NISO RP-2005-01

NISO Metasearch Initiative

Ranking of Authentication and Access Methods Available to the Metasearch Environment

A Recommended Practice of the National Information Standards Organization

Standards Committee BA (Task Group 1)
Access Management

September 13, 2005

Published by the National Information Standards Organization
Bethesda, MD

Contents

Introduction ..... **ii**
Background ii
Access Management Task Group iii
Members iii
Part I: Methodology and Recommendation **1**
Current Situation 1
Methodology 2
Use Cases 3
Environmental Factors 4
Ranking of Authentication Methods 6
Recommendation 7
Next Steps 8
Part II: Detailed Reviews of Authentication Methods **9**
Athens 10
Cookies 12
IP-Filtering 14
Kerberos 16
LDAP (Lightweight Directory Access Protocol) 18
NCIP (NISO Circulation Interchange Protocol) 20
Proxy Servers 23
Referring URL 26
Shibboleth 28
SIP / SIP2 (Standard Interchange Protocol) 31
Username / Password 33
X.509 Authentication Certificates 34

Figures

Figure 1: Access Management Process in Metasearch 1
Figure 2: The Access Management Process 2
Figure 3: Relative Ranking of Authentication Methods 7

Tables

Table 1: Authentication Methods 2
Table 2: Summary List of Environmental Factors 5

Introduction

This report provides an evaluation and ranking of existing authentication methods, as they could be used in a metasearch environment, and recommends metasearch-related authentication best practices in today's environment. It is intended for several purposes:

- To familiarize libraries, resource providers, and metasearch providers with issues and solutions available using current technology.
- As a basis for setting priorities for new development in access management and providing a sensible foundation for making choices in a complex and rapidly changing environment.
- To suggest directions for working within specific communities to develop metasearch compatible technologies.

This document is not, however, a comprehensive training manual on implementing authentication and access methods. It is a starting point for those new to access management technology or a refresher for those familiar with the area.

Background

Metasearch—also called parallel search, federated search, broadcast search, and cross-database search—has become commonplace in the information community's vocabulary. All speak to a common theme of allowing search and retrieval to span multiple databases, sources, platforms, protocols, and vendors at once. Metasearch services rely on a variety of approaches to search and retrieval including open standards (such as NISO's Z39.50), proprietary APIs, and screen scraping. However, the absence of widely supported standards, best practices, and tools makes the metasearch environment less efficient for the system provider, the content provider, and ultimately the end-user.

To move toward industry solutions, NISO sponsored a Metasearch Initiative to enable:

- metasearch service providers to offer more effective and responsive services
- content providers to deliver enhanced content and protect their intellectual property
- libraries to deliver services that distinguish their services from Google and other free web services.

The groundwork for NISO's Metasearch Initiative was laid in two important events:

- A two day strategy meeting in May 2003 defined the metasearch state-of-the-art and built consensus on ways to move forward.
- A metasearch workshop in October 2003 informed librarians, content providers, and aggregators about metasearch.

Following these meetings, NISO established three Task Groups / Standards Committees to address the different Metasearch needs areas:

- **Access Management** (Standards Committee BA / Task Group 1)
- **Collection and Service Descriptions** (Standards Committee BB / Task Group 2)
- **Search and Retrieval** (Standards Committee BC / Task Group 3)

Access Management Task Group

The Access Management Task Group was charged with gathering requirements for Metasearch authentication and access needs, inventorying existing processes now in place, and developing a series of formal use cases describing the needs. Specific deliverables were

- A definitions document of Access Management and Metasearch terms
- Defined distinctions in Access Management between user access and agent access
- Understanding basic requirements of constituents
- An inventory of methods and techniques in use today
- Use cases describing authentication and access needs
- Defined statistics that must be kept to satisfy access management systems

Members

The following committee members contributed to this report:

Mike Teets, Chair
OCLC

Katie Anstock
formerly Talis Information, LTD

Susan Campbell
CCLA

Frank Cervone
Northwestern University

Paul Cope
Auto-Graphics, Inc.

David Fiander
University of Western Ontario

Ted Koppel
Ex Libris, Inc.

Peter Murray
OhioLink

Mark Needleman
SIRSI Corporation

Ed Riding
DYNIX Corporation

R. L. Scott
U.S. DOE, OSTI

Tim Shearer
UNC-Chapel Hill

David Yakimischak
formerly JSTOR

Part I: Methodology and Recommendation

Current Situation

The Access Management Process (AMP), for the purposes of this report, can be defined as *the communication between a user and metasearch engine or metasearch engine and a resource*. The AMP communication protocol defines the way a metasearch engine or resource authenticates, authorizes, and issues credentials based on the presented credentials, attributes, and entitlements of the user or metasearch engine, as depicted in Figure 1.

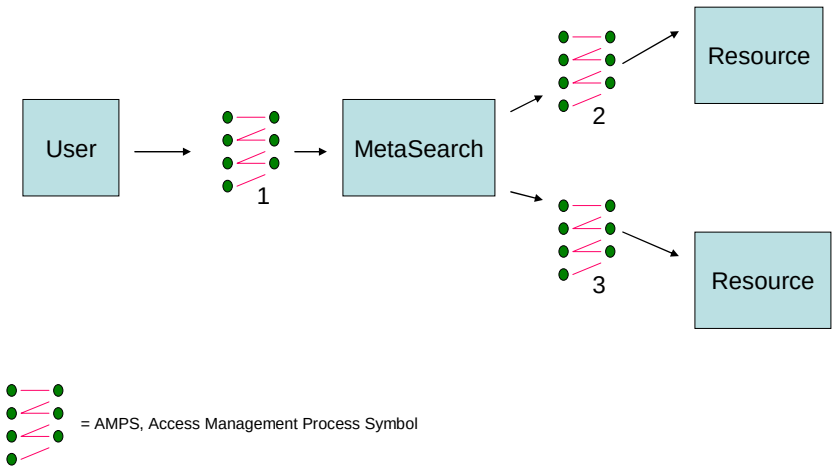


Figure 1: Access Management Process in Metasearch

This AMP process in a metasearch environment can require multiple steps and be quite complex as illustrated in Figure 2. The process involves multiple "actors" (end user, authenticator, authentication release authority, authorizer, metasearch engine, data source) and any actor can play many roles. A variety of methods are currently in use to perform the steps of access management, including proprietary protocols, and the authentication done at one stage cannot necessarily be passed on directly to the next stage.

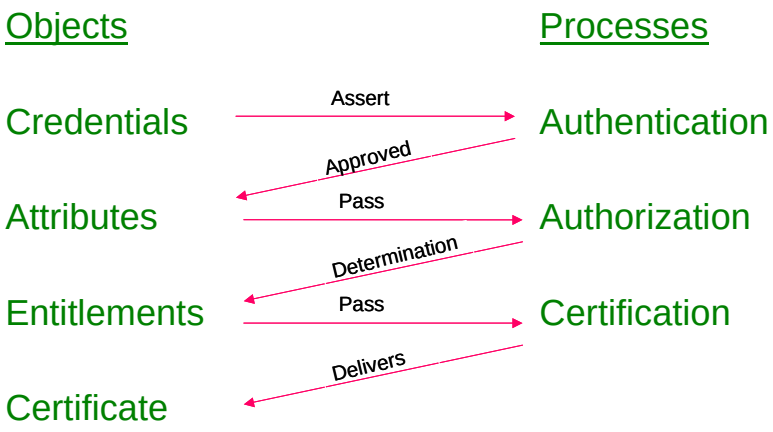


Figure 2: The Access Management Process

The access management process in the current environment can be a very resource intensive operation—even more so than the search and retrieval operation that follows it. As metasearch use continues to grow, improved authentication / certification mechanisms are needed that would reduce performance impact and sustain or even increase security controls.

Methodology

The Task Group used the following method to evaluate AMP solutions and to develop their recommendations:

1. **Survey of authentication methods in use**
   Existing authentication methods were identified through surveys and interviews of metasearch providers. Table 1 lists the methods that were considered for further evaluation and ranking. Detailed discussions of these methods are included in Part II of this report.

