Originally presented at XML 2005 in Atlanta, GA, USA in November 2005. Has been modified slightly since. Companion paper will be available at http://2005.xmlconference.org/proceedings eventually.

Abstract: “This talk will explore how the Security Assertion Markup Language (SAML) and Liberty Alliance standards are being used to solve the problem of secure, personalized, seamless transactions that remain privacy-sensitive.”

Welcome to the Participation Age

Enterprise
Collaborative Industry Networks, Outsourcing, New Business Models

Developers
Java, Open Source, Standards Development

Consumers
Blogs, Instant Messaging, Personalized Content on Devices, Social and Job Networking, Online Gaming

Public Sector
Inter-Agency Collaboration, Healthcare Networks, Political Campaigning, International Coalitions

The opportunities

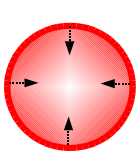
Definition of identity:
Traits (fingerprints)
Characteristics (Sun employee)
Preferences (window vs. aisle)

Definition of digital identity:
Name
Associated attributes

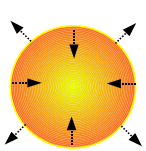
What can have an identity?
Persons (real people) in their roles
Legal entities (companies, campuses, agencies, departments, . . .)
Things (air quality monitoring sensor, traffic counter, . . .)
RFID tags; Digital Assets, Smart Cards
Software services, agents, . . .

Can Identity Keep Up with Distributed Applications?

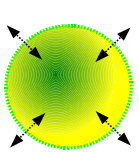
- Perimeters are dissolving
- Access is anytime, anywhere, through any device
- We need security, control, manageability, and accountability



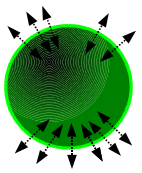
The Era of the Firewall
Keep data inside the firewall



The Era of the Intranet/Internet
Manage data inside and outside the firewall



The Era of the Extranet
Manage data through the firewall

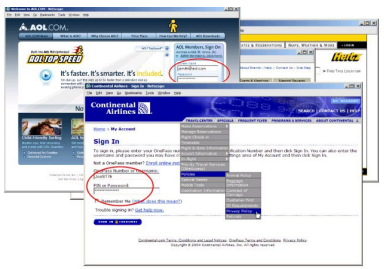


Nothing But Net
Just access and entitlement

The challenges:
Opening access to revenue resources
Legal regulations and compliance
Stewardship and ethical considerations
Reduce complexity & costs
Based on organizational attitudes represented by policy
Intellectual Property protection (copyright adheres to people, not devices!)

Issues with Digital Identity Today

- Users have a proliferation of logins and passwords
- Redundantly stored attributes get out of synchronization
- Security, privacy, and cost are concerns
- When identity is not as “distributed” as the applications that need to use it, business opportunities are missed



The risks:
Most added-value services need identity
The most basic element in a high-value relationship with customers, employees, citizens or business partners
Has to be managed with great care to proactively fight fraud and identity theft
Common mechanisms to handle identities are required

“Distributed” = shared = federated

Requirements for Federated Identity

- Standard **formats** for identity information
  - > Able to represent all existing authentication and attribute technologies
- Standard, secure, privacy-enabled **protocols** for exchanging identity information between components of distributed applications
  - > Technology-neutral, well-specified, and interoperable
- A way to set up **trust relationships** between entities that share identity information
  - > Within technical, business, and legal frameworks

Formats: XML-based, of course

Protocols: we know browsers and client devices of all types will be involved, and we have the opportunity to use web services as well

Trust relationships must include *you*, satisfied that there is appropriate stewardship of information about you!

No need to keep your cash in a mattress; you select a bank based on trustworthiness (and value-added features)
Will take it as a given that it's reasonable to ask online service providers to hold and share your data on your behalf, in appropriate confidence

Security and Identity Standards in the Land of XML: Some Examples

- **XML Signature:** fine-grained data origin authentication
- **XML Encryption:** fine-grained confidentiality
- **XKMS:** outsourced key management
- **SPML:** user provisioning services
- **XACML:** authorization policy expression and evaluation
- **WS-Security:** end-to-end secure SOAP messaging
  - > Using security tokens of various types, including SAML

A number of technologies and standards have been widely used for many years to solve the problems of security and identity, among them directory standards like LDAP and security standards such as Kerberos and the various types of public key infrastructure (PKI). With XML gaining popularity as a solution for more loosely coupled computer-to-computer communication, several standards efforts were launched that applied XML and web services technologies to these solutions -- codifying existing practice but not inventing, for example, new authentication methods or cryptographic key distribution frameworks. In large part, these efforts have been complementary rather than overlapping.

