

Directory Integration

Using directory services to successfully manage partner relationships

By Paul Sholtz
New Architect Magazine
July 2002

Integrating your information systems more closely with strategic business partners' can be beneficial. However, the key to any successful partner integration project is effective identity management, along with secure provisioning of user access rights. Businesses must control who has access to what information, and make sure that each partner organization can only access the information that it has privileges to use.

Directory systems help provide identity management and secure user access provisioning. Thus, effective directory integration and management is key for any large-scale partner integration project.

Partner Relationships

Business collaboration is driven by several market forces. Supply chain management is one such example—a company that integrates with its immediate upstream suppliers can manage inventory assets more efficiently by confirming orders, pricing information, and shipping

dates. Private marketplaces, business-to-business exchanges, and other collaboration hubs are also good examples. These systems let participating organizations extend their relationships with current suppliers and dramatically reduce the costs associated with procurement administration.

The growing availability and maturity of Web services is also making it easier to integrate information systems between business partners. Companies are slowly moving their applications out from behind the firewall and onto the edges of their networks, where they can participate in dynamic, Internet-based transactions with customers and other business partners. Web services are rapidly transforming our notion of what an application really is. What was once a static clump of binary code is now a dynamic set of federated services specifically tailored to each user. The number of organizations using Web services to dynamically collaborate with business partners is small, but steadily growing. Companies like Merrill Lynch, Thomas Wiesel Partners, and Wachovia Securities have already had some notable successes.

How directory integration impacts your partnership management initiatives depends on the nature of your relationships. If you're only providing one set of services to a business partner, chances are your partner will only be authenticating against one directory. This makes system administration a relatively straightforward task, and directory integration is probably unnecessary in this case.

Directory Integration

As your business relationships grow, however, you may want to make additional information services available to your partners. The applications that provide order confirmation, pricing data, and shipping date information services may all run on different systems in your enterprise, and users may need to authenticate against a different directory access with each service.

In such cases, directory integration could go a long way toward providing your business partners with a more pleasant end-user experience, and would save your enterprise time and money on system administration. Moreover, there may be businesses whose relationships with your company are characterized by several different roles—they may provide products for your supply chain in one role, and they may be a customer of your products and services in a different role. In these cases also, users may be accessing a variety of information services provided by your enterprise and authenticating against several directory services.

Directory Basics

A directory is a specialized type of database. Just like with a normal database, you can use directories to store and retrieve nearly any type of information. Directories, however, are usually optimized for being read from rather than written to. They are best suited for storing relatively static data, such as usernames or X.509 digital certificates.

A directory isn't typically deployed just as a single, stand-alone server, but rather as part of a broader network service. The notion of a directory service encompasses the directory itself along with a specific network protocol that clients use to access the directory. Directory services may also include a replication scheme for distributing data among participating servers and provide guarantees about the availability and security of the service on the network.

Directory services are nothing new, anyone who has been on the Internet has used a directory service at one time or another. DNS and email are perhaps the best-known examples.

DNS is a good example of a global directory service. It's distributed across a global network of cooperating machines and the DNS namespace is uniform. Directory services with a uniform namespace provide clients with the same view of the data no matter where the client is in relation to the service. In other words, no matter where you are on the Internet, DNS returns the same information in response to a query for `newarchitectmag.com`.

DNS also tolerates temporary inconsistencies when directory information is updated—up to 48 hours in the case of some zone transfers—as long as the directory information replicates to all participating servers and comes into sync eventually. Tolerance of temporary data

Directory Integration

inconsistencies during updates and replication is another feature that distinguishes directories from traditional data management systems.

Powerful as it is, DNS is only one of a whole range of directory services commonly used on computer networks. The most common directory service used in the enterprise for managing user identities, credentials, and access rights is a standard called the lightweight directory access protocol (LDAP).

LDAP began as a low-cost, PC-based front end for accessing X.500 directories. Based on the ISO/OSI networking standards, X.500 is an exhaustive, heavyweight directory model that defines a global namespace for addressing and locating objects in the directory. It also consists of a network protocol, known as Directory Access Protocol (DAP), for querying and updating directory information. Although comprehensive, X.500 is difficult to implement and maintain because of its complexity, and full-blown DAP clients were often too large to install and run on smaller computer systems like PCs.

LDAP was developed at the University of Michigan, with support from the Internet Engineering Task Force, in the early 1990s to make it easier for PCs and other modest computer systems to browse and access the information stored in an X.500 directory. One of the biggest obstacles to acceptance of X.500 and simplifying access to the directories involved the DAP protocol—specifically DAP's

reliance on the OSI network protocol stack. Due in large part to its complexity, the OSI stack never really gained widespread acceptance. LDAP simplifies directory access by letting clients connect to directory services directly over the much more widely adopted TCP/IP stack, making the directory services available on LANs, WANs, and the global Internet. LDAP also dropped many of the more esoteric and infrequently used features of the DAP protocol, which made the specification much easier to implement in client software.