Table 1: Authentication Methods

Method	Description
Athens	a proprietary access management system for controlling access to web-based subscription services
Cookies	a small bit of data that a web server directs a web browser to store, which is then returned by the browser to the server in subsequent resource requests
IP Filtering	a method for controlling access to a server based on the Internet Protocol (IP) address of the incoming connection
Kerberos	an IETF-defined network authentication protocol that utilizes a trusted third party, called a “keyserver”, to perform the authentication of clients on a TCP/IP network using symmetric-key cryptography
LDAP (Lightweight Directory Access Protocol)	an IETF-defined protocol for accessing directory type information utilizing a simplified (as compared to X.500) protocol
NCIP (NISO Circulation Interchange Protocol)	a protocol for the exchange of messages between and among applications to enable them to perform the functions necessary to lend and borrow items, to provide controlled access to electronic resources, and to facilitate co-operative management of these functions
Proxy Server	an intermediary server that is used to provide additional security between a client and the end server by filtering or caching transactions in both directions
Referring URL	a method for enabling authentication based on the URL of the source which provided the link

Method	Description
Shibboleth	an implementation of OASIS SAML by Internet2 for the exchange of information about users between a web browser and web server in a secure and privacy-preserving manner
SIP/SIP2 (Standard Interchange Protocol)	a protocol to allow self-service machines in the library to exchange data with the library automation system
Username & Password	a method of authentication requiring the matching of a username with its associated password
X.509 Digital Certificates	a mechanism of utilizing public-key certificates for authentication

2. **Use cases**
   A comprehensive set of use cases were developed and then simplified to three metasearch specific cases. These are summarized in the **Use Cases** section.

3. **Environmental factors**
   A set of environmental factors was identified that are critical success factors in metasearch. These factors are discussed further in the **Environmental Factors** section.

4. **Ranking of methods against use cases and environmental factors**
   Each method was ranked on a ten point scale indicating how well it addressed each use case and environmental factor.

5. **Aggregation and modeling of rankings**
   The rankings were aggregated and modeled graphically to identify the best solutions. This model is explained and illustrated in the **Ranking of Authentication Methods** section.

6. **Recommendation**
   The committee concluded its evaluations with a best practice recommendation as described in the **Recommendation** section.

Use Cases

Detailed use cases were developed that included an understanding of:

- **Primary actor** – the principal actor that calls upon system services to achieve a goal
- **Stakeholders' behavior** – the behaviors related to satisfying the stakeholders' interests
- **Preconditions** – what must always be true at the beginning of the use case scenario
- **Indicators of success** – what must be true for the successful completion of the scenario
- **Main success scenario** – the typical success path or flow for a successful scenario
- **Alternate flows** – other scenarios, branches, or decisions that may represent successful or failed scenarios
- **Technology requirements** – any technology specific requirements for conducting the use case scenario
- **Special requirements** – additional behavioral or technical requirements related to the use case
- **Frequency of occurrence** – how often or frequently the use case scenario may need to be repeated
- **Open issues** – known issues for success in the use case

These detailed cases were then combined into three broadly defined situations in which the type of authentication or authorization system required by an information service provider affects a member of a subscribing organization (or community) attempting to access the information service via a metasearch engine:

1. **In-Domain User –** A user attempts to access a licensed database via the metasearch engine from a location that is on the network of the licensing organization (an "in-domain" user)
   For most licensed information service products, the in-domain user is the simplest case. Someone who is permitted to access the physical resources of the licensing organization is assumed to be authorized to use the networked resources licensed to the organization. In the case of a resource that is licensed to just a particular group within a larger organization,[1] an "in-domain" user is one who is on the network of the subgroup that has licensed the material.
2. **Out-of-Domain User –** A user attempts to access a licensed database via the metasearch engine from a location that is not on the network of the licensing organization (an "out-of-domain" user).
   The value of electronic resources to the end user is almost entirely one of convenience. Thus, while in-domain use is the simplest to handle, it is out of domain use that is usually of most importance to the users themselves; they want to be able to access the resources not just in the library, but in departmental offices or labs, or anywhere on campus. Further, they must be able to access the resources from home.
3. **Credentialed Access** – A user attempts to access a licensed database via the metasearch engine that relies on some sort of credential to manage resource access.

Environmental Factors

Although the authentication methods can be examined purely in terms of the user when evaluating suitability for a given use case, environmental factors play a critical role as well. These factors must be applied within three different contexts: the metasearch service provider, the information service (i.e. database) provider, and the licensing organization and its users.

Eleven environmental factors were identified as critical success factors in metasearching.

Suitability / Effectiveness – Is this authentication method suitable or effective at providing access control? Service providers will evaluate this in terms of reliability and security. Users will evaluate in terms of ability to access the licensed resources.

Ease of Implementation – How easy is it to implement this authentication method? This factor can lead to very different rankings for service providers versus licensing organizations. For example, IP filtering can be very simple for a university to "implement," since all that is required is that a list of IP addresses or ranges be reported to the service provider. The provider, on the other hand, must maintain a database of authorized IP ranges and check all incoming connections against that database.

Licensing Cost – How expensive is it to license any infrastructure necessary to implement the authentication method? For the most common systems deployed today, there is zero, or minimal, licensing cost. Newer and proprietary systems (such as Kerberos or SIP) may require users to acquire software licenses.

Implementation Cost – How expensive is it to implement the authentication method? This is indirectly related to the ease of implementation. Systems that require client software to be installed on end-user computers (such as the X.509 digital certificate infrastructure) will be more expensive than more passive systems like IP filtering.

[1] For example, the Law school has a license to Lexis/Nexis, which is restricted to members of that program only, rather than to the entire university.

Software Expertise Required – How much networking or programming expertise is required to implement and maintain the system? In some cases, individual end users may require a certain level of software expertise (for example, can the user successfully modify the proxy and security configuration of their web browser).

Security – How secure is the authentication method? Is it susceptible to spoofing, forging identities, or cracking?

Maintainability – How much ongoing work is required to maintain the authentication system? What types of changes within the licensing organization require changes to the configuration of the system?

Robustness – How robust is the authentication method? The working group members generally interpreted robustness as a combination of security, maintainability and scalability. One authentication method is more robust that another if it can be set up and then left to run, with little ongoing attention required, beyond monitoring its performance.

Scalability – How scalable is the authentication method? Does it cope well with large numbers of users, licensing organizations, or parallel connections?

Simplicity of Understanding – How simple is the authentication method to understand for the people involved? Having a clear model of how the authentication method works can often simplify support issues.

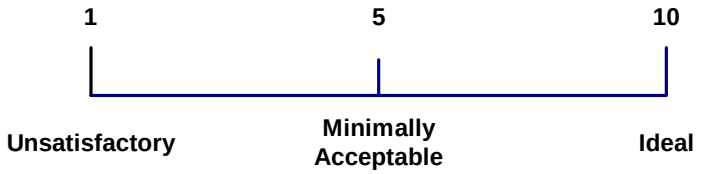
Market Acceptance / Preexisting Implementations – How common is the authentication method? Does the licensing organization already have the necessary infrastructure in place to support the method? Does the information service provider have other clients already using the authentication method?

Table 2: Summary List of Environmental Factors

1. Suitability/Effectiveness	7. Maintainability
2. Ease of Implementation	8. Robustness
3. Licensing Cost	9. Scalability
4. Implementation Cost	10. Simplicity of Understanding
5. Software Expertise Required	11. Acceptance/Preexisting Implementation
6. Security	

Ranking of Authentication Methods

Each authentication method was ranked separately on use cases and the environmental factors using a ten-point scale:



All of the rankings were combined into an average and the rankings were graphed on a scatter plot with the X axis representing Use Case rankings and the Y axis representing Environmental Factor rankings (Figure 3). While ranking each method, the group was mindful of the different organizational contexts of metasearch applications. For instance, an access method such as Kerberos or Shibboleth might fit well in a college campus setting and deserve a higher mark if only considering that environment. However, some of the very attributes that make that method very effective in a campus setting make it inappropriate in a public library setting. The group's goal was to identify the best methods for universal adoption.

Methods to the right of the chart in Figure 3 are considered better at satisfying the requirements of the use cases. Methods near the top of the chart performed better on the environmental factors. In general, the rankings should be considered to be relative ones, rather than absolute. For example, Shibboleth satisfied use case requirements better than Referring URL did, while IP Filtering ranked better on environmental factors than Shibboleth.

