A standards-based approach to Federated Identity

Mike Lopez

mike.lopez@manukau.ac.nz

Dr. Samuel Mann

smann@tekotago.ac.nz

John Peppiatt

john.peppiatt@manukau.ac.nz

Andrew Sewell

asewell@tekotago.ac.nz

Christopher Stott

chris@manukau.ac.nz

Abstract

Federated Identity allows users to access multiple services at different organisations with the same credentials. In this paper, we summarise key work currently being carried out on Federated Identity. We evaluate several existing and suggested schemes and propose a new standards-based platform-neutral design pattern that uses current mature technologies and is suitable for the implementation of federated identity in a business-to-business context. The design pattern is verified with a practical implementation at two polytechnics.

Keywords: Distributed systems, federated identity, federated security, identity management, service-oriented architecture.

1 Introduction

In this paper, we use the term Federated Identity to refer to an approach to authentication that allows users to access multiple services at different organisations with the same credentials. The term is sometimes also used to refer to the aggregation of a person's user information stored across multiple distinct identity management systems. This as a non-goal in our design and we aim specifically to make it possible for a business to defeat any such aggregation attempts.

This paper follows the design science research process recommended by (Peffers *et al.*, 2006). We begin by defining the problem, identifying work being carried out by other researchers and setting out those standards that let us achieve a standards-based solution. We then derive the objectives, describe the design and development, and conclude with the demonstration and evaluation of the artefacts.

This quality assured paper appeared at the *19th Annual Conference of the National Advisory Committee on Computing Qualifications* (NACCQ 2006), Wellington, New Zealand. Samuel Mann and Noel Bridgeman (Eds).

1.1 Problem Identification

In recent years, business to business interactions over the Internet have become more common. Current technology allows businesses to offer services to their trading partners through a combination of Web sites and "rich client" software that uses Web services.

In order to offer these services securely, each service needs to know something about the identity of the party consuming the service so it can adjust its functionality and content accordingly. Authentication of identity typically involves the consumer making some claim of identity and supplying evidence to support that claim. Usually the claim is a login and the evidence a password. Other forms of authentication are possible, but the login/password approach remains the dominant paradigm.

At present, a business offering services to its partners typically creates separate logins and passwords for each user of the service. As a result, users that work with multiple services often have to contend with multiple logins, one for each service they access. To make things worse, many of these services also force users to change their passwords frequently.

The result of this is that the number of passwords a user has to remember rapidly exceeds the capacity of human memory. This can lead users to adopt coping strategies that present a serious threat to security; users may pick easy to remember passwords, use the same password on multiple systems or write down a list of their passwords, sometimes even attaching the list to the computer screen;

This "password overload" has been noted in academic sources (Clear, 2002) and widely reported in general literature; for example, a survey (Jedras, 2005) of 1700 corporate users carried out for RSA Security reported that 15% keep a paper list by their computers and that 15% use the same password for everything they do.

There is a significant administrative cost in maintaining these multiple identities; users join and leave businesses and help desks have to deal with forgotten passwords. Attempts to automate the management of forgotten

passwords by providing hints can seriously compromise password integrity.

Rather than blaming the users for their coping strategies, Federated Identity aims to provide a solution to this password overload by enabling a user to use the same credentials safely and securely for multiple purposes and services. With this vision, a user should be able to log on once, with their normal business credentials, authenticate themselves with a password, (or possibly smartcard or biometric data) and then consume a wide range of services without further authentication.

There remains a problem though; an unscrupulous business partner with access to these credentials could discover sensitive information or compromise the business. Solutions to issues of trust and trust management have to be found before a practical implementation of Federated Identity can be achieved.

2 Current work

Many researchers are working on solutions to Federated Identity. We briefly summarise below some of the key projects.

2.1 Liberty project

The Liberty project (Wason, Cantor, Hodges, Kemp, & Thompson, 2005) is a collaborative project from a consortium of over fifty organisations that is led by Sun Microsystems Inc.

The central idea in this approach is that a number of organisations offering services over the Web may combine in a "Circle of trust". Users of these services may then choose to combine their accounts so that a single login can be used within the circle. Service providers share user information and the consequent privacy issues are addressed by allowing each user to explicitly opt-in to the scheme.

