRequest>` sent to the IDP

SP can do this initially, or subsequently if its policy dictates that a resource requires different authentication context than that obtained with the first assertion (e.g. 'Up-authentication')
Based on their authentication to the IDP, a principal may establish sessions at multiple SPs (through separate Single Sign-Ons to each of them).

At some later time, the principal may wish to terminate his or her session either with an individual session participant, or with all session participants SPs in a given session managed by the session authority IDP.

For the latter case (e.g. Global log-out), either the IDP or a particular SP can request that all the principal's other sessions can be ended.
Single Logout Profile

- Can be initiated at either the IdP or a particular SP
- Profile allows the SLO protocol messages be delivered through either synchronous (e.g. SOAP) or asynchronous “front-channel” (e.g. HTTP Redirect, POST, or artifact) bindings.
- An identity provider acting as a session authority may itself act as a session participant in situations in which it is the relying party for another identity provider's assertions regarding that principal.
Single Logout Sequence

User

IDP

User clicks 'Logout'

IdP logout web page is displayed

Repeat for each SP

Single logout confirmed

Single logout request

Process logout

Single logout response
Single Logout (SP initiated)

1. `<LogoutRequest>` issued by Session Participant

2. Identity Provider determines session participants: Are any other system entities participating in this session?

3. `<LogoutRequest>` issued to other session participant, if another session participant exists.

4. Principal's local session is terminated, and `<LogoutResponse>` returned.

5. `<LogoutResponse>` issued to originating Session Participant.

Whether the user Agent is involved depends on the binding.

Steps 3 and 4 are repeated for each “other” session participant discovered in Step 2.
Assumption is that an IDP has established some form of persistent identifier for a principal with a SP, allowing them to share a common identifier for some length of time.

Subsequently, the identity provider may wish to notify the service provider of a change in the format and/or value that it will use to identify the same principal in the future.

Alternatively the service provider may wish to attach its own "alias" for the principal in order to ensure that the identity provider will include it when communicating with it in the future about the principal.

Finally, one of the providers may wish to inform the other that it will no longer issue or accept messages using a particular identifier.
An identity provider wishing to change the value and/or format of the identifier that it will use when referring to the principal, or to indicate that a name identifier will no longer be used to refer to the principal, informs service providers of the change by sending them a `<ManageNameIDRequest>` message.

Protocol can be combined with a synchronous binding, such as the SOAP binding, or with asynchronous "front-channel" bindings, such as the HTTP Redirect, POST, or Artifact bindings.

A front-channel binding may be required, for example, in cases in which direct interaction between the user agent and the responding provider is required in order to effect the change.
Name Identifier Management Profile

1. <ManageNameIDRequest> issued by requesting provider

2. <ManageNameIDResponse> returned by responding provider

- The grayed-out user agent illustrates that the message exchange may pass through the user agent or may be a direct exchange between system entities, depending on the SAML binding used to implement the profile.
- Either provider can request that the federation be terminated by including a <Terminate> element rather than a new value for the NameId in the <ManageNameIDRequest>
When an SP that shares an identifier for a principal with an IDP wishes to obtain a name identifier for the same principal in a particular format or federation namespace, it can send a request to the IDP using the NameIDMappingRequest.

For example, a SP that wishes to communicate with another SP with whom it does not share an identifier for the principal can use an IDP that shares a (different) identifier for the principal with both SPs to map from its own identifier to a new identifier with which it can communicate with the second SP.

Except in special cases, the IDP will return an encrypted (for the target SP) Name Identifier in its NameIDMappingResponse.
The requester MUST use a synchronous binding, such as the SOAP binding [SAMLBind], to send the request directly to the identity provider. The requester MUST authenticate itself to the identity provider, either by signing the `<NameIDMappingRequest>` or using any other binding-supported mechanism.
The artifact resolution protocol provides a mechanism by which SAML protocol messages can be transported in a SAML binding by reference instead of by value.

Both requests and responses can be obtained by reference using this specialized protocol.

A message sender, instead of binding a message to a transport protocol, sends a small piece of data called an artifact using a binding.

The most common use for this mechanism is with bindings (e.g. HTTP Redirect) that cannot easily carry a message because of size constraints, or to enable a message to be communicated via a secure channel between the SAML requester and responder, avoiding the need for a message signature.
1. `<ArtifactResolve>` issued by requesting provider

2. `<ArtifactResponse>` returned by responding provider

- The `<ArtifactResolve>` message is used to request that a SAML protocol message be returned in an `<ArtifactResponse>` message by specifying an artifact that represents the SAML protocol message.
- The original transmission of the artifact is governed by the specific protocol binding that is being used.
Liberty Identity Web Services Framework

Platform for permissions-based attribute sharing
Liberty’s Architecture

Liberty Identity Federation Framework (ID-FF) & Security Assertion Markup Language (SAML) 2.0

Enables identity federation and management through features such as identity/account linkage, simplified sign on, and simple session management.

Liberty Identity Services Interface Specifications (ID-SIS)

Enables interoperable identity services such as personal identity profile service, contact book service, geo-location service, presence service and so on.

