ID-WSF Basics
A pragmatic look at Liberty identity web services and the business needs addressed

Liberty 2.0 Workshop, 22 January 2007
Eve Maler, Sun Microsystems
eve.maler@sun.com
http://www.xmlgrrl.com/blog
Real-life example 1: Sun-BIPAC

- BIPAC offers customized political services to Sun employees online
  - Sharing unrestricted content: easy
    - Just look for sun.com referrer/IP address
  - Sharing legally restricted content: not so easy!
    - The service needs stronger authentication, along with the user's citizenship, shareholder, and employment status
    - ...and Sun and its employees need to keep from exposing their actual identities to BIPAC, to comply with regulations and give users confidence about their “political privacy”

- Ultimately achieved with Liberty identity services – which BIPAC is now rolling out to more customers
Liberty published standards in context

**ID-SIS**: Service Interface Specs
- ID-SIS plus ID-WSF equals “Liberty Web Services”
- Defines particular useful services
- Personal profile, geolocation...

**ID-WSF**: Identity Web Services Framework
- Focused on application-to-application interaction

**ID-FF**: Identity Federation Framework
- “Liberty Federation”
- Focused on human-to-application interaction
- Now converged with SAML V2.0
The human-to-app story

- Single sign-on, single logout, etc. take place among:
  - The user (with actions mediated by a **client** of some kind)
  - An **identity provider** (IdP)
  - A **service provider** (SP) that serves as a **relying party** (RP)
- These actions are communicated primarily with XML over HTTP(S)
Why app-to-app interaction?

- Get around browser payload limitations
- Allow identity-enabled actions to happen silently (mediated by policy) when you're not around
  - All the way from *pay my bills automatically*...
  - *...to let the emergency-room doctor access my medical records from another country if I'm in a coma*
- Allow multiple services to cooperate securely
  - Providing both personalization and access control
- To achieve this, Liberty uses SOAP-based protocols
Design goals

- A standards-based architecture for identity web services
  - Ecosystems of services that expose interfaces on behalf of individual users' identities
- A flexible foundation layer for application development
  - Across security domains and computing platforms
  - Across time, allowing for service location flexibility
- The option of maximum privacy and security
  - Identity information requests access-controlled
  - Minimal disclosure of identity information
  - Protection against disclosure of identifiers
**High-level protocol architecture**

- Makes use of existing standards wherever appropriate
- All Liberty infrastructure components and foundational services can be replaced by your own
  - At some cost to interoperability, naturally
- Extensibility and modularity are built in to let you easily create your own identity-based services

**Categories of services**

- **ID-SIS services**
- **Third-party services**
- **Foundational identity services**
- **Identity-enabled web services infrastructure**
- **Internet / Web / web services infrastructure**
Major benefits of ID-WSF's design

- Authentication, authorization, and application of usage policy against consumers of identity data
- User privacy through use of pseudonyms
- Dynamic service discovery and addressing
- Common web services transport mechanisms to apply identity-aware message security
- Abstractions and optimizations to allow anything – including client devices – to host identity services
- Unified data access/management model for developers
- Flexibility to develop arbitrary new services
- Support for social identity applications
An all-singing, all-dancing sample flow
Kicking off an app-to-app interaction

- It usually starts with a user (possibly not you!) logging in and asking for some service behavior involving your identity.
- During SSO, the IdP informs the SP where to find your **Discovery Service (DS)**
  - A hub for locating, and possibly getting coarse-grained authorization to use, various identity services of yours.
- In a typical deployment, the IdP and DS form one tightly coupled software component.
The locate-and-access dance

- The SP dons the role of a web service consumer (WSC)
  - A WSC is the requestor endpoint, and a web service provider (WSP) is the responder endpoint
  - **Tip:** Mentally add “of identity data” to remember which is which
- The WSC asks the DS where a particular WSP is, and asks for access
  - WSPs will typically do fine-grained WSC authorization themselves
- One example of a WSP is the ID-SIS Personal Profile (PP) service for name, address, etc.
Getting information-sharing approval

- What if the PP service needs to check with you before responding?
  - It can ask your DS where to find an **Interaction Service (IS)** for you so it can bother you real-time
    - According to your own policy preferences for what's important enough to bother you with
  - The PP is acting as a WSC
    - Doing the locate-and-access dance itself, just like BuyPuppyStuff did
  - The IS uses non-Liberty means to (e.g.) SMS you for approval
Observations

- These logical components were included for maximum privacy and flexibility, but not every deployment needs them all!
  - And the worst case is still optimized so that devices sensitive to “protocol chattiness” can handle it
- Any identity service can “recursively” use the discovery and access system provided by the DS to call another one
- At any point a service can (attempt to) contact the user for informed consent, policies, more attributes...
- Throughout, the user might be known only by a pseudonym
Real-life example 2: Radio@AOL
(credits: Conor Cahill and John Kemp)

- The ultimate in user control: your personal device serves up your preferences

- Locate and access DS

- Locate and access radio service

- A Liberty-enabled User Agent or Device (LUAD)

