Internet-Scale Identity Systems: An Overview and Comparison

Overview

An Internet-scale identity system is an architecture that defines standardized mechanisms enabling the identity attributes of its users to be shared between applications and Web sites. This enables a streamlined and optimized online experience for users, greater protection from identity theft, and opportunities for customization and personalization that do not require the user to manually configure account information. There are a number of different technologies and standards initiatives designed to deliver an Internet-scale identity system including SAML, OpenID, Information Cards, and OAuth. This white paper provides an overview of key initiatives and explores the similarities, differences, and synergies among them.
# Table of Contents

Introduction ................................................. 3
Identity Systems ............................................ 3
    SAML .................................................... 3
    Information Cards ..................................... 4
    OpenID ............................................... 4
    OAuth ............................................... 5
An Identity Conversation .................................. 5
    Today’s Conversation ................................ 5
    An Improved Conversation ......................... 6
Analysis of Identity Systems .............................. 7
    Identity Type ........................................ 7
    Authentication ....................................... 8
    Identity Flow ........................................ 9
    Trust Model ......................................... 10
    Discovery .......................................... 11
Converged Use Cases ...................................... 12
    OpenID with Information Card Authentication & SAML SSO 12
    OpenID Account Registration, Authentication and SSO with OAuth Service Authorization .. 13
    SAML Authentication and With OpenID SSO ......................... 14
Summary: Coexistence or Convergence .................. 15
Introduction

An Internet-scale identity system is an architecture that defines standardized mechanisms enabling the identity attributes of its users to be shared between applications and Web sites. This enables:

- A streamlined and optimized online experience for users
- Greater protection from identity theft
- Opportunities for customization and personalization that do not require the user to manually configure account information.

A number of different technologies and standards initiatives are designed to deliver an Internet-scale identity system. This white paper provides an overview of the key initiatives and explores the similarities, differences, and synergies among them.

For the purpose of establishing context, the following are definitions of specific terms used throughout this white paper:

**Identity Provider (IDP)** — A type of service provider that creates, maintains, and manages identity information for users and provides user authentication on behalf of other service providers.

**Relying Party/Service Provider (RP/SP)** — A system entity that decides whether or not to take an action based on information provided by another system entity such as an Identity Provider.

**User** — A person who uses an information system and its resources for any purpose.

**Active Client** — A smart client or service that assists the user in controlling and directing identity transactions.

**Selector** — An active client using the information card metaphor.

Identity Systems

This section presents an overview of the Internet-scale identity systems that have emerged.

**SAML**

Security Assertion Markup Language (SAML) is an XML-based framework for the communication of security-focused identity information and is standardized under the OASIS Security Services Technical Committee (SSTC). The SAML 2.0 specification set is composed of rules for the structure of identity assertions, protocols for moving assertions from place to place, bindings of protocols to typical message transport mechanisms, and profiles that tie all the above into interoperable patterns for common use cases (e.g. Browser Single Sign On, Web Services Security, etc.).
SAML is sometimes criticized for its heaviness — specifically, for the complexity of the specifications and the relative constraint of its security rules. The complexity and security rules are a result of SAML’s development for sophisticated business scenarios where SAML has consequently experienced significant success.

Recently, the SAML community has seen significant interest in extending SAML to meet less stringent requirements for low-sensitivity use cases. For example, the SAML “Simple Sign Binding” allows the XML Signature requirements of the core SAML specification to be ignored or replaced with a simpler signing mechanism.

SAML’s key distinctions are its broad scope, robust security and privacy model, and demonstrated interoperability between multiple vendor implementations through the Liberty Alliance’s Conformance Program (Now a part of the Kantara Initiative).

**Information Cards**

Information Cards is based on the OASIS IMI specification version 1.0, where a user leverages a selector to control the delivery of identity information between parties. Windows CardSpace is one example of a selector: a Windows client application delivered in the Windows 7, Vista, and XP operating systems. When a Web application requires user identity information, it activates CardSpace, which takes over the interface to guard against phishing attacks, examines the identity sought by the Web application, compares the identity request to a set of Information Cards controlled by the user, and determines whether any existing Information Cards contain data suitable to meet the request from the Web application. After choosing an appropriate Information Card IDP (potentially with the user’s assistance), CardSpace then forwards on the request from the Web application IDP, and similarly proxies the returned identity attributes.