Liberty Identity Web Services Framework (ID-WSF)

Provides the framework for building interoperable identity services, permission based attribute sharing, identity service description and discovery, and the associated security profiles.

Liberty specifications build on existing standards (SAML, SOAP, WS-Security, XML, etc.)
What is ID-WSF?

- A framework for locating and invoking identity-based Web services to provide a simplified & customized online experience

- Identity-based Web services:
  - Are associated with a Principal's Identity (e.g. My Calendar Service)
  - Can be Invoked using a Principal’s Identity

- Permissions-based Attribute Sharing
  - Invoking Services under control of user
  - Service Requestor doing so on behalf (either directly or indirectly) of user.
What is an identity service?

- A service that presents external interface to some aspect of my online identity
- Typically exposed as a SOAP-based web service
- Allows for greater control of my identity by reducing duplication throughout the network
- Increases privacy because fewer personal information items are released, e.g.:
  - An “Inbox” service might allow me to receive "permission-based" marketing without releasing my email address
  - A “Payment” service would allow payments to be made without releasing my credit card number
ID-WSF – New Concepts

- **Web Services Client (WSC):** typically, the invoker/consumer of an identity-based service.
- **Web Services Provider (WSP):** typically, the provider of an identity-based service.
- **Data Services Template (DST):** provides an extensible framework to produce new Identity-based Services above the protocol stack, allowing interoperability e.g.: ID-Personal Profile and ID-Employee Profile.
- **Discovery Service (DS):** Facilitates the registration and subsequent discovery of Identity-based services.
- **Interaction Service (IS):** allows WSPs to obtain authorizations and information directly from users.
- **Authentication Service (AS):** Authenticates Principles and provides appropriate credentials for accessing ID-WSF systems (analogous to IdP in ID-FF).
1) ID-FF/SAML2 can be used to bootstrap into ID-WSF
   • SP gets Assertion which can include bootstrap information for invoking DS
   • SP then acts as WSC to invoke ID-WSF services

2) Authentication Service (AS) provides a SOAP interface into the IdP to perform “ID-FF like” operations (non-web)
   • Results in ID-FF/SAML2 assertion provided back to client
   • Client can then invoke DS

3) WSF specifies how SAML Assertions can be used to communicate identity information between WSF actors
ID-FF/SAML2 and ID-WSF together

**ID-FF**: The SP uses ID-FF to obtain the identity credential for Jane.

**ID-WSF**: The SP (acting as a WSC) uses IF-WSF to invoke services at the WSPs on Jane’s behalf.
1. Register RO for Geo Svc

IDP

2. Unauthenticated P
   GET's
   some info that requires
   P's geolocation

DS

3. Use SAMLv2 to req SSO of
   P

4. Ret Authn assn w/P's DS
   RO

SP/WSC

5. Req RO of P's Geo Svc

6. Ret ROv2 of P's Geo Svc +
   SAML Assertion

7. Query P's Geo Svc
   using info from RO and including
   SAML assertion

WSP

P's Geo Svc

P browser
ID-WSF Core Components

- Discovery Service
- Service Invocation (SOAP Binding)
- Interaction Service
- Data Services Template
- Security Mechanisms
- Authentication Service
- Privacy
Registry for services associated with an identity
WSPs register the identity services they host at the DS so that WSCs can subsequently discover them
Translates and protects tokens/identifiers as necessary to allow one entity to safely communicate with a second entity
Allows for multiple providers of the same service
Data specific discovery
  - Retrieve the wallet service that has a credit card
  - Retrieve the profile service that has an age
• A resource offering is the association of a particular user's resource and a service instance.
• This association is necessary as there is a many-to-many relationship between resources and service instances.
• Typically, a single service instance will serve many resources. For example, a personal profile service provider would typically serve up many profiles behind a single service instance, as having a separate protocol endpoint for each profile would be impractical.
• Thus, a ResourceOffering element is defined to associate a resource with a service instance that provides access to that resource.
Resource Offering cont'd

- ResourceOffering's (RO) are fundamental to WSF – they are the *currency* for WSF transactions
- WSPs register the Ros for the services they host at the corresponding principal's Discovery Service using the DS specification's `<Modify>` message
- WSCs query the DS for RO of a particular service type (e.g. Contact book) using the DS `<Query>` message
- DS determines if it has any matching ROs and, if so, returns any in its `<QueryResponse>` message
- DS may also return credentials for the WSC to use in its subsequent query to the WSP
Resource Offering

<ds:ResourceOffering entryID="aolradio" xmlns:ds="urn:liberty:disco:2003-08">
  <ds:ServiceInstance>
    <ds:ServiceType>http://radio.aol.com/xsd/2.0/ns/</ds:ServiceType>
    <ds:Description>
      <ds:CredentialRef>i2e02078eb93d07101902b8b60a01</ds:CredentialRef>
    </ds:Description>
  </ds:ServiceInstance>
</ds:ResourceOffering>
Liberty Identity Web Services Framework (ID-WSF) messages are designed so that they may be mapped onto various transport or transfer protocols.