This approach is well suited to both consumer to business and consumer to government transactions. It is less well suited though for business to business transactions. For the latter, it is usually the organisation itself that chooses its business partners and services rather than individual users. Moreover, few businesses are willing to share details of their user base with their business partners.

Further work on the project is required before it is ready for generic business to business use.

2.2 Shibboleth

Shibboleth (Scavo & Cantor, 2005) is a product of the US Internet2 initiative and is targeted particularly at the education and government sectors. In the UK, the Joint Information Systems Committee (JISC) is evaluating the possibility of adopting this system

Shibboleth does not define any authentication process, leaving this to individual organisations but focuses on protocols for passing identity information between organisations. These protocols are based on SAML (Security Assertion Markup Language), an international standard (A. Anderson & Lockhart, 2005) developed by the OASIS Security Services Technical Committee.

A key element of Shibboleth, the Browser/Artifact protocol, allows an identity provider to vary its responses from service to service. However, Shibboleth, like the Liberty project, takes the view that interactions are between a user and an organisation; there is no recognition that a user may be acting on behalf of an organisation.

2.3 Microsoft Passport

Microsoft's Passport Network is a scheme in which users register with a third party (Microsoft) to set up an account identity. Under a co-branding agreement, organisations can then trust Microsoft to carry out the authentication of users on their behalf, with Microsoft passing back a user account identity.

The scheme is widely used by Microsoft and partner sites such as hotmail, but has had limited uptake by other parties. From a business-to-business perspective there are two key weaknesses:

- Anyone can sign up for a Microsoft passport, not just users of an organisation's business partners
- There have been several known security issues. For example, a vulnerability in the password change service (Lemos, 2003).

2.4 Web Services standards

The WS-SECURITY standard (Lawrence, Kaler, & al., 2006) defines an approach to secure messaging that provides end-to-end security, rather than the point-to-point security provided by transport layer implementations.

The WS-TRUST draft proposed standard (S. Anderson et al., 2005) builds on the secure messaging defined in WS-SECURITY to define mechanisms for issuing and exchanging security tokens and establishing and accessing trust relationships.

The WS-FEDERATION draft proposed standard (Bajaj, Della-Libera, Dixon, Dusche, Hondo and Hur, (2003) describes mechanisms that allow different security realms to federate by allowing and brokering trust of identities, attributes and authentication between participating Web services.

Together, these standards describe a robust framework for implementing federated identity. They fall short of a generic solution in two ways:

- They apply only to service-based applications and are not supported by current browsers.
- Specific implementations are described in profiles and both service provider and consumer must support the profile. There is a tension between the need for a generic solution and the desire to specify in detail the exact mechanisms used. For example, the Kerberos token profile (Kaler, Hallam-Baker, Lawrence, Monzillo, &

Nadalin, 2006) describes how Kerberos tokens are to be used; however, both the service consumer and the provider must support this profile.

2.5 Summary

Each of the schemes discussed above addresses part of the federated identity problem, but none of them offers a generic solution in a business-to-business context.

Although work is still proceeding to improve the level of inter-operation, many of these schemes require potential business partners to agree to implement a common set of technology and platforms.

There is little recognition of the difference between users acting on their own behalf and acting on behalf of an organisation.

3 Current standards and practice

A key goal of our project was to evaluate the extent to which a practical implementation of Federated Identity could be achieved by using existing mature standards and practices in a technology-neutral manner. The relevant standards are briefly described below.

3.1 Challenge/response

The HTTP 1.1 standard RFC2616 (Fielding et al., 1999) describes a generic extensible framework for a challenge-response authentication protocol. Briefly: when an attempt is made to access a protected resource, the host challenges the browser and sets out the authentication schemes it will accept. The browser then gathers user credentials and resends the request with these credentials. The protocol also defines a "realm" which allows the protection space of the host to be partitioned. The browser uses the combination of the host and realm to cache credentials so that it may respond silently to future challenges for the protection space.

Within this framework, the standard defines two explicit authentication schemes: "Basic" and "Digest". These are described in RFC2617 (Franks et al., 1999).