In some senses, it is the X-axis position on the graph that should be considered more important, as it represents an authentication method's ability to meet the needs of the users. In many cases, a poor ranking on the Environmental Factors axis has more to do with the current implementation environment than with the method itself. For example, Shibboleth was the second-highest ranked method in terms of ability to meet the needs of the use cases, but it scored very poorly on the environmental factors because at the time the ranking was begun, it had only been deployed in test environments and few vendors supported it. Since the UK's Joint Information Systems Committee (JISC), has announced plans to move from Athens, its current authorization system, to Shibboleth within the decade, Shibboleth's environmental rankings for *Acceptance/Preexisting Implementations* and *Ease of Implementation* are expected to improve dramatically.[2]

Please remember that these rankings are focused on authentication in the metasearch environment alone. These should not be considered a generic ranking of the strength of the evaluated models.

[2] JISC. *The Future position on Athens and Shibboleth.* 9 Aug 2004. Accessed 26 Nov 2004. http://www.jisc.ac.uk/index.cfm?name=jisc_athens_shibboleth_pos_news050804.

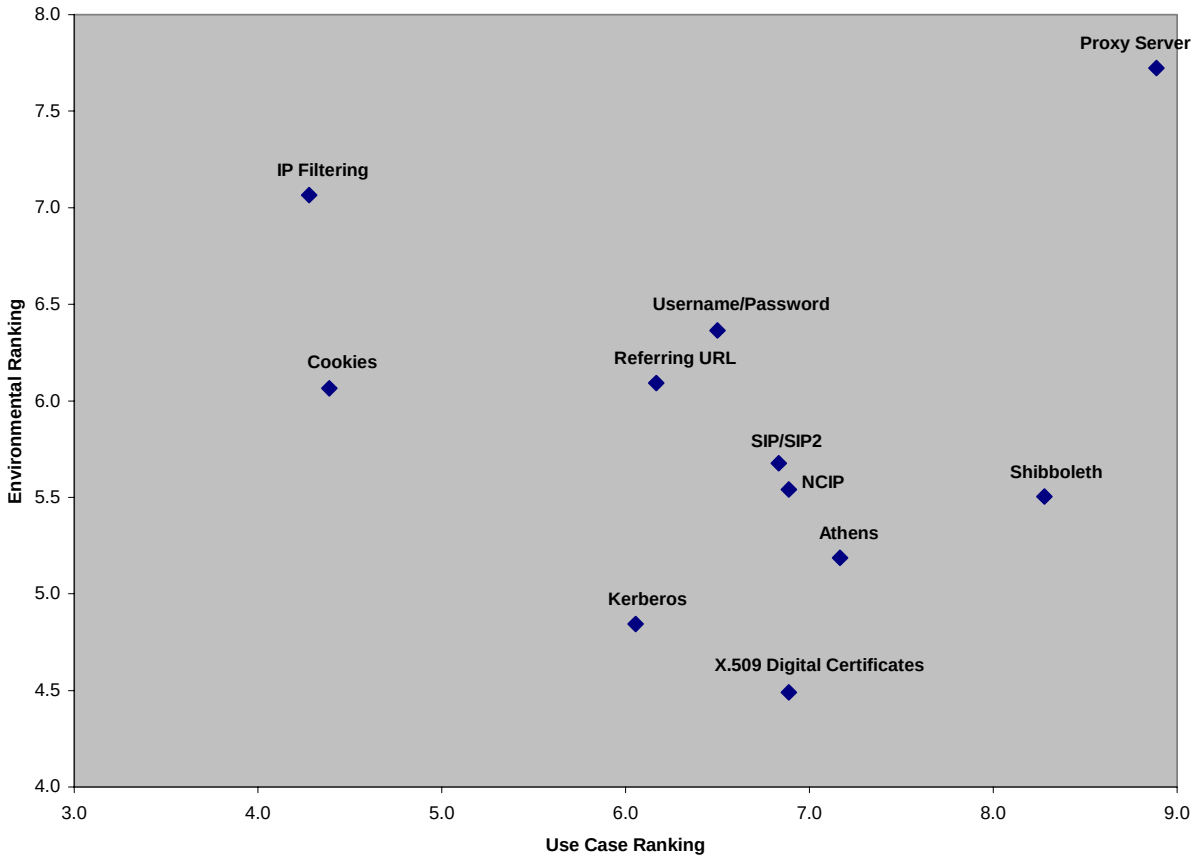


Figure 3: Relative Ranking of Authentication Methods

Recommendation

The NISO Metasearch Initiative Task Group on Access Management recommends that institutions in the process of acquiring new electronic resources should implement either:

- **IP-Authentication with a Proxy Server (either "traditional" or "rewriting"),** or
- **Username/Password authentication**

to control access to their electronic resources and support metasearch.

These were the two highest ranked authentication methods, in terms of both environmental factors and user acceptability systems, are the most widely supported by vendors, have the lowest implementation and maintenance costs, and are the simplest for smaller or less technically sophisticated organizations to implement. They also ensure that remote (i.e. off-site) users can access the resources of the institution with little difficulty.

While Athens and Shibboleth were both evaluated as more "usable," and IP Filtering had a higher ranking in terms of environmental factors, support for these methods was unbalanced: Athens and Shibboleth are not broadly deployed in the current environment (Shibboleth is emerging); and IP Filtering doesn't provide an acceptable level of usability for off-site users, who are often primary beneficiaries of an institution's networked resources.

Next Steps

During the development of this study, the Task Group determined that while Shibboleth had many features making it a desirable alternative, the current Shibboleth implementation model does not allow for mediated access to controlled resources, as required by a user performing a metasearch of several distributed resources.

Members of the NISO Metasearch Initiative and the metasearch community have started working with the Shibboleth developers to ensure that Shibboleth 2, the next version of that specification, will provide facilities that will allow surrogates to authenticate to service providers as the user that initiated the request.

The Internet2 Shibboleth project team, recognizing the growing need for access management in distributed environments such as metasearch, grid computing, and information portals, has begun the work of implementing the OASIS SAML 2.0 specification. Cross participation between the Shibboleth project team and the NISO Metasearch Initiative access management task group has been established and work has started on the creation of use cases that express the needs of a metasearch environment.

Part II:

Detailed Reviews of Authentication Methods

Athens

Description:	a proprietary access management system for controlling access to web-based subscription services
Developer:	Eduserv Technologies Ltd. (Bath, United Kingdom)
Website:	http://www.athens.ac.uk

Overview:

Athens was first implemented in 1996. It is used in the UK by all HEIs (Higher Education Institutes), the NHS (National Health Service), and many FE (Further Education) institutions as a means of authenticating and authorizing users to access electronic resources. It is a centralized service hosted by Eduserv in the UK, funded by the JISC (the Joint Information Systems Committee).

Athens is, fundamentally, a central repository of organizations, usernames and passwords with associated rights. It has extensive account management facilities for organizations to create and manage usernames and passwords, and to allocate rights to individual usernames.

The service also offers single sign-on whereby when a user signs in to the resource, they are subsequently signed in to all Athens authenticated resources.

Athens now offers the Athens Devolved Authentication (Athens DA) service whereby the institution has an alternative method of authentication—LDAP for example—and once authenticated the user is then signed in to all Athens resources. Many HEIs are implementing this as an alternative to the "traditional" Athens service.

Note the Athens service is funded until July 2006. There are moves within the JISC to implement Shibboleth as a replacement/alternative to Athens.

Traditional Athens workflow:

The user goes to an electronic resource that offers Athens authentication. At the Athens authentication prompt, s/he enters their username and password (see Username / Password AMP, below, for further information).

The user is authenticated against the central Athens repository, which stores what electronic resources the user can access.

With single sign-on, the user then goes to another electronic resource, which offers Athens authentication, and the user is authenticated.

Pluses:

- Once users have an Athens username and password, they can access all Athens authenticated resources to which their home institution subscribes, so if a user has signed into the metasearch engine using their Athens username and password they are subsequently authenticated for all Athens resources (260 resources currently).
- Athens targets are available from anywhere (on or off a campus).
- Single sign-on means users are seamlessly passed from one e-resource to another.

