Liberty Alliance & ‘Touching the Browser’

Paul Madsen, Entrust
Chair of Liberty Trust, Security, and Privacy Subteam
Agenda

- Liberty Alliance overview
- Touching the Browser
- Liberty model
- Liberty SSO
- Extended SSO
Agenda

- Liberty Alliance overview
- Touching the Browser
- Liberty model
- Liberty SSO
- Extended SSO
Identity Crisis

Joe's Fish Market.Com

Tropical, Fresh Water, Shell Fish, Lobster, Frogs, Whales, Seals, Clams
What is Federated Identity?

- A federated identity is one whose scope extends beyond the original application(s) for which it was originally created.
- Existing identities can be leveraged for other applications, simplifying management for enterprises and end-users.
- Mechanisms for enabling this within an enterprise already exist - new requirements for cross-domain transactions demand new standards for the protocols and exchange formats.

Making identity ‘portable’
What is the Liberty Alliance?

• A business alliance, formed in Sept 2001 with the goal of establishing an open standard for federated identity management
• Global membership consists of consumer-facing companies and technology vendors as well as policy and government organizations
• The only open organization working to address the technology and business issues of federated identity management
Liberty Alliance Membership

- More than 170 global member organizations
- Driven by end-users, government orgs and vendors
Defining Liberty

**Liberty Alliance IS...**

- a member community delivering technical specifications, business and privacy best practices
- developing an open, federated identity standard that can be built into other companies’ branded products and services
- providing a venue for testing interoperability and identifying business requirements
- driving convergence of open standards

**Liberty Alliance IS NOT**

- a consumer-facing product or service
- developed and supported by one company
- based on a centralized model for identity
Liberty & SAML

- **Liberty builds heavily on SAML**
  - Security Assertions Markup Language

- **SAML is an XML-based framework for exchanging security information**
  - XML schema and definition for security assertions
  - XML schema and definition for a request/response protocol

- **An OASIS standard**
  - Vendors and users are both involved
  - Codifies current system outputs rather than inventing new technology

- **Excellent traction in the marketplace**
Liberty Roadmap

Phase 1
Simplified sign-on and identity federation (ID-FF)

Phase 2
Web Services Framework (ID-WSF)

Future Phases
Enhancements to Federation and Services Infrastructure (ID-SIS)

To be released October 2003

Released Summer 2002

INCREASED BUSINESS VALUE

INCREASED INTEROPERABILITY
Agenda

- Liberty Alliance overview
- Touching the Browser
- Liberty model
- Liberty SSO
- Extended SSO
Touching the Browser?

- A model in which User Agent’s baseline capabilities can be **dynamically extended**
- Functionality downloaded as needed
  - May be invisible to User
  - May be cached for subsequent use
- Plug-ins, ActiveX, Java applets
- Contrasts with ‘**working with what you get**’
Issues

- **Locked-down user agents**
  - For security & virus protection

- **Download size**
  - Mitigated by caching

- **User aversion**
  - Dreaded ‘trust this’ query?

- **Inconsistent functionality**
  - Is Java enabled
What might ‘Touching the Browser’ mean for federated identity

**Functional Areas**

- **Discovery** – Can user agent facilitate determination of appropriate providers?
- **Protocols** – Can user agent play active role in messaging?
- **Attributes** – Can user agent store and release Principal’s attributes?
- **Security** – Can user agent provide security protections beyond the base set?
Agenda

- Liberty Alliance overview
- Touching the Browser
- Liberty model
- Liberty SSO
- Extended SSO
Liberty model

- Liberty default model is to ‘work with what you get’
- Touching the Browser model complicated by the variety of User Agents we’d need to deal with
- Liberty does not preclude dynamic extension capabilities but does not require them
- Account for different User Agents in order to leverage their different capabilities
Liberty profiles

- Liberty must support a variety of User Agents
  - Old browsers, new browsers, Phones, PDAs, etc.
- User Agents differ in the functionality and capabilities
- Liberty defines base protocols for enabling federated identity messaging between providers
- Abstract protocols are profiled for the ‘real-world’
- The various Liberty profiles make different expectations of User Agent capabilities
Baseline Liberty Requirements

- HTTP 1.0 or HTTP 1.1
- SSL 3.0 or TLS 1.0 or any subsequent protocols which are backwards compatible
  - either directly or via a proxy
- Minimum maximum URL length of 256 bytes.
- A WAP browser user agent MUST support WML 1.0, 1.1, 1.2 or 1.3 in addition to the above requirements.

This is the ‘what we get’
Optional ‘requirements’

➡ **Cookies**
- Enables Identity Provider discovery
- Prevents the ‘Who is your IDP question?’

➡ **Javascript**
- Streamlines the Form POST profile by automatically submitting forms
- Prevents the Principal from having to click on Submit/Continue buttons

➡ **SOAP**
- LEC profile stipulates that User Agent actively sends SOAP messages
Single Sign-On

- Simplest aspect of federated identity
- An individual is able to access a remote service based on an authentication event that occurred elsewhere
- Liberty ID-Federation Framework builds on SAML SSO protocols and messages
- Authentication Web site (Identity Provider) communicates a SAML assertion to that effect to the relying Web site (Service Provider)
User Experience

Step 1: Federate (link) Accounts

- **Airlines, Inc.**
  - Mileage Account #
  - Password
  - Login
  - Please Login
  - Mileage Account #
  - 4215-2212
  - Password
  - ******
  - Login

- **Welcome**
  - Link this account with your car rental account?
  - Yes
  - No

- **Rental Car Co.**
  - Please Login
  - Account #
  - 624159
  - Password
  - ******
  - Login

- **Welcome**
  - Link this account with your airline account?
  - Yup
  - Nope

Step 2: Single sign-on

- **Airlines, Inc.**
  - Mileage Account #
  - 4215-2212
  - Password
  - ******
  - Login

- **Welcome back Mr. Madsen**
  - Book a flight
  - Rent a Car

- **Rental Car Co.**
  - Welcome back Mr. Madsen
  - Your Status: Gold
  - Preferences: Mid-Sized Sedan
Pseudonyms

- SSO requires that sites *talk* about the User
- Privacy concerns rule out a global identifier
- Liberty defines mechanism for opaque identifiers
Liberty SSO Protocol Flow

- Instead of the SP directly authenticating the user, the SP queries the IdP and the IdP issues an authentication assertion.
- SP must ‘trust’ the IDP.

1. Initial authentication
2. Browse to SP
3. User authentication request
4. Authentication Assertion issued
5. Authentication Assertion sent

Identity Provider

Service Provider
Authentication Assertion

- Assertion ID
- Issuer
- Issue Instant (timestamp)
- Validity time limit
- Audience Restriction

Authentication Statement
- Authentication Method
- Authentication Instant

User account info (IdP pseudonym)
User account info (SP pseudonym)
Digital Signature of assertion
Issues

- **User will access SP resources based on their authentication to the IDP**
  - The strength of this authentication is critical
  - Liberty defined a syntax by which SP can indicate its preferences and by which the IDP can assert the details
  - Liberty doesn’t stipulate but likely default mechanism will be Password

- **Browser gap between SSL sessions**
  - Liberty makes extensive use of HTTP redirects
  - Liberty stipulates SSL but there will be two separate sessions

- **Not Liberty issues per se but they’re there nonetheless**
Agenda

- Liberty Alliance overview
- Touching the Browser
- Liberty model
- Liberty SSO
- Extended SSO
Extended SSO

- Certificate-based authentication to IDP would address weak password issue
  - Client-auth SSL theoretically possible but key management & roaming limitations make impractical

- Message signing capability in browser would address SSL gap issue
  - XML Signature support in the browser?

- Extend the browser with certificate-based authentication and message signing capabilities
Entrust TruePass™

- Entrust TruePass is a Web based client/server solution
- TruePass Client is a small Java applet (~150kb) that gets downloaded in a hidden frame of the HTML page
- Digital Certificate based strong authentication; leverages server authenticated SSL session
- Digital Signature Support
- All digital ID lifecycle management operations are transparent to the user
Entrust TruePass authentication

- Applet downloaded to browser
- User signs in (to applet) with strong password
- Applet signs challenge string with user’s private key.
- User is authenticated to server
Extended SSO

Initial strong authentication

Strong binding through digital signature
Summary

- Liberty does not stipulate mechanisms that would require ‘touching the browser’.
- Liberty chose the ‘work with what you get’ model.
- However, Liberty does not preclude extending the browser’s capabilities.
- An extended User Agent can coexist with Liberty specifications – optimizing baseline capabilities as appropriate.