THE Open GROUP

White Paper: Assuring Interoperability for The Directory-Enabled Enterprise

The views expressed in this Open Group White Paper are not necessarily those of any particular member of The Open Group.

Document Number: W902

Any comments relating to the material contained in this document may be submitted to

The Open Group
Apex Plaza
Forbury Road
Reading, Berkshire
RG1 1AX, England
Tel: +44 1189 508311
Fax: +44 1189 500110

Executive Summary

This White Paper sets out The Open Group's approach to delivering assurance of interoperability of Directory products. It describes what has been achieved to date, and what is planned for the future.

Directory is a powerful tool. It can provide a standard repository for data shared by operating systems and applications. It can give external organizations controlled access to selected information. It can help simplify enterprise administration and achieve massive cost savings. This all adds up to a vision – the vision of The Directory-Enabled Enterprise.

The key to realizing that vision is interoperability. Increasingly, Directory servers are bundled with other products such as databases and operating systems. This, coupled with the trend to Enterprise decentralization, makes a single-supplier policy for Directory procurement completely impractical. The typical enterprise will have a hundred or more Directory servers from perhaps ten different vendors.   Customers and vendors want the servers to work with each other, and want the enterprise's applications to work with them all.

The Open Group can make a major contribution by delivering assurance of interoperability through testing and certification. In order to succeed in the marketplace, product vendors must convince buyers that their products will indeed interoperate and perform as advertised. This customer confidence is essential, but difficult to achieve. By delivering assurance of interoperability, The Open Group will create the necessary customer confidence. This in turn will grow the market for all vendors of LDAP server products and LDAP-enabled applications.

In co-operation with the Directory Interoperability Forum, The Open Group has defined a testing and certification program to deliver assurance of directory interoperability. The first deliverable of that program – The Open Brand for LDAP 2000 – is to be launched at the June 2000 Open Group Conference.

This White Paper is in three parts.

The first two parts set out The Open Group's approach by explaining our current understanding of The Directory-Enabled Enterprise, of the issues for interoperability of directory services and applications, and of how interoperability can be assured.

The first part is the Business Scenario for The Directory-Enabled Enterprise. Based on the experiences and requirements of real enterprises, it:

- explores the business and technical environment in which directories are deployed
- analyzes the processes in which they are used
- identifies the human and computing actors that participate in those processes
- summarizes the requirements, and
- looks at the resulting technology architecture model.

Executive Summary

Our understanding at this point is incomplete. The business environment and the technological possibilities are constantly changing. The version of the Business Scenario in this White Paper is a snapshot that will be further developed and evolved over time. It will form the basis for the evolution and development of the testing and certification program.

The second part of the White Paper – Understanding Interoperability – discusses directory service categories and application data access patterns. This leads to the identification of a number of interoperability considerations in the areas of management, operations, schema, and security.

The third and final part of the White Paper describes how The Open Group plans to deliver assurance of interoperability for directory services and applications.

The starting point is the Open Brand for LDAP 2000. The Open Brand allows the vendor to use a "seal of approval" to indicate to buyers that their product conforms to open standards. Vendors with The Open Brand for LDAP 2000 guarantee that their products conform to the standards for LDAP Version 3.

The next step is The Open Group "Works With LDAP" program for directory applications. Vendors with the "Works With LDAP" mark will guarantee that their products interoperate with conformant LDAP Version 3 servers.

The Open Brand for LDAP will evolve as directory standards mature – in particular to provide better security and to cover server-server interoperation in addition to client-server interoperation. The "Works With LDAP" program will evolve to complement changes to The Open Brand for LDAP as our understanding of directory applications develops.

The final part of the White Paper concludes by describing the process and roadmap for evolution of The Open Brand for LDAP 2000 and the "Works With LDAP" program.