At Web sites that support Information Cards, a user can authenticate using a previously established secret key rather than a password. Complex cryptography is possible because the selector manages the keys and performs authentication to the site on behalf of the user. Selectors also enable self-asserted personal information to be exchanged, if it is useful to the Web site and allowed by the user.

While CardSpace is specific to Microsoft Windows, selectors have been created for other platforms through initiatives such as Higgins and Azigo. Interoperability between participants is coordinated through organizations like the Open Source Identity Systems (OSIS) group and the Information Card Foundation (ICF).

**OpenID**

OpenID is both an identifier format and an extensible set of protocols for passing identity information. Users enter their identifier at the site they wish to access and are redirected to the appropriate Identity Provider. During this session, the user can move between multiple OpenID Relying Party sites without re-entering their password at the Identity Provider, because the Identity Provider simply checks the session and transparently returns control to the requesting site. The OpenID base protocol is very simple, but can be augmented by additional extensions to pass registration attributes or to classify the strength of authentication method used at the Identity Provider. OpenID identifiers may have several forms, including a URL, an i-name, or a DNS domain name. The next version of the OpenID protocol may also ratify the use of an email address.
OAuth

OAuth 1.0a is an IETF standard that began as a delegation protocol built to allow users to granularly grant service and data access to external relying parties. A common usage of OAuth is the “OpenID/OAuth Hybrid Model” which combines an OpenID authentication (and possibly a registration) with a subsequent request for permission to consume ongoing services. As with most Internet-scale Identity models, OAuth uses HTTP redirection to communicate between the parties. OAuth has two different common flows: 3-legged OAuth involves redirection to a user-facing approval page, while 2-legged OAuth is used for already-approved data and simply involves a request and response.

An Identity Conversation

Service providers need information about a user in order to deliver a meaningful experience. The route by which the RP/SP obtains identity information can be considered a conversation between the various parties involved.

The section below is an overview of the model this conversation takes today as well as the associated issues. Following that is a second overview of the model when a dedicated Identity Provider is added to the conversation.

Today’s Conversation

Almost all identity interactions on the Web today take the form of a two-way conversation between a user and an RP/SP that the user is visiting:

User ➔ RP/SP: Hello, I’d like Service “A.”

RP/SP ➔ User: Before I can provide Service “A,” I require Identity Attribute “X” (such as name, email address, password, zip code or credit card number etc.)

User ➔ RP/SP: Here is Identity Attribute “X.”

RP/SP ➔ User: Here is Service “A.”

There are a number of issues with the above model:

- Users bear the burden of tracking and managing their identity attributes in order to deliver them as required to the RP/SP. Risky behavior (e.g. writing down passwords, weak passwords, reusing passwords across different sites, using a single email address for all requests, etc.) can severely compromise the security and privacy of their identity.
- When a user forgets, loses, or is otherwise unable to provide required identity attributes (such as a password), the RP/SP must provide mechanisms to help the user remember or retrieve them. Providing such mechanisms (e.g. password reset) is a significant cost and support burden for RPs.
- RP/SPs are required to store passwords and personal information for their users, creating an assumption of liability and reduced focus on the core value of the service they provide.
- Users expect to provide identity information to the service providers with which they interact. The constant barrage of requests for their identity information frequently leads users to adopt a conditioned response of either always providing identity information or never providing identity information.
• Most users are ill-suited to properly assess the trustworthiness of sites requesting their identity information. Consequently, they have become targets for phishing and other sophisticated online attacks that rely on this susceptibility.

• For many RP/SPs, self-asserted identity attributes are of questionable worth as the RP/SP must still validate the attributes (e.g. Social Security number) or risk accepting fraudulent attributes (e.g. credit card number). In such cases, it is ideal for RP/SPs to receive the user’s identity attribute claims from a trusted third party.