There are special opportunities and challenges in doing security at these higher application levels. E.g., XML needs to be “canonicalized” (normalized) before signing, but allows you to easily sign subelements in place. Web services for traditional security services such as authentication can make it easier to apply security in an SOA, but new security protocols might have holes that we have noticed yet.

* XML-Sig and XML-Enc: Methods of digital signing and encryption suited to opportunities and challenges of XML. Selective signing and encryption of XML; representation of signed and encrypted content in XML form.

* XKMS: Set of XML web services for registering and looking up cryptographic keys. Technology-neutral layer above SPKI, PKIX, etc. Allows client devices to offload key management tasks to an external service.

* SPML: Platform-independent method of provisioning user accounts.

* XACML: Way to express access control policies in XML and protocol for interacting with a policy decision point to get authorization decisions.

* WS-Security and companion “security token profiles” (including one that uses SAML): Define how to apply digital signing and encryption to SOAP web service messages for end-to-end security protection.

The two biggies in the identity space are SAML and Liberty.

SAML: The Universal Solvent for Identity Information

- **SAML:** Security Assertion Markup Language
- Developed and maintained at OASIS by the Security Services Technical Committee (SSTC)
  - > Members include both vendors and users
- Work began in January 2001; V1.0 was stable by mid-2002
  - > Worked quickly and built in many extensibility points
  - > Others immediately began building on top of it
- All IdM vendors now offer some degree of SAML support

Introducing SAML and Liberty in this talk so you can understand their general mechanism for solving problems of identity. Typically you wouldn't start an implementation from scratch; you would either buy a solution (which likely doesn't require any coding at all) or build your own solution on top of one of the open-source projects.

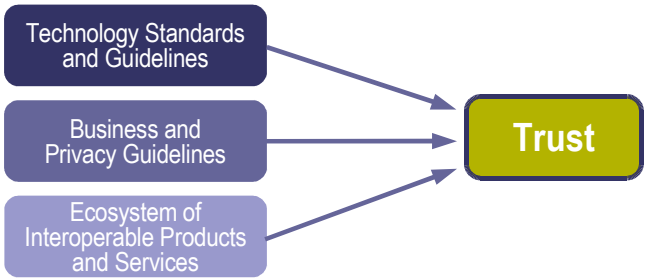
SAML's first use case was single sign-on.

SAML V1.0 had design goals of:
Finishing quickly
Codifying existing security mechanisms more than inventing new ones
Providing extensibility points

Its extensibility mechanisms were designed to get the basics out quickly and let others show the way to additional useful scenarios. Liberty was one of the first to build on top of it. The Internet2 Shibboleth project was also an early adopter/extender of SAML.

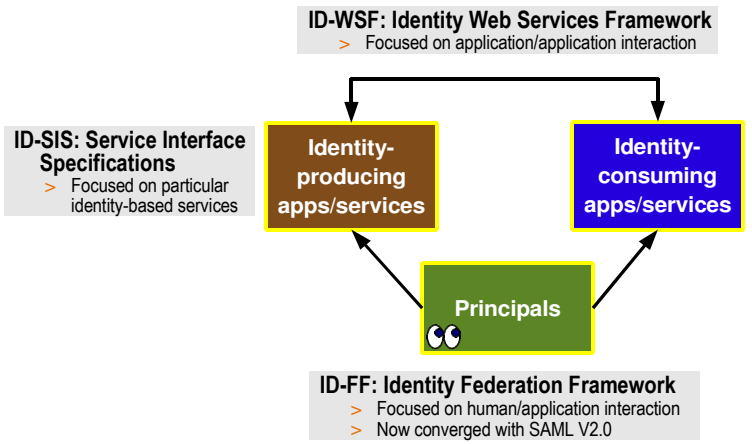
Liberty: Pervasive Identity in Network Applications and Services

- Alliance of more than 150 companies, nonprofits, and government agencies
- Produces open, privacy-enabled standards and guidelines for federated network identity, targeting all network devices



Liberty Alliance standards: The Liberty Alliance Project is an alliance of more than 150 companies, nonprofits, and government organizations from around the globe. It develops open standards for federated network identity, with an emphasis on supporting all existing and emerging network devices. It produces technology specifications such as the Identity Federation Framework and Identity Web Services Framework, along with technical, business, and legal guidelines for adoption and deployment. It also provides interoperability testing and certification services.