Do not intrinsically address specific aspects of message exchange such as: to which system entity the message is to be sent, message correlation, the mechanics of message exchange, or security context.

WSF defines a mapping onto SOAP 1.1, an XML-based messaging protocol.

Neither does SOAP itself define the specific message exchange aspects mentioned above, but does offer an extensibility model that may be used to define message components that do address such message exchange specifics.

SOAP extensibility is effected by adding message components to the portion of the SOAP message called the Header.
SOAP Binding headers

- **Message Correlation**: SOAP does not define a mechanism to correlate one SOAP message with another message, such as in a request-response paradigm.

- **Provider and affiliation declaration**: Participants in ID-* interactions may declare themselves by their Provider ID, as well as their Affiliation ID if appropriate.

- **Processing context**: An ID-* requester may need to express additional context for a given request, for example indicating that the requester expects to make such requests in the future when the Principal may or may not be online.

- **Consent Claims**: ID-WSF-based entities may wish to claim whether they obtained the Principal’s consent for carrying out any given operation, such as updating a Principal’s Personal Profile entry.
Usage Directives: ID-WSF-based entities may wish to indicate their policies for handling data at the time of data request, and entities releasing data may wish to specify their policies for the subsequent use of data at the time of data release.

Timeout: The <Timeout> header block is defined in this specification to allow the receiver of an ID-* message to indicate that processing of the received message failed due to a timeout condition.

Credentials context: The receiver of an ID-* message might indicate that credentials supplied in the request did not meet its policy i allowing access to the requested resource.
Example

```xml
<S:Envelope>
  <S:Header>
    <sb:Correlation S:mustUnderstand="1"
      messageID="uuid-23235ad23-21390feda-230809-89898"
      timestamp="2003-06-06T12:10:10Z" />
    <sb:Provider providerID="example.com"... />
    <sb:UsageDirective id="directive1000"... >
      <PrivacyPolicyReference ... >
        Privacy Policy Information
      </PrivacyPolicyReference>
    </sb:UsageDirective>
    <wsse:Security>
      <samlp:Assertion... >
        Assertion data goes here
      </samlp:Assertion>
    </wsse:Security>
  </S:Header>
  <S:Body>
    Request Messages go here
  </S:Body>
</S:Envelope>
```

- **Uniquely identifies message and when it was generated and, if necessary, what message it is related to.**
- **Provides Directives on how the data will be used by the client. The Policies themselves are not defined by Liberty.**
- **Authenticates requestor and, in many cases, the principal on behalf of who the requestor is making the request.**
- **SIS Service Request (typically includes the resource ID returned from discovery service).**
Status codes

- The `<Status>` element is used to convey status codes and related information between WSP and WSC.
- SOAP Binding defines base status codes (other specs may define additional)
- Examples
  - `CannotHonourUsageDirective` - The receiver is unable or unwilling to honor the stipulated usage directive.
  - `InappropriateCredentials` - The sender has submitted a request that does not meet the needs of the receiver. The receiver may also indicate credentials that are acceptable to them.
  - `ProcessingTimeout` - The sender is indicating that processing of the request has failed due to the processing taking longer than the `maxProcessingTime` specified on the request `<Timeout>` header block.
Interaction Service

- Enables WSP Interaction with User
  - Typically WSP does not have direct user access
  - Real-time consent, data, and/or decision Collection
- Multiple Methods
  - Allow trusted party (SP) to interact
  - Request that SP re-direct user’s browser to WSP
  - Direct interaction without involving SP
Interaction Service

Jane using a browser

Buy some electronics

CoolToys.com

Charge $5,235.03

Transaction Approved

Hey, CoolToys.com wants to... Is this OK?

Yes, that's fine.

Ask Jane if ... Is OK?

Jane says it's OK.

Hmm... Jane's rules say that if the charge is over $300 to confirm with her...
<InteractionRequest xmlns="urn:liberty:is:2003-08">
    <ResourceID data="d8ddw6dd7m28v628"/>
    <Inquiry title="Profile Provider Question">
        <Help moreLink="http://location.example.com/help/consent">
            Example.com is requesting your address. Please pick one of the provided options.
        </Help>
        <Select name="locationchoice">
            <Label>Do you want to share your address with Example.com?</Label>
            <Value>no</Value>
            <Item label="Not this time" value="no">
                <Hint>We won’t give out your address but we’ll ask you again next time.</Hint>
            </Item>
            <Item label="Yes, once" value="yes">
                <Hint>We will share your address but will ask again next time.</Hint>
            </Item>
        </Select>
    </Inquiry>
</InteractionRequest>
Data Service Template

- Data Service Template (DST) provides generic mechanisms for interacting with data services.
- Data Service Template provides protocols for the query and modification of data attributes related to a Principal that are exposed by a data service.
- Defines some guidelines, common XML attributes and data types for data services.
- Defines how a WSC can subscribe to be notified of changes to particular data attributes if and when it changes (or other criteria).
- Different SIS services may chose to build on the common DST layer.
Security Mechanisms