An Improved Conversation

Many of the above issues can be addressed by introducing a new actor into the conversation — an Identity Provider (IDP). The IDP removes the need for the user to provide identity attributes when requested by an RP/SP. The IDP acts on the user’s behalf to provide identity information required by the authorized RP/SPs with which the user interacts.

Introducing the IDP changes the previous conversation as follows:

User \(\rightarrow\) RP/SP: Hello, I’d like Service “A.”

RP/SP \(\rightarrow\) User: Before I can provide Service “A,” I require Identity Attribute “X.”

User \(\rightarrow\) RP/SP: Which IDPs can meet this requirement? IDPa is a match.

User \(\rightarrow\) IDPa: RP/SP requires Identity Attribute “X.”

IDPa \(\rightarrow\) User: Here is Identity Attribute “X.”

User \(\rightarrow\) RP/SP: Here is Identity Attribute “X.”

RP/SP: Let’s see what I know about IDPa. OK, good enough.

RP/SP \(\rightarrow\) User: Here is Service “A.”

The above example represents a logical sequence; few identity interactions will include all of the above steps or follow this order. As an example, the flow of identity is not always “through” the user, but rather could flow directly from the IDP to the RP/SP. Even then, the assumption is that the user has implicitly made the request and so the exchange could be considered compliant with their privacy policies.

We can revisit the previously stated issues in the context of the restyled identity conversation:

• The burden on the user can be significantly reduced through SSO (fewer authentication operations and fewer passwords to remember) and automated identity attribute exchange (reduced form filling).
• RP/SPs can outsource expensive identity management and support functions to Identity Providers, who can more readily factor such costs into their business model.
• RP/SPs can focus on their service offering rather than deal with the increasingly legislated requirements of managing and protecting personal data.
• Users may become more discerning and less casual in the management of their identity attributes. Armed with appropriate tools, they may feel less overwhelmed and consequently less likely to choose either of the two easy choices — release identity information to any request or never release identity information.
• The burden of evaluating the authenticity of the RP/SP is removed from the user and placed on systems designed for the task.
• Depending on the nature of the identity information and the requirements of the RP/SP, identity can be provided by a trusted third party rather than be supplied by the user. The RP/SP may consequently be able to place greater confidence in its validity.

Analysis of Identity Systems

All of the identity systems described in this white paper can be analyzed in terms of the above conversational framework.

The choices the systems make for the various steps can have implications for the security and privacy the systems are able to deliver — and consequently for the sensitivity of the use cases they can support. We will see how these same choices define areas of applicability where each system may offer unique value.

Identity Type

Identity systems typically focus on the exchange of certain types of identity attributes — either unique user identifiers or more general attributes (such as email address, telephone number, user profile information, etc.).

While the communication of an identifier on its own can enable Single Sign-On, richer use cases generally require additional information.

<table>
<thead>
<tr>
<th>Identity Type</th>
<th>Identifiers</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML</td>
<td>Allows for a variety of identifier types, from pseudonyms to one-time anonymous.</td>
<td>Defines a general XML-based syntax for communicating identity attributes, and a protocol by which attributes can be exchanged.</td>
</tr>
<tr>
<td>Information Cards</td>
<td>Allows for IDPs, either self-issued or a third party, to issue a Private Personal Identifier (PPID) by which RP/SPs can recognize the user for authentication.</td>
<td>In addition to the PPID claim, RP/SPs can indicate their desire for more general attributes. CardSpace focuses on personal profile attributes (e.g. name, address, email, etc.).</td>
</tr>
<tr>
<td>OpenID</td>
<td>Default assumption is that users are identified by a personal URI.</td>
<td>The OpenID SREG extension focuses on exchange of attributes typically requested at account registration time.</td>
</tr>
<tr>
<td>OAuth</td>
<td>The service is identified rather than the user. An opaque one-time identifier is used to map between user accounts on each side.</td>
<td>OAuth token can be used for access to RESTful API services — the token doesn’t contain attributes but presence of the token in the HTTP header allows the API to be called and the attributes to be retrieved.</td>
</tr>
</tbody>
</table>
The nature of the identifiers used by RP/SPs to identify users is a key differentiator of identity systems. Most significantly, the nature of the identifier impacts the privacy characteristics of the systems with respect to guarding against inappropriate correlation of user actions at different RP/SPs.