With "Basic" authentication, the username and password are sent in trivially encoded plain text. This presents a significant security risk unless a secure transport layer is used; Transport Layer Security (TLS 1.0) or Secure Sockets Layer (SSL 3.0), as described in RFC2246 (Dierks & Allen, 1999) and RFC3546 (Blake-Wilson, Nystrom, Hopwood, Mikkelsen, & Wright, 2003) provides the necessary security. Basic authentication is widely supported across browsers and platforms and is generally considered secure when used with these protocols.

With "Digest" authentication, a cryptographic hash or "digest" is sent in place of the password. This provides the necessary protection for passwords, but user identifiers are still sent "in the clear". Because of this, a secure transport layer is less necessary but still recommended. Digest authentication is however less widely supported by browsers and platforms. It offers a good level of protection, but has the weakness that the host must know the password. This can compromise the integrity of the underlying operating system authentication; most systems prefer not to store passwords so that system integrity is not lost if a "password file" is compromised.

The http challenge response protocol is designed to be extensible and several proprietary extensions exist. For example, Microsoft has implemented an "NTLM" scheme to enable integrated windows authentication. This is a secure option, but limits the choice of platforms and browsers to those offered by Microsoft. Other common implementations include Kerberos (Kohl & Neuman, 1993) and Negotiate (Surati & Muckin, 2002).

3.2 Forms authentication

"Forms authentication" is a technique that is widely used on many platforms by web sites for authentication. In this scheme, a "cookie" (Kristol & Montulli, 2000) or Uniform Resource Identifier (URI) query string parameter (Berners-Lee, Fielding, & Masinter, 1998) is used to represent an authenticated user session. Requests that arrive without such an identifier are redirected to a "Login" page that gathers user credentials, authenticates the user, allocates a session identifier, and redirects back to the referring URL[1] with the identifier attached.

For this scheme to work, the login page needs to be able to verify a user's credentials and must therefore have access either to a data store of these, or to a service that can carry out the authentication. Conceptually, the login page can be hosted in three possible places:

- By the organisation hosting the site
- By the organisation using the site
- At a third party.

One implementation of this third option is Microsoft's Passport scheme in which both the login and the user data store are hosted by the third party (Microsoft). Most organisations host the login page at the same site as the service. Both of these approaches suffer from the weakness that few businesses want to expose information about their user base to third parties.

As with HTTP Basic authentication, the forms model requires a secure transport layer.

3.3 Portal architecture

With a "portal" approach, rather than contact individual sites directly, the user's first contact is with a central site that manages access to a range of sites and services. The user logs in to this portal, establishes a session, and can then navigate through the portal consuming services.

In our vision, these services might be implemented locally on the portal's system, or might be implemented

[1] The term "Uniform Resource Locator" (URL) refers to the subset of URIs that, in addition to identifying a resource, provides a means of locating the resource by describing its primary access.

on some other system, perhaps on the other side of the world and using different technologies from the portal.

In this case, there must be some mechanism for the portal to communicate user identity to the remote host, perhaps by attaching additional information to the request. The request then arrives with this information at the remote site which uses it to identify the user and permit access without further interaction.

This model requires that the site hosting the service trusts the portal to carry out the necessary identification and authentication.

3.4 Motivation and value

A practical solution to the Federated Identity problem would remove a major barrier to the widespread adoption of business-to-business interactions over the Internet.

A number of competing solutions are emerging, but solutions which require a commitment to a particular platform or technology are unlikely to succeed unless appropriate technology bridges can be created.

An approach based on mature standards would remove this barrier to adoption.

4 Objectives and approach

Our key design goals were that:

- Each organisation would be free to choose whatever authentication scheme and technology it wished and could change these at any time without affecting its partners.
- A user would have a single sign on (SSO) and could then access multiple services at different sites without further active participation in the authentication process.
- Credentials would always be kept within an organisation's boundary and the use of the scheme would not expose the organisation to any significant security risk.
- Each organisation would have complete control over the choice of partners with which it would work and over the information it would supply to each of these. It could actively defeat any attempt by services to aggregate data about its users or their usage patterns[2].
- A common pattern would be used for both web sites and web services. It would work well with all mainstream platforms and technologies and be simple to implement.

We set up our demonstration to use Forms and Portal models with Basic, Digest, and Windows authentication schemes. We used a secure transport layer for all services. Our demonstration was limited to browser access to Web sites because it is easier to emulate browser behaviour in a rich client than to replace the existing browser base.