Minuses:

- There is a maintenance overhead in the issuing and maintenance of the Athens service for institutions.
- All users are stored internally on local services as well as in the central repository so there is a need to synchronize.
- The technology on which Athens is based is proprietary; it is designed to work with web-based electronic resources, i.e. human readable, rather than Z-targets or other machine-to-machine communication methods.

Metasearch Athens Workflow

A user signs in to the metasearch engine with his/her institutional username and password.

Behind the scenes, the user is also signed in to Athens and will therefore subsequently be able to access all Athens electronic resources seamlessly for that session.

Pluses:

- Users have only their institutional username and password to remember.
- The user can be signed on to multiple applications simultaneously (e.g. the learning management system as well as the metasearch engine) as well as to electronic resources.

Minuses:

- There is a maintenance overhead in the issuing and maintenance of the Athens service for institutions.
- All users are stored internally on local services as well as in the central repository so there is a need to synchronize.
- The technology on which Athens is based is proprietary; it is designed to work with web-based electronic resources, i.e. human readable, rather than Z-targets or other machine-to-machine communication methods

Cookies

Description:	a small bit of data that a web server directs a web browser to store, which is then returned by the browser to the server in subsequent resource requests
Developer:	The Internet Engineering Task Force (IETF)
Specification:	RFC 2965, *HTTP State Management Mechanism*, October 2000. http://www.ietf.org/rfc/rfc2965.txt

Overview:

From the introduction of Netscape 3.0 and Internet Explorer (IE) 3.0 in the mid-90s, browsers have provided a means of interacting with a web server by recording small bits of identifying information in (theoretically) temporary files on a client PC. This mechanism is known as setting a cookie in the user's browser. Many types of websites now commonly use these cookies for a multitude of purposes.

The standard developed from an initiative at Netscape in the mid 90s. This de facto standard is rather simple, but very flexible and can be expanded as needs are defined. This flexibility is a double-edged sword. It means that there is no standard means of representing either authentication or authorization information via cookies and it further means that information in a cookie is not necessarily restricted to the use of the application that generated the cookie in the first place.

Cookies are a bit of an anomaly as an authentication method because cookies do not provide the mechanism for gaining initial authentication. Cookies function as a token identifier signifying that a user has passed some other, preexisting authentication mechanism and been granted authorization for access.

A major problem with cookies however, is that while a cookie is initially set in the browser as the result of a specific user's request, there is no inherent linkage of the cookie to user. Furthermore, there is no requirement that the cookie be deleted once the original user's session has ended. As a result, subsequent presentation of the cookie is not guaranteed to be from the user to whom the cookie was originally granted. As a consequence, without strong security controls on the client side, it is very likely that a cookie can be reused by a user to whom it was not originally granted. Furthermore, unless there is an aggressive fraud detection mechanism in place, cookie-based access management tokens are subject to manipulation and/or redistribution.

Workflow:

Cookies are sent from a server via response headers in an HTTP transaction that instructs the user's browser to set a cookie with a particular name and value. For example, the following HTTP response header would set a "username" cookie:

Set-Cookie: NAME=username; VALUE=jhn321; expires=DATE; path=PATH; domain=DOMAIN_NAME.

Subsequent cookies could be used to store password and various aspects of authorization information for this user. (JavaScript executed within the browser context can also read and set cookies.)

Once the appropriate cookie has been set, subsequent browser requests to the same server (or domain, depending on the nature of the cookie parameters) would include the cookie name and value in subsequent HTTP request headers. Based on the information returned, the server can then make a determination of the authorization a user has and provide access as appropriate.

Pluses:

- Authorization of the user is specific to the individual user.
- Can be used to provide authorization to specific resources.
- Well-known and documented interface.
- Developing applications to use cookies is relatively simple compared to other methods.
- Can be used to remote user access and control.
- Allows for site personalization by storing cookie information persistently in the browser cache.
- Provides a handy means for storing information in a state-less environment.
- Cookies can be used on the server side to track the movement of the user through a site and may potentially provide statistics on how long and how often particular pages are viewed.

Minuses:

- Does not provide a mechanism for initial authentication.
- Data security during system interchange is not required to be secure, i.e., username and password information could be sent in the clear.
- Size and number of cookies that can be stored on the client side are limited.
- Allowing cookie information to be stored persistently in the browser cache is an inherent security risk as cookies are typically stored as plain text in a browser cache directory where anyone could potentially view or modify them.
- Cookies can be disabled, completely or partially, on the client side.
- In a state-less environment, potential exposures to unauthorized use could be easily exploited since there is no mechanism to guarantee that the user presenting the cookie is the person that was originally authenticated.
- Cookies can be used on the server side to track the movement of the user through a site and could potentially be used to spy on user activities.

Recap

The use of cookies for establishing authorization is ubiquitous on the Web. It is no surprise that it is also extensively used in library information systems. However, while this method is simple to implement and use for authorization, it has many detractors due to the inherent lack of security within the model. A primary concern is that once initial authentication has been established, subsequent presentation of the cookie is not guaranteed to be from the person to whom the cookie was originally granted.

IP-Filtering

Description:	a method for limiting access to a server based on the Internet Protocol (IP) address of the incoming connection
Specification:	There is no one standard or specification for the IP Filtering methodology. It is usually done through configuring a firewall, proxy server, or other network traffic management software.
Website:	For more information on IP addresses, visit the IANA (Internet Assigned Name Authority) website: http://www.iana.org/

Overview:

Since the advent of the Web and the offering of protected resources via the Web, IP Filtering has been a means of restricting access to a set of qualified users—the qualification being that these users come from a recognized set of IP addresses. In fact, this method is widespread, being used by virtually all information providers.

It is fairly straightforward, requiring the library or leasing entity to communicate all IP addresses (or ranges thereof) representing workstations within the physical locations under the library or leasing entity's jurisdiction. Any changes in IP-addresses (new workstations, new subnets, etc.) must also be communicated to the vendor.

Workflow:

A user attempts to access a resource using a link for "internal" users. Because he is situated at or is using a sanctioned IP-address, he is allowed immediate access, without login. If attempting to access the resource from an unsanctioned IP-address (outside the library or campus jurisdiction), the connection is refused, and the user is forced into some other method of validation, such as login.

Users are allowed or denied access based on their IP-address.

Pluses:

- Method is easily and well-understood by both library staff and vendor personnel.
- IP addresses "seem" to be easily quantified and communicated.
- Vendors can store IP addresses and easily link them to a purchasing entity (library) for billing, access control, and statistical purposes.
- Requires no login by the "internal" user.
- Very good solution for "onsite" users.

Minuses:

- Challenge to keep vendors up-to-date on all IP addresses or ranges needed by the library.
- Does not accommodate "remote" users (those attempting to access the protected resource from outside the library's jurisdiction).
- Access is not terribly secure as the IP address can be spoofed or internal machines can be compromised with unauthorized or misconfigured proxy servers.

Recap

IP-Filtering provides a well-understood solution for those who can always predict the IP addresses of their users, but not for users attempting to access the resource from outside the library's typical domain.

Kerberos

Description:	an IETF-defined network authentication protocol that utilizes a trusted third party, called a "keyserver", to perform the authentication of clients on a TCP/IP network using symmetric-key cryptography
Developer:	Massachusetts Institute of Technology (Cambridge, Massachusetts)
Specification:	*The Kerberos Network Authentication Service* (version 5), RFC 4120, July 2005. http://www.ietf.org/rfc/rfc4120.txt
Website:	http://www.mit.edu/~kerberos/

Overview:

Kerberos is a network authentication protocol developed at MIT. It was designed to promote secure identity authentication over insecure networks by having each party to the authentication process prove its identity to a third party (the Kerberos Authentication Server). If the server is satisfied as to the identity of each party, credentials (known as tickets) can be passed, allowing secure, authenticated communication.

Workflow:

A user is presented with an appropriate "login" screen. That process creates a "ticket" that is sent to an authorization server.

The Authentication Server (AS) examines ticket. If satisfied, says "yes" and processes ticket.

The AS creates an encrypted message ticket and sends it to the network destination.

The Destination examines the AS credential (to ensure it is a secure transaction) and the user's ticket (that describes the user). If satisfied, the network connection is made and communication begins.

NOTE: There are all sorts of complicated variations to this theme:

(a)If the person and network destination are distant, they may be in different "realms." Cross-realm functionality must assure security across different networks.