Identifiers used by Identity Systems fall into three general categories:

- **Global Identifier** — All RP/SPs use the same identifier to refer to a given user. While attractively simple, the global model makes collusion trivial between RP/SPs.
- **Pair-wise Pseudonyms** — Pairs of providers establish and subsequently use unique identifiers for users. This model inhibits collusion between RP/SPs as they share no common identifier for a user; however, correlation based on other identity characteristics (e.g. phone number) is possible.
- **One-time Identifier** — Each time a user accesses an RP/SP they have a different identifier, consequently preventing that RP/SP from correlating their actions across subsequent visits. In such a model, because identifying the user accessing an account at the RP/SP is necessarily impossible, the RP/SP generally customizes service based on other identity attributes of the user (e.g. role in an enterprise scenario, frequent flyer status for a B2C scenario, etc.).

SAML supports all of the above identifier types and defines a number of mechanisms tailored to support pseudonyms.

OpenID 1.1 primarily relies on global URIs and inherits the attendant privacy risks. Advocates at the time of 1.1 claimed the risk was mitigated by a user’s ability to present different OpenIDs to different RP/SPs as necessary. While that is true, the implication was that the user bears the burden of remembering and managing multiple URI personas. OpenID 2.0, on the other hand, includes “directed identity,” which is the ability to use a pair-wise pseudonym.

Information Cards and the IMI specification rely on a pair-wise pseudonym called PPID (private personal identifier) for self-asserted cards, and optionally can use either PPID or a privately specified identifier for managed cards.

OAuth uses an opaque one-time identifier to map the identity in a consuming service to the identity at the service provider which owns the requested API data, but does not as part of the protocol itself uniquely identify the user.

**Authentication**

Single Sign-On enables a user to authenticate to an IDP and then have their digital identity asserted by the IDP to RP/SPs, thereby enabling user access to resources at that RP/SP. For some identity systems, the act of authentication by which the user presents some credentials (e.g. password, OTP, X.509 certificate, IP address, etc.) to the IDP is deemed out of scope — the focus is instead on the protocols and syntax by which this act can be communicated to RP/SPs to enable SSO. Both SAML and OpenID take this approach; neither defines mechanisms by which a user authenticates to the IDP, but rather mechanisms to allow the RP/SP to request that the IDP authenticate the user, and mechanisms to allow the IDP to respond that the user has been authenticated. SAML’s Authentication Context allows the IDP to describe how the authentication occurred (e.g. Biometric or SIM card) but not how to perform it. OpenID is currently exploring comparable functionality.
Alternatively, the IMI specifications defines a mechanism by which a user (using a client with capabilities beyond a default browser) can authenticate to an IDP in order to enable subsequent SSO and other identity transactions. Perhaps IMI’s most notable functionality is a model where there is no IDP per se. The user authenticates to RP/SPs simply by presenting a previously established identifier and demonstrating possession of a secret cryptographic key to enable protection from the many issues surrounding passwords (i.e. phishing).

**Identity Flow**

Identity systems can be characterized by the channel through which identity information flows from the IDP to the RP/SP. Fundamentally, there are two high level choices:

- **Front Channel** – Identity information flows through the user-agent from IDP to RP/SP and thereby makes possible direct user mediation
- **Back Channel** – Identity information flows direct from IDP to RP/SP

In principle, the front channel promises improved privacy through the stronger control point enabled by real-time user consent for attribute exchange. Against this must be balanced the reality of insecure clients and the inability of this model to support use cases where the user is offline and not present to actively mediate the identity flow. Message size limitations of the front channel method can be addressed through either HTML Form mechanisms or through smart clients capable of more advanced messaging.