Table of Contents

PART 1: THE BUSINESS SCENARIO

Contents

PART 2: UNDERSTANDING INTEROPERABILITY REQUIREMENTS

PART 3: DELIVERING ASSURANCE OF INTEROPERABILITY

Contents

Table of Figures

Part 1: The Business Scenario

> *What a difference a few more bucks for first-rate architecture make to everyone and everything it impacts.* - Malcolm Forbes

Business Scenario Problem Description

Background of Scenario

This business scenario describes an idealized enterprise that is based on a number of real organizations. It does so in order to present a generic picture of the requirements for and deployment of enterprise directories.

It was produced by combining and generalizing specific scenarios for The UK National Health Service and Shell International, and adding input from Kaiser Permanente, and Siemens, and incorporating input and comments from members of The Open Group Directory Program, who are drawn from a wide range of Directory customer and vendor organizations.

The present version is a starting point, rather than a final product. It is based on the minimum range of input that is reasonable for a generic scenario. It is intended that the Program Group will add to it, using input from other organizations who are using Directory in different ways.

The UK National Health Service is the largest single health-provider organization in the world, with 1million employees. Recent implementations of high-profile networks and services — in particular, *NHSnet*, a computer intranet linking major NHS organizations; *GPnet,* a computer intranet linking together all English General Practitioners (GPs); and *NHSdirect*, a telephone inquiry service that allows the general public to obtain medical advice on minor ailments without the need to consult a General Practitioner — have created a pressing need for secured, accurate, and reliable internal Directory Services. In addition, a recent UK Government White Paper has called for an electronic patient record to be associated with each UK citizen throughout his or her lifetime. Internal Directory Services are seen as an essential underpinning for such a system.

Shell International is a world-wide family of companies, with a dynamic structure. It has constantly changing relationships with a host of other individuals and organizations that are intimately involved in its business operations and are connected via an “extranet”. Its business operations are centered on oil and gas, and are diversified into other areas also. Many of its activities are characterized by high-value, high-risk transactions: for example, the value of a supertanker cargo is enormous; the risk associated with a shipwreck is environmentally massive and may not even be quantifiable in business terms.

Kaiser Permanente is America's largest not-for-profit health maintenance organization, serving 8 million members in 11 states and the District of Columbia.

Siemens is one of the largest electrical engineering and electronics companies in the world. Its product range includes Directory servers, and it is developing a corporate PKI directory for internal use.

The main drivers for this business scenario are desires and needs to:

- provide higher quality service to customers
- improve the efficiency and effectiveness of the business processes
- control and manage risk
- provide public access to information, and
- maintain security and confidentiality of information.

In the cases of public access to information, and security and confidentiality of information, there are often legal requirements (such as the confidentiality requirements imposed by US health insurance act regulations).

Directory can contribute directly to higher quality service provision, to process efficiency and effectiveness, and to public information access. Its contribution to risk control and management and to information security is an indirect one, as a component of a Public Key Infrastructure (PKI) that provides improved security.

Purpose of Scenario

This Business Scenario was produced by The Open Group Directory Program Group. The Program Group aims to help customers and suppliers realize their vision of The Directory-Enabled Enterprise. It achieves this aim by delivering understanding of requirements and assurance of interoperability for Directory services and applications.

The purposes of this Business Scenario are:

- to communicate the Group’s understanding of The Directory-Enabled Enterprise, and
- to inform the work of the Group in developing testing and certification programs to provide assurance of directory interoperability.

Objectives

This business scenario supports the following specific objectives.

1. Any authorized person should be able to access contact information for any person in the enterprise, or other information to which he or she is entitled, at any time, anywhere.

2. Authorized users should be able to access services and applications at any time, anywhere, including when they are away from their normal location.

3. Authorized computers should be able to access connection information to other computers for electronic data transfer of information any time, anywhere.

4. The integrity and privacy of information being accessed or being transmitted in messages should be protected.

5. Where desired, authorship and timing of messages should be incontrovertibly verifiable.

6. Directory services should rapidly reflect permanent or temporary changes to the organization’s structure.

Not all organizations have all of these objectives, and the importance given to different objectives in different organizations varies. But every organization that is, or aims to be, “Directory-Enabled” will have some of these objectives, and they are listed here as the key objectives of The Directory-Enabled Enterprise.