So on one level, you can think of LDAP as a lightweight information access protocol much like many other common Internet protocols. You can use LDAP to access and browse X.500 directory information (by default, on port 389) much like you can use HTTP to access and browse Web documents (by default, on port 80).

But LDAP is more than just a lightweight version of the bulky DAP networking protocol. LDAP inherits its data model essentially unchanged from X.500. The LDAP directory service model is based on the concept of entries. An entry is a collection of descriptive attributes. Each attribute has a name, type, and one or more values. For example, the attributes that describe a person might include the person's name (common name, or cn in our example), telephone number, and email address.

For example, the entry for Paul Sholtz might have the following attributes:
cn: Paul Sholtz

Directory Integration

email: paul@doublegemini.com
telephoneNumber: 650-555-1212

LDAP arranges directory entries into a hierarchical, tree-like structure, starting at a root and then branching down into individual entries. At the top levels of the hierarchy, entries represent large organizations. Underneath these, the directory entries might represent smaller organizations. The hierarchy usually ends with entries for individual people or resources. See "Hierarchy of Entries in an LDAP Directory Service" for more information.

Each entry is uniquely identified by a distinguished name. A distinguished name consists of a unique identifier specifying a particular entry at some level of the hierarchy (for example, in "Hierarchy of Entries in an LDAP Directory Service," paul and john are different user IDs that identify different entries at the same level) and a path that traces the entry back to the root of the tree.

The distinguished name for the paul entry might look like this:

```
uid=paul,ou=employees,o=doublegemini.com
```

Here, uid represents the unique identifier that specifies the entry, ou represents the organizational unit in which the entry belongs, and o represents the larger organization containing the entry.

Although many commercial directory products on the market today support LDAP—most notably Active Directory from Microsoft—it isn't unusual to find legacy directories scattered throughout the enterprise. NT 4.0 domains, NDS, and Lotus Notes are all common examples. Understanding LDAP nevertheless gives us a good picture of what directories are good for and where their limitations lie. Directories can manage relatively static information, such as user authentication credentials, but they weren't designed to replace relational databases, traditional file systems, or DNS.

Enterprise Challenges

Directories are the lifeblood of the modern enterprise. They store critical user account and identity information for enterprise applications, services, network operating systems, and messaging systems. Directories also often store network configuration information for computers, printers, routers, and corporate security policies.

Yet managing directories can prove a serious challenge for an IT organization—largely because many enterprise applications ship with their own proprietary authentication subsystems. Just because you authenticate yourself to Windows2000 doesn't mean that Oracle Financials knows who you are (or how to access your authentication credentials, which are stored in Active Directory).

Forrester research and Gartner Group recently concluded in separate studies that the average Fortune

Directory Integration

1000 company maintains over 181 separate directories. Needless to say, this creates a significant burden for system administrators, who must duplicate their efforts to create, modify, and remove directory information in multiple locations. It also impedes their ability to maintain directory data integrity when updates occur. Companies that efficiently integrate their directories stand to realize substantial competitive gains. The Burton Group estimates that a 25,000-user company can spend \$360,000 annually on directory changes if the company has only seven user directories.

Directory Integration Techniques

Consistent identity management is important within a single enterprise, and it's essential if you're scaling your operations out beyond the firewall into partner organizations. There are four important techniques for integrating and unifying the information stored in directories: virtual directories, synchronization, metadirectories, and information brokering. These techniques range from simple system administration aids to full-blown data synchronization services that help you maintain the integrity of directory information. The integration solution that's right for your organization depends on what your business requirements are.

Virtual Directories: These are a useful tool for system administrators. The console interface is linked directly to the managed directories on the network, letting administrators manage multiple directories through a

centralized point of control. Although helpful, virtual directories are generally used only for system administration purposes and don't typically provide the more complex management services, like directory synchronization, that help maintain data integrity.

Microsoft offers a directory development API called the Active Directory Services Interface (ADSI) for accessing and managing directory products from different vendors. Independent software vendors can use it to build virtual directories. ADSI is a set of COM programming interfaces that integrate with any directory service that offers an ADSI service provider. The NT 4.0 domain directory, NDS, Lotus Notes, and LDAP directories integrate with ADSI. See "ADSI Integration" for more information. Some popular virtual directory products on the market include Entevo's DirectAdmin, Computer Associates's Unicenter TNG Directory Management Option, and IBM's Tivoli User Administration.

Synchronization: As mentioned, replication is an important function provided by the directory services. If directory information is updated at one server in a distributed cluster, the changes need to be replicated across all of the other participating servers in the cluster. This is fine if all of the directory servers in your organization use the same general replication protocol—LDAP, for instance. But what happens when you make changes in an NT 4.0 domain that need to be replicated in an LDAP directory? That's where synchronization comes in.