<table>
<thead>
<tr>
<th>Front Channel</th>
<th>Back Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAML</strong></td>
<td>Supported by the SOAP Binding, providers can communicate directly.</td>
</tr>
<tr>
<td>Supported by various browser mediated bindings. The Enhanced Client Profile also supports “smarter” clients than default browsers.</td>
<td></td>
</tr>
<tr>
<td><strong>Information Cards</strong></td>
<td>No back channel. All communications are proxied through the Selector.</td>
</tr>
<tr>
<td>IMI mediates all identity flow — the RP/SP presents its identity request to the Selector, which forwards the request to the IDP. The returned identity flows the same route in reverse.</td>
<td></td>
</tr>
<tr>
<td><strong>OpenID</strong></td>
<td>OpenID 3.0 may introduce support for direct communication.</td>
</tr>
<tr>
<td>The OpenID authentication protocol relies on redirects and HTML Form POSTs through the browser.</td>
<td></td>
</tr>
<tr>
<td><strong>OAuth</strong></td>
<td>OAuth’s 2-legged flow allows services to access information which is already authorized or for which approval is not needed.</td>
</tr>
<tr>
<td>OAuth’s 3-legged flow allows users to approve data flow between services.</td>
<td></td>
</tr>
</tbody>
</table>
Another aspect of identity flow is the point from which identity interactions are initiated. Most typical is a pattern that begins when the RP/SP, in trying to determine access rights or personalization, sends a request to the IDP for the desired identity. A very different set of use cases requires the flow of identity to be initiated from the IDP, and not as a result of a request from the RP/SP. This flow pattern is necessary for portal use cases in which the user, after authenticating to the IDP, can be presented with a list of available RP/SPs with which they have the option of an identity transaction (e.g. a user has seamless access to a partner Web site from a travel portal). The assumption that a particular list of available RP/SPs is presented has implications for the trust model as it requires the IDP to assert identity assertions for the user only to RP/SPs on the list.

Trust Model

<table>
<thead>
<tr>
<th>RP/SP Initiated</th>
<th>IDP Initiated</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML</td>
<td>Yes</td>
</tr>
<tr>
<td>CardSpace</td>
<td>Yes</td>
</tr>
<tr>
<td>OpenID</td>
<td>Yes</td>
</tr>
<tr>
<td>OAuth</td>
<td>Yes</td>
</tr>
</tbody>
</table>

A useful definition of “trust” for the purpose of this white paper is: A reasonable expectation of confidence in an actor’s behavior.

In the context of identity interactions between IDPs and RP/SPs, for one to trust the other means that the first has a reasonable level of confidence that the second will behave in a certain manner in a given context. This may include agreement upon security infrastructure, privacy policies, etc.

Depending on the application, “reasonable” can mean very different things. For low-value and no-risk applications like blog commenting, an RP/SP might establish reasonable confidence in an IDP based solely on the fact that the IDP is able to correctly follow the OpenID SSO protocol. Consequently, such an RP/SP may choose to be indiscriminate in choosing its IDP partners. If there is little or no risk of choosing a bad (malicious, hacked, etc.) IDP, the RP/SP is motivated to accept any IDP. Likewise, an IDP assumes no risk by asserting identity to an arbitrary RP/SP, because the value or sensitivity of the application creates no great risk to be distributed.

For more sensitive applications (e.g. SSO from a bank IDP to a stock analysis RP/SP), the RP/SP must have more stringent criteria for determining which IDPs to engage with in identity interactions. If the risk of accepting an assertion from an IDP is non-trivial, the RP/SP will manage this risk by being more careful in selecting identity partners. Likewise, an IDP may choose to assert identity to only a certain set of RP/SPs in order to protect itself against risk. Importantly, if IDPs and RP/SPs are anything but completely indiscriminate in their choice of partners, security mechanisms are required to determine whether any particular provider meets selected criteria.
Any of the identity systems discussed in this white paper could support a range of trust models; however, each has been designed or optimized for a particular subset of use cases. OpenID defines a relatively low level of security requirements and is appropriate to low-sensitivity applications. For example in OpenID 1.1 and 2.0, an RP/SP and IDP, even with no prior relationship, can use Diffie-Helman cryptographic techniques to establish a shared secret key for authenticating subsequent protocol messages. Such a dynamic mechanism is appropriate for OpenID’s default assumption of indiscriminate providers.