- Sec Mech spec combines and profiles different security specifications (SSL/TLS, WS-Sec, STP) to ensure required security characteristics for SOAP messages.
- This includes:
  - validation of the message: transport or message level authentication
  - the communication of info that could aid in performing an authorization decision
  - Mechanisms for confidentiality and non-repudiation
Security Mechanisms cont'd

- WSC SOAP messages secured through a combination of transport level (e.g. SSL) & message level (e.g. WS-Security) protection mechanisms
- Liberty defines URIs for such combinations
  - *urn:liberty:security:2004-12:TLS:SAMLV2* indicates that the WSC will authenticate to the WSP through a SAML 2.0 Assertion embedded within the SOAP Header, the message sent over a TLS-Protected pipe
- When a WSP registers a ResourceOffering at a DS, it indicates what combinations it requires/supports by specifying appropriate URIs
- When a WSC queries the DS for the principal's services, it can include which URIs it can support
- DS filters ResourceOfferings appropriately to ensure that an intersection of capabilities can be found
Authentication Service

- Allows general identity authentication over SOAP
- SASL Based SOAP Authentication
  - General purpose authentication exchange mechanism
  - Existing defined support for multiple mechanisms
    - CRAM-MD5, PLAIN, X.509, SECURID, etc.
    - Extensible for future methods/mechanisms
- Client->Server or Server->Server Authentication
- Can bootstrap to Discovery Service
Authentication Service Negotiation

Client:  
<SASLRequest mechanism="CRAM-MD5  PLAIN"/>

Server:  
<SASLResponse serverMechanism="CRAM-MD5">  
  <Status code="continue" />  
  <Data>  
    923992ad4d2...  
  </Data>  
</SASLResponse>

Client:  
<SASLRequest mechanisms="CRAM-MD5">  
  <Data>  
    A87B3237d73282...  
  </Data>  
</SASLRequest>

Server:  
<SASLResponse>  
  <Status code="success" />  
  <ResourceOffering>  
    Resource Offering for the DS goes here...  
  </ResourceOffering>  
  <Credentials>  
    <samlp:Assertion ...>  
      Assertion data goes here  
    </samlp:Assertion>  
  </Credentials>  
</SASLResponse>
Liberty-enabled User Agents or Devices are SOAP capable clients

(LUAD-)WSCs may need to interact with 'vanilla' SPs (that may not be SOAP/WSF capable)

The ID-WSF Single Sign-On Service is a profile of the ID-FF Single Sign-On and Federation Protocol to address this mismatch

The mechanism is based on two steps.

First, a (LUAD-)WSC wishing to interact with some SP can use the Authentication Service at an Identity Provider to obtain security tokens.

Next, the (LUAD-)WSC invokes the Single Sign-On Service at the Identity Provider in order to obtain an authentication assertion to convey to the SP, thus enabling Liberty-SSO-enabled, vanilla, web-based interactions with that SP.
Privacy - Usage Directives header

- Allows for indication of associated privacy policy in both information request or reply.
- A `<UsageDirective>` appearing in a request message expresses intended usage.
- A `<UsageDirective>` appearing in a response expresses how the receiver of the response is to use the response data.
- A `<UsageDirective>` in a response message containing no response message data, a fault response for example, may be used to express policies acceptable to the responder.
- A message containing Usage Directive can be signed using XMLDsig and thus bind together the released personal information and associated policy.
Example Usage Directive

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <xml ns:sb="urn:liberty:wsf:soap:p-bind:1.0">
    <xml ns:pp="rn:liberty:idpp:1.0 />
    <S:Header>
      <sb:UsageDirective
        id="directive1000"
        ref="#datarequest001"
        S:mustUnderstand="1">
        <cot :PrivacyPolicyReference
      </sb:UsageDirective>
    </S:Header>
    <S:Body>
      ...
    </S:Body>
  </S:Envelope>
This header block is used to explicitly claim that the Principal consented to the present interaction.

Liberty defines one well-known URI Liberty implementers and deployers MAY use to indicate positive Principal consent was obtained with respect to whatever interaction is underway or being initiated.

This URI is known as the "Principal Consent Obtained" URI (PCO). The value of this URI is: urn:liberty:consent:obtained

This URI does not correspond to any particular Consent Agreement Statement. Rather, it simply states that consent was obtained. The full meaning and implication of this will need to be derived from the execution context.
XML Walk Through
1. SASL Request to IDP

POST /soap/IDPAS HTTP/1.1
Host:
Accept: text/xml
Expect: 100-continue
User-Agent: Sen
Content-Length: 370
Content-Type: text/xml

SOAPACTION:
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:sa="urn:liberty:sa:2004-04"
    xmlns:sb="urn:liberty:sb:2003-08"
    xmlns="http://liberty.ws/soap-security">
  <S:Header>
    <sb:Correlation
        messageID="C8797D0D-9020-07FC-AF0A-5622C01F4A61"
        timestamp="2004-09-27T14:50:45Z"/>
  </S:Header>
  <S:Body>
    <sa:SASLRequest
      mechanism="PLAIN ANONYMOUS CRAM-MD5"
      advisoryAuthnID="012345678901234"/>
  </S:Body>
</S:Envelope>
2. SASL Response