Liberty Technology Standards Scope



Some enterprise use cases from Yvonne Wilson (Sun) with security/risk management focus:

* BIPAC
  - reduce risk by eliminating personally identifiable information (PII)
* Phone billing and lookup
  - SSO gets more people using the app,
  - reduce risk, personal info not used in pw
* HR apps
  - SSO main benefit, they have all PII anyway
  - **We'll use the employer-401(k) federation example throughout**
* Government consulting engagements
  - Meet need to separate gov data from commercial
    Reduce risk, build gov trust by separation
* Collect telemetry from customers
  - cost savings, customers give us less data

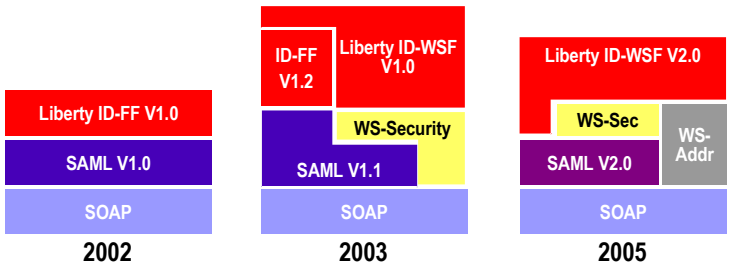
Risk reduction comes about from stuff like:
 * less data being sloshed around, (less theft)
 * possibly faster ability to implement dual-factor authN (just do at IDP instead of every app),
 * fewer pw -> users less likely to write them down so less likely to be compromised, users may be more likely to choose better passwords
 * industry std - less likely that app programmers will invent their own mechanisms which often have security glitches
 * credentials only collected by one entity so less risk of capture and compromise
 * one place to cut off access in event of attack
 * better visibility into access control grants & actions - so better auditability
 * access control can be more dynamic, up-to-date to better reflect job changes

Some consumer use cases:

- People and companies with the right authorizations can look at and add items to your **calendar** (an identity-based service)
- You can get an authorization from your **bank** to buy a book from an **online bookseller** without revealing your identity, and have the **shipping company** send it to you without knowing what's in the package

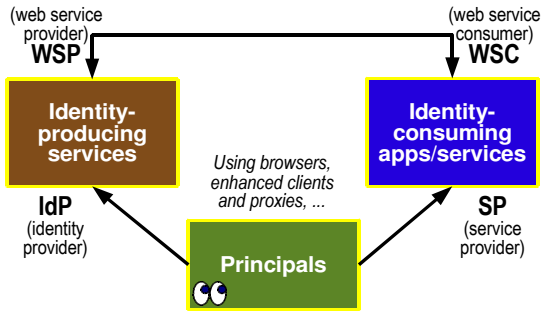
SAML and Liberty Co-Evolution

- ID-FF converged with SAML V2.0
- Internet2 Shibboleth requirements were also accounted for
  - > Its **OpenSAML.org** plans to ship early-access support for SAML V2.0 very soon



Major Entities Involved in Assertion Exchange

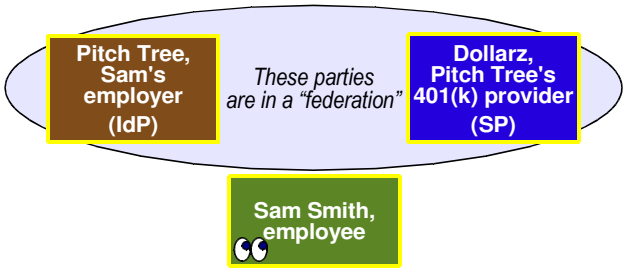
- Terminology comes from SAML and Liberty
- Not intended to be exhaustive!



As Saki said, "A little inaccuracy sometimes saves tons of explanation." Liberty techies will probably shudder at this slide but it's helpful for introducing the most major terms and concepts.

