The Venn of Identity

Options and Issues in Federated Identity Management

Federated identity management lets users dynamically distribute identity information across security domains, increasing the portability of their digital identities. It also raises new architectural challenges and significant security and privacy issues.

Eve Maler
Sun Microsystems

Drummond Reed
Cordance

Despite aging and psychological and cosmetic changes, who you are as a person is fairly constant—Eve and Drummond will remain Eve and Drummond over time. The same isn't true of your digital identity. Currently, eve@xmlgrrl.com is tied to Eve, for example, but might later be tied to someone else or disappear entirely. This is just one of the challenges people have with digital identities.

Federated identity management is a set of technologies and processes that let computer systems dynamically distribute identity information and delegate identity tasks across security domains. Federated identity is the means by which Web applications can offer users cross-domain single sign-on (SSO), which lets them authenticate once and thereafter gain access to protected resources and Web sites elsewhere.

However attractive its benefits, federated identity imposes costs as well, entailing new and increased security and privacy risks because it shares valuable information across domains using loosely coupled network protocols. Such risks require mitigation, which can range from preventing message replay to collecting user consent for data sharing in both online and offline scenarios.

Here, we describe the federated identity model and discuss its security and privacy risks and architectural challenges. We also profile three popular federated identity protocols for implementing the model: the Security Assertion Markup Language (SAML),[1] the OpenID specification,[2,3] and the InfoCard specification underlying Microsoft's Windows Cardspace.[4]

Identity management and single sign-on

Although many objects have digital identities—from RFID-tagged equipment to software applications to companies—digital identities for humans raise the most interesting issues and challenges, whether in an enterprise context or on the open Internet.

Large-enterprise IT departments view identities as the user accounts for employees (and others) that they manage through a user store (often based on the Lightweight Directory Access Protocol). As enterprises grow, management teams must synchronize their account stores, both to ensure proper account provisioning and to govern users' application access. Due to mergers, acquisitions, and joint ventures, however, enterprises often find that managing identities this way is costly and brittle.

Most Web sites and applications view identities as the accounts they host on behalf of their users, who access email, buy goods, engage in social activities, and so on. While Web applications manage user accounts much as their enterprise counterparts do, users tend to think of these identities as personal resources under their own control. And, unlike in enterprise scenarios, the biggest problems with Web identity are borne by users: they must create and remember their usernames and passwords for each site, populate each profile with the same data, and remember each site's arcane rules.

Federated identity offers solutions to many problems shared by both environments, and SSO is often the first federated identity capability that organizations add. SSO offers Web users a friendlier experience through a more consistent and less frequent login process, and gives employees more time to make products or provide services. Further, combining SSO with account linking lets Web portals unify diverse online interactions—a popular feature that can, for example, let e-government initiatives present many different agency sites as a unified whole. Finally, SSO can simplify the architecture of each participating site.

SSO involves sharing information about when and how users authenticate using a particular identity. It can also involve sharing user attributes such as employee roles and shipping addresses. With all this information in hand, receiving sites can make sophisticated authorization decisions, such as ensuring that managers see only their direct reports' salaries, and present customized user interfaces, such as automatically calculating shipping costs and schedules based on the user's address.

Overview: Federated identity model

Whenever a human is involved in an identity interaction, the federated model involves four logical components:

- The *user* is a person who assumes a particular digital identity to interact with an online network application.
- The *user agent* is a browser or other software application that runs on anything from a PC to a mobile phone to a medical device. A user's online interactions always take place through an agent, which can passively allow identity information flow or actively mediate it.
- The *service provider* (SP) site is a Web application—such as an expense-reporting application or an open source community—that offloads authentication to a third party, which might also send the SP some user attributes. Because the SP relies on external information, it's often called a relying party (RP).
- The *identity provider* (IdP) is a Web site that users log in to and that sometimes stores attributes of common interest to share with various SPs.

In SSO, data about identification, authentication, and sometimes attributes flows from the IdP to the SP. However, SSO has several variants, each of which dictates different flows and data choices. To illustrate this, we'll offer examples with a user named Alice. In one common variant, Alice begins her browsing at an SP, such as an investment management site, which she might visit frequently by using a browser bookmark. If Alice wants to access protected resources there, the SP must send an explicit authentication request to Alice's bank (the IdP). This pattern is known as *SP-initiated SSO*. An alternative pattern is *IdP-initiated SSO*, in which an IdP, such as a health insurance site, acts as a portal through which Alice accesses various SPs, such as online pharmacies and billing statement aggregators. In either case, if Alice's relationship with an SP predates her IdP relationship, the IdP and the SP accounts must be linked (with her permission) to make SSO successful.