HTTP/1.1 100 Continue
HTTP/1.1 200 OK
Date: Mon, 27 Sep 2004 18:50:59 GMT
Server: Jetty/4.2.21 (Windows 2000/5.0 x86 java/1.4.2_04)
Content-Type: text/xml
Content-Length: 466

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:sb="urn:liberty:sb:2003-08"
    xmlns="urn:liberty:sa:2004-04">
    <S:Header>
        <sb:Correlation s:mustUnderstand="1"
            id="thisCorrHdr.2345"
            refToMessageID="C8797D0D-9020-07FC-AF0A-5622C01F4A61"
            timestamp="2004-02-03T22:12:27Z"/>
    </S:Header>
    <S:Body>
        <SASLResponse serverMechanism="PLAIN"
            xmlns="urn:liberty:sa:2004-04">
            <Status code="continue"/>
        </SASLResponse>
    </S:Body>
</S:Envelope>
3. Authenticate

POST /soap/IDPAS HTTP/1.1
Host: 
Accept: text/xml 
Expect: 100-continue 
User-Agent: Sen 
Content-Length: 455 
Content-Type: text/xml 
SOAPAction: 
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"/>
  <S:Header>
    <sb:Correlation xmlns:sb="urn:liberty:sb:2003-08"
      messageId="D93D6E95-A1F0-4A50-7938-1D1FA9D77918"
      refToMessageID="i48b4353f50aca1494665d61b93498c885449c868"
      timestamp="2004-09-27T14:51:00Z"/>
  </S:Header>
  <S:Body>
    <sa:SASLRequest xmlns:sa="urn:liberty:sa:2004-04"
      mechanism="PLAIN">
      <sa:Data>ADAxMjM0NTY3ODkwMTIzNAAwMTIzNDU2Nzg5MDEyMzQ=</sa:Data>
    </sa:SASLRequest>
  </S:Body>
</S:Envelope>
4. Authentication Service provides Resource Offering for Discovery Service

HTTP/1.1 200 OK
Date: Mon, 27 Sep 2004 19:01:51 GMT
Server: Jetty/4.2.21 (Windows 2000/5.0 x86 java/1.4.2_04)
Content-Type: text/xml
Content-Length: 1169
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <sb:Correlation s:mustUnderstand="1"
      xmlns:sb="urn:liberty:sb:2003-08"
      id="thisCorrHdr.3456"
      messageId="e44b8753f05abb1499657e61b83378c775219a768"
      refToMessageID="D93D6E95-A1F0-4A50-7938-1D1FA9D77918"
      timestamp="2004-02-03T22:12:27Z" />
  </S:Header>
</S:Envelope>
4 ... Authentication Service provides Resource Offering for Discovery Service

```xml
<S:Body>
  <SASLResponse serverMechanism="PLAIN" xmlns="urn:liberty:sa:2004-04">
    <Status code="sa:OK"/>
    <ResourceOffering entryID="1">
      <ResourceID>http://example.nokia.com:8080/soap/012345678901234</ResourceID>
      <ServiceInstance>
        <ServiceType>urn:liberty:disco:2003-08</ServiceType>
        <ProviderID>http://example.nokia.com:8080/soap/</ProviderID>
        <Description>
          <CredentialRef>2sxJu9g/vvLG9sAN9bKp/8q0NKU=</CredentialRef>
          <Endpoint>http://example.nokia.com:8080/soap/IDPDS</Endpoint>
        </Description>
      </ServiceInstance>
    </ResourceOffering>
    <Credentials notOnOrAfter="2004-09-28T18:28:44Z">
      <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
        AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU=">
        ...
      </saml:Assertion>
    </Credentials>
  </SASLResponse>
</S:Body>
</S:Envelope>
```
5. Contact Discovery Service for Service

POST /soap/IDPDS HTTP/1.1
Accept: text/xml
Content-Length: 679
Content-Type: text/xml

SOAPAction:
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <sb:Correlation xmlns:sb="urn:liberty:sb:2003-08"
      messageId="F5683B9F-DF73-AA9E-D01A-82CFFE8F6341"
      timestamp="2004-09-27T14:51:01Z"/>
      <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
        AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="/>
      ...
    </wsse:Security>
  </S:Header>
</S:Envelope>
5… Contact Discovery Service for Service

```xml
<S:Body>
  <Query xmlns="urn:liberty:disco:2003-08">
    <ResourceID>http://example.nokia.com:8080/soap/012345678901234</ResourceID>
    <RequestedServiceType>
      <ServiceType>urn:nokia:ws:samples:hello</ServiceType>
    </RequestedServiceType>
  </Query>
</S:Body>
</S:Envelope>
```
HTTP/1.1 200 OK
Date: Mon, 27 Sep 2004 19:01:53 GMT
Server: Jetty/4.2.21 (Windows 2000/5.0 x86 java/1.4.2_04)
Content-Type: text/xml
Content-Length: 1066