The Classic Web Single Sign-On Scenario

- Sam Smith needs access to protected resources at employer Pitch Tree and 401(k) provider Dollarz
- Both sites benefit from sharing some information about Sam
- All parties would benefit from SSO: Sam authenticating only to Pitch Tree and then using a protected resource at Dollarz



Sam needs to log in (authenticate himself) to Pitch Tree to do his work, and he needs to authenticate himself to Dollarz to see and modify his retirement account details. Sam's employer Pitch Tree is a useful authoritative source for verifying Sam's identity, and it also happens to store attributes about Sam that Dollarz would generally needs. Dollarz has a trust relationship with Pitch Tree since it's a supplier of a particular service to the company. Since they have this trusted

The term "federation" can get confusing. Identity is federated when it is shared or distributed. IdPs and SPs are also said to be in a "federation" when they set up a business and technical relationship.

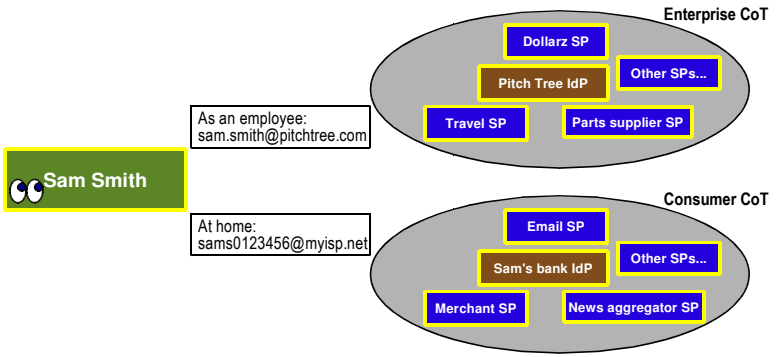
SSO can be achieved in a tightly coupled way but it requires much more technical coordination between these two independent entities than for a SAML-based solution using XML protocols.

The essential ingredients are as follows:

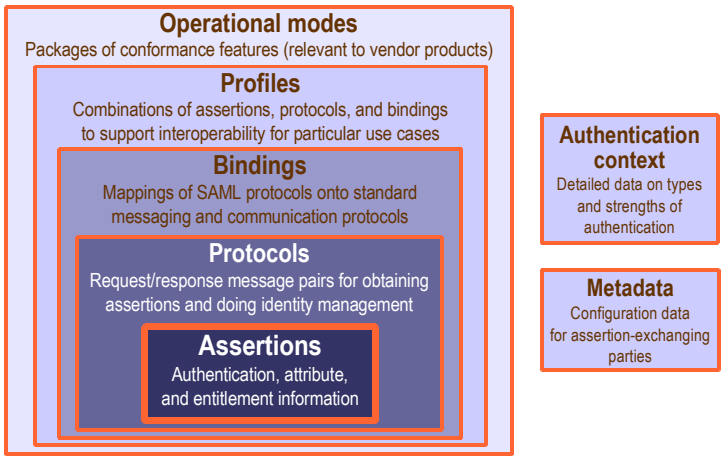
* The user has to authenticate at the IdP at some point, thus allowing the necessary SAML assertion to be created. (That's Sam, logging in to the Pitch Tree site.)
* The IdP has to convey the assertion to the SP. (That's Pitch Tree, sending an assertion about the facts of Sam's credential-checked login session -- possibly including some attributes about him -- to Dollarz.)

Building Circles of Trust

- An IdP can serve as the hub for any number of SP spokes
- Each circle may have its own privacy and security needs



SAML Components



Profiles are the “minimum unit of interoperability”, but operational modes are the “minimum unit of conformance”. Each one requires support for a particular set of profiles (IdP vendors care about this; SPs don't typically care)

SAML Assertions

- An **assertion** is a declaration of fact (according to someone)
- SAML assertions contain one or more statements about a subject:
  - Authentication statement: **"Sam authenticated with a smartcard PKI certificate at 9:07am today"**
  - Attribute statement (which itself can contain multiple attributes): **"Sam is a manager and has a $5000 spending limit"**
  - Authorization decision statement (now deprecated)
  - Your own customized statements...

Example of the Common Portions of an Assertion

```
<saml:Assertion
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  Version="2.0"
  IssueInstant="2005-11-15T14:07:00Z">
  <saml:Issuer>
    www.pitchtree.com
  </saml:Issuer>
  <saml:Subject>
    <saml:NameID Format=
     "urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
      sam.smith@pitchtree.com
    </saml:NameID>
  </saml:Subject>
  <saml:Conditions
    NotBefore="2005-11-15T14:07:00Z"
    NotOnOrAfter="2005-11-15T14:37:00Z">
  </saml:Conditions>
    ... statements go here ...
</saml:Assertion>
```

