Introduction

OCLC has completed its investigation of the two proposed electronic access protocols for the ILL Policies Directory. The first is *X.500*, a directory protocol standard developed by the International Telecommunications Union (ITU). The second is *Lightweight Directory Access Protocol* (LDAP), a derivative of the X.500 protocol developed as an open source project by the Open System Interconnection – Directory Services (OSI-DS) and the Internet Engineering Task Force (IETF).

In order to proceed further on the project, a consensus must be reached concerning which protocol to implement. In preparation for the April 2, OCLC/IPIG conference call this document presents and compares both protocols and outlines OCLC's preferred approach.

X.500 Protocol

The X.500 protocol was first approved in 1988 and then enhanced in 1993 under the auspices of the International Telecommunications Union (ITU). Its purpose was to provide an international standard for directory systems.

Model

The X.500 protocol architecture consists of a Client-Server communicating via the Open Systems Interconnection (OSI) networking model. The Client is called the Directory Service Agent (DUA) and the Server is called the Directory System Agent (DSA).

There are two sub-protocols used to communicate between systems. The communication protocol between a DUA (Client) and a DSA (Server) is called the Directory Access Protocol (DAP). The communication protocol between one DSA (Server) and another DSA is called the Directory System Protocol (DSP). The X.500 uses the DSP sub-protocol to give a "distributed" and "global view" of the data. That is, not all the data is stored on one server but distributed among multiple servers. However, when a client accesses a X.500 system via DAP, the data is gathered from one or several servers using DSP and presented as one global view of the data.

Data

Data within the X.500 architecture is stored in *Objects* and *Attributes*. This is analogous to *Tables* and *Columns* in database parlance or *Records* and *Fields* in file parlance. Objects are identified by unique identification numbers called Object Ids. or OIDs. Attributes are contained within objects and represent specific data elements, such as name, address, etc.

The data is accessed via a *Directory Information Tree* (DIT). A DIT is a hierarchal structure that consists of a root with many nodes or branches (similar to a file directory structure). For example, a telephone DIT would consist of a root with nodes for each country that contains nodes for each area code that contains nodes for each phone number.

Data is encoded within X.500 in Abstract Syntax Notation (ASN.1), Basic Encoding Rules (BER) format.

Data is updated (added, changed or deleted) by transactions described by the protocol.

Security
The X.500 protocol uses the X.509 Public Key Infrastructure (PKI) specification (i.e. digital certificates) for authentication.

Replication
The X.500 protocol provides for database replication. This means that directory data can be replicated or copies distributed to multiple servers for the purpose of load distribution and system contingency.

Session
A typical X.500 session may proceed like the following:

- Client: Connects and requests access to the server; this is called the *Binding* operation.
- Server: Server authenticates the client and completes the binding operation.
- Client: Requests a service from the server, such as search for an entry in the directory, and presents any parameter data.
- Server: Performs service and may connect to another X.500 server then communicates a response.
- Client: Receives response and unbinds or terminates the connection.

Specification
The X.500 protocol is described in a series of specifications:

- X.501: *The Models* – Concepts and models.
- X.509: *Authentication Framework* – Authentication of clients and servers.
- X.511: *Abstract Service Definition* – Functional services (i.e. search, modify, etc.)
- X.518: *Procedures for Distributed Operation* – Operations that span multiple servers.
- X.519: *Protocol Specifications* – Describes the overall X.500 protocol and the sub-protocols: Directory Access Protocol (DAP), Directory System Protocol (DSP), Directory Operational Binding Protocol (DOP), and Directory Information Shadowing Protocol (DISP).
- X.520: *Selected Attribute Types* – Attribute type or data element definition.
- X.521: *Selected Object Class* – Object class definition.
- X.525: *Replication* – Replication operation among multiple servers.

Lightweight Directory Access Protocol (LDAP)

To simplify the complex X.500 protocol and lessen the heavy load on clients to support Directory Access Protocol (DAP), *Lightweight* Directory Access Protocol (LDAP) was developed. Originally, LDAP was designed to be an alternative to the client side protocol of X.500 (DAP). It enabled clients to use simple TCP/IP networks to connect to intermediate servers. These intermediate servers would then connect to X.500 servers using DSP over OSI networks. The LDAP protocol was later expanded to also replace the server side (DSP) of the X.500 protocol.

Model

The LDAP protocol architecture consists of a Client-Server communicating via the TCP/IP networking model. Normally LDAP servers are independent and only communicate with LDAP clients.

Instead of presenting the directory service as a global view like X.500, LDAP uses a referral mechanism. When a client requests data from an LDAP server that does not contain the data requested, the server responds with another URL of a server that does contain the information, which is similar to the World Wide Web.

Data

The LDAP architecture, like the X.500 architecture, stores data in Objects and Attributes. However, LDAP identifies Objects by a unique name instead of a number. LDAP also similarly uses a Directory Information Tree to access the information and ASN.1/BER for encoding. However there is a current effort to use XML encoding instead of BER.