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <QueryResponse xmlns="urn:liberty:disco:2003-08">
      <Status code="ok"/>
      <ResourceOffering entryID="1">
        <ServiceInstance>
          <ServiceType>urn:nokia:ws:samples</ServiceType>
          <ProviderID>http://example.nokia.com:8080/soap/</ProviderID>
          <Description>
            <CredentialRef>2sxJu9g/vvLG9sAN9bKp/8q0NKU=</CredentialRef>
            <Endpoint>http://example.nokia.com:8080/soap/HelloWS</Endpoint>
          </Description>
        </ServiceInstance>
      </ResourceOffering>
    </QueryResponse>
  </S:Body>
</S:Envelope>
<Credentials notOnOrAfter="2004-09-28T18:28:44Z">
  <wsse:BinarySecurityToken EncodingType="wsse:Base64Binary"
    wsu:Id="ia1575535f5b0712dbff7033db0721e4f838390f3"
    xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext"
    xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
    AZoOuAM4BdMxtKugmt1qiwZze11vQb/m5udOPOTa8Y5L
  </wsse:BinarySecurityToken>
</Credentials>

</QueryResponse>
</S:Body>
</S:Envelope>
7. Send service request

POST /soap/HelloWS HTTP/1.1
Accept: text/xml
Expect: 100-continue
Content-Length: 528
Content-Type: text/xml

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header>
            <wsse:BinarySecurityToken EncodingType="wsse:Base64Binary"
                wsu:Id="ia1575535f5b0712dbff7033db0721e4f838390f3"
                xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext"
                xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
                AZoOuAM4BdMxtKugmt1qiwZze11vQb/m5udOPOTa8Y5L
            </wsse:BinarySecurityToken>
        </wsse:Security>
    </S:Header>
    <S:Body>
        <RequestHello xmlns="urn:nokia:ws:samples">
            <HelloString>Hello Web Service Provider!</HelloString>
        </RequestHello>
    </S:Body>
</S:Envelope>
HTTP/1.1 200 OK
Date: Thu, 23 Sep 2004 19:02:39 GMT
Server: Jetty/4.2.21 (Windows 2000/5.0 x86 java/1.4.2_04)
Content-Type: text/xml
Content-Length: 270

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ex:HelloResponse xmlns:ex="urn:nokia:ws:samples">
      <ex:ResponseString>
        Hello Web Service Consumer!
      </ex:ResponseString>
    </ex:HelloResponse>
  </S:Body>
</S:Envelope>
ID-WSF uses SAML assertions as security token format

ID-WSF defers to OASIS Web Services Security (WSS) for how security information is carried in SOAP Headers.

Additionally, ID-WSF points to the SAML Token profile (WSS STP) for how SAML assertions are carried in a WSS <Security> header.

Different versions of ID-WSF reference different combinations of these other specifications.
Multiple versions of SAML will soon exist in marketplace – possibility of multiple SAML assertion versions (e.g. 1.1 and 2.0) must be addressed within ID-WSF.

When registering a service, a WSP indicates which version of SAML assertions it supports.

When querying for services, a WSC indicates which version of SAML assertions it supports.

DS mediates/filters appropriately when returning ResourceOfferings to WSC.
Example: AOL’s Implementation

- ID-WSF based services
  - Authentication Service
  - Discovery Service
  - Radio & Photo Services
- Intelligent clients on connected devices
  - Direct WSCs
  - Client only configured with address of IdP (authentication svc)
Identity Web Services: Radio@AOL

- Authentication Service
- Discovery Service
- Radio App Server
- Radio Data Server

Liberty-based messages

Service specific messages

Data Flow

Authentication Messages

Service Discovery

Application Messages
Liberty Conformance Program
Conformance Program

- The conformance program is designed to validate core functionality via interoperability testing so that purchasers of Liberty-based technology can focus on other details specific to their market and/or deployment.
  - gives assurance that systems will, in fact, work with one another.
- Vendors choose to attend regularly scheduled Conformance events to validate their implementations are interoperable with others.
- Companies provided with Test Procedures documents, Non-Disclosure Agreement (NDA), and logistics information.
- Vendors judged as having successfully completed the conformance event’s test procedures are permitted to license the use of the “Liberty Interoperable” Logo.
Static Conformance Requirements (SCR) document defines required and optional features of implementations

- **ID-FF SCR** defines 4 profiles:
  - IdP, SP, SP Complete, LECP
  - Plus 2 Extended Profiles for IdP and SP

- **ID-WSF SCR** defines 9 profiles:
  - WSC, WSP, DS, IdP, IS, PAOS
  - WSC, WSP, IdP come in two “flavors”:
    - ID-FF authentication
    - LUAD (AS) authentication