Example of an Authentication Statement

```
  <saml:AuthnStatement
    AuthnInstant="2005-11-15T14:07:00Z"
    SessionIndex="0">
    <saml:AuthnContext>
      <saml:AuthnContextClassRef>
urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
      </saml:AuthnContextClassRef>
    </saml:AuthnContext>
</saml:AuthnStatement>
```

Authentication Context Classes

- Internet Protocol
- Internet Protocol Password
- Kerberos
- Mobile One Factor Unregistered
- Mobile Two Factor Unregistered
- Mobile One Factor Contract
- Mobile Two Factor Contract
- Password
- Password Protected Transport
- Previous Session
- Public Key – X.509
- Public Key – PGP
- Public Key – SPKI
- Public Key – XML Signature
- Smartcard
- Smartcard PKI
- Software PKI
- Telephony
- Nomadic Telephony
- Personalized Telephony
- Authenticated Telephony
- Secure Remote Password
- SSL/TLS Cert-Based Client Authn
- Time Sync Token
- Unspecified
- Your own customized classes...

Example of an Attribute Statement

```
<saml:AttributeStatement>
  <saml:Attribute
    NameFormat="http://pitchtree.com">
    Name="Role"
    <saml:AttributeValue>
      Mgr
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute
    NameFormat="http://pitchtree.com">
    Name="PurchaseLimit"
    <saml:AttributeValue xsi:type="pitchtree:type">
      <pitchtree:amount currency="USD">
        5000.00
      </pitchtree:amount>
    </saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
```

Attribute Profiles

- Basic
  - Simple string-based SAML attribute names
- X.500/LDAP
  - Common convention for SAML attribute naming using OIDs, expressed as URNs and accompanied by usage of **xsi:type**
- UUID
  - SAML attribute names as UUIDs, expressed as URNs
- DCE PAC
  - DCE realm, principal, and primary group, local group, and foreign group membership information in SAML attributes
- XACML
  - Mapping of SAML attributes to an XACML attribute representation
- Your own customized attribute profile...

Themes in the XML Expression of SAML

- URIs as category names for various options
  - No native use of “QNames in content”
  - SAML standardizes a starter set of URIs in each case, but anyone can develop and use other URIs
  - For example, `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress` for email-based name identifiers
- Controlled extension points
  - Abstract schema types – for example, for “base” name identifier
  - Wildcards in selected locations – for example, `<AttributeValue>` allows arbitrary XML element content

SAML Profiles

- Web SSO, including authentication (and often attribute) information:
  - Using standard commercial browsers
  - Using enhanced clients
- Identity federation – setting up privacy-enabled agreements among providers for referring to a subject
- Direct attribute retrieval
- Single logout – coordinated logout from multiple providers
- Your own customized profiles...

Enhanced HTTP clients (such as handheld devices) or proxies that know how to interact with IdPs but are not SOAP-aware (some smartphones on the market are already Liberty-enabled)

Identity federation:
Using a well-known name or attribute
For anonymous users by means of attributes alone
Using a privacy-preserving pseudonym

Direct attribute retrieval using several common attribute/directory technologies, as we saw with the attribute profiles

SAML Artifacts

- Small, fixed-size, structured data object pointing to a SAML protocol message
  - Typically larger and variably sized
- Allows for “pulling” SAML information rather than having to push it
  - Designed to be embedded in URLs and conveyed in HTTP messages (usually securely with HTTPS)
  - Helps to manage costs of digital signing and high-bandwidth interactions

Web SSO Profile: Branches in the Information Flow

- Does Sam start out at the Pitch Tree site (IdP) or the Dollarz site (**SP**)?
  - > The SSO assertion has to be conveyed from IdP to SP regardless
  - > If he visits Dollarz first, you need an “assertion, please” phase in the SAML flow
- If Dollarz needs to ask for the assertion, does it push (**POST**), allow to be pulled (artifact), or redirect the request?
- Does Pitch Tree push (POST) or allow to be pulled (**artifact**) the response?

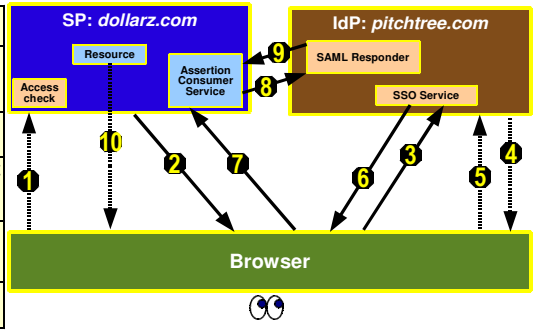
Options in bold are shown in the next flow diagram

The bold choices shown on this slide are what we will assume in the following flow diagram example.

