

Enterprise Directory Services

Technology Components

Technology Component 1: User Agent

Access to the directory is through a user agent, also known as a user interface or a client. This does not refer only to a user accessing the directory, it includes whatever mechanism an application uses to communicate with the directory. This can be the traditional client application a user is presented during a login session. It can also be a service, either built-in or external, to an application that passes, for example, LDAP calls.

Technology Component 2: Access Protocols

The user agent requires a protocol to communicate with the Directory System Agent (DSA) to request services (See Figure 1-7). User Agent services such as Read, Modify, Search, List, etc. request the DSA, using filters, to scan the Directory Information Base (DIB) and return the results. The user agent always initiates this communication. There are many access protocols. Some examples include Novell Directory Access Protocol (NDAP), X.500 Directory Access Protocol (DAP), Open Database Connectivity (ODBC), and Java Naming and Directory Interface (JNDI). The Lightweight Directory Access Protocol (LDAP) has recently become the most popular and universally supported access protocol. As the name implies, LDAP is a lightweight front-end to many directories from many vendors. Initially LDAP was designed to increase performance and conserve bandwidth while accessing X.500 directories. Most directory and application vendors have chosen LDAP as an access protocol to communicate with their own directory or for their application to communicate with an external directory.

Technology Component 3: Directory System Agent (DSA)

This is considered the directory server software. It performs all reads, writes, deletes, modifications, etc. to the directory information base (DIB) on behalf of the user agent via the access protocol.

Technology Component 4: Directory Information Base (DIB)

The DIB is a database where directory information and objects are stored. Portions of this information can be distributed and replicated to other servers in the enterprise, in order to enhance performance and provide fault tolerance. Performance is improved by servicing requests for data from the DIB that is closest to the source of the request. Fault tolerance is achieved by replicating the data to multiple locations. If a DIB is unavailable, another server holding a replica of the information can then service the request. This strategy is found in X.500, NDS, and pure LDAP solutions. Although the techniques differ slightly, the concept is basically the same in each solution.

Technology Component 5: Chaining or Referral Protocols

Chaining and referral protocols are the communications between DSAs, allowing a request to be passed from one DSA to another to obtain the information. Therefore, information in a directory can be accessed without needing to know the exact location of that specific piece of information. When a DSA receives a request, it queries the DIB. If the DIB does not contain the information requested, the DSA can then pass the request to another DSA to perform the lookup to its DIB and so on until a DSA can be found that has the requested information in its DIB. There are two techniques to accomplish this: chaining (found in X.500 directories using a Directory System Protocol - DSP) and referrals (found in Novell's NDS and Netscape's LDAP directory).

Technology Component 6: Shadowing and Replication Protocols

To provide fault tolerance, multiple directories can be replicated to each other. This is usually only achievable with like directory types from the same manufacturer. Distributing copies of directory information to other servers allows another server to service requests when the destination server has become unavailable.

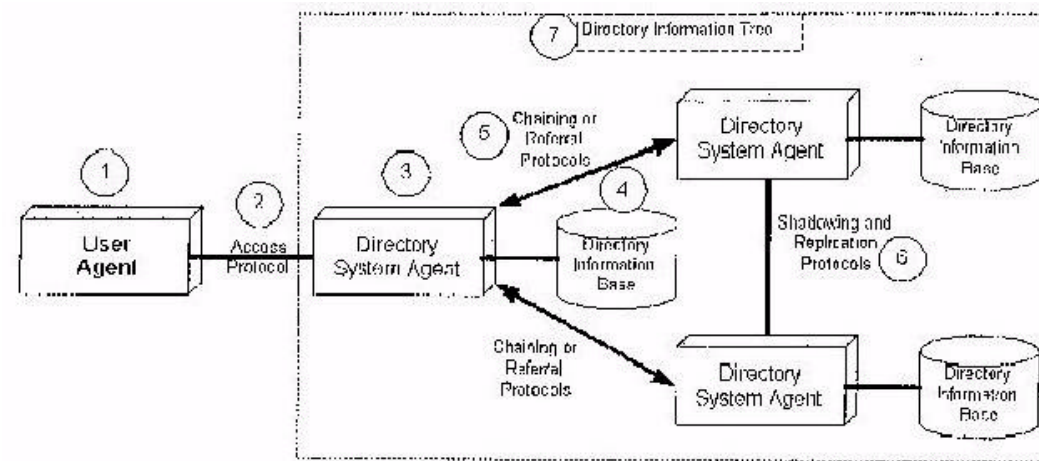


Figure 12-23. Directory Architecture

Technology Component 7: Directory Information Tree (DIT)