- No conformance program for ID-SIS
Conformance cont'd

- Interoperability Conformance Testing Program
  - Achieve the Liberty Interoperable mark.
  - Four events per year
  - Success requires interop with at least two other implementations in each complementary profile
    - E.g., an IdP must test with at least two SP and two LECP implementations
  - So far, about 20 companies and 30 different products have gone through this successfully

- The Liberty Conformance Review Team (LCRT) is the Administrator of the testing events. The LCRT’s members are the official observers of Conformance events.
Conformance cont'd

- Liberty recently announced that its Conformance Program will be extended to include support for SAML 2.
- SAML 2 program will begin in July, tests will be held quarterly.
- Customers of WSF implementations must be confident of lower-level SAML interoperability. SAML 2 testing is therefore key to Liberty.
- Liberty SAML 2 program will test interoperability conformance, i.e., that implementations interoperate with each other.
- OASIS is also developing a conformance testing program for SAML to assure that products conform fully to the complete SAML specification.
- LAP and OASIS programs will consequently be complementary.
The Liberty Technology Expert Group (TEG) also conducts Interoperability testing on specs before releasing as final. This process is 'internal' and does not result in any logo or mark. Purpose is to identify any issues with the specs before issuing as final. Schedule of events dependent on spec release schedule.
Liberty Identity Service Interface Specifications
Liberty’s Architecture

Liberty Identity Federation Framework (ID-FF) & Security Assertion Markup Language (SAML) 2.0

Liberty Identity Services Interface Specifications (ID-SIS)

Liberty Identity Web Services Framework (ID-WSF)

Liberty specifications build on existing standards (SAML, SOAP, WS-Security, XML, etc.)
Overview of Liberty Service Interfaces

- **Multiple elevations** (service interfaces) built on the same foundation frameworks (ID-FF & ID-WSF)

- **First service tracks:**
  - **Identity Service Interface Specifications (ID-SIS)**
    - Personal Profile Service
    - Employee Profile Service
    - Geo-location Service
    - Presence Service
    - Contact Book Service
    - Gaming Profile Service (Q3 2005)
    - NEW! Content SMS/MMS messaging Service

**Identity Federation Framework (ID-FF)**
Enables Identity federation and management through features such as identity/account linkage, simplified sign on, and simple session management.

**Identity Web Services Framework (ID-WSF)**
Provides the framework for building interoperable identity-based web services.

- **Discovery, Interaction, Invocation**

<table>
<thead>
<tr>
<th>Service</th>
<th>SAML</th>
<th>HTTP</th>
<th>WS-Security</th>
<th>WSDL</th>
<th>XML Enc</th>
<th>XML Sig</th>
<th>WAP</th>
<th>XML</th>
<th>SSL/TLS</th>
<th>SOAP</th>
</tr>
</thead>
</table>

ID-Service Interface Specifications

- Family of interoperability specifications for identity-based web services
- Use WSF for the plumbing, concentrate on application logic
- May use WSF Data Services Template as a model
- 3 or more Liberty members can start a new group
Personal Profile Service

- Personal Profile (ID-PP) and Employee Profile (ID-EP) introduced within ID-WSF (although logically part of ID-SIS)
- Defines attributes for describing Principal demographic data elements
  - Individual and Employee respectively
- Based on Data Services Template (ie. uses DST defined atomic query/modify)
- Uses XPATH to specify target of query and modify operations
ID-PP continued

- ID-PP defines data hierarchy for representing Personal data attributes
- Extensive examples
  ```xml
  <PP>
  <InformalName>thewanderer</InformalName>
  <CommonName>
    <CN>Zita Lopes</CN>
    <LCN xml:lang="es">LKj343asas</LCN>
    <AltCN>Maria Lopes</AltCN>
    <AltCN>Zita Mª Lopes</AltCN>
    <AnalyzedName nameScheme="">
      <PersonalTitle>Dr.</PersonalTitle>
      <FN>Zita</FN>
      <SN>Lopes</SN>
      <MN>Maria</MN>
    </AnalyzedName>
  </CommonName>
  [etc...]
  ```
- Minimal implementation required to drive ID-WSF conformance testing
Current Services Work

- **Contact Book Service**: A common method for users to manage and share personal or business contacts regardless of contact book provider, enabling service providers to access or automatically update, at the user’s request, information like billing or shipping address.

- **Geo-location Service**: An interoperable way to automatically identify a person’s location, at the user’s request, to provide services like weather, news, travel or currency updates or directions to a chosen location.

- **Presence Service**: A common way for users to share presence information, such as whether they are online, offline, on the phone or in a meeting, with any service provider for the purpose of communicating availability.

- **Gaming**: A common way to enable end-users to maintain player profile.

- **NEW! Content SMS/MMS messaging Service**
A Liberty Enabled identity based location service does…

- It enables access to location data, such as coordinates and/or civil address data, in a secure, interoperable and privacy-protected manner.