When the architecture separates the identity information's source from its usage, everyone benefits:

- Alice can log in once—with one set of credentials—and access multiple Web sites without revealing her credentials to all of them.
- SPs can delegate many account-management tasks (such as password resets) and receive accurate just-in-time user data.
- IdPs can focus on improving authentication methods and adding attractive features to account-management interfaces.

But this loose coupling of identity tasks also introduces several security, privacy, and architectural challenges.

Security considerations

Like all outsourcing, federated identity can offer better service at a lower cost, but it also entails new risks. First, federated identity involves crossing security domains. Ideally, all parties should secure their communication channels against replay attacks, man-in-the-middle attacks, session hijacking, and other threats that allow malicious use of user information or Web resources. In an HTTP context, security architects consider Secure Sockets Layer/Transport Layer Security (SSL/TLS) with mutual authentication as a security baseline. Still, application deployers often avoid, overlook, or only partially implement this step.

User authentication is another weak link in the Web identity chain. Currently, most sites rely on username/password pairs because this method poses the smallest initial burden for users and site administrators. However, it's notoriously weak and susceptible to phishing attacks.

For SPs, federated identity is less expensive than implementing a high-quality authentication infrastructure because it offloads the authentication task to an IdP. However, IdP-based SSO can magnify the costs of a stolen password because it expands the scope

Most sites rely on username/password pairs because this method poses the smallest initial burden for users and site administrators. It's notoriously weak to phishing attacks.

of malicious activity. Most SSO protocols offer ways to mitigate this risk. For example, they might limit to a minute or less the valid lifetime of the security token that an IdP sends to SPs; some protocols also offer a single logout (SLO) feature that offers users near-simultaneous sign-out of all SSO-accessed Web sites. Also, while most protocols let the parties to federated identity interactions choose the authentication method used, they usually offer a way for IdPs to describe the method they applied in each instance so that SPs can consider it when making authorization decisions.

Privacy issues

Sharing personally identifiable information is a great concern in managing privacy, protecting data, and complying with regulations. With federated identity, however, sharing such information is often a key goal, which raises interesting privacy issues. It's possible, for example, for an SP site to learn a user's globally unique digital identifier during SSO, even if it's not necessary to know "who the user is."

Given this, we must consider privacy and minimal disclosure at a foundational level—that is, at the identifiers that serve as digital identity labels. Federated identity systems often manage many types of identifiers assigned by different IdPs in various contexts. Such identifiers might be

- absolute (context-independent and omnidirectional) or relative (context-dependent and unidirectional);
- single-part unique values, hierarchical segments, or multipart aggregated keys;
- raw, hashed, or encrypted; or
- anonymous, pseudonymous, or "veronymous" (fully revealing the user's real-world identity).

Pseudonyms are an important technique for preserving privacy, especially when multiple Web services cooperate to provide an aggregated offering that necessitates user-attribute sharing. If an IdP communicates with an SP about Alice using a pseudonym unique to the IdP-SP-Alice relationship—rather than Alice's social security number or email address—it prevents multiple SPs from correlating Alice's activities and thwarts eavesdroppers (unless they use sophisticated timing attacks). Even if SPs know Alice by a pseudonym, however, she still might share enough attributes to identify herself partially or fully to an SP. For example, combining as little as a postal code and an annual income figure can often be personally identifiable. In this case, a system of informed user consent can help safeguard against excessive disclosure.

In the real world, we use some identifying documents for various third-party purposes without the document issuer's knowledge. For example, Alice might show her driver's license to a bartender to prove she's of legal drinking age. To achieve similar "unlinkability," federated identity protocols must apply special data flows and careful encryption to prevent IdP visibility into a user's SP relationships.[5]

Architectural challenges

Federated identity's loosely coupled nature presents interesting design challenges.

IdP discovery. To provide portal-style IdP-initiated SSO, administrators can typically configure an IdP to contact its partner SP sites when Alice wants to visit them. But with SP-initiated (bookmark-enabled) SSO, we encounter the IdP discovery problem—that is, how does an SP know where to send its authentication request when Alice visits and wants an identity-based service? This "where are you from?" problem has a few possible solutions.

If the SP is in a prearranged IdP partnership (a "circle of trust" that often involves business contracts and legal liability agreements), we can statically configure it with the IdP's location. If the SP must choose from multiple IdPs—that is, if it has no established IdP relationships, its circle of trust includes multiple IdPs, or it belongs to several circles of trust—Alice might have to input her IdP's location. This scenario is known as simplified (rather than single) sign-on. Here, the process isn't seamless, which exacts a significant cost when attention and usability are at a premium. Another option is to give Alice a user agent that's smart enough to know the answer. As Web browser limitations become more risky and devices such as smart phones gain popularity, the role of "smart clients" is becoming increasingly important.