(b) Pre-authentication can take place for enhanced security against easy (bad) passwords.

(c) Tickets can be forwarded from server to server.

(d) Tickets can be renewable past their expiration date.

(e) Tickets can be post-dated if necessary.

Pluses:

- High security level because both parties must be authenticated.
- Individual identity may not be shared across network, only that person is known to and approved by the Kerberos Authentication Server.
- Software is freely distributed.

Minuses:

- Does not provide for authorization to specific resources. Authenticates a network identity, but does not indicate the privileges associated with that identity.

- Although standard uses "credentials," these are taken to mean the person's identity and their punched "ticket." There is no way to carry person-specific attributes, such as department, faculty status, etc., for analysis by target to give privileges.
- Relatively complex installation and set-up for smaller academic and many public libraries. Need fairly sophisticated Unix (and other) knowledge. Although software is free, knowledge cost is high.
- Doesn't appear to allow the logging and chargeback capability needed in some metasearch use scenarios.

Recap

Kerberos appears to be rather complex and somewhat beyond the support reach of all but large academic campuses, which are often blessed with Unix and technical experts. Even if it were simpler, it appears to be inadequate for metasearch. Kerberos authenticates the person (only), but does not carry credentials or other data elements that would allow downstream information suppliers to make decisions about access.

LDAP (Lightweight Directory Access Protocol)

Description:	an IETF-defined protocol for accessing directory type information utilizing a simplified (as compared to X.500) protocol
Developer:	The Internet Engineering Task Force (IETF)
Specification:	*Lightweight Directory Access Protocol: Technical Specification* (version 3), RFC 3377, September 2002. http://www.ietf.org/rfc/rfc3377.txt

Overview:

LDAP, Lightweight Directory Access Protocol, was initiated and designed by the IETF (Internet Engineering Task Force) as a way to make use of existing X.500 directories. A common misconception is that LDAP is a directory. It is in fact a protocol to access and update any directory. The term lightweight is a relative term in the context of broad access to information retrieval systems. LDAP fits into a class of services generally known as pluggable authentication modules. Implementing authentication and authorization in LDAP is an exercise in extensive interoperability testing. There are a host of relatively proprietary LDAP implementations wrapped into broader commercial services. OpenLDAP, (http://www.openldap.org/) is an open source implementation of LDAP that includes a full implementation suite of services. Unfortunately, many of the commercial services that may house existing user identifications are tied to the commercial implementations, which do not cover OpenLDAP in their support agreements. As a relatively low level software component, broad adoption is limited by the necessity to bi-laterally interoperability test with each service partner.

Workflow:

NOTE: Implementation assumes that a directory of user profiles exists or is created to store authentication information.

Resource provider selects LDAP implementation software and connects to data store.

Client service (and metasearch service) must select either the same software or interoperability test each LDAP transaction with the resource provider.

Once services are available, user connects to a resource through a known identity such as username/password, email address/password, or screen name/password.

Metasearch providers essentially function as an LDAP server to the user and as a client to the resource provider.

Authentication requests through LDAP are typically simple pass/fail response.

Pluses:

- Authorization is typically a shared directory with other services so the user has fewer “keys” to remember. Such “keys” might include their university student number, email address or other well known identity.
- Well-known and documented protocol.
- Uses existing IT directory infrastructure.

Minuses:

- When identities are based on well known identities, such as email address, LDAP may be considered less secure.
- Often used in relatively closed environments (inside a single campus or service); not widely used across many services external to the primary campus or service.
- Not broadly adopted in information retrieval systems.
- Interoperability testing between each service using a hosted LDAP service is necessary.

Recap

While the term LDAP and its purpose are widely known, its use as a client-server authentication protocol in a heterogeneous environment is not common. LDAP works well in closed campus environments where all of the services are within the control of a single domain. Use of LDAP as an authentication method through a metasearch service to a resource provider is not recommended.

NCIP (NISO Circulation Interchange Protocol)

Description:	a protocol for the exchange of messages between and among applications to enable them to perform the functions necessary to lend and borrow items, to provide controlled access to electronic resources, and to facilitate co-operative management of these functions
Developer:	National Information Standards Organization (Bethesda, Maryland)
Maintenance Organization:	Colorado State Library (Denver, Colorado)
Specification:	ANSI/NISO Z39.83-2002, *Circulation Interchange Part 1: Protocol (NCIP).* http://www.niso.org/standards/resources/z3983pt1rev1.pdf ANSI/NISO Z39.83-2002, *Circulation Interchange Part 2: Protocol Implementation Profile 1*. http://www.niso.org/standards/resources/z3983pt2.pdf
Website:	http://www.cde.state.co.us/ncip/

Overview:

NCIP or the NISO Circulation Interchange Protocol (ANSI/NISO Z39.83) was developed to replace the SIP and SIP2 protocols (see below for more on SIP).

The standard is now in production in some limited areas and its use will be expanding as vendors begin to implement the standard into new products and retrofit it to existing systems.

NCIP expands on the concepts that are part of the SIP/SIP2 protocol by utilizing message sets and replacing text strings for message interchange. It uses XML data structures for data transfer or exchange between different computer systems.

The standard is extremely flexible and can be expanded as needs are defined. This flexibility leads to some basic implementation concerns of how each vendor implements their individual solutions and how each vendor will interface to other vendor's systems.

The NCIP standard consists of messages that are needed for typical Integrated Library System (ILS) processes. These processes can be loosely defined as three basic modules or components. First, there is user authentication (also used by the other two modules). Second, there is self-service or self check out. Finally there is the ILL (book borrowing and loaning) or direct consortial borrowing (DCB). Vendors can implement any or all of the components as will be relevant to their individual systems and requirements.

A system that requests information from another system is considered a Requestor. A system that responds to a request is considered a Responder. Some systems may act only as requestors, some as responders only, and some will be both depending on the specific vendor's needs or functions of their systems. Individual message request and response arguments within the three modules will also be implemented as the vendors' systems dictate.

Primary interest for this committee is the user authentication component of the standard.

Workflow:

A user is presented with an appropriate "login" screen. The user enters his/her user-id/barcode and password or credentials. User's credentials are passed from the "application" client (requestor) to target

NCIP service (or responder). Messages are exchanged per the protocol standard and the user authenticated.

Authentication of the user results in a "valid" user or an invalid user. Valid users are those that exist in the system; invalid are non-existent users. Additional information about the user is also available and may be passed based on the services and messages supported and requested by the application client or requestor and the target or responder.

Users are allowed access or denied based on the user's existence or by further testing using additional message to get value judgments regarding the user.

Pluses:

- Authorization of the user is specific based on user credentials.
- Ties well to ILS system for user authentication.
- Can tie authorization to status of user in the ILS (no fines, exceeds fines limits).
- New standard—lots of interest.
- XML based for simplified data exchange and processing.
- Implementation should be considered "easy" or the fallout of a system, which is tied to an ILS system; thus no extra staff effort is required to build and maintain user records and the user permissions.
- Very good solution to address remote user access and control.

Minuses:

- Does not provide for authorization to specific resources. NCIP is designed to provide the identity and status of the user relative to the ILS system or circulation (CIRC) process.

  To clarify: Assuming that a patron is "enabled" in the ILS for CIRC, that it is what the NCIP target will tell the other system when the user is authenticated. It is not likely that a CIRC system will be tracking authorization levels for specific resources. It is possible for NCIP to transmit attributes defined in a patron record, such as patron class or department, and those attributes could be used by a service provider in making access control decisions. Since NCIP is primarily for CIRC authentication of a user or access control, its use may be limited for actual authorization of specific resources.

- Not expected to be used by resource providers for authentication or authorization into their systems if they do not authenticate at the specific user level.

  To clarify: Some content providers like OCLC are starting to use NCIP like an ILS vendor would. They use NCIP to authenticate patron or staff members. There will be some content providers that must authenticate specific users and thus NCIP will meet their needs. However, if a vendor authenticates based on institution IDs or libraries—and not to specific user IDs or accounts—such as Gale or EBSCO do, then it is less likely this standard will be used. OCLC, although a data provider, needs to track or grant access or authenticate specific staff within an organization so permissions can be provided. Whereas a vendor such as Gale or EBSCO allows all users from Library X to see the resource or resources. There are even special cases where special library codes are used to provide different interfaces (for example, children vs. adults) rather than authorizing to specific user.