Views of Environments and Processes

Business Environment

Enterprises range from small, unified companies to large, complex, distributed organizations. Directories are valuable to enterprises of all kinds, but the value increases with the organization’s size and complexity. The organizations on whose experience this scenario is based are generally at the upper end of the complexity scale.

The UK NHS has a complex structure. A unit of NHS organization is called a Health Community. At its core is a Health Authority. A Health Authority typically has several Hospitals, of which some will work for other Health Authorities. An NHS Trust is a hospital that may have multiple functions, and be servicing more than one Health Authority. A Health Authority may have, say, 65 Practices, each practice having up to 15 General Practitioners (GPs). There are 90-100 Health Authorities in England, 400-450 trusts, about 1,000 Primary Care Groups, about 10,000 GP practices, and about 30,000 GPs. There are also other bodies, such as central government departments, teaching hospitals, laboratories, and mental health trusts. The relationships between these bodies are many-to-many. In principle, anyone in any of them can need to find people or information in any other. NHS bodies also need to co-ordinate with many outside bodies, such as voluntary bodies (charities), local government social services departments, and the police.

Shell is 40% UK-owned, 60% Netherlands-owned. It is domiciled in both countries. Corporate leaders live in both countries; there is no single corporate headquarters. The corporation is organized as a set of global businesses: for example, the aviation fuel business which has 200 people in various locations who rarely meet. Its shape is constantly in flux. Acquisitions and deacquisitions happen frequently. Joint ventures come together and shut down quickly. Operating agreements are made and terminated. Cross shareholdings are formed and dissolved. Locations, organizations, and employees — there are about 100,000 employees — fluctuate rapidly.

The key organizations and entities in the business environment — particularly as relevant to the processes discussed in this Scenario — are illustrated in Figure 1.

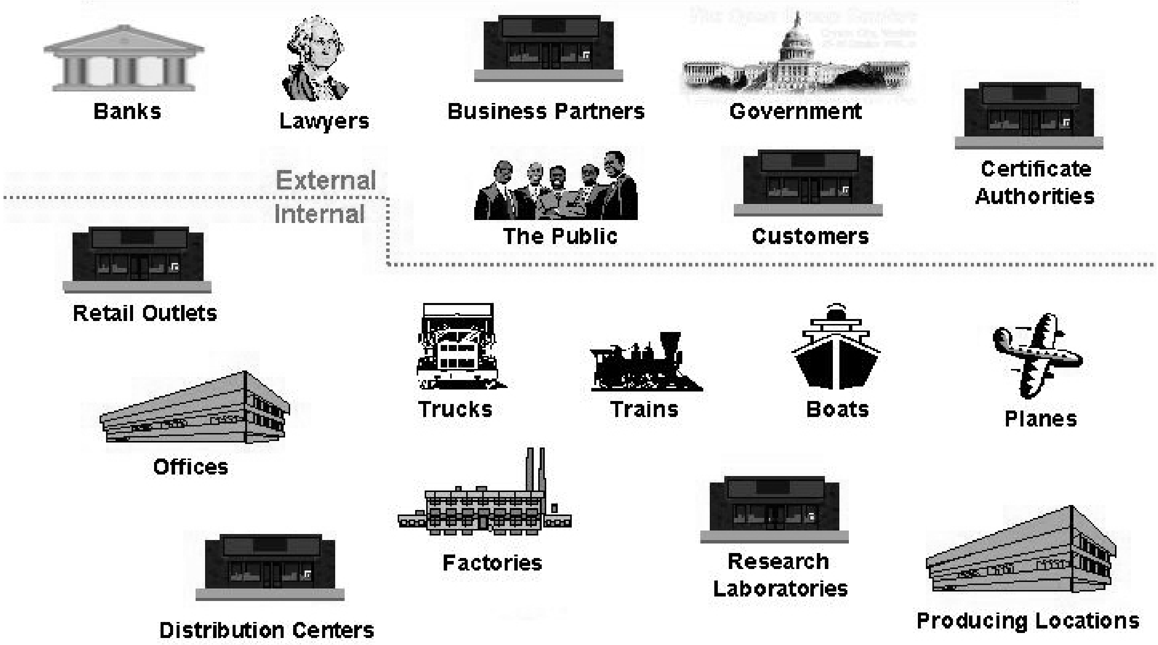


Figure 1: The Business Environment

Not all organizations will have all of these internal components and external relationships. But the figure illustrates a number of points typical of complex modern-day enterprises, each of which has particular implications for the provision of enterprise directory services.

- The organization has a number of facilities of different kinds and in different locations. These include shops, offices, warehouses, factories, and laboratories. They also include special-purpose facilities such as hospitals, oil rigs, construction sites, and police stations.
- In addition to fixed locations, an organization may have directory-users who are mobile — in trucks, trains, boats, planes, etc.
- Users can roam between different locations inside and outside the organization.
- Organizations have business relations with other organizations of various kinds, including business partners, banks, legal advisors, and government departments.
- As well as having established business relationships, an organization may interface to the general public, members of which may access its information and services, either anonymously or after establishing their identity.
- The shape of the organization and its business relationships can change dynamically.
- The distinction between those “inside” and those “outside” the organization may not be easy to draw.

Technical Environment

The key entities in the technical environment – particularly as relevant to the processes discussed in this Scenario – are illustrated in Figure 2.

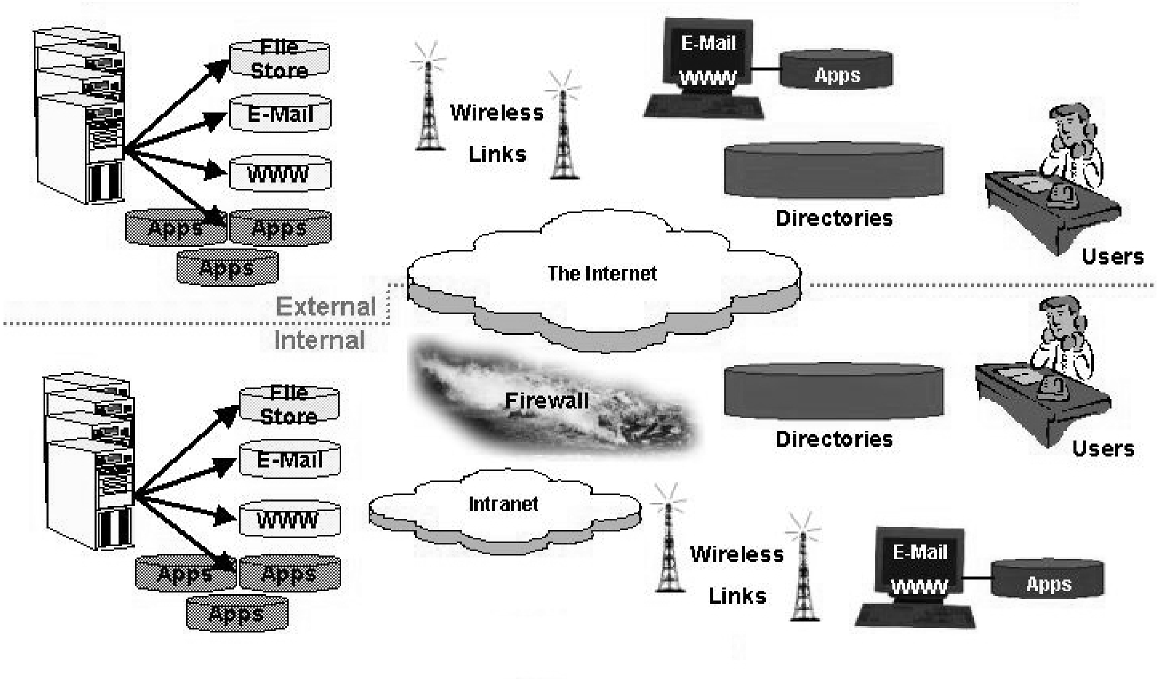


Figure 2: The Technical Environment

Host computers can be of various kinds: mainframes, minicomputers, server PCs. The operating systems that they run include:

- proprietary mainframe and minicomputer O/Ss (MVS, VMS, AS/400, etc.)
- UNIX Operating Systems (see http://www.opengroup.org/regproducts/catalog.htm for complete lists of registered UNIX products)
- Linux
- PC Network operating systems (such as Netware), and
- MS-Windows operating systems.

Some of these operating systems — notably Windows 2000, Netware and some UNIX operating systems — incorporate Directories. These directories are a part of the operating systems' management infrastructure, but in many cases they can also be used for other purposes, including to store details of users and equipment within the organization.

Host computers are used as general-purpose filestores and databases, as mail servers, and as web servers. They also run specific applications to support activities such as:

- personnel management

- workflow management
- finance
- procurement
- supply chain and catalog management, and
- customer relationship management.

Some of these applications make explicit use of directories, generally via the LDAP protocol.

Users access host information and applications from PCs, workstations, and terminals. They may work on the “fat client” (having major applications permanently installed) or the “thin client” (major applications are kept on the hosts and downloaded only as needed) principle.

Hosts and client computers are connected by networks. The Internet Protocol Suite is universally seen as the right choice of protocol for use both within the organization and for external communication. Proprietary protocols are still to be found but are being phased out.

It is estimated that the number of devices connected to the Internet via wireless will overtake the number with fixed connections within the next few years. Wireless connections are currently used for equipment in trucks, ships, etc. Increasingly, intelligent devices carried by the user (PDAs, WAP ‘phones, etc.) will be a factor. Convenience may in time also lead to static devices being given wireless connections.

Network links vary in bandwidth. Present-day wireless links, in particular, often have low capacity. Availability of bandwidth often imposes practical limitations on communications.

Communication takes place between users, clients, and hosts both within and outside the organization. This often implies a need for enterprise directories to provide information about external users, and for external directories to provide information about users within the enterprise.

Many organizations use firewalls to filter traffic between their sites and the general Internet, and to control external access to their systems. The systems and networks running the Internet Protocol within the firewall are often called an *intranet*.

Directory systems are of various kinds. A number of organizations maintain corporate X.500 directories, which in general also have LDAP capabilities. There are specialist LDAP directory server products. There are other data storage products that support LDAP. As noted above, there are LDAP directories bundled with some operating systems. Finally, there are “metadirectory” products that provide a uniform (LDAP) directory interface to collections of directory and other data storage products.

Administrative Environment

Directory and other systems and services within an enterprise may be provided and managed by the enterprise itself, or may be outsourced.

The blurring of boundaries between organizations may lead to confusion among users about who has responsibility for administration of the systems that they are using. For example, a professor of medicine in a teaching hospital may access an NHS directory via the UK academic network, and will likely assume that his local university computing department has administrative responsibility for it.

In many organizations, systems are generally operated by people whose main job is not systems management. In some cases, they are not in an office environment with network engineers readily available to support them.

Process Descriptions

The processes that are analyzed in this scenario are:

- information look-up
- secure e-mail
- access to information and applications
- roaming
- directory update, and
- directory federation.

Information Look-Up

Human users and also elements of the IT infrastructure use directories to obtain information that allows them to contact and access users and objects. In the case of a human user, this is carried out using a directory search engine, which may be part of another application (such as an e-mail client). The directory search engine accesses the directory across the network (typically via LDAP). An element of the IT infrastructure accesses the directory in the same way.

In distributed directories, the first directory that is accessed may not contain the requested information, but may obtain it from another directory. The X.500 DSP protocol is defined to support this kind of operation (known as *chaining*