Identifier schemes. With federated identifiers, we can represent the same identity across multiple naming authorities and resolve them at different scopes. For example, Alice's identity might be interpreted as a particular IdP's user, a particular group's member, or a unique individual.

The most common federated identifiers are data network IDs, such as IP addresses and DNS names. The Web's URI and Internationalized Resource Identifier (IRI) standards let Web content creators combine these identifiers into hierarchical global identifiers. A new addition to this series is the Extensible Resource Identifier standard, designed expressly for digital identity.[6] XRI provides an abstraction layer for URIs and IRIs, similar to the one DNS provides for IP addressing. We can, for example, resolve a personal XRI such as the *=drummond.reed* i-name into multiple URIs that represent Drummond in different contexts, including his blog, Skype ID, and so on. Federated identity systems can also use IDs adapted from other types of networks, such as phone numbers, instant messaging

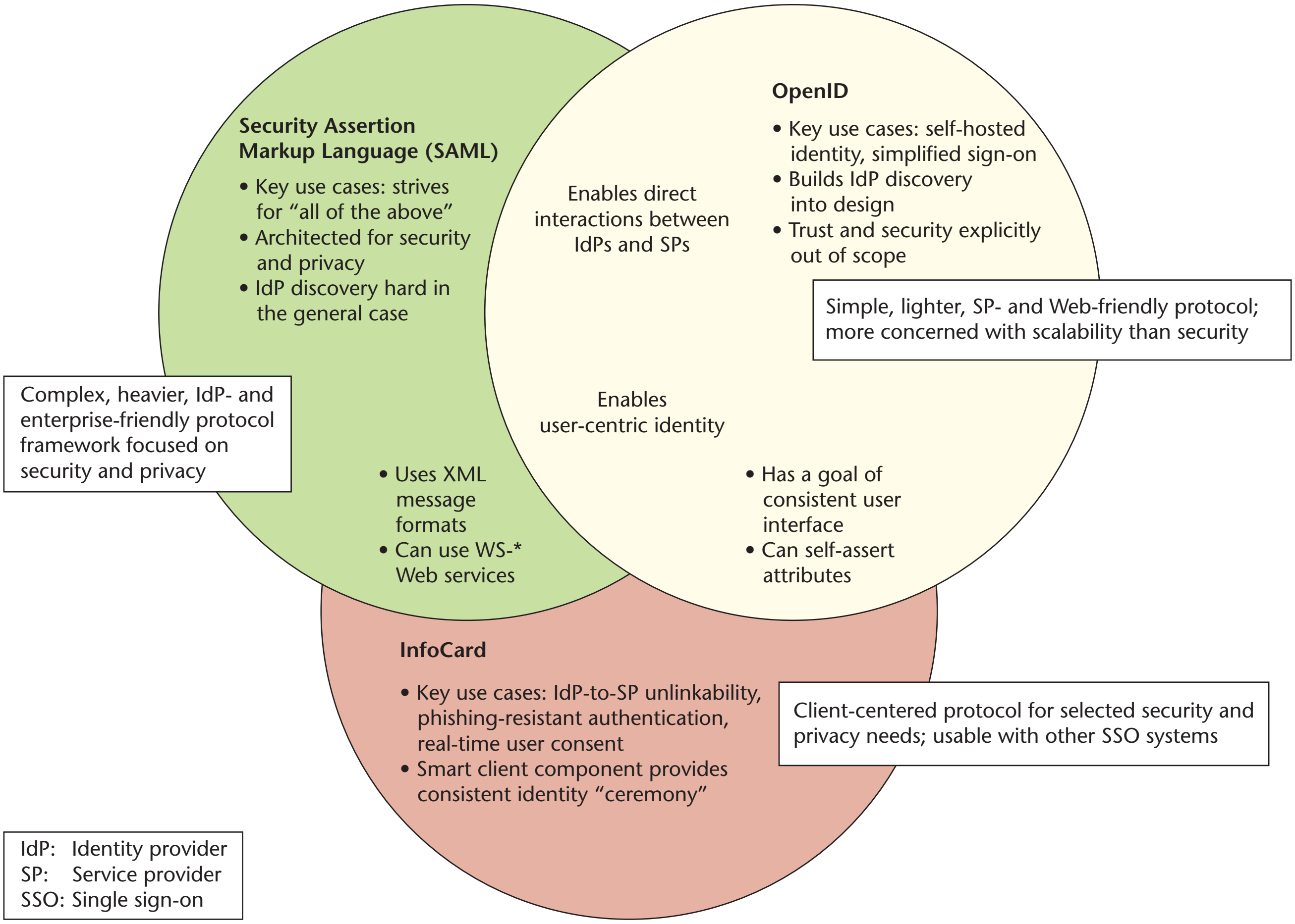

Security Assertion Markup Language (SAML)
• Key use cases: strives for "all of the above"
• Architected for security and privacy
• IdP discovery hard in the general case
OpenID
• Key use cases: self-hosted identity, simplified sign-on
• Builds IdP discovery into design
• Trust and security explicitly out of scope
Enables direct interactions between IdPs and SPs
Simple, lighter, SP- and Web-friendly protocol; more concerned with scalability than security
Complex, heavier, IdP- and enterprise-friendly protocol framework focused on security and privacy
Enables user-centric identity
• Uses XML message formats
• Can use WS-* Web services
• Has a goal of consistent user interface
• Can self-assert attributes
InfoCard
• Key use cases: IdP-to-SP unlinkability, phishing-resistant authentication, real-time user consent
• Smart client component provides consistent identity "ceremony"
Client-centered protocol for selected security and privacy needs; usable with other SSO systems
IdP: Identity provider
SP: Service provider
SSO: Single sign-on