- Eve's radio preferences service

- Stream AAC music and song/artist info

- Use preferences to choose streams and select presets

- And oh, by the way – a WSP (...of identity data)

- Ultravox stream server

- Radio@AOL
Protocol architecture piece-parts

Legend:
- Liberty Alliance standard
- External standard
- Third-party (possibly a standard)

TCP/IP, UDP, SSL/TLS, HTTP, SOAP 1.1, SAML assertions

WS-Addressing
Core
SOAP Binding
WSDL
Security Mechanisms
SAML Token Profile
WS-Security
Security Mechanisms
SAML2 Metadata
Security policy URIs

Description

Third-party svcs

ID-SIS

Third-party svcs

Interaction Service

People Service

Authn, SSO, Identity Mapping Services

Discovery Service

Data Services Template

Subscription/Notification Framework

WS-Addressing SOAP Binding

Authn, SSO, Identity Mapping Services

SAML Token Profile

SAML Profile

WS-Addressing Core

WS-Security

Security Mechanisms

Security Mechanisms

People Service

Interaction Service

TCP/IP, UDP, SSL/TLS, HTTP, SOAP 1.1, SAML assertions

Liberty Alliance standard

External standard

Third-party (possibly a standard)
Major features of ID-WSF

Already touched on:
- Bootstrapping from SSO
- Service discovery
- User interaction
- Smart clients

To be touched on shortly:
- Web service identity model
- Privacy mechanisms
- Person-to-person federation
- Design patterns for common development tasks

Additional features:
- Service invocation and message construction
- Security policy
- Identity mapping mechanisms
- Identity provider services
Web service identity model

- A model for carrying the identity of various parties to a transaction within web service messages
  - Sender (human)
  - Recipient
  - Invoker (service on behalf of human)
  - Target identity owning the resource (human)
    - In querying for your own mail from an email service, you are the target identity
    - In looking up your colleague's calendar, your colleague is the target identity
- WS-Security SOAP headers and SAML assertions are profiled to carry this info in “identity tokens”
Privacy mechanisms

- Ensuring that your data is shared on your terms by:
  - Capturing your usage directives and consent status in service messages
  - Allowing for interaction with you at critical junctures to obtain your consent and privacy policies
    - Interaction Service, Interaction Redirect

- Inhibiting correlation of your activities by:
  - Offering different pseudonyms to different parties
    - Identity Mapping Service

- Protecting your data in transit
  - WS-Security for confidentiality
Real-life example 3: PayByTouch

(credits: Greg Whitehead and Robert Aarts)

- Provides the option of strong authentication at transaction-time, based on authentication quality or user policy.

- Browsers properly equipped with plugins could support fingerprint-based authentication.

- See also TeliaSonera/FT “FIDELITY” project for similar Wallet service: http://www.celtic-fidelity.org
Person-to-person federation

- The **People Service (PS)** lets you create reusable groups and roles involving other people's identities
  - And use them to control access to your resources
  - Even if multiple IdPs are involved

- Whereas today in (say) Flickr, you can create lists only for “friends” and “family” with Flickr IDs
  - And you can't reuse these lists with other services
    - Though you can issue “foreign” guest invitations by email

- The PS is useful for business scenarios too
  - Managing team access to resources in joint-venture projects
  - Identity proofing when a colleague loses their token
Design patterns for development

- Many identity services at the ID-SIS layer need:
  - Provisioning, retrieval, and ongoing updates of identity data (“CRUD” – create, retrieve, update, delete)
  - Subscription and subsequent notification if something changes
- Templates and guidelines are provided for rapid service development offering these common features
Major open-source implementations

- Sun's [http://OpenSSO.dev.java.net](http://OpenSSO.dev.java.net)
  - SAML, ID-FF, ID-WSF in Java; SAML in PHP ("Lightbulb")
- Entrouvert's [http://LaSSO.Entrouvert.org](http://LaSSO.Entrouvert.org)
  - SAML, ID-FF, ID-WSF in C with SWIG bindings for Python, Perl, Java, PHP
- Symlabs' [http://ZXID.org](http://ZXID.org)
  - SAML, ID-FF, ID-WSF (and WS-Fed) in C with Perl/PHP wrappers
  - ID-WSF C client and Java server
Final food for thought: Liberty and Web 2.0

- SAML, Liberty, XRI, and OpenID protocol designers have been discussing the proposition:
  - Can we move from *incompatibility* to *equivalence* to *compatibility* to *convergence*?
- “Lightbulb” integration of OpenID discovery and metadata with SAML has shown one possibility
  - Existing specs for XRI SSO and Lightweight SSO may give way to an “OpenID-SAML profile”
- Additional ideas:
  - Leveraging existing attribute exchange technology in new “identity schemas” work
  - OpenID-enabled People Service
ID-WSF Basics
Thanks for your attention! Questions?

Eve Maler
eve.maler@sun.com