As pointed out, some content providers (such as e-book providers) who need to or wish to track usage or access to specific patrons or users will find NCIP provides a viable authentication solution.

- New standard—limited implementations.
- Cost to the library for "service" from ILS or associated vendor. This will be true with most similar services.
- Data security during system interchanges needs to be addressed.
- As with any remote system or server authentication, the problem of target availability needs to be considered. For example, if a user tries to access a system that uses NCIP remote authentication, if that remote server or target is not available the user cannot be authenticated and access may be denied. This is true for all similar remote authentication methods.
- Does not address in-domain users. Libraries wishing to provide "in-domain" terminals, which do not require a user to be authenticated, will need to have some other form of authentication in place.

Recap

When implementing a user authentication scheme that will be tied closely to an ILS, this protocol would be worth considering for user authentication. Legacy systems will be the last to see support (if at all) of the protocol, so alternate methods of authentication or a protocol such as SIP/SIP2 may be required to get the same results.

Alone this protocol does not address authorization of specific resources, although the flexibility of the protocol could be incorporated as needed and information shared using the protocol.

Proxy Servers

Description:	an intermediary server that is used to provide additional security between a client and the end server by filtering or caching transactions in both directions
Developer:	The Internet Engineering Task Force (IETF)
Specification:	Proxy server implementation relies on multiple IETF internet protocols. For an overview of the technologies involved and references to relevant specifications, see: *Internet Web Replication and Caching Taxonomy*, RFC 3040, January 2001 http://www.ietf.org/rfc/rfc3040.txt

Overview:

Proxy servers play an intermediary role between a client and a server. The client passes its request to the proxy server. The proxy server then (often) passes the request along to the "real" server as though it were the client. The "real" server passes responses back to the proxy server as though fulfilling a request for a client. Then the proxy server takes on the role of server again and redirects the response back to the "real" client. Essentially the proxy server plays the role of middleman.

The roles a proxy server can play are varied and include:

- Improved performance – Proxy servers can be used to cache web content. In this scenario, the request is passed to the proxy server. If the proxy server has an appropriate version of the requested file(s) it will reply directly. If not, it will pass the request on, as described above.
- Security – In organizations with a firewall (a means of blocking incoming and outgoing traffic), a proxy server can be used to limit off-site access to certain internal users, or to allow access to only certain external websites. This is the most common use of proxy servers in corporate environments.
- Augmented IP Filtering – In the metasearch environment, the proxy server can be used to extend access for users who meet the licensing agreement but are on a machine that does not have a resource-registered IP address. In this role, the proxy server, which does have a registered IP address—and may therefore access the resource(s)—stands in for the remote user. In the metasearch authentication and authorization world, proxy servers typically play only an authorization role; once users has proved who they are, they may inherit the privileges that the authorized IP (that of the proxy server) has been granted.

It is worth noting that there are presently two basic kinds of proxy servers.

- Traditional Proxy Server – This type of server typically relies on client configuration. The client can either be manually configured to send traffic through a proxy server or can be set to perform automatic configuration (essentially a JavaScript file with hosts/domains; as links are followed, they are looked up and proxied if a match is found).
- Rewriting Proxy Server – This type of server relies on configured URLs. Essentially every URL that is to be proxied gets a URL prefixed to the "real" URL. The prefix leads requests to the proxy server itself, which picks up the "real" URL and then performs as a traditional proxy server.

Workflow:

NOTE: There is an assumption that the institution relies primarily on IP filtering to meet its licensing agreements.

Traditional Proxy Server:

A student has configured her client with the URL for an automatic proxy configuration. Upon firing up the client, a fresh version of the auto-config file is downloaded.

Upon performing a metasearch from off campus the client detects, based on the fresh auto-config file, that the URL is in a domain that should be funneled through the proxy server. The request is forwarded to the proxy server. [Optional: To reduce proxy traffic, a routine checks the IP address. If off-campus it continues; if on-campus it merely redirects the client directly to the remote server.]

The proxy server checks to see if the user has authenticated (usually through a cookie). If not, it pushes the user to an authentication routine. Upon evidence of successful authentication, the proxy server performs its middleman role.

Rewriting Proxy Server:

A student performs a metasearch from off campus. She clicks on a URL (or form action) that is prefixed with the URL to the proxy server.

The request is forwarded to the proxy server. [Optional: To reduce proxy traffic, a routine checks the IP address. If off-campus it continues; if on-campus it merely redirects the client directly to the remote server.]

The proxy server checks to see if the user has authenticated (usually through a cookie). If not, it pushes the user to an authentication routine. Upon evidence of successful authentication the proxy server performs its middleman role.

Pluses:

Traditional Proxy Server:

- Once configured, things are relatively seamless for the end user.
- Only have to authenticate once per session (depending on the scheduled life of cookies).
- Configured clients work “anywhere” on the Web. For instance, when clicking on a link to an auto-configured resource from Google Scholar or a sister institution, the user still gets routed through the proxy and the link “works” for the user (assuming the link goes to something that the institution has licensed).

Rewriting Proxy Server:

- No need for users to configure their client. If they are using your site, things seem to work as if by magic.

Minuses:

Traditional Proxy Server:

- Users must configure their browsers:
  - If using multiple browsers (IE, Firefox, Safari), they must configure each version of each type of client they use.
  - If users are in a public space (such as a public library unrelated to their “home” institution), they may not have permissions to configure the client. With permissions, they are configuring

that client for all walk-in patrons to that library, which may be inappropriate. Finally, if cookies are the mechanism for proving that one has authenticated, subsequent patrons may get "illegal" access until the cookie has expired.

- Clients typically support one configuration. If the user belongs to multiple institutions that provide proxy access, they must configure to belong to one institution, and then re-configure to belong to each of the others.

- Off-site users that are attempting to access licensed resources from behind a firewall (e.g. part-time students doing research from their places of employment) may not be able to configure their browsers, since they may need to "daisy-chain" two different proxy servers.
- Not all remote resources "play well" with traditional proxy servers.
- Single point of failure for off-campus access in this scenario. If the proxy server is down, the whole mechanism fails.

Rewriting Proxy Server:

- Rewriting proxy servers have more trouble than traditional proxies with performing the middleman role when cookies must be passed between the remote server and the end user's client.
- This requires pre-configured URLs. If the user follows a link that is not preconfigured (i.e. a bookmark or a URL created by an institutional staff member who does not understand the technology) they will fail to be able to use the service.
- Off-site users that are attempting to access licensed resources from behind a firewall (e.g. part-time students doing research from their places of employment) may not be able to access resources, depending on what websites or access ports are permitted through the firewall.
- Not all remote resources "play well" with rewriting proxy servers.
- Single point of failure for off-campus access in this scenario. If the proxy server is down the whole mechanism fails.

Recap

Proxy servers can greatly improve remote access to resources when the main authorization method is IP filtering. They are widely deployed; there are quite a few conventional and open source options; and they are a fairly low barrier technology. Though proxy servers do address authorization they do not necessarily deal with the issue of authentication. Implementers must still have users prove who they are before allowing them to make use of the proxy. Traditional proxies can be frustrating for end users because they must configure their clients. Rewriting proxies can be frustrating for end users because they run into web pages (created by uninformed staff) with non-prefixed links , and because their personal bookmarks and URLs that they type in will not work with the rewriting proxy server.

The rewriting proxy server works best for occasional or casual users of an institution's resources: for example, a shared home computer that is only occasionally used to access licensed information resources. The traditional proxy server's demands that the user change "advanced" browser configuration settings and log in to the proxy server every time the browser is started are too inconvenient for this situation.

The traditional proxy server is recommended for environments in which there is a strong constituency of users that are normally off-site and who usually use their computers for research. For example, clinical medical researchers who work in the teaching hospitals but rely on the university's purchased information resources. For these users, configuring the browser and logging in to it are not onerous, and the traditional proxy server's ability to support bookmarks and URLs that have been typed in by the user are essential.

