In Pursuit of Liberty?

An exploration of the Liberty Alliance Project.

by Randy Mahrt

January 2003

Abstract

The Liberty Alliance Project is a consortium of industry leading business and technology companies that have banded together to create an open standard specification for securely sharing user identity information. In today's world most users have unique user ids and passwords for each of the services they interact with on the enterprise network or internet. The specification offers a solution to the problem by suggesting that users can choose to link their various accounts together facilitating single sign-on and global logout. There are currently a few vendors offering proprietary solutions in this space. The largest of these is Microsoft .NET Passport. There appears to be fierce competition between members of the consortium and Microsoft. The Liberty specification is relatively new and, it will take some time to see whether the industry will start building solutions based on the specification.

The Liberty Alliance Project

The Liberty Alliance Project offers an open technical specification for identity management on the Internet. Network identities are administered by the user and securely shared with the organizations of the user's choosing. The vision of the Alliance is "a networked world across which individuals and businesses can engage in virtually any transaction without compromising the privacy and security of vital identity information."[1] This paper explores the Liberty specification version 1.0 that was released on July 15, 2002. The specification employs a Federated Network Identity model to deliver single sign-on, global logout and identity federation. Over 60 member companies covering a broad range of industries currently sponsor the Liberty Alliance Project.

What's the problem?

A prevalent identity model used today requires an Internet user to maintain identity information at each site they interact with. This one-to-one relationship means that a user's identity information is fragmented and strewn about over the Internet. This fragmentation provides a weak model for privacy and security of network identity information. In addition, it makes maintaining userids/usernames complex.

Numerous Internet sites offer the ability for users to set personalization preferences. This service can greatly improve the effectiveness and efficiency of the site. Users often take advantage of this personalization by offering some personal information in the process. For example the site http://www.yahoo.com allows users to create a personalized site http://my.yahoo.com that a user can customize to display information like news, sports and stocks of personal interest. In order to take advantage of personalization the user must sign up. In order to sign up the user chooses a userid and password, sets a secret question, and provides personal information like birthday, name and zip code. The service also comes with an email account. Many sites offer similar personalization services.

In order to transact on the internet most eCommerce sites require users to create accounts before they can use the site to make purchases. The user provides personal *confidential* information to create the account. For example at the site http://www.amazon.com/ a user can create an “Amazon” account (userid and password) that allows address and credit card information to be stored so the user does not have to type in the information every time an order is placed.

Users who take advantage of these internet services have personal and confidential information like name, address, email, credit card, social security and driver’s license numbers stored on each of the sites that they interact with. This presents a significant security issue because there are no prevalent industry wide standards for storing and sharing of this information. This also presents a complex user experience because the user has to remember which user id and password to use at each site.

A typical user can have in excess of 15 different userids with accompanying passwords. Some tech-savvy internet users may have many more. In order to cope with the situation users may resort to using the same userid and password on as many sites as possible. Others may use the Post-It note system to keep track of all their disparate userids and passwords. Both methodologies make the user identity information less secure. A *Single Sign-on* scenario where the user signs-on once and has access to all of the sites and services he needs without the need to sign-on at each individual site would be preferable.

A Few Good Men

There are currently a few industry solutions that provide single sign-on capabilities two of which are Microsoft .NET Passport and AOL Time Warner’s Magic Carpet. Most use a central user repository and are proprietary in nature.

Microsoft .NET Passport offers a viable solution to the multiple userid/password situation. It is a proprietary system developed by Microsoft that provides single sign-on to all participating sites. Passport is required for users to log into Hotmail and MSN. It is also integrated into Windows XP. This means that millions of users utilize Passport every day. The .NET Passport solution has a few inroads into other platforms but primarily offers single sign-on functionality on the Microsoft platform.