SAML and IMI define a more robust security model necessary for high-sensitivity use cases. Consequently, SAML and CardSpace are more likely to rely on less dynamic cryptographic mechanisms such as X.509 certificates.

Recently, the OpenID community has explored layering security in order to support higher sensitivity use cases, and the SAML community is defining profiles that diminish the security settings from those defined in the core SAML 2.0 specifications.

**Discovery**

Discovery mechanisms are functions by which appropriate IDPs can be identified as candidate providers, for a given combination of desired user identity, RP/SP identity, and policy requirements. Discovery is necessary in any architecture in which there are multiple IDPs, or for which the location of IDPs is potentially dynamic. In situations where there is only ever a single Identity Provider (e.g. an enterprise) there is no need for a discovery operation, because any RP/SPs will implicitly know where to obtain the user’s identity attributes.

Discovery is similar to a Web search for an identity. The inputs to the engine are the user in question and the type of identity being sought by the RP/SP. The output is a list of IDPs that meet the criteria.

Discovery can be preceded by a registration step: a step by which IDPs register themselves as providing a particular identity service for a given user. Such a registry could be located on the client or on a network endpoint.

<table>
<thead>
<tr>
<th>Registration</th>
<th>Discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAML</strong></td>
<td>No, Defines an optional cookie mechanism by which RP/SPs can discover IDPs. Allows alternatives.</td>
</tr>
<tr>
<td><strong>Information Cards</strong></td>
<td>Manifested through the installation of managed cards into the Selector. Implicit. RP/SPs request identity attributes of CardSpace rather than querying for the location of attributes.</td>
</tr>
<tr>
<td><strong>OpenID</strong></td>
<td>No, Implicit. The RP/SP discovers the IDP by prompting the user for a URI or email.</td>
</tr>
<tr>
<td><strong>OAuth</strong></td>
<td>Explicit. Identity services pre-register for a consumer key and secret. Explic. The service being leveraged is directly labeled and accessed.</td>
</tr>
</tbody>
</table>
Converged Use Cases

The following use cases explore how different identity systems can work in concert.

OpenID with Information Card Authentication & SAML SSO

Alice uses isp.com for Internet access as it offers a host of additional services for her family. Rather than trying to remember multiple IDs and passwords, Alice has created an account at signon.com so that she can use the same credentials at any OpenID enabled Web site. Alice also likes to play online games which is one of the services she takes advantage of at isp.com. She is unaware that the games service is actually provided by games.com.

Alice has bookmarked a games Web site made available via her ISP at isp.com/games. As Alice has not yet authenticated to isp.com and does not have an active session she is redirected to the isp.com’s login page. isp.com supports OpenID in addition to local authentication. Alice’s OpenID provider is at signon.com so she enters her OpenID URL -alice.signon.com.

isp.com initiates an OpenID authentication with signon.com and then redirects Alice to signon.com. signon.com has implemented phishing resistant authentication and supports self-asserted information card authentication. Alice uses her Identity Selector to authenticate to signon.com.

After successfully authenticating Alice, signon.com use the OpenID protocol to redirect Alice back to isp.com

isp.com establishes a local session for Alice and then redirects her to games.com using the SAML 2.0 protocol. SAML is used as a result of a partnership between isp.com and games.com which requires a trust model that enables them to dynamically exchange user profile information including age range and billing information. games.com needs age information to prevent minors from playing R-rated games and billing information to ensure Alice is invoiced on her isp.com monthly statement for the games she plays.
OpenID Account Registration, Authentication and SSO with OAuth Service Authorization