Figure 1. Three popular federated identity protocols. The Security Assertion Markup Language (SAML), OpenID, and the InfoCard protocol used by Windows Cardspace have both common traits and clear differences.

addresses, and postal addresses, as well as account IDs, such as employee, government, and customer numbers. However, sharing such identifiers with other parties invites correlation and privacy issues.

User empowerment. Fully empowering human beings to control their identities—from informed consent for identity data sharing to actually controlling the accumulated data that represents them online—is an ongoing business and technical challenge. The philosophy behind this empowerment goal is known as *user-centric identity*, and it represents a diverse set of use cases and technical approaches.

One approach is to give users total control over their identities, even letting them host their own IdPs (or letting them dictate where they're hosted), as well as control authentication and attribute exchange. This approach indeed empowers users. However, it also requires complex design to avoid both privacy problems and trust issues with authentication quality and attribute veracity. Without corroboration by a trusted third party, no SP is obliged to believe Alice when she asserts "I'm old enough to legally buy alcoholic beverages."

Another approach gaining currency is to gather user consent for data sharing at the moment the SP requests it, exposing the request's nature and extent so that the user can make the most informed decision possible. This approach might offer SP advantages (assisting in liability auditing, for example), but it also places demands on user attention, could require special user-agent technology, and assumes a rich policy and permission-tracking environment.

Of course, even when they're not online, users want to ensure that data sharing happens according to their instructions. For example, Alice might wish to allow certain people to see and add entries to her calendar when she's offline. This also requires a rich policy environment.

Federated identity protocols

Currently, there are three popular protocols for implementing various aspects of federated identity:

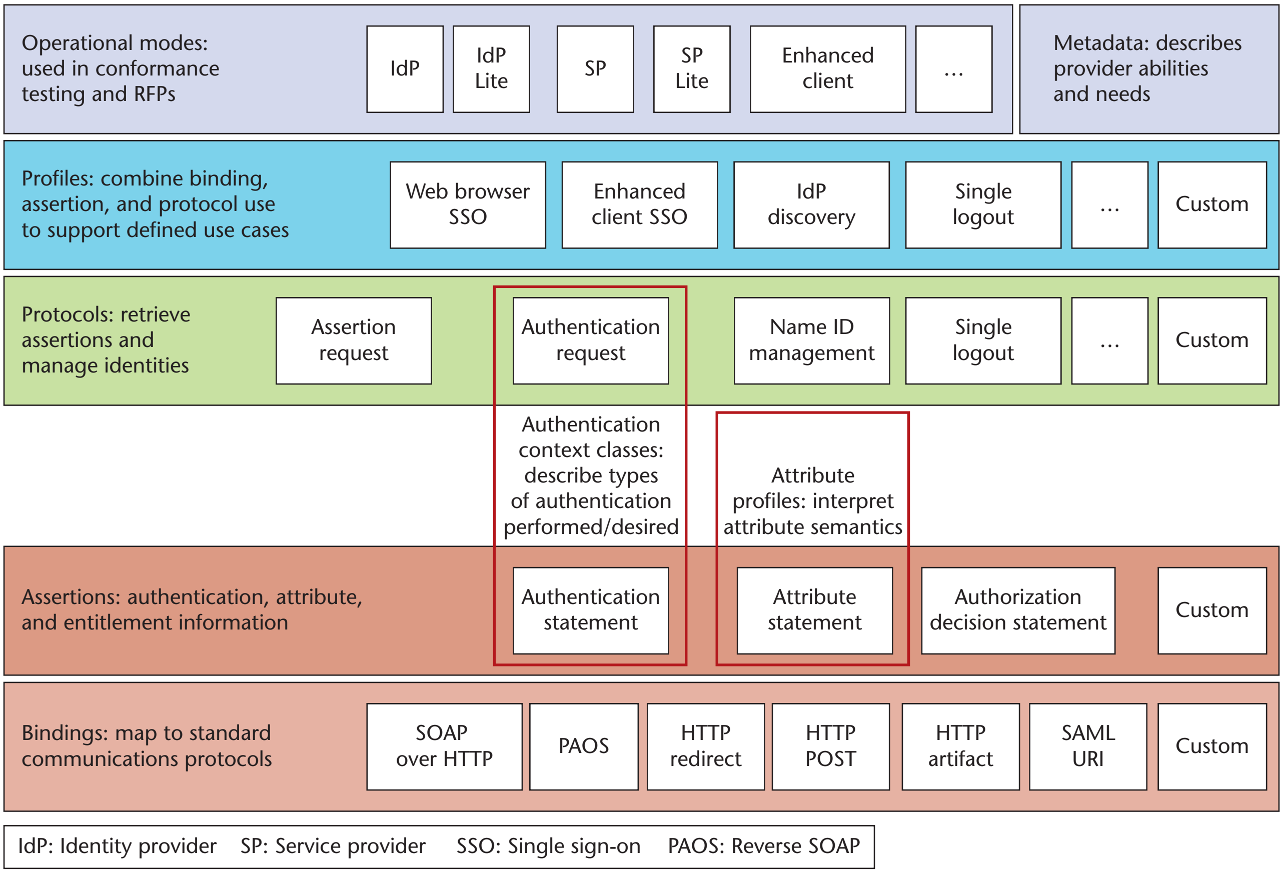

Operational modes: used in conformance testing and RFPs
IdP
IdP Lite
SP
SP Lite
Enhanced client
…
Metadata: describes provider abilities and needs
Profiles: combine binding, assertion, and protocol use to support defined use cases
Web browser SSO
Enhanced client SSO
IdP discovery
Single logout
…
Custom
Protocols: retrieve assertions and manage identities
Assertion request
Authentication request
Name ID management
Single logout
…
Custom
Authentication context classes: describe types of authentication performed/desired
Attribute profiles: interpret attribute semantics
Assertions: authentication, attribute, and entitlement information
Authentication statement
Attribute statement
Authorization decision statement
Custom
Bindings: map to standard communications protocols
SOAP over HTTP
PAOS
HTTP redirect
HTTP POST
HTTP artifact
SAML URI
Custom
IdP: Identity provider
SP: Service provider
SSO: Single sign-on
PAOS: Reverse SOAP

Figure 2. The Security Assertion Markup Language framework. SAML assertions consist of XML packets containing information such as a target user's identifier, authentication status, and attributes.

SAML, OpenID, and the Identity Selector Interoperability Profile specification (often referred to as InfoCard) underlying Windows Cardspace. SAML is the most mature and comprehensive technology, with versions standardized in 2002 and 2005. OpenID, a community effort, emerged in 2005. It's now in its second generation and continues to be in active development. Cardspace has a Microsoft pedigree, and its protocol was made available more recently. Figure 1 summarizes the protocols' commonalities and distinctions.

Security Assertion Markup Language

SAML is an Oasis and ITU standard (ITU-T X.1141) that offers an XML-based framework for exchanging security and identity information across domain boundaries. SAML is something of a universal identity solvent, and its architecture serves various needs, including supporting non-human identity holders. Its design is driven by strong requirements for trust, high-value transactions, and privacy. Although the framework is flexible and some of its components are reused by other technologies (including Cardspace and various OpenID extensions), SAML offers its own solutions for solving common use cases.

As Figure 2 shows, SAML's core is composed of *assertions*—XML packets containing the identity holder's identifier, authentication status, and attributes. Assertions and protocol messages can be signed, encrypted, and combined into profiles—that is, patterns that solve particular use cases like SSO and account linking.

SAML offers a broad range of solutions for IdP- and SP-initiated SSO, account linking through a federated identifier, SLO, attribute exchange, and long-term federated identifier management. It mitigates many security and privacy risks, such as by offering pseudonyms in several forms. Although SAML doesn't include a prepackaged solution for preventing IdPs from tracking user SP visits, developers can further profile it to meet this requirement.

SAML ties into the Liberty Alliance's Identity Web Services Framework standard, which covers use cases involving offline users and identity-based Web services.[7] ID-WSF includes a customizable interaction service. In our case, for example, Alice can ask her stockbroker to send her text messages when she's offline and needs to approve stock trades that meet her previously configured buy order.

SAML offers a solution for IdP discovery based on a common-domain cookie, but administrators typically prefer other methods. SAML is often deployed in circles of trust, anchored by an IdP hub representing a large user community—such as users of a particular mobile phone network—and a defined set of trusted SP "spokes." In this scenario, administrators can configure IdP information into SP sites prior to any SAML interactions and users can experience true SSO.