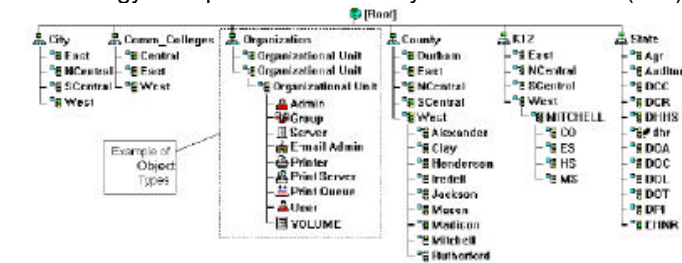
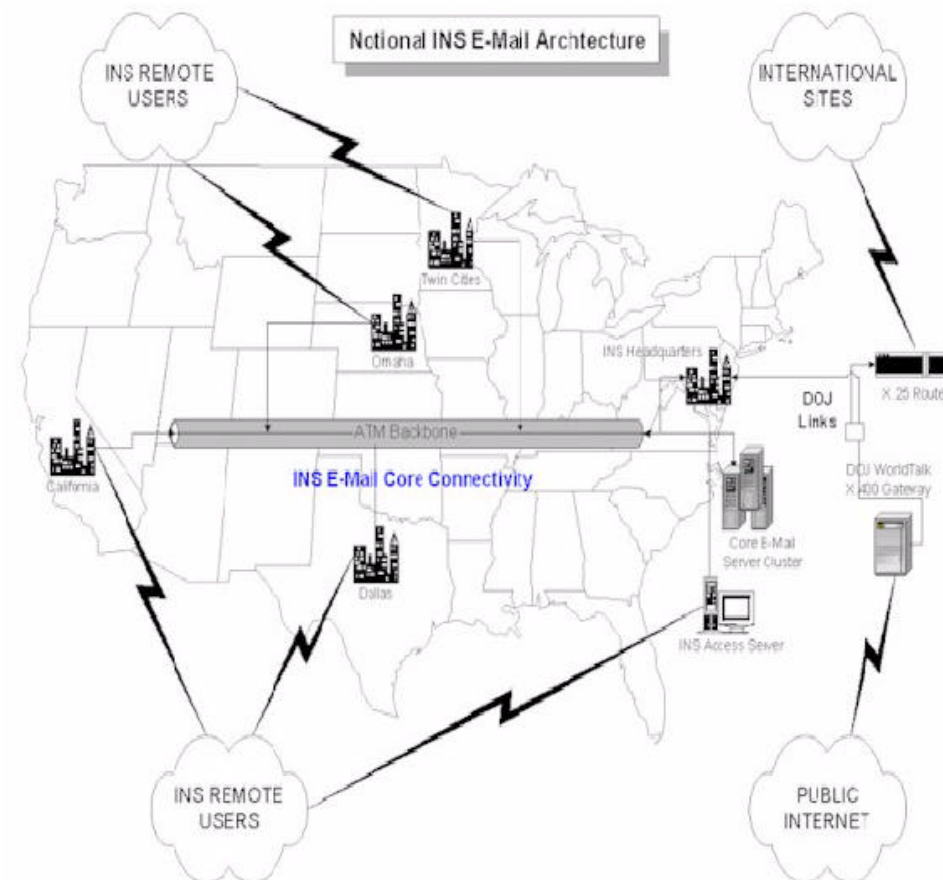
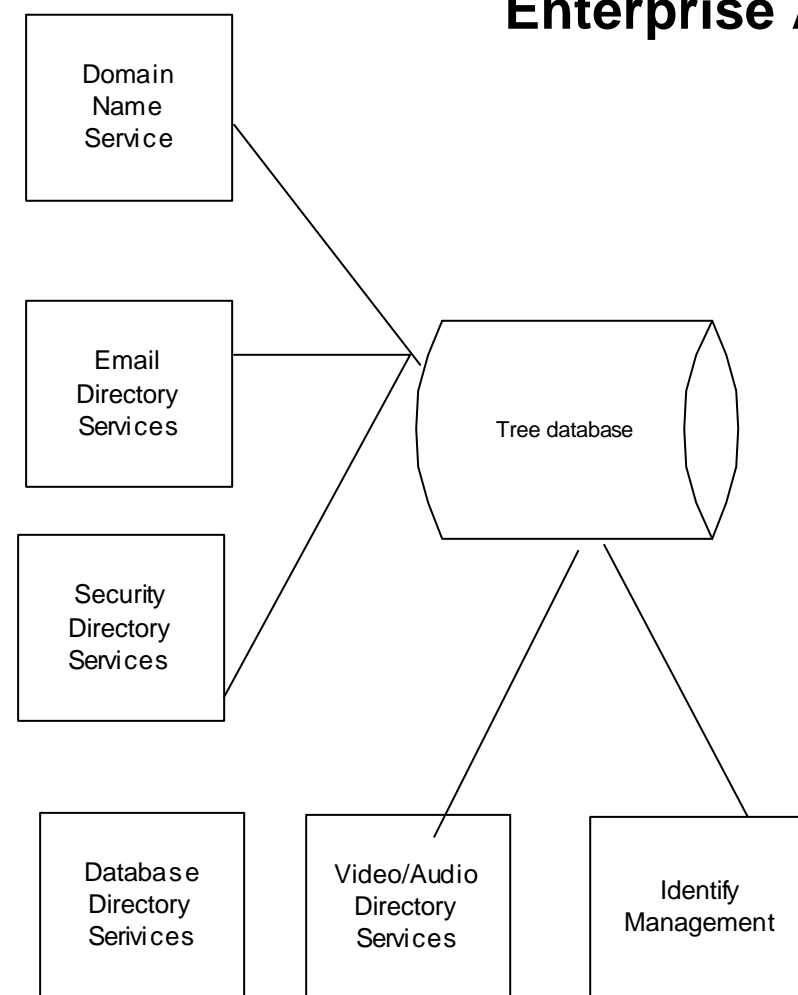