Alice is a member of PrettyPuppy.com, a site dedicated to owners who love their dogs. She uses her Google Mail account to authenticate to PrettyPuppy, rather than having to maintain a separate username and password. While visiting PrettyPuppy, Alice sees a link for Scrapbooks.com advertising a ‘Custom Dog Scrapbook’ that imports her private PrettyPuppy dog pictures from her account and the accounts of her friends. Alice clicks on the link to Scrapbooks.com to join and is given a choice, she can either use her Google Mail account at Scrapbooks.com or create a new account. Alice chooses to use her Google Mail account (alice_m@gmail.com), the same account she uses for PrettyPuppy. She is then asked to consent to allow Scrapbooks.com to retrieve photos from PrettyPuppy to put into her scrapbook. When Alice returns to Scrapbooks.com to order a second album at a later date, she enters her Gmail address and password into a pop-up window from google.com and then gains access to see her Custom Dog Scrapbook. When she visits PrettyPuppy a few minutes later, she is able to access the site without any further authentication or authorization.

Scrapbooks.com has created a first-time user registration experience using using OpenID with the Simple Registration (SReg) OpenID extension. Scrapbooks.com acts as an OpenID SP and requests enough information to create Alice’s new account without prompting Alice directly for the information. As soon as the account is created, scrapbooks.com launches a three-legged OAuth request asking Alice to allow scrapbooks.com to consume private dog photos from prettypuppy.com. During the OAuth request, scrapbooks.com is acting as an OAuth consumer.

Google.com acts as an Identity Provider for both PrettyPuppy and Scrapbooks.com, providing Scrapbooks.com with a first name, a last name, a “directed identity” OpenID identifier and an email address during user registration, and supplying both Scrapbooks.com and PrettyPuppy with only a directed identity for all subsequent site access Requests.

PrettyPuppy.com is acting as an OAuth Provider, responding to Alice’s consent by delivering an access token to scrapbooks.com that can only be used for the photos related to Alice and her friends.

Note that there is an extension to the OpenID protocol called the “OpenID/OAuth Hybrid Extension” that can make this authentication and authorization story simpler in the case where the OpenID provider and the OAuth provider are the same entity.
SAML Authentication and With OpenID SSO

Alice owns a company called AlicesArtifacts.com. Alice doesn’t believe in storing passwords in the cloud, but wants to use cloud services for email and calendar. She has therefore set up a SAML Identity Provider at her office, and uses the premier version of Google Apps for your Domain (GAFYD) so that her employees can authenticate to their local Active Directory and then federate to retrieve their email and calendar from the Cloud.

Since Alice travels frequently, she has decided to use Tripit.com for managing itineraries. Tripit supports federation, but only via OpenID, not SAML. Luckily, GAFYD can bridge the gap between her Identity Provider and Tripit.

She visits Tripit and indicates she would like to create an account using Google Apps for your Domain. She is then taken to Google to authenticate, but because her domain is set up to federate, Alice instead authenticates to her home Identity Provider and federates back to GAFYD via SAML. Once Alice is authenticated to GAFYD, she is returned to Tripit via OpenID, where she obtains an account and is able to single sign-on into Tripit.

Tripit.com is acting as an OpenID SREG relying party, which is used both for account creation and for ongoing federated authentication and single sign-on.

Google Apps For your Domain acts as both a SAML relying party and an OpenID Provider, using SAML to authenticate users, but also acting authoritatively as a source for OpenID identity requests.

AliceM.com is acting as a SAML Identity Provider, collecting credentials locally and issuing a token to Google Apps for your domain.
Summary: Coexistence or Convergence

The protocols and technologies in this document create the promise of a secure, privacy enabling Internet-scale identity system composed of heterogeneous technologies operating together in a compatible and cohesive manner. Such coexistence implies determination of the areas in which current identity systems like SAML, OpenID, CardSpace, and OAuth are duplicative in functionality and scope — this is necessary to determine where and how these systems can be compatible.

This white paper demonstrates that these systems have unique characteristics and strengths, and suggests some representative scenarios in which these strengths complement rather than compete.

These identity systems will coexist and all offer sufficiently unique capabilities to allow them to flourish at a certain degree of independence. Notwithstanding the unique capabilities, there is a significant degree of duplication of functionality among the various systems. Convergence among the systems would eliminate such duplication and result in a simpler identity landscape.