5 Design and Development

The Challenge/Response and Forms authentication protocols implicitly assume that the host[3] organisation site has access to a user's credentials (or equivalent). We achieve this by placing the authentication service within the partner organisation's security context and redirecting unauthenticated requests to that service.

The key problem that remains to be solved is how to communicate the user identity safely and securely between the partner's security context and the host service.

We achieve this with a "visa" which is essentially a service-oriented abstraction of an equivalent to a Kerberos ticket.

In order to achieve the goal of keeping user credentials within an organisational boundary, we implemented a two-stage authentication protocol (Figure 1).

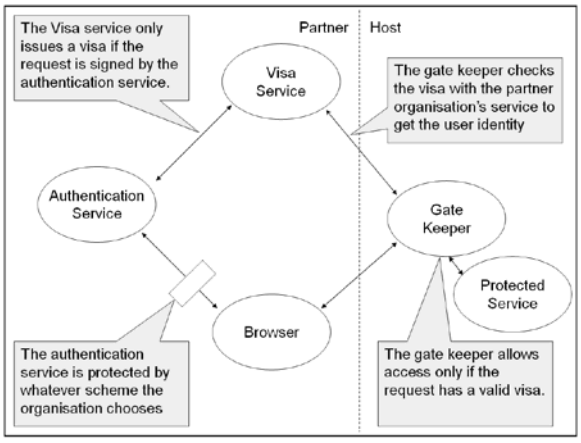

Figure 1

In the first stage, a user authenticates with an authentication service within the partner organisation's boundary. The service response includes a "visa" (an opaque token) that is cryptographically signed by the issuer. The visa is cached for a moderate session time (20 minutes).

In the second stage, a host organisation presents this visa as part of a cryptographically signed request for user identification. To retrieve the user information, the host organisation must be on the trusted services list of the partner organisation, the signature must be valid and the session unexpired.

A two-stage authentication process is widely used to cross security context boundaries. The key point of difference is that we take a service-oriented approach by asking the partner organisation for user details, rather

[2] The Privacy Act, 1993 prevents an organisation from using the same identifier to identify a user to different partner organisations (Principle 12).

[3] We use the term "host" organisation to refer to the organisation hosting a service and "partner" organisation to an organisation whose users access the service

than a data-centric approach in which the data is encrypted in a ticket.

This adds considerable flexibility: a partner organisation may vary the user information it releases based on the identity of the host service requesting the information.

5.1 Authentication service

The authentication service consists of a "logon" page that is placed under the built-in authentication control of the partner organisation. This is all that is required to ensure that the scheme automatically adjusts to whatever authentication scheme the partner organisation uses. The page issues a visa and redirects the browser back to the host service.

To emulate a portal architecture, we added a "portal" page that simply presented the user with a list of trusted services and appropriate links. It redirected the user to the login page to acquire the necessary visa.

5.2 Visa service

The Visa service is the mechanism by which a host organisation discovers the identity of an authenticated user. The service allows anonymous access and relies on cryptographic techniques for security.

The service issues a visa only if the request originates from the configured authentication service of the partner organisation and responds to requests for user identity only from services that are configured as trusted.

In both cases, a requester must demonstrate knowledge of the private key associated with its configured public key.

5.3 Gate Keeper

The gate keeper at the host organisation was implemented as a pipeline module. We chose this approach because it makes it easy to place an entire machine under access control without code changes to individual sites.

The gatekeeper inspects all incoming requests, managing the authentication process and allowing authenticated requests to pass through to the protected services.

The gatekeeper works with a configured list of trusted partner organisations, together with appropriate public keys and authentication URLs.

We used a combination of URL parameters and cookies to convey standard information. Generally URL parameters were used on the initial request. These parameters were then stored as cookies for subsequent requests. We defined the following parameters:

Organisation: The name of the partner organisation. This is required for all schemes and must be on the list of trusted partner organisations.

Visa: An opaque authentication token issued by a partner organisation's Visa issuing service. The gate-keeper uses this token to request user identity from the partner.

In the absence of a Visa, the gate-keeper redirects the caller to the partner organisation's authentication service.

5.4 Single Sign On

When a browser authenticates with the partner organisation's authentication service, it will cache the user's credentials against the URL of the authentication service. These credentials are not transmitted to the various host services as the user navigates among them.