LDAP adds another feature called the LDAP Data Interchange Format (LDIF). It is a method for LDAP clients and servers to communicate schemas and updates via a text-based format. This enables LDAP users to easily discover the data layouts of unknown schemas and to perform batch updates to a directory.

Security

The LDAP protocol utilizes the Simple Authentication and Security Layer (SASL) specification for identification and authentication. The SASL layer is flexible in that it enables other security *mechanisms* (such as Kerberos or GSSAPI) to be implemented or *plugged in*. Since LDAP uses TCP/IP it can be transported over Secure Socket Layer (SSL) connections.

Replication
The LDAP protocol, like X.500 also provides for database replication. That is, updates are replicated via the protocol to mirror LDAP sites.

Session
A typical LDAP session may proceed like the following:

- Client: Connects and requests access to the server; this is called the *Binding* operation.
- Server: Server authenticates the client and completes the binding operation.
- Client: Requests a service from the server, such as search for an entry in the directory, and presents any parameter data.
- Server: Performs service and communicates a response or a referral to another LDAP server.
- Client: Receives response and unbinds or terminates the connection and may connect to a referred server.

Specification
The LDAP protocol is described in a series of specifications:

- RFC 2251: *LDAPv3 Protocol* – LDAP protocol definition.
- RFC 2252: *LDAPv3 Attribute Syntax Definitions* - Attribute type or data element definition.
- RFC 2253: *LDAPv3 UTF-8 String Representation of Distinguished Names* – UTF-8 character encoding of directory entry keys.
- RFC 2254: *The String Representation of LDAP Search Filters* – Query mechanisms for use in URLs and APIs.
- RFC 2255: *The LDAP URL Format* – Uniform Resource Locator construction.
- RFC 2222: *Simple Authentication and Security Layer (SASL)* – Transport Layer to plug-in security mechanisms.

Heavyweight vs. Lightweight

Which is better, the heavyweight represented by X.500 or the lightweight represented by LDAP? The protocols are very similar in data structure but vary in these areas:

- LDAP uses the ubiquitous TCP/IP layer, where X.500 use the less common OSI layer.
- X.500 presents the Directory information as a single monolithic view, where LDAP presents the Directory information as a main view with possible referrals.

Performance

Performance can be measured in two ways:

1. Speed of a transaction - LDAP would consistently have the advantage because of its lightweight structure.
2. Extent of a transaction – LDAP would still have the advantage if the extent of a transaction were local to the LDAP server. However if the target of transaction were on another server, X.500 would have the advantage. This is because LDAP would return a referral response to the Client and the Client would in turn forward the transaction to the referred server. The X.500 protocol would automatically forward the request.

Scalability

In general scalability is not too much a factor between the protocols but is more dependent upon the vendor product. Some basic LDAP products have smaller capacities but most LDAP and X.500 products can support millions or billions of entries with several thousand users.

Cost

Cost is based upon the server product, implementation and support. LDAP has the definite advantage in this arena. The LDAP products are either free, included with the operating system or less expensive then X.500 servers. LDAP is also cheaper to implement due to free document specifications and faster installation times. LDAP is more ubiquitous and therefore has an abundance of resources in the form of books, web sites and forums. X.500 support is available usually only via consultants or paid support contracts.

What about both?

A popular implementation is to use X.500 as the *backbone* for the directory and use LDAP as the front-end interface. This allows a better performing Client with a more integrated Server network.

Preliminary Conclusions

Based on our investigation and the lack of available X500 solutions, OCLC believes it can implement the Policies Directory faster and more easily using LDAP. LDAP fully meets the searching and update needs of the IPIG and should the National Library of Australia choose to implement the proposed X500 tree, the OCLC-hosted directory can be added to the network. Thus, LDAP may be the more practical solution at this stage.

Below is a list of vendors that have been contacted by OCLC. While a specific LDAP vendor has not be determined yet, we are confident we will be able to find a vendor that will provide the needed functionality. We are actively working with the vendors currently.

Companies Investigated

LDAP
IBM
AOL
Oracle

X500
Computer Associates

After an initial meeting with Computer Associates, OCLC has tried to correspond with Computer Associates repeatedly to no avail.
This lack of responsiveness and a lack of an Aix based system, has led OCLC to not choose Computer Associate's eTrust product for an X500 solution.

Unfortunately, OCLC can find no other X500 based products available in the marketplace. If a member of the IPIG community knows of another vendor, OCLC would be glad to investigate that vendor for an X500 solution.

Sources

- Directories and X.500: An Introduction; http://www.nlc-bnc.ca/publications/1/p1-244-e.html
- How to build an Enterprise Directory with LDAP and X.500; http://www.messagingdirect.com/publications/IC-6040.html
- Understanding and Deploying LDAP Directory Services; Timothy A. Howes, Gordon S. Good, Mark Smith; Macmillan Publishing, USA; 1998.
- LDAP: A Next Generation Directory Protocol; http://www.intranetjournal.com/foundation/ldap.shtml
- Which Directory Offers the Best LDAP Server?; http://developer.novell.com/whitepapers/ldap/
- Directory Services Markup Language (DSML) – LDAP & XML; http://www.dsml.org