Figure 12-24. Directory Tree Hierarchy

The Directory Information Tree is a logical, hierarchical, representation of the enterprise directory as an inverted tree. Figure 12-24 is an example of a directory tree. This can be made up of millions of objects that, without a facility like this, would not normally be linked in any other way. This allows users and administrators to navigate through the tree without regard to the fact that they are spanning potentially hundreds of servers. Access controls determine what can be accessed or even seen in the directory tree.

Enterprise Architecture

The term "enterprise directory services" refers to central stores of information about people associated with an institution. This centralized data is usually authoritative, meaning that the set of people responsible for entering data correctly and maintaining its integrity are the only sources for that data. For example, an enterprise's Human Resources department has the responsibility for correctly maintaining information about who is an employee of that enterprise, while the telecommunications department is responsible for assigning telephone numbers, and the computer services department is responsible for assigning an email address. If each of these three departments - Human Resources, telecommunications, and computer services - maintain separate lists, each department winds up duplicating information held in other areas. The inevitable result is that Human Resources has a correct list of names, but out of date or incorrect email and telephone numbers; likewise, telecommunications may not be aware that someone has left the enterprise or changed their last name.



Standards

4.2.3 LDAP

The Lightweight Directory Access Protocol (LDAP) is a standard describing access to directory services. LDAP was derived from the OSI (Open Source Initiative) Directory Services model X.500 known as "DAP" (Directory Access Protocol). DAP runs over the OSI network protocol stack. LDAP's overall data and namespace model is quite similar to X.500. The major difference is that the LDAP protocol is designed to run directly over the TCP/IP stack, which makes it "lightweight". The current version, LDAP V3 (IETF RFC 3377), includes important security enhancements.

LDAP is a protocol, not a database. A protocol describes messages used to access certain types of data. It is possible to store data in a variety of backend data stores and use the LDAP protocol as a standardized querying interface. LDAP also provides a data model that standardizes the naming and organization of the data. Finally, LDAP servers are designed to optimize read functions, since the main purpose of this service is to answer queries regarding relatively non-volatile data.

The LDAP information model structures data as a tree - the Directory Information Tree (DIT). An entry in the DIT corresponds to a node in the tree, and contains information about an object Class. ObjectClasses have both required and optional attributes, and attribute typing defines the encoding and matching rules to be used during searching. The LDAP information model is also called the LDAP schema.

LDAP provides globally unique naming. By following a path from a node back to the root of the DIT, a unique name is built and is referred to as that node's distinguished name (DN). Figure 9 shows an example DIT. Following a path from the gray dotted arrow to the base DN, the unique distinguished name "uid=jhc,ou=people,dc=uab,dc=edu" is built.

Figure 9: DIT Example

Access via the LDAP protocol is implemented by bindings (authentication), queries, and updates. Authorization to access data can be managed using access control lists (ACLs).

Products

LDAP Directory Server Implementations

A variety of LDAP directory server implementation choices are available. There are a number of factors to consider when selecting your implementation, including price, familiarity with supported platforms, or integration with existing infrastructure.

OpenLDAP is an open source LDAP Directory Server implementation. Appendix D.2 describes how to install and configure OpenLDAP for use with H.350.

SunOne (iPlanet) Directory Server software began under the name 'iPlanet' (formerly Netscape) and has since moved to 'Sun One' under new management. This is a popular product and is relatively easy to install and configure. Appendix D.1 describes how to install and configure SunOne iPlanet for use with H.350.

Active Directory is Microsoft's proprietary directory service for Windows 2000 and 2003. Active Directory supports the LDAP protocol. Appendix D.3 describes how to install and configure OpenLDAP for use with H.350.

Novell Directory Service (eDirectory, formerly called NDS) is Novell's proprietary directory service for Netware. NDS supports the LDAP protocol.

Practical Enterprise Architecture		Technology Architecture									
		Directory Services									
		<table border="1"> <tr> <td>SUBJ</td> <td>FSCM NO</td> <td>DWGNO</td> <td>REV</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table>	SUBJ	FSCM NO	DWGNO	REV					
SUBJ	FSCM NO	DWGNO	REV								
		John Wu		SPEC	OF 13						