Directory Integration

Synchronization is the automatic update process that ensures that directory information is consistent across all participating directories in your organization—even if they use a different data model and schema. Synchronization differs from replication in that it can provide simple data translation services as well, which ensures that information is updated uniformly across all directories in the enterprise, regardless of their format. For instance, you can use synchronization to coordinate directory information stored in LDAP and NT 4.0 domains.

You can use two different models to synchronize directories: one-to-one and one-to-many. The one-to-one method only synchronizes two directories at a time, and is implemented using either one-way or two-way update semantics. In one-way synchronization, one directory serves as the master source of information and always propagates update information directly into the second directory. The second directory never propagates update information back to the first directory. In two-way synchronization, the two directories can update each other as necessary.

Directory Service Manager for NetWare (DSMN) from Microsoft uses one-way synchronization to coordinate NT domain information with NetWare. DSMN propagates changed user account information from an NT domain into a corresponding NetWare Bindery directory. By contrast, Netscape's Directory Server uses two-way synchronization to coordinate user account information

with NT. Changes in NT are propagated to Directory Server, and changes in Directory Server are propagated back to NT.

One-to-many synchronization scales beyond two directories. You choose one directory to serve as the centralized enterprise directory, and configure the other directories on the network to route information into or out of the enterprise directory as needed. See "One-to-Many Synchronization" for more information.

Metadirectory: A problem with synchronization in a large enterprise environment is that the same user may use different logons for separate directory systems throughout the enterprise. For instance, I may authenticate to NT as psholtz and authenticate to NDS simply as paul.

Having multiple identifiers for the same user can be a problem for synchronization programs. Synchronization often requires a single unique identifier for each user that's accepted across all connected directories. For instance, in the above example, two-way synchronization would simply create a new account in NDS called psholtz and a new account in NT called paul, rather than synchronizing the appropriate information between the existing accounts. Metadirectories can address this problem. These are similar to the centralized enterprise directories described in the one-to-many synchronization technique above, except that user objects are imported into the metadirectory using a technique called joining.

Directory Integration

Joining can correlate the user attributes of one particular user from each of the participating directories with the corresponding user object in the metadirectory. The metadirectory then holds all of its underlying directories' user attributes.

Joining permits some relatively sophisticated synchronization policies. For example, you can synchronize one set of user attributes with one participating directory (such as NT) and synchronize a completely different set of user attributes with another participating directory (such as NDS).

Information Brokering: This technique is more about optimization than integration. Gathering all of the directory information from across the enterprise into one place can make metadirectories top-heavy. Avoid this problem by keeping some information locally in the underlying directories and propagating that information to the metadirectory only when the metadirectory needs it. The metadirectory therefore doesn't store information that it doesn't need, saving replication and synchronization costs between directories. This technique is called information brokering.

Be aware that information brokering can cause problems if you don't deploy it carefully. The information broker can bog down the network with search requests if there isn't enough data stored directly in the metadirectory.

Emerging Trends

Identity management and directory services are an exciting area of IT innovation right now, and I've only barely scratched the surface of this topic here. Some other emerging trends and technologies that you should bear in mind are password synchronization and single sign-on (SSO) systems. Also, watch for Internet-scale identity management systems like Microsoft Passport and the Liberty Alliance. These types of systems may dramatically change the way users authenticate themselves to enterprise applications and services in the near future. Keeping on top of these emerging technologies will help you identify new ways to create value for your business partners while cutting your own system administration costs.

-

Contact Paul Sholtz
Phone: 917.438.7087
E-mail: paul.sholtz@doublegemini.com

Directory Integration

Glossary

ADSI: A set of Microsoft ActiveX controls that abstract the capabilities of directory services from different network providers to present a single set of directory service interfaces for accessing and managing network resources.

Information Broker: Technique used to optimize performance on large metadirectories by storing some information locally in participating directories. The metadirectory will search for and locate the information in the local directory if necessary.

Join: The process of joining disparate enterprise directory services into a single, centralized metadirectory. Join is useful in environments where individuals use authentication credentials that aren't consistent across the participating directory services.

Open Systems Interconnect (OSI) Network Model: The standard model for networking protocols and distributed applications defined by the International Organization for Standardization (ISO) in 1984. The OSI network model defines seven layers: physical, data link, network, transport, session, presentation, and application. The layers provide clearly defined functions that can improve internetworking connectivity between computers and other network-enabled devices.

X.500: X.500 is the OSI directory service. It defines a comprehensive directory service, including an information model, a namespace, a functional model, and an authentication framework. X.500 also defines the Directory Access Protocol (DAP) that clients use to access the directory. DAP is a full OSI protocol that contains a large amount of functionality, much of which goes unused by most applications.

X.509: A common format for specifying the content of digital certificates. Digital certificates are digitally signed statements by an independent and trusted third party vouching for the identity of a particular principle. They're commonly used for authentication purposes in client-server applications.