Getting to grips with ID-WSF

Jonathan Scudder, Sun Microsystems
Topics to cover

- The big picture
- Web services and humans
- Invokers and senders
- Identity-based web services
- Putting it all together
- A rundown of the specs
Federation progression

Identity silos

Federation - Single Sign On

SOA behind the door?
ID-WSF as a foundation

Builds on federation (SAML 2.0)
Defines identity service framework
Standard identity services (ID-SIS) using framework
Real people & WS

Trust users, not clients

Involve users in making decisions

Interaction services in real time
Important roles in ID-WSF

- Invoker
- Sender
- Discovery service
- Interaction service
- Identity-based service
**Invoker**
The user who is behind the action. May be hiding behind a heap of intermediary clients and servers.

**Requester**
The intermediary who is sending the request at this leg of the journey. Requester can also be the invoker.

**Disco/Interaction**
Tells requester where the invokers service can be found, and provides the means to get user permission.

**Identity Service**
The identity-related service that the invoker needs, and that the requester is trying to get in touch with.
ID-WSF terms

- **WSC** - Web Service Consumer
  - Requester and possibly invoker

- **WSP** - Web Service Provider
  - Identity or non-identity based web service

- **DST** - Data Services Template
  - Template for defining identity services

- **AP** - Attribute provider
  - Type of identity service that allows retrieval of attributes (rather than modify etc.)
ID-WSF flash animation

http://www.projectliberty.org/resource_center/specifications
Core identity services

- Authentication Protocol (SASL)
- Authentication Service
- Single Sign On Service
- Identity Mapping Service
Airline “FlyMe” has a number of registered users, and is an Identity provider for a car hire company “Bangers-R-Us” and hotel chain “Nice Hotels”.

User “Bob” has an account with all three service providers, and has federated these accounts.

Bob wants to hire a car and book a hotel. He logs on to “bangers-r-us.com” and authenticates via the “FlyMe” site.
Bob selects the car he wants to hire, and proceeds to the checkout. Rather than fill out his address and contact details again at the hotel chain, he ticks off the “share my contact details” checkbox on the order form.

The Bangers-R-Us site now calls the FlyMe discovery service and adds itself as an endpoint for retrieving Bob’s attributes. Permission is given to share this info.

Bob then accesses the NiceHotels.com site and finds himself a nice hotel near the beach.
When it comes to filling out his contact details on NiceHotels.com, Bob selects the “use FlyMe to get my contact details” checkbox.

NiceHotels queries the Discovery Service, gets an endpoint which points at Bangers-R-Us, and then calls the endpoint. Bangers-R-Us uses the token supplied to check that the user session is valid, and then returns the relevant information.

Bob has a great weekend...
What if Bob didn’t want to share his information with all sites?

The Bangers-R-Us or Discovery Service could ask the user for permission independantly of the requesting service (Nice Hotels).

How does Bangers-R-Us share his information?

Bangers-R-Us implements an Identity Service specification based on ID-WSF Data Service Template.

How does Nice Hotels know how to talk to Bangers-R-Us?

The identity service spec would be agreed beforehand. Alternatively, standard identity services from ID-SIS could be used to achieve interoperability.

How does Bangers-R-Us know that Bob is the one requesting his information?

The request contains a SAML assertion for Bob, retrieved from the Authentication Service (FlyMe IDP)
Other scenario
Non-identity web services

- Assume site which is part of a federation
- User authenticated to IDP
- External (non-identity) web services called
- Allow these web services to check the invoker identity by using the SSO service
Web Service makes decision based on confirmed user token
Coping with corporate fusion

- Large users groups with information in different locations and systems
- Hide location of user data
- Use discovery to find the appropriate service
- Use a standard identity service (eg: data service template) as the common interface
Web Service Client

Corporate IDP

Web Service Provider
Company B
Implements identity service
ID-SIS Employee info?

Data

Web Service Provider
Company A
Implements identity service
ID-SIS Employee info?

Data

Discover endpoint for appropriate service
Specification rundown

- ID-WSF SOAP binding
- ID-WSF Security Mechanisms
- ID-WSF Discovery Service
- ID-WSF Data Services Template
- ID-WSF Subscriptions and Notifications
- ID-WSF Interaction Service
- ID-WSF Profiles for LUAD
- ID-WSF Authn, SSO, Identity Mapping
- ID-WSF People Service
ID-WSF SOAP Binding
Details how services are invoked with a SOAP framework. Defines use of WS-Adressing, headers and processing rules.

ID-WSF Security Mechanisms
Describes security requirements, and defines token usage.

ID-WSF Discovery Service
Core identity services that provides references to a user’s services. Eg: “Jonathan’s Calendar Service”. Can issue security tokens.
ID-WSF Data Services Template

Example template for building Secure Identity Services within the ID-WSF framework. Read this if you are going to implement an identity service!

ID-WSF Subscriptions and Notifications

Describes how users can subscribe and be notified when certain events happen at other providers.

ID-WSF Interaction Service

Specifies protocols and profiles for involving the user in making decisions.
ID-WSF Profiles for LUAP

Describes and defines how Liberty-smart clients can interact directly with ID-WSF components.

ID-WSF Authn, SSO and Identity Mapping

Describes protocol for authenticating ID-WSF entities, an Authn Service for performing this authentication, and a Single Sign On service for retrieving SAML assertions.

ID-WSF People Service

 Defines how users can get and use identifiers for other users. Proxy/delegation.
Thanks for listening!