Web SSO Profile

SP-Initiated – SP POST Binding – IdP Artifact Binding

1. (*Sam asks for access to SP resource*)
2. SP sends form to browser
3. Browser form POSTs <AuthnRequest> to IdP
4. (*IdP challenges Sam*)
5. (*Sam provides credentials*)
6. IdP generates assertion/artifact and sends by redirect
7. SP receives and interprets
8. SP sends <ArtifactResolve>
9. IdP sends <ArtifactResponse> with SSO assertion
10. (*SP gives access to resource as appropriate*)



1. Sam attempts to access some protected area of the Dollarz website. We will assume he hasn't logged in here, so there is no current login session.
2. The Dollarz site sends an HTML form back to Sam's browser. It contains a SAML AuthnRequest characterizing Sam as the principal whose information is required.
3. The browser, either due to some action on the part of Sam or via an “auto-submit,” issues the POST to the Pitch Tree site's SSO Service.
4. If Pitch Tree doesn't already have a valid record of a recent-enough (and strong-enough) login by Sam, it challenges him for his credentials by means of the browser.
5. Sam logs in (we'll presume successfully for this example).
6. Pitch Tree's SSO service generates a SAML assertion describing Sam's authentication details and an artifact that can be used to pull a response message containing this assertion. The artifact is labeled with the “source identifier” of the Pitch Tree SAML responder. It could use redirection or an HTML form to send the artifact; we assume the latter here. The form control name is SAMLart.
7. The Dollarz Assertion Consumer Service gets the source identifier; because of its pre-existing trust relationship with Pitch Tree, is configured to understand this as a reference to the Pitch Tree SAML Responder.
8. The Dollarz Assertion Consumer Service goes ahead and contacts the Pitch Tree SAML Responder with an ArtifactResolve request containing the artifact it received.
9. Pitch Tree's SAML Responder now sends an ArtifactResponse message containing the needed assertion.
10. At this point, Dollarz has what it needs to decide whether to give Sam the protected Dollarz resource he wanted. Implementations vary, but typically it would set up a session for Sam as necessary, alert his browser to this fact by sending a redirection message with a cookie to it, and delivering the resource when asked to by the browser's HTTP GET.

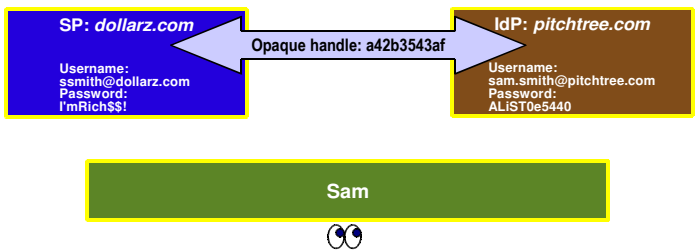
This flow has quite a few moving parts, but Sam can be blissfully unaware of most of it. If you look past the binding choices and the question of which entity initiates the sequence, what adds complexity is the need to ensure confidentiality of the information so that only the intended audience has access to each message.

Account Linking for Privacy and Flexibility

- SSO involves only one-way identity information flow
  - Dollarz need not have an account for Sam at all
  - But it likely does, since it's not a trivial relationship
- Correlation of Sam's two accounts needs a two-way linkage
- Privacy requires obfuscation of sensitive information
  - Like Sam's login and password at each account

Name Identifier Management Profile

- Different kinds of opaque handles can be set up, modified, and removed
  - > For Pitch Tree and Dollarz, a *persistent pseudonym* is the right answer
  - > Another option for anonymous transactions is a *transient* identifier



For More Information

- From the **OASIS SSTC**:
  - Executive Overview, Technical Overview, presentations
  - saml-dev@oasis-open.org discussion list
  - http://www.oasis-open.org/committees/security
- From the **Liberty Alliance**:
  - Circle of Trust Legal Framework document
  - Implementation Guidelines
  - Business Guidelines for Mobile Deployments
  - Privacy and Security Best Practices
  - And much more...
  - http://www.projectliberty.org

FEDERATED IDENTITY MANAGEMENT:
An Overview of Concepts and Standards

Eve Maler
eve.maler@sun.com
http://www.xmlgrrl.com/blog