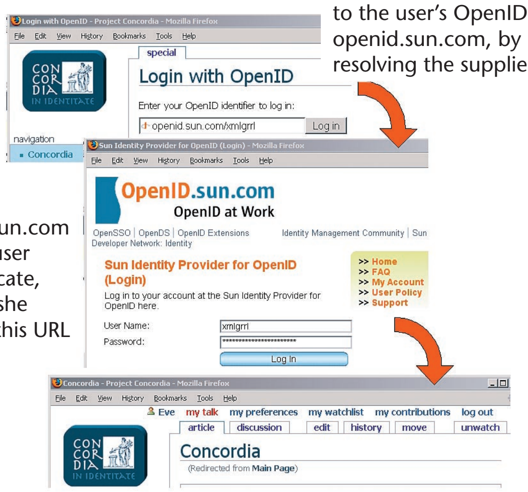
The InCommon Federation illustrates another IdP discovery style. Federation members are universities and other institutions, all of which serve as IdPs to each other. InCommon uses Shibboleth, a SAML-based protocol and implementation, to help students, faculty, and staff share valuable online resources. For example, Cornell University, in the role of an IdP, might want its students and faculty members to access the protected resources of Stanford University, in the role of an SP, and vice versa. When Alice, a Cornell student, arrives at the Stanford Web site to look up a research paper there, Shibboleth lets her choose Cornell from a drop-down list in order to authenticate.

OpenID

Whereas SAML has a broad scope, Brad Fitzpatrick originally developed OpenID for use in the LiveJournal online community as a lightweight, decentralized way to authenticate commenters and avoid blog-comment spam.

OpenID operates like closed-loop email-address authentication: when Alice leaves a comment on Bob's blog, she provides her own blog's URL, which Bob's blog uses to redirect back to Alice's blog—or, according to her blog's instructions, to her preferred IdP—with an authentication request. As Figure 3 shows, this interaction provides simplified sign-on, letting the user log in using an OpenID identifier (that is, the URL itself) at any site that consumes OpenIDs. Because users can choose and even host their own OpenID providers, OpenID exemplifies one important approach to the user-centric identity philosophy. The Web currently has an OpenID ecosystem based largely on open source implementations; many such sites freely offer OpenIDs to users, and several thousand sites accept them.

OpenID is evolving rapidly. Its first version supported only URL identifiers, but it's since expanded to support XRIs and their Extensible Resource Descriptor Sequence format,[8] allowing more sophisticated discovery of IdPs and their capabilities. OpenID's newer versions also let federated identity parties exchange basic user attributes. OpenID has yet to tackle more complex use cases, such as federated identifier management or SLO.


Login with OpenID - Project Concordia - Mozilla Firefox
File Edit View History Bookmarks Tools Help
special
CON COR DIA IN IDENTITATE
Login with OpenID
Enter your OpenID identifier to log in:
openid.sun.com/xmlgrrl
Log in
navigation
Concordia
Sun Identity Provider for OpenID (Login) - Mozilla Firefox
OpenID.sun.com
OpenID at Work
OpenSSO | OpenDS | OpenID Extensions
Identity Management Community | Sun Developer Network: Identity
Sun Identity Provider for OpenID (Login)
Log in to your account at the Sun Identity Provider for OpenID here.
>> Home
>> FAQ
>> My Account
>> User Policy
>> Support
User Name:
xmlgrrl
Password:
Log In
Concordia - Project Concordia - Mozilla Firefox
Eve
my talk
my preferences
my watchlist
my contributions
log out
article
discussion
edit
history
move
unwatch
Concordia
(Redirected from Main Page)

projectcordia.org delegates authentication to the user's OpenID provider, openid.sun.com, by resolving the supplied OpenID

openid.sun.com has the user authenticate, proving she "owns" this URL

After the user confirms to openid.sun.com that she wants to share her information with projectconcordia.org, she is logged in there

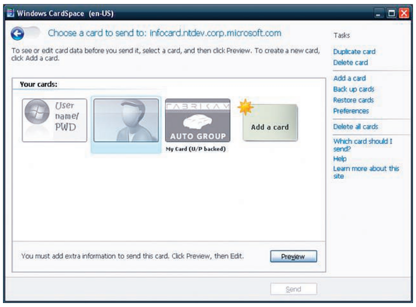
Figure 3. OpenID. Originally developed for an online community, the rapidly evolving OpenID treats Web addresses as user identifiers.

In OpenID, discovery takes place when users supply their universally resolvable identifiers to an SP. Given this, IdPs and SPs that have never met can converse successfully—a type of scalability consciously modeled on that of the Web itself. This can present a privacy challenge: as an architecture biased toward broadly sharing user information, OpenID allows and even encourages different SPs to correlate a user's activity. However, OpenID Authentication version 2.0, released in December 2007, also supports pseudonymous login—that is, Alice could provide her IdP's identifier rather than her own. OpenID also lets the IdP view the SPs that Alice visits. To control this information's dissemination to a third-party IdP, her only option would be to run her own OpenID IdP.