The .NET Passport platform claims more than 200 million accounts worldwide. Avivah Litan from Gartner Inc. has research that "suggests that most people were automatically enrolled when using another Microsoft product, many of them unwittingly, and that there is far lower actual demand for the application."[2] In February 2002, "her research showed just 14 million U.S. users who knew they had signed up with Passport, and 84% did so only because it was required to use other Microsoft applications. Only 2% of the users were actually using Passport for the function it was designed – to manage multiple online identities."[3]

According to Mr. Asaravala from New.Architect there are some development costs involved in using Passport:

> While Passport membership is free to end users, participating businesses must pay an annual fee of $10,000, plus a vaguely defined compliance test fee of $1,500. According to Microsoft, the latter covers the cost of having an outside vendor verify a Passport implementation, and is usually–though not always–a one time fee.
>
> From the developers standpoint, a Passport subscription amounts to a license to use the Passport development libraries in a production environment. Subscribers need not use a Microsoft Web server or even a Microsoft operating system-the libraries are available for Solaris and Linux systems running the Apache and iPlanet Web servers, in addition to Windows and IIS. In order to activate the Passport, developers must go through their sites and add API calls to each page or resource that needs authentication.[4]

AOL Magic Carpet offers single sign-on using a screen name. According to Mr. Asaravala "sites that support screen names form a circle of trust allowing users to travel from one participating site to the next without having to log in more than once per session. As of this writing, however, the circle is limited to sites in AOL Time Warner's porfolio, and it's unclear when the final, public version of the system will be released."[5]

It's a Matter of Trust

The Liberty Alliance specification utilizes the concept of a *Federated Network Identity* as depicted in Figure 1. It is based on a circle of trust and consists of users and identity and service providers.

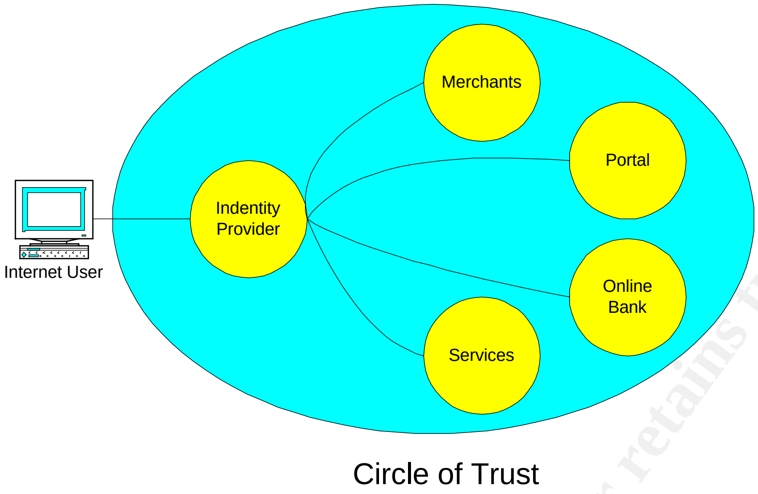

Figure 1 - Federated Network Identity

Users in the federated network can be members of an enterprise or individuals that have a need to interact with network resources. They form a relationship with an identity provider. Users are given the choice to opt into and out of these relationships and services that providers make available.

Identity providers are in the business of forming trust relationships with various service providers. The user chooses to be affiliated with the identity provider with the understanding that personal information could be shared with the trusted service providers.

Service providers are organizations that offer services to users. They form partnerships or trust relationships with identity providers. This is a very broad category that includes many of the companies currently utilizing the web today.

According to the documentation, "To become circle of trust members, providers are required to establish bilateral agreements on selecting certificate authorities, obtaining X.509 credentials, establishing and managing trusted public keys and managing life cycles of corresponding credentials."[6]

It's a Matter of Choice

The specification gives users the ability to choose who they associate with. The following simplified example illustrates the process that a user could go through to link or federate accounts (userid's) on two separate sites. The two sites have formed a trust relationship (Figure 2). The bank is the identity provider and the 401k site is the service provider. A user has a previous relationship with two internet companies.