The first request to each host service in a browser session will be redirected to the authentication service, which will authenticate silently using the credentials cached in the browser and issue a visa.

6 Demonstration

The proof of concept code was written with Microsoft's Visual Studio.Net, but equivalents could be implemented easily in any programming language and environment.

We implemented a simple test page and placed it under the control of the gatekeeper. We set up several partner authentication services at two polytechnics. The demonstration included authentication against Windows servers, Novell directory services, and LDAP servers.

We used Internet Explorer 6.0 and Mozilla Firefox 1.0.7 browsers for our testing.

7 Evaluation

The design pattern does not require an organisation to adopt any particular authentication scheme. It achieves this flexibility by placing the authentication service under the control of whatever (http compliant) scheme the organisation chooses.

In consequence, an organisation may change its chosen scheme without coding changes by reconfiguring the protection of the authentication service. Such changes are accommodated "on the fly" by partner organisations.

A user's credentials are kept in browser memory and transmitted over a secure connection to the organisation's own authentication service; they are never sent to the organisation's partners or to any third parties.

Standard browser behaviour will associate the credentials with the organisation's authentication service and will silently re-supply them when challenged, thus providing a single sign on.

An organisation's visa service has complete control over what user information (if any) is released to the various host services. It can release different information to each, including, for example, pseudonyms rather than real user identities. This allows an organisation to actively defeat any attempt by the hosts the aggregate user data or usage patterns.

7.1 Ease of implementation

Davis (1986, 1989)identified perceived usefulness (PU) and perceived ease of use (PEU) as critical factors in technology acceptance. From the PU perspective, the key

point of difference between our approach and those referred to herein is the recognition *a priori* of the critical difference between consumer to business (or government), and business-to-business schemes: a business takes decisions on behalf of its users and thus needs appropriate tools and architecture to comply with privacy legislation and protect its business interests.

ICT professionals have made a considerable investment of time in the languages, environments, technologies and toolkits they know. To optimise PEU, ICT professionals should be able to implement the solution within this trusted framework and the implementation itself should be straightforward.

In a Microsoft.Net environment, no code changes are required to place web sites under the control of the gatekeeper module; this can be achieved with standard configuration files. Web sites that need to know the user identity can access this through the standard IIdentity interface. Our test page required just one line of code to do this.

Similarly, the authorisation and visa services can be implemented in a Microsoft.Net environment simply by installing the software and placing it under whatever access control mechanism the partner organisation uses.

The gatekeeper module required 5 classes and used 770 lines of source code and the partner organisation services 7 classes and 638 lines of code. It would be relatively trivial to write equivalents for other environments, such as Java.

7.2 Security

The use of a cryptographically secure "visa" service effectively extends the chosen scheme to allow it to be trusted by partner organisations.

As a protection against replay attacks, we implemented the visa as a one-off token, tying it to the IP address of the requester; this allowed us to safely store the token in the cookie collection. User identification never leaves the server, unless exposed by the web site itself.

When used in conjunction with a secure transport layer, we would consider this scheme to be highly secure. To justify this claim we set out the following analysis:

Gatekeeper

The gatekeeper permits access to protected sites and services only when the request has a visa and where the visa service of the trusted authentication service issues a positive signed response.

Visa service

The visa service will only issue a visa on receipt of a signed request from its configured authentication service. It will issue user data only when it has a signed request from a configured trusted host that specifies a visa allocated for that service. It will issue this response only once (to defeat replay attacks).

The signing of all requests and responses demonstrates knowledge of the private keys corresponding with the configured public keys.

7.3 Recommendations

In general, we would recommend using an integrated scheme (such as NTLM) when the browser is being used from within the organisation's boundary, and a scheme such as Basic or Digest when accessed from the Extranet.

Although Forms authentication schemes are easily supported and have good browser support, we would not recommend their use because of their vulnerability to security risks such as "phishing".

The recommended pattern is designed to be complementary to approaches such as Project Liberty or Shibboleth, rather than an alternative. An organisation may work within a "Circle of trust" with some key partner organisations while simultaneously inter-working with other partner organisations on a basis of more limited trust.

We believe we have shown that it is possible to create a robust authentication scheme for Federated Identity that is based on existing mature standards.