OpenID's IdP discovery model also prevents true SSO, in which the SP can directly visit the IdP site without asking the user to indicate its location. (Although many SAML deployments offer true SSO, developers can also use SAML in an OpenID-like fashion, letting users give the SP a resolvable identifier that provides the IdP's location.[9])

The InfoCard Protocol and Windows Cardspace

Windows Cardspace is a new Microsoft .Net component designed to give users a consistent digital-identity



Windows CardSpace (en-US)

Choose a card to send to: infocard.ntdev.corp.microsoft.com

To see or edit card data before you send it, select a card, and then click Preview. To create a new card, click Add a card.

You must add extra information to send this card. Click Preview, then Edit.

Figure 4. An identity selector interface that uses the InfoCard protocol. A specialized user agent offers users a consistent digital identity built around software-based identity cards.

experience using a specialized user agent; Microsoft has documented the protocol implemented by Cardspace in the InfoCard specification. The Cardspace experience centers on collections of user data called information cards, presented in a wallet-like software interface called an identity selector. Each card represents a different identity; when an SP asks for credentials, the user chooses an identity from the selector (see Figure 4).

When an SP requests authentication and attributes, the identity selector transmits the set of claims requested about the user inside a digitally signed security token. This set of claims corresponds closely to the notion of a SAML assertion, and, in fact, one of the supported token types is a SAML token. Cardspace supports two types of cards:

- Self-asserted cards represent a small, fixed attribute set whose values are determined solely by the user (somewhat like OpenID identities). In Microsoft's implementation, such claims are stored directly on the user's device.
- Managed cards represent IdPs' extensible sets of claims about the user. An identity selector typically retrieves the claims from the issuing IdP each time the user selects a specific card in response to an SP's request. Managed cards are somewhat like typical SAML-federated identities in that the IdP governs their identifiers, claims, and lifetimes.

In both cases, users' identity selectors will only let them choose a card that meets an SP's policy requirements.

Deploying any special client technology imposes a cost, but it also permits more elegant solutions to problems such as IdP discovery. Such client technology is a central feature of SAML's Enhanced Client Profile, and the ID-WSF Advanced Client specifications include a similar solution. The InfoCard model addresses this by eliminating the SP's need to connect to the IdP: for self-asserted cards, the client device itself can be the IdP; for managed cards, the IdP stores its address on the card for the identity selector's own use.

A managed card reflects a user's close relationship with an IdP, and an identity selector can use this to enhance Web authentication's phishing resistance. As a gatekeeper between IdP–SP communications, an identity selector also lets users prevent the IdP from learning which SPs they're patronizing. Finally, an identity selector applies user-centric principles in collecting user consent.

Currently, the InfoCard protocol is compatible only with the WS-* Web services protocols, which center on WS-Trust.[10] However, the Eclipse Foundation's open source Higgins Project (www.eclipse.org/higgins) is working on an Apache-like approach for identity, aiming to make identity selectors and the information card model available in a plug-in API architecture that works with multiple protocols.

Interoperability issues

Interoperability is an ongoing challenge for federated identity. Even within a single protocol, interoperability among online partners can be difficult because of protocol options, conformance variations, and the architecture's cross-platform nature. Protocol overlaps further complicate the picture:

- SAML and OpenID both address simplified sign-on, but not identically;
- InfoCard and SAML both offer smart-client solutions, but optimize them for different purposes; and
- OpenID and InfoCard both claim to offer user-centric identity, yet the term refers to multiple and sometimes incompatible goals.

Nonetheless, many deployers are likely to begin using these technologies together as they grow in popularity—authenticating to a SAML or OpenID IdP using an information card, or using OpenID instead of SAML as a first step in launching a Liberty identity-enabled Web service.

The first step in solving interoperability problems is understanding the distance between technologies. As an example, Jeff Hodges offers a useful, in-depth comparison of OpenID and SAML.[11] The industry is also beginning to coordinate cross-technology interoperability to promote unified identity handling across networks—much the same as TCP/IP promoted unified distribution of IP packets. Project Concordia (www.projectconcordia.org), for example, explores deployers' multiprotocol issues and develops scenarios involving technology combinations, while Identity Commons' Open Source Identity System working group (http://osis.idcommons.net) helps test interoperability among open source identity projects such as Higgins, Bandit (www.bandit-project.org), and OpenXRI (www.openxri.org).

Federated identity presents complex challenges in terms of technical issues and human needs. Important requirements often seem mutually exclusive. Some security aspects, such as full auditing of system-resource access, can conflict with user privacy issues, such as keeping a user's true identity private. At the same time, user empowerment styles, such as having users act as real-time data flow intermediaries, can conflict with user conveniences, such as achieving totally silent SSO.

Two current development efforts aim to solve such conflicts. NTT Laboratories' Sasso project seeks to let users strongly authenticate to a regular browser-based SSO session using standard-issue mobile phone SIM cards over standard SAML protocols.[12] In addition to avoiding a security/deployer and convenience/custom-protocol trade-off, the Sasso approach offers user-centric features such as real-time consent gathering and self-hosted identity.

In another project, Identity Commons established an Identity Rights Agreements working group (http://wiki.idcommons.net/moin.cgi/IdentityRightsAgreementsCharter) that patterns itself on the highly successful Creative Commons model for copyright licenses (www.creativecommons.org). The Identity Rights Agreements goal is to create a small set of standardized agreements—each represented by a recognizable icon—by which sites can offer and individuals can select the terms under which they agree to share personally identifiable information.

Finally, the annual ACM Digital Identity Management Workshop series has been presenting new digital identity research for several years. In 2007, the workshop's focus was user acceptance of digital identity paradigms in Web 2.0 online applications (http://www2.pflab.ecl.ntt.co.jp/dim2007). Topics included strengthening authentication while increasing usability; models for assessing linkability of online data to individuals; and establishing trust using reputation rather than traditional identification and authentication. □

References

1. *Security Assertion Markup Language (SAML) V2.0*, Oasis, 2007; http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf.
2. *OpenID Authentication 2.0*, OpenID Foundation, 2007; http://openid.net/specs/openid-authentication-2_0.html.
3. *OpenID Attribute Exchange 1.0*, OpenID Foundation, 2007; http://openid.net/specs/openid-attribute-exchange-1_0.html.
4. *Identity Selector Interoperability Profile 1.0,* Microsoft, 2007, http://download.microsoft.com/download/1/1/a/11ac6505-e4c0-4e05-987c-6f1d31855cd2/Identity-Selector-Interop-Profile-v1.pdf.
5. A. Pfitzmann and M. Hansen, *Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management—A Consolidated Proposal for Terminology v0.31*, 15 Feb. 2008; http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.31.pdf.
6. *Extensible Resource Identifier (XRI) Syntax 2.0*, Committee Specification, Oasis, 2005; www.oasis-open.org/committees/download.php/15377.
7. *Identity Web Services Framework 2.0*, Liberty Alliance, 2006; www.projectliberty.org/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications_including_errata_v1_0_updates.
8. G. Wachob et al., eds., *Extensible Resource Identifier (XRI) Resolution 2.0*, Committee Draft, Feb. 2008; http://docs.oasis-open.org/xri/2.0/specs/xri-resolution-V2.0.html.
9. J. Hodges, *OpenID-SAML Lightweight Web Browser SSO Profile*, IdentityMeme, 21 Sept. 2007; http://identitymeme.org/doc/draft-hodges-saml-openid-profile-02.html.
10. A. Nadalin et al., eds., *WS-Trust 1.3*, Oasis Web Services Secure Exchange Technical Committee, 19 Mar. 2007; http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html.
11. J. Hodges, *Technical Comparison: OpenID and SAML*, IdentityMeme, 17 Jan. 2008; http://identitymeme.org/doc/draft-hodges-saml-openid-compare.html.
12. T. Abe, H. Itoh, and K. Takahashi, "Implementing Identity Provider on Mobile Phone," *Proc. ACM Workshop on Digital Identity Management* (ACM DIM), ACM Press, 2007, pp. 46–52.

***Eve Maler** is a principal engineer at Sun Microsystems, where she develops interoperability strategies and leads partner engagements related to Web services, security, and identity. She has a BA in linguistics from Brandeis University. She was one of the coinventors of the XML and coauthor of* Developing SGML DTDs: From Text to Model to Markup *(Prentice Hall). Maler also contributed to the development of standards such as SAML, Liberty Alliance, the Universal Business Language, and DocBook, and cofounded interoperability efforts, such as Project Concordia. Contact her at eve.maler@sun.com; www.xmlgrrl.com.*

***Drummond Reed** is chief architect of Seattle-based Cordance Corporation and VP infrastructure of Boston-based Parity Communications. His research interests include technology, standards, and applications for XML-based persistent identity and trusted data sharing. He is cochair of the Oasis Extensible Resource Identifier (XRI) and XRI Data Interchange (XDI) technical committees, and a founding board member of the OpenID Foundation, OpenXRI.org, and Identity Commons. Contact him at drummond.reed@cordance.net; http://xri.net/=drummond.reed.*

SECURITY & PRIVACY
Risking Communications Security
Software Susceptibility
Under Surveillance
Infrastructure Security
Attacking Systems