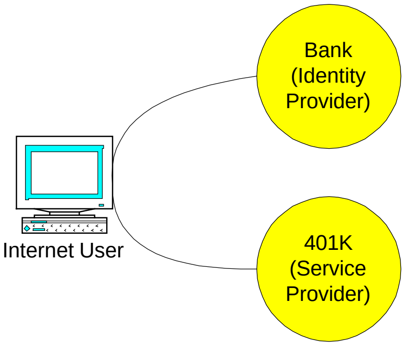

Figure 2 - User has a relationship with two internet companies

The user logs on to the bank, using his username and password, to check his account balance (Figure 3). The user is considered authenticated at this point. The bank offers the user the opportunity to associate his logon identity with the 401k site that provides the ability to manage retirement savings. This is called an *introduction*. The user has the option to accept or reject this offer. He accepts the offer. At this point the user has consented to linking account information but the accounts have not actually been linked.

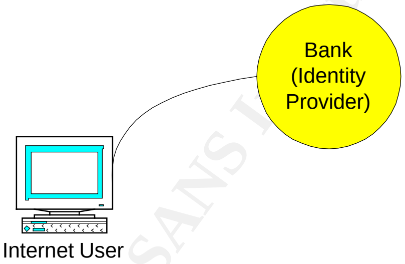

Figure 3 - User accesses bank site

While still being signed on to the bank site the user then accesses the 401k site upon which he would be presented with the option of federating his local identities from the two sites (Figure 4). The 401k site is able to present this option because the user consented to introduction. He decides to federate his identities. The user is then asked to log on to the 401k site at which point his identities would be federated between the two sites.

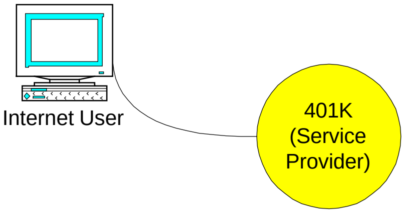

Figure 4 - User accesses the 401k site

To federate the two username accounts each site creates an *opaque handle* that uniquely describes the user. They each create entries in their user directories for each other and note each other's handle for the user. The opaque handle is not the username. The user may, in fact, have unique usernames for each site. Because of this there is no need for a globally unique user id.

After successfully completing this scenario the user would be able to experience single sign-on between the two sites (Figure 5) meaning that if he was signed on to one site and transverses to the other federated site he would not be asked to sign-on again.

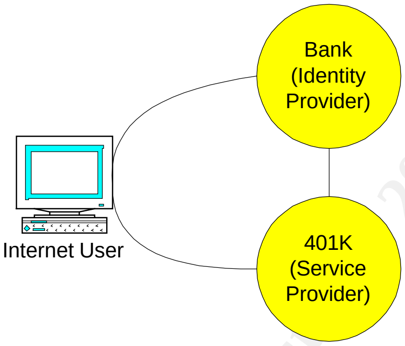

Figure 5 - Single Sign-on

Identity federation can become complex when there are multiple identity and service providers participating in the circle of trust. Each one-to-one link relies on an opaque handle to identify the user in each respective system. Many scenarios are discussed in detail in the Liberty documentation.[7]

The specification also provides for federation termination. The user has the ability to defederate his identities. Service and identity providers are both able to initiate the process. When the user initiates the defederation request at an identity provider the identity provider is states to the service providers that it has trust relationships with that it will no longer provide user identity information behalf of the user. When defederation is initiated at the service provider the service provider states to the identity provider that it will no longer accept identity information on behalf of the user.

Under the Covers

The liberty architecture is based on three main components: metadata and schemas, web redirection and web services.

The metadata and schemas are the data and formats exchanged between identity and service providers. In order to facilitate single sign-on, users must first authenticate themselves. Authentication is the act of determining that the user is who he says he is. When a user logs into the corporate network with a userid and password he is authenticating himself to the network. The specification does not prescribe the specific details for the process of establishing an identity. It does, however, provide an *Authentication Context* for identity providers to provide authentication assertion and additional information to service providers.

An authentication context can include information like identification methods, authentication mechanisms and credential details. There are many factors to an authentication context.

An *Authentication Context Statement* defines the specific factors used in an authentication. Similar context statements are grouped into *Context Classes* to simplify the task of assessing and comparing authentication assertions. The use of context classes gives a common framework for identity and service providers to communicate about authentication.

The specification uses XML schemas to define the authentication context. Liberty has supplied 10 authentication classes that support common and future authentication methods[8].

The protocols that Liberty defines are an extension of the *OASIS Security Assertions Markup Language* (SAML)[9]. SAML is a XML-based framework that enables the exchange of authentication and authorization information. Simple Object Access Protocol (SOAP) is used as the transport mechanism.

In order to accomplish single sign-on authentication assertions are communicated between the identity and service provider. A simplistic user scenario could go as follows: The user authenticates to the identity provider. The identity provider in turn provides an authentication assertion to the user. The assertion would be based on the authentication class used to authenticate. The user then presents the authentication assertion to a service provider to gain access to a particular service. The communication between the identity and service provider can be implemented by using web redirection or web services.

A typical single sign-on implementation utilizing a web redirection (query string parameters) implementation would follow a number of steps as depicted in Figure 6. In order for the http transactions to be considered secure Secure

Sockets Layer (SSL) 3.0 would need to be utilized (https). Information is either passed as part of the URI as query string parameters or in Form-Post.

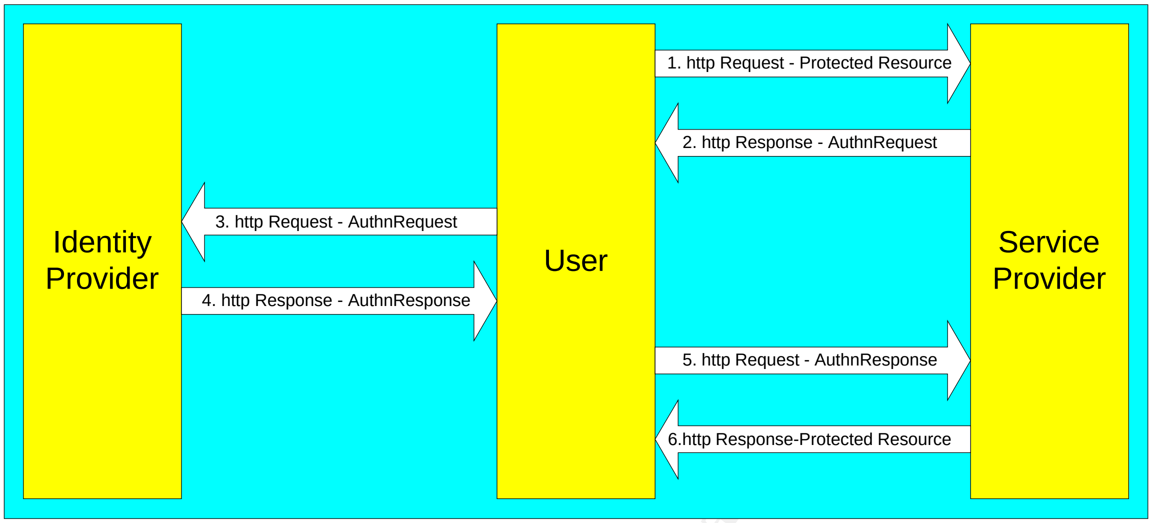
Identity Provider
User
Service Provider
1. http Request - Protected Resource
2. http Response - AuthnRequest
3. http Request - AuthnRequest
4. http Response - AuthnResponse
5. http Request - AuthnResponse
6.http Response-Protected Resource

Figure 6 - Single Sign-on (Login) Message Flow

Step 1 – The user requests a **protected** resource at the service provider. If the user is not authenticated the service provider determines the appropriate identity provider to redirect the user to. This information is stored in the user's profile. The user's opaque handle is a key to locate the user's identity provider. The original URI that the user was trying to access is passed as part of the redirect.

Step 2 – The user is redirected to the identity provider's single sign-on service with the AuthnRequest message as a query component.

Step 3 – The user accesses the identity provider's single sign-on service with the attached AuthnRequest information. The identity provider processes the AuthnRequest. The user goes through the process of being authenticated.

Step 4 – The identity provider responds with an AuthnResponse. This contains an authentication assertion.

Step 5 – The user requests the original URI from the service provider with the AuthnResponse. The service provider processes the assertion information that was passed.

Step 6 – The service provider responds with the original protected URI request.

Notice that all communication between the service provider and identity provider takes place via the user's browser. Information is communicated via query string parameters.

An alternative implementation could use web services (SAML & SOAP). The identity provider would only pass an authentication artifact back to the user in step 4. The user would present the *artifact* to the service provider in step 5. The service provider would then communicate directly with the identity provider to obtain the full authentication assertion by passing the assertion artifact. A SAML request within a SOAP message would be used to request the full assertion. This implementation would be more secure but would require a communication channel between the identity and service providers.

There are a number of documented profiles[10], in addition to what has been discussed here, that can be used to implement single sign-on. These include a browser artifact, browser POST, WML POST and enabled client proxy.

The user is also able to logout of all sites that he has initiated secure sessions with. Logout can be initiated either at the identity or service provider. In either case, the identity provider is responsible for communicating logout to all the sites that it provided authentication assertions to on behalf of the user.

A web based redirection single sign-off (logout) would follow the steps outlined in Figure 7.

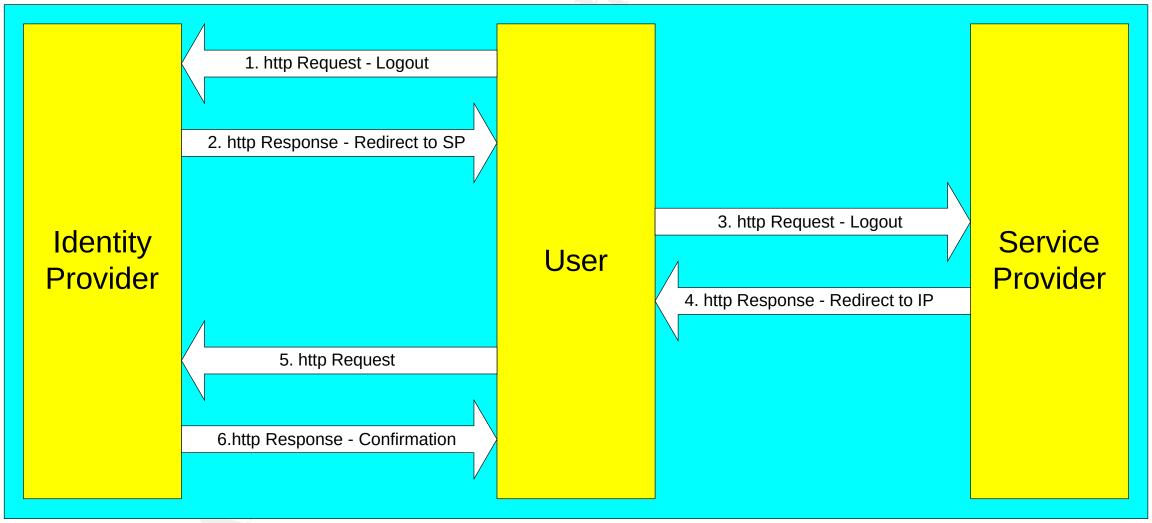

Figure 7 - Single Sign-off (Logout) Message Flow

Step 1 – The user requests a logout from the identity provider. All service providers that were given an authentication assertion must be notified of the logout request. Steps 2–5 would be iterated for each service provider logout.

Step 2 – The identity provider responds with a redirect to the service provider logout.

Step 3 – The logout redirect is sent to the service provider. The service provider logs the user out.

Step 4 – The service provider responds with a response that redirects the user back to the identity provider. No success of failure message is conveyed. The sole purpose of the redirect is to send the user back to the identity provider.

Step 5 – The user is redirected back to the identity provider so a confirmation message can be sent to the user's browser.

Step 6 – The identity provider responds with a logoff confirmation response stating that full logout was successful.

Single sign-off can also be implemented using SOAP messages to communicate logout between the identity and service providers.[11]

It's a Matter of Time

There is an industry need for users to have the ability to administer and securely share personal and confidential information with the systems and organizations they interact with. The v1.0 specification is a good start to offering identity federation, single sign-on and single sign-off. The standards group has been working on a v1.1 that includes some minor enhancements and fixes to some known security issues. It is due to be released on January 15, 2003. In addition v2.0 is slated for later this year.

The next version of specification will expand the ability to include features for permission-based attribute sharing. This will enable the more complete sharing of personal identity information according to users preferences.

In order for the specification to make a significant impact on the industry it needs to make its way out into the real world. Software vendors need to build products that allow businesses to utilize the framework and businesses need to use the products to deploy real systems.

The specification seems to be gaining some industry acceptance. According to Mr. Wong, in September, "Sun Microsystems ... unveiled a new open-source software development tool designed to help businesses start testing and building online identification systems using the new Liberty Alliance standard."[12] Mr. Wong says "about half a dozen companies – including Sun, Novell, RSA Security and Entrust – have announced they are planning to support Liberty in their software products."

Sun now offers identity server software called "Sun ONE Identity Server 6.0" that is based on the standard. The product supports version 1.0 out-of-the-box. It also provides a comprehensive identity management system, which streamlines access management by simplifying the creation and administration of identities as well as the management and enforcement of authentication. In addition, the Sun ONE Identity Server 6.0 leverages industry standards such as SAML and SOAP.

Of course Microsoft has a significant number of users using the .NET Passport system and it is still unclear whether it will work with the Liberty specification. According to Mr. Fisher, "In a surprising move, Microsoft Corp. on Thursday announced that it will open up a portion of the source code of its Passport identity service on a limited basis."[13]

Some in the industry are even saying that the consortium is divided in their thinking. Mr. Galli and Fisher say, "A growing rift among members of the Liberty Alliance authentication project is placing the technology's future in question."[14]

The problem is centered on single sign-on and the fact that .NET Passport has an established base of customers using the system. Some of the Liberty members have conceded defeat to Microsoft on the Windows platform.

In fact, Jonathan Schwartz, executive vice president of Sun's software group said, "There is no way we can compete with them there. They have that market tied down really tight." This faction of members is waiting for some pervasive computing device not based on Microsoft platform to propel the Liberty specification into the lime light. Schwartz said, "I don't think it will be very long before we have a pervasive non-microsoft client. Have you seen the latest cell phones with color screens and keyboards and cameras? That's the way it will go."[15]

Another alliance faction does not subscribe to the same thinking. "We don't have to concede anything to Microsoft," said Justin Taylor, chief strategist for directory services at Novell Inc., of Provo, Utah. "Liberty is much more attuned to enterprise users today than Passport is. Microsoft is trying to move into the enterprise, but we feel that we're strong in that area."[16] Others members also share the feeling that Liberty is poised to successfully deliver value to the enterprise.

These differing opinions may signal a divide in the thinking of the consortium leadership. When there is division among the members there is more room for failure. A city divided cannot stand.

Many Liberty member companies have pledged their support for the specification saying that they will develop systems in 2003 that will utilize the specification. According to the Liberty site General Motors has this to say about the specification:

> GM is working internally to prepare its systems for Liberty-compliant technology. By preparing its systems now, GM hopes it will able to take advantage of Liberty-enabled products as soon as they are available, making GM's customers, employees, and vendors among the first to reap the benefits of federated identity. As federated identity becomes the industry standard, GM will find itself one step ahead of the game in terms of offering a simple, secure and better way for its customers, vendors and employees to access GM's products and services online.[17]

It is hard to predict the fate of the Liberty specification. Time will tell all.

References

Asaravala, Amit. “A Question of Identity.” New.Architect. January 2003. http://www.newarchitectmag.com/documents/s=7766/na0103b/index.html.

Beatty, John. “Liberty Protocols and Schemas Specification.” July 2002. http://www.projectliberty.org/specs/liberty-architecture-protocols-schemas-v1.0.pdf.

Box, Don et al. “Simple Object Access Protocol (SOAP) 1.1” World Wide Web Consortium Note, May 2000. http://www.w3.org/TR/SOAP.

Fisher, Dennis. “Liberty Alliance Spec Won’t Cure Security Mess.” eWeek. July 2002. http://www.eweek.com/article2/0,3959,373876,00.asp,

Fisher, Dennis. “Microsoft Opens Passport Source to Developers.” eWeek, October 2002. http://www.eweek.com/article2/0,3959,625314,00.asp

Fisher, Dennis & Galli, Peter. “Liberty Alliance Waves White Flag at Passport.” eWeek, December 2002. http://www.eweek.com/article2/0,3959,740753,00.asp.

General Motors. http://www.projectliberty.org/newsletter/public/#case.

Hallam-Baker, Philip. et al., “Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML).” OASIS. April 2002. http://www.oasis-open.org/committees/security/docs/draft-sstc-core-31.pdf,

Hodges, Jeff. “Liberty Architecture Overview.” July 2002. http://www.projectliberty.org/specs/liberty-architecture-overview-v1.0.pdf.

“Interopability Prototype for Liberty.” Sun Microsystems, Inc. http://developer.java.sun.com/developer/codesamples/liberty.html.

Kannappan, Lena & Lachance, Matthieu. “Liberty Architecture Implementation Guidelines.” July 2002. http://www.projectliberty.org/specs/liberty-architecture-impl-guidelines-v1.0.pdf.

“Liberty Alliance Ringing In Single Sign-On.” July 2002. http://www.techweb.com/tech/security/20020717_security

Madsen, Paul. “Liberty Authentication Context Specification.” July 2002. http://www.projectliberty.org/specs/liberty-architecture-authentication-context-v1.0.pdf.

Mauldin, Hank. “Liberty Glossary.” July 2002. http://www.projectliberty.org/specs/liberty-tech-glossary-v1.0.pdf.

Mishra, Prateek. “Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML).” OASIS. January 2002. http://www.oasis-open.org/committees/security/docs/draft-sstc-bindings-model-11.pdf.

Platt, Darren. et al., “SAML Requirements and Use Cases,” OASIS. December 2001. http://www.oasis-open.org/committees/security/docs/draft-sstc-saml-reqs-01.pdf.

Rouault, Jason. “Liberty Bindings and Profiles Specification.” July 2002. http://www.projectliberty.org/specs/liberty-architecture-bindings-and-profiles-v1.0.pdf.

“Sun One Identity Server.” Sun Microsystems, Inc. http://wwws.sun.com/software/products/identity_srvr/home_identity.html.

Taschek, John. “Liberty Alliance or Passport?” eWeek. June 2002. http://www.eweek.com/article2/0,3959,266840,00.asp.

Taschek, John. “Microsoft Opens Up Passport Code,” eWeek. October 2002. http://www.eweek.com/article2/0,3959,643397,00.asp.

“The Liberty Alliance Project release version 1.1 Specifications.” November 2002. http://www.webservices.org/index.php/article/articleprint/778/-1/5.

Wade, Will. “Liberty Consortium Says ID Standard Will Stick.” American Banker. January 2003. http://www.americanbanker.com.

“Wireless Application Protocol Wireless Markup Language Specification Version 1.3” Wireless Application Protocol Forum, Ltd. February 2000. http://www.wapforum.org/.

Wong, Wylie. “Sun unlocks Liberty Alliance tool.” ZDNet News. September 2002. http://zdnet.com.com/2100-1104-958526.html.

[1] Hodges, p.5.
[2] Wade, p.1.
[3] Wade, p.1.
[4] Asaravala, p.2.
[5] Asaravala, p.1.
[6] Hodges, p.17.
[7] Hodges, pp.22-26.
[8] Madsen, pp.11-25.
[9] P. Mishra, p1.

[10] Rouault, pp.12-27.
[11] Rouault, pp.34-45
[12] Wong, p1.
[13] Fisher, p.1.
[14] Galli & Fisher, p.1.
[15] Galli & Fisher, p.1.
[16] Galli & Fisher, p.1.
[17] Liberty Newsletter, p.1.