Referring URL

Description:	a method for enabling authentication based on the URL of the source which provided the link
Developer:	The Internet Engineering Task Force (IETF)
Specification:	Referring URLs are incorporated into the HTTP header, as defined in: *Hypertext Transfer Protocol – HTTP/1.1*, RFC 2616, June 1999 http://www.ietf.org/rfc/rfc2616.txt See also: *HTTP Authentication: Basic and Digest Access Authentication*, RFC 2617, June 1999. http://www.ietf.org/rfc/rfc2617.txt

Overview:

Referring (also known as Referrer) URL is a method for protected resource providers to recognize qualified users based on the web page that launches the user to the protected resource. Typically, the user must somehow present credentials in order to access the launching page, and then, when the resource is accessed from that page, the URL of the referring page is transmitted as an HTTP request header. The protected resource provider recognizes this referring URL as matching up with URLs from specific organizations, and allows access.

Implementation is fairly straightforward, requiring the library or leasing entity to communicate the URLs from which its qualified users will access the resource. Not all protected resource vendors provide this service, but more offer it than any other single method of access control, beyond IP-Filtering and Proxy Server.

Workflow:

A user is prompted to login, and upon successful login, is allowed access to a Referring or launching web page.

When the user attempts to access the protected resource from this screen, she is allowed access.

From the service provider's perspective, users are allowed or denied access based on their Referring URL.

Pluses:

- Method is easily and well-understood by both library staff and vendor personnel.
- Vendor can store Referring URLs and easily link them to a purchasing entity (library) for billing, access control, and statistical purposes.
- Accommodates both internal and external users.

Minuses:

- It is a challenge to discover and secure all of the launching points for access to a protected resource (MARC 856 tags in the online catalog, A-Z list web page, link-resolvers, metasearch interface, etc.)
- It is a challenge to keep vendors up-to-date on all Referring URLs used by the library or licensing entity.

- Access is not terribly secure as the Referring URL can be spoofed.
- Not supported by all protected resource providers.

Recap

Referring URL provides a well-understood solution for those who can always predict and secure the web pages from which their users are launched to access protected resources. It provides access for in- and out-of-domain users, and is used by many vendors and libraries. Because of its lack of absolute security and the unpredictable types and numbers of pages from which a user can access a resource, it is not embraced by all.

Shibboleth

Description:	an implementation of OASIS SAML by Internet2 for the exchange of information about users between a web browser and web server in a secure and privacy-preserving manner
Developer:	Internet 2 / MACE (Middleware Architecture Committee for Education)
Specification:	*Shibboleth Architecture Protocols and Profiles*, version 1.3, working draft, 7 August 2005. http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-latest.pdf *Shibboleth Architecture Conformance Requirements*, version 1.3, working draft, February 24, 2005. http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-conformance-latest.pdf *Shibboleth Architecture 1: Technical Overview*, working draft 02, June 8, 2005. http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf Shibboleth v1.3 software and supporting documentation: http://shibboleth.internet2.edu/release/shib-latest.html
Website:	http://shibboleth.internet2.edu/

Overview:

The Internet2 Shibboleth project provides an open source implementation of the OASIS SAML 1.1 standard for web sign-on and user attribute exchange. The Shibboleth System defines a policy framework that supports bilateral trust between organizations and a multilateral trust among organizations such as the higher education community and its partners. Key concepts within Shibboleth include: federated administration; access control based on attributes; active management of privacy; standards-based; framework for multiple, scaleable trust and policy sets; and a standard (yet extensible) attribute-value vocabulary.

Shibboleth leverages campus identity and access management infrastructures (as an Identity Provider or IdP) to authenticate individuals and then sends information about them to the resource site, enabling the Service Provider (SP) to make an informed authorization decision. Because only information (attributes about the person requesting access) is exchanged, the Shibboleth system allows institutions with different technology architectures and security systems to easily collaborate without using proxies or managing thousands of external or transitory accounts. It also simplifies the process of integrating a service, such as access to a licensed library resource with an institution's authentication systems. In this way, it provides a unified service environment by leveraging the institution's single sign-on system to enable access to any Shibboleth SP.

In the primary usage case, when a user attempts to access a resource at a remote domain, the user's own home security domain can send certain information about that user to the service provider site in a trusted exchange. These attributes can then be used by the resource to help determine whether to grant the user access to the resource. The user may have the ability to decide whether to release specific attributes to certain sites by specifying personal Attribute Release Policies (ARPs), effectively preserving privacy while still granting access based on trusted information.

Shibboleth is most often deployed by cooperating parties that come together to form a Federation. When joining a federation, members agree to abide by federation policies. The federation manages and distributes metadata describing the members and provides the Public Key Infrastructure (PKI) certificates needed to validate a member's identity and participation in the federation during Shibboleth transactions. At the time of writing, federations are planned or have been created in the higher education communities in the U.S., Canada, the U.K., France, Germany, Switzerland, Finland, the Netherlands, and Australia.

In the summer of 2005, the Shibboleth project released Shibboleth v1.3. The Shibboleth package is now fully compliant with the SAML v1.1 specification. In addition, this package is also compliant with the Shibboleth profile of SAML (http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-conformance-latest.pdf).

Workflow:

When users first try to access a resource protected by Shibboleth, they are redirected to a service which asks them to specify the organization from which they want to authenticate.

If the user has not yet locally authenticated to a [Web Initial Sign On] service, the user will then be redirected to their home institution's authentication system.

After the user authenticates, the Shibboleth components at the local institution will generate a temporary reference to the user, known as a handle, for the individual and send this to the service provider (SP) site. The SP site can then use the handle to ask for attributes about this individual. Based on these attributes, the SP can decide whether or not to grant access to the resource. The user may then be allowed to access the requested materials. (http://shibboleth.internet2.edu/shib-tech-intro.html, 17-Mar-2005)

Shibboleth is an implementation of the OASIS Security Assertion Markup Language (SAML) Browser profile and as such, all of the transactions must act and behave like a normal browser-client/web server interaction. Since the metasearch engine does not (and should not) know the user's credentials, as soon as the metasearch engine tries to behave like a browser contacting a database engine, the Shibboleth model breaks down—it cannot perform the Web Initial Sign On authentication step. In addition, protocols that are not based on an underlying browser-based HTTP transaction (e.g. Z39.50, XML gateways) are out-of-scope for Shibboleth. (Such a protocol is not out of scope for SAML; the appropriate profile would need to be agreed to and implemented).

Pluses/Minuses: Since Shibboleth does not currently support metasearch, this section intentionally left blank. See Next Steps in Methodology and Recommendation above.

Recap

Although Shibboleth can be used in its present form for the browser-to-metasearch-engine portion of a transaction, it cannot be used from the metasearch engine to the destination search services. What could work in a metasearch environment is a delegated form of a Shibboleth interchange. Ideally, the metasearch engine as an intermediary would obtain the necessary tokens from the user's IdP and transmit them to the SP as part of a yet-to-be-defined delegation protocol.

With the v1.3 release recently completed, the Internet2 Shibboleth project has begun the work of implementing the OASIS SAML 2.0 specification. The Shibboleth project team believes SAML 2.0 features can be used to provide good support of attribute-based access management in many n-tier situations. (An "n-tier" application, meaning "some number of tiers," is one that is distributed among three or more computers in a network. Metasearch can be described as a 3-tier application: the user's browser, the metasearch engine, and the target resource.) Many commercial and open-source implementations of the SAML standard exist and have been demonstrated to interoperate. The Shibboleth team played a primary role in producing the SAML standard and seeks to leverage SAML to develop

multi-tier support so that there can be multiple interoperable implementations, offering the community a wide choice of products and support models.

The team's plan is to identify specific use cases of n-tier scenarios and address them. Shibboleth project members have expressed interest in investigating the NISO metasearch use cases, and working with the NISO initiative to define acceptable approaches to the problems. The Shibboleth project will probably ask to constrain the variety of possible trust frameworks that can be used. The project hopes that the next release of Shibboleth would contain support for the agreed upon NISO metasearch use cases.

SIP / SIP2 (Standard Interchange Protocol)

Description:	a protocol to allow self-service machines in the library to exchange data with the library automation system
Developer:	3M Library Systems
Specification:	*3M Standard Interchange Protocol*, version 2.0, document Revision 2.10, updated September 17, 1998.

Overview:

The Standard Interchange Protocol (SIP) or Standard Interchange Protocol Version 2 (SIP2) was developed originally by 3M to allow self-service machines in the library to exchange data with the library automation system. The standard is the most common method in place for exchange of information to and from library automation systems or ILS. The Version 2 of the standard expanded on the concepts that are part of the original SIP protocol

This standard has been in production for some time and as a result is available in most ILS systems now in production and is even available for some of the older system or systems no longer being supported.

SIP/SIP used a text based messaging system with standard commands and responses to communicate between computer systems. The standard consists of messages that are needed for typical ILS processes. These processes can be loosely defined as two basic modules or components. First, there is user authentication also used by self -service. The second component is to support self-service or self-check out devices. Vendors can implement either or both of the components as will be relevant to their individual systems and requirements.

A system that requests information from another system is considered a Requestor. A system that responds to a request is considered a Responder. Some system may act only as requestors, some as responders only and some will be both depending on the specific vendor's needs or functions of their systems. Individual message request and response arguments within the two components will also be implemented as the vendors' systems dictate.

Primary interest for this committee is the user authentication component of the standard.

NOTE: Realizing that SIP2 was limited in its functional ability to expand, the NCIP standard was developed as the replacement protocol for SIP/SIP2. NCIP expands the message sets and replaces text strings for message interchange and uses XML data structures for data transfer or exchange between different computer systems. (See the NCIP write-up above.)

Workflow:

A user is presented with an appropriate "login" screen. The user enters his/her user-id/barcode and password or credentials. User's credentials are passed from the "application" client (requestor) to target SIP/SIP2 service (or responder). Messages are exchanged per the protocol standard and the user authenticated.

Authentication of the user can results in a "valid" user or an invalid user. Valid users are those that exist in the system; invalid are non-existent users. Additional information about the user is also available and may be passed based on the services and messages supported and requested by the application client or requestor and the target or responder.

Users are allowed or denied access based on the users existence or if further testing is done using additional message to get value judgments regarding the user.

Pluses:

- Authorization of the user is specific based on user credentials.
- Ties well to ILS system for user authentication.
- Can tie authorization to status of user in the ILS (no fines, exceeds fines limits).
- Well established standard available on most ILS systems.
- Implementation should be considered "easy," or the fallout of a system which is tied to an ILS system, thus no extra staff effort is required to build and maintain user records and the user permissions.
- Very good solution to address remote user access and control especially with older or unsupported ILS systems which do not have NCIP functionality available to them.

Minuses:

- Older standard no longer being enhanced with new functionality.
- Cost to library for "service" from ILS or associated vendor. This will be true with most similar "services."
- Data security during system interchanges needs to be addressed.
- As with any remote system or server authentication, the problem of target availability needs to be considered.
- Does not address in-domain users. Libraries wishing to provide "in-domain" terminals, which do not require a user to be authenticated, will need to have some form of IP authentication in place.

Recap

When implementing a user authentication scheme that will be tied closely to an ILS, this protocol would be worth considering for allowing user authentication. Legacy systems will be most likely to support SIP/SIP2. If a choice of SIP2 or NCIP is available for your implementation, it would be best to select NCIP for its added flexibility.

Username / Password

Description:	a method of authentication requiring the matching of a username with its associated password
Developer:	N/A

Overview:

Typically when an institution (as opposed to an individual) is given a username/password (u/p) by a vendor, it is because that vendor cannot support other authentication/authorization mechanisms such as IP Filtering or Shibboleth. Many vendors, however, support u/p (especially for individual subscriptions). The vendor gives the licensee a username and password with which to log into their system. The licensee therefore gives it to the metasearch engine provider.

Workflow:

A user performs a metasearch.

The metasearch engine, with regards to this u/p target, sends along the u/p to the target system.

Assuming a valid u/p, the target then allows the engine access to the system on behalf of the licensee.

Pluses:

- Once configured (if configurable), things are relatively seamless for the end user.
- U/p targets are available from anywhere (on/off a campus).

Minuses:

- There are many ways targets accept u/p; configuring/automating the authentication can be challenging.
- Security can often be a problem, for instance when the u/p is scripted into a URL.
- Single point of failure.
- If a target uses one u/p for each individual user, this method typically cannot work with metasearch systems. When it can it usually means that the metasearch system process will require some interruption while the individual user provides their unique credentials. (However, some systems have the ability for the end user to profile in their individual ids).
- If one u/p is used for the entire site, there is no way to get decent statistics about what user is doing what—all searches are attributed to a single account.
- Expiration of the username or a change of the password can cause interruptions in service until the metasearch system has been updated.

Recap

Username/Password is a fairly low barrier means of dealing with authentication and authorization in a metasearch environment. However, inserting the u/p into the click stream may present challenges, and security can be limited.

X.509 Authentication Certificates

Description:	a mechanism of utilizing public-key certificates for authentication
Developer:	ITU Telecommunication Standardization Sector (ITU-T); Adapted by The Internet Engineering Task Force (IETF).
Specification:	*Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*, ITU-T Recommendation X.509 (03/00) [Note: There are several technical amendments that go with this standard.] Available from: http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-X.509-200003-I *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 3280, April 2002. http://www.ietf.org/rfc/rfc3280.txt
Website:	http://www.ietf.org/html.charters/pkix-charter.html

Overview:

X.509 attribute certificates (ACs) provide a mechanism for demonstrating that the identity associated with a public key certificate has certain security attributes, or, that the identity is a member of certain groups. Such a certificate might indicate that the "bearer" is

- a member of a particular university's community,
- an undergraduate,
- majoring in Chemistry,
- enrolled in 3rd year organic chemistry, or
- any other attributes that may be deemed appropriate by the institution that issued the AC.

X.509 attribute certificates can only be used for authentication and authorization within the context of a deployed public key infrastructure (PKI). This PKI must cover all of the potential users within the client institution, and vendors that support the use of ACs as an authorization method must negotiate a trust relationship between their systems and the client's PKI servers.

Workflow:

A user logs into her workstation and authenticates to the institutional public key infrastructure.

When she attempts to access a resource that uses ACs for authorization, her workstation transmits the appropriate AC to the information provider.

The information provider verifies the AC with the PKI and that the AC contains the attributes necessary for a user to be granted access to the resource. If this is the case, then the user is granted access to the resource. If the AC is invalid, or does not contain the appropriate attributes (for example, the resource is restricted to members of the Law School), then access will be denied to the user.

Pluses:

- Given the existence of an institutional PKI, and vendor support, implementation is simple.

- Vendor database authentication is strongly linked to institutional systems; remote databases seem to be participating in the institution's single sign-on infrastructure.
- Once the trust linkages have been created between the client and vendor PKIs, little further maintenance is required. Revocation of authorization is handled by the institution via the normal user administration processes.
- Access to databases is based on the user's identity. No proxy server is necessary for off-campus/out of branch access.
- The system is highly secure.

Minuses:

- Depends on deployment of a public key infrastructure throughout the client institution as well as PKI support on the vendor's part. This effort is extensive, and typically will not be driven by the library's needs. That is, if there is no pre-existing PKI, the library's authorization needs are likely to be deemed insufficient reason to deploy one.
- The ability to provide "guest" accounts for in-library users who are not part of the institution's user community (e.g. walk-in general public access to an academic library) may be limited or difficult to achieve within the strong security environment of the PKI.

Recap

If an institution has already deployed an X.509 public key infrastructure, if all members of the institutional community are enrolled in the PKI, and if the vendors support X.509 attribute certificates, then implementing X.509 ACs for database authorization is simply a matter of establishing a trust relationship between the institution and the vendors. Once established, such a PKI and web of trust between the institution and its service providers ensures seamless, convenient access to the institution's information resources on the part of the institutional community, regardless of the community members' physical locations.

Of course, few institutions, and even fewer vendors, have implemented X.509 broadly, and the costs of deploying a full PKI are well beyond the resources of most institution's libraries. Nor is it likely that the authorization needs of the library will be consider sufficient reason to begin such a large project, although they will probably be considered one of several significant stakeholders in such a project if it was being considered by the institution. For certain types of institutions, like the public library, it is unlikely that an X.509-based authorization system will ever be feasible, since there is no parent organization that can afford, or mandate the use of, a PKI that includes all members of the institution's constituency.