LAP is not defining Location service logic

- It adds privacy, security and interoperability qualities to location specific information exposure towards any location services business logic

LAP is not defining Location specific protocols

- It adds a web-services based exchange of information. Then, that information can be further conveyed to any servers and/or network elements by using any location specific protocol technologies

Multiple technologies exist by which location can be determined

- (e.g. mobile-based, GPS, etc…) …
  - But, these are out of the scope of Liberty as well
Geo-location Use Case

Geo-location information is made Available, through various means, to myLocation.com (OOS of LAP).

myLocation.com registers at DS as Provider of Loc information for Bob (using Liberty WSF protocols).

Weather.com discovers that myLocation.com hosts Bob's Location information (using Liberty WSF protocols).

Bob Accesses weather.com for personalized weather information.

Bob Accesses weather.com for personalized weather information.
A Liberty Enabled identity based Presence service creates:

- Identity-based specification that helps Liberty-enabled products to realize presence service using existing protocols.
- Interaction for the information exchange between Presence Service Provider and other SP (e.g. Location, contact book, game) within the same Circle of Trust.

LAP is not defining yet another new Presence service protocol:

- It adds privacy, security and interoperability qualities in Liberty-enabled environments to existing presence services or protocols.
Contact Book Use case

- Sean logs onto a Circle of Trust Service Provider (using the Federated Identity framework) to order gifts for friends/family and checks out.
- Sean tells the Service Provider that different billing addresses will be needed.
- The Service Provider invokes the Contact Book Service.
- The Service Provider returns list of users in Sean's Contact Book (however, one card is marked "Private" and is not returned).
- Sean matches the appropriate contact with each gift ordered.
- Sean modifies one incorrect address entry for one card.
- The Service Provider performs "modify" action on the change and writes new attribute into Sean's Contact Book via the Contact Book Service.
- Sean finishes his order, and logs out.
Gaming Service

- Gaming is one of the fastest growing markets
- The Internet and mobile networks are increasingly used for buying games, playing, and interacting
- Online gaming – probably the next breakthrough area

<table>
<thead>
<tr>
<th>Global games market value by sector ($ million)</th>
<th>2002</th>
<th>2003</th>
<th>2007</th>
<th>2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Handheld</td>
<td>1307</td>
<td>1501</td>
<td>1925</td>
<td>1206</td>
</tr>
<tr>
<td>Console</td>
<td>7187</td>
<td>6047</td>
<td>6445</td>
<td>5358</td>
</tr>
<tr>
<td><strong>Total HW</strong></td>
<td><strong>8494</strong></td>
<td><strong>7548</strong></td>
<td><strong>8370</strong></td>
<td><strong>6564</strong></td>
</tr>
<tr>
<td>Software</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Handheld*</td>
<td>2326</td>
<td>2238</td>
<td>2693</td>
<td>1602</td>
</tr>
<tr>
<td>Console*</td>
<td>14922</td>
<td>16449</td>
<td>13969</td>
<td>13077</td>
</tr>
<tr>
<td>PC*</td>
<td>3707</td>
<td>3806</td>
<td>3135</td>
<td>2617</td>
</tr>
<tr>
<td>Broadband</td>
<td>290</td>
<td>497</td>
<td>2137</td>
<td>4106</td>
</tr>
<tr>
<td>Interactive TV</td>
<td>133</td>
<td>249</td>
<td>1955</td>
<td>4130</td>
</tr>
<tr>
<td>Mobile</td>
<td>243</td>
<td>587</td>
<td>3783</td>
<td>6928</td>
</tr>
<tr>
<td><strong>Total SW</strong></td>
<td><strong>21621</strong></td>
<td><strong>23826</strong></td>
<td><strong>27672</strong></td>
<td><strong>32460</strong></td>
</tr>
<tr>
<td><strong>Total HW &amp; SW</strong></td>
<td><strong>30115</strong></td>
<td><strong>31374</strong></td>
<td><strong>36042</strong></td>
<td><strong>39024</strong></td>
</tr>
</tbody>
</table>

*Sales and rental

Source: Informa Media Group
Content Messaging Service

- Define a Web services interface that enables SMS/MMS messages over Web services using Liberty Identity Web Services Framework, to/from:
  - Service Providers
  - SMSC (Short Message Service Center) and
  - MMSC (Multimedia Messaging Service Center)
- Provide SMS/MMS users with enhanced privacy
Summary

- Liberty architecture provides standards-based platform for building identity-centric applications
- Three components
  - ID-FF – federation of identities across domains and SSO
  - ID-WSF – platform for SOAP-based identity attribute sharing
  - ID-SIS – family of interoperability specifications for identity services
Resources

- Liberty Developer Resource Center
  - www.projectliberty.org/resources/resources.html
- SAML
  - www.oasis-open.org/committees/security
- SOAP
  - www.w3.org/2000/xp/Group/
- SSL/TLS
  - www.ietf.org/html.charters/tls-charter.html
Contributors

- Conor Cahill, AOL
- Carolina Canales-Valenzuela, Ericsson
- Frederick Hirsch, Nokia
- Paul Madsen, NTT
- Prateek Mishra, PrincipalIdentity
- Rob Philpott, RSA Security
- Jeff Smith, NTT
- Eric Tiffany, ISTO
- Greg Whitehead, Trustgenix