8 Communication

Communication is being done through this paper. Researchers who would like access to the software or would like to participate in the further development or evaluation of this approach should contact the lead author.

We believe that we have shown that it is possible to create a secure implementation of Federated Identity using existing mature standards in a platform-neutral and language-agnostic manner.

9 References

Anderson, A., & Lockhart, H. (2005). Saml 2.0 profile of xacml v2.0. *2006*. Retrieved March 6th 2006, from http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf

Anderson, S., Bohren, J., Boubez, T., Chanliau, M., Della-Libera, G., Dixon, B., et al. (2005). *Web services trust language (ws-trust)*. Retrieved March 6th, 2006, from ftp://www6.software.ibm.com/software/developer/library/ws-trust.pdf

Bajaj, S., Della-Libera, G., Dixon, B., Dusche, M., Hondo, M., Hur, M., (2003). *Web services federation language (ws-federation)*. Retrieved March 6th, 2006, from ftp://www6.software.ibm.com/software/developer/library/ws-fed.pdf

Berners-Lee, T., Fielding, R., & Masinter, L. (1998). *Uniform resource identifiers (uri): Generic syntax*. Retrieved March 6th, 2006, from http://www.ietf.org/rfc/rfc2396.txt

Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., & Wright, T. (2003). *Transport layer security (tls) extensions*. Retrieved March 6th, 2006, from http://www.ietf.org/rfc/rfc3546.txt

Clear, T. (2002). Design and usability in security systems - daily life as a context of use? *SIGCSE Bulletin, 34*, 13-14.

Davis, F. (1986). *A technology acceptance model for empirically testing new end-user information systems: Theory and results.* MIT Sloan School of Management, Cambridge, MA.

Davis, F. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly 13*(3), pp. 319-339. .

Dierks, T., & Allen, C. (1999). *The tls protocol version 1.0*. Retrieved March 6th, 2006, from http://www.ietf.org/rfc/rfc2246.txt

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., (1999). *Hypertext transfer protocol -- http/1.1*. Retrieved March 6th, 2006, from http://www.ietf.org/rfc/rfc2616.txt

Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., et al. (1999). *Http authentication: Basic and digest access authentication*. Retrieved March 6th, 2006, from http://www.ietf.org/rfc/rfc2617.txt

Jedras, J. (2005). Users suffering password overload. Retrieved 24th April 2006, from http://www.rsasecurity.com/company/news/in-the-news/articles/Comp_Canada.pdf

Kaler, C., Hallam-Baker, P., Lawrence, K., Monzillo, R., & Nadalin, A. (2006). Web services security kerberos token profile 1.1. *2006*. Retrieved March 6th 2006, from http://www.oasis-open.org/committees/download.php/16788/wss-v1.1-spec-os-KerberosTokenProfile.pdf

Kohl, J., & Neuman, C. (1993). *The kerberos network authentication service (v5)*. Retrieved March 6th, 2006, from http://ietf.org/rfc/rfc1510.txt

Kristol, D., & Montulli, L. (2000). *Http state management mechanism*. Retrieved March 6th, 2006, from http://www.ietf.org/rfc/rfc2965.txt

Lawrence, K., Kaler, C., (2006). Web services security: Soap message security 1.1 (ws-security 2004). Retrieved 6th March,2006, from http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

Lemos, R. (2003). Passport problems could cost microsoft. Retrieved March 6th 2006, from http://news.zdnet.com/2100-1009_22-1000655.html?tag=nl

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., (2006). The design science research process: A model for producing and presenting information systems research. *First International Conference on Design Science Research in Information Systems and Technology* Retrieved April 25th, 2006, from http://ncl.cgu.edu/designconference/DESRIST%202006%20Proceedings/4A_2.pdf

Scavo, T., & Cantor, S. (2005). Shibboleth architecture technical overview. Retrieved March 6th 2006, from http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf

Surati, S., & Muckin, M. (2002). *Http-based cross-platform authentication via the negotiate protocol*. Retrieved March 6th, 2006, from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsecure/html/http-sso-2.asp

Wason, T., Cantor, S., Hodges, J., Kemp, J., & Thompson, P. (2005). *Liberty id-ff architecture overview*. Retrieved March 6th, 2006, from http://www.projectliberty.org/specs/draft-liberty-idff-arch-overview-1.2-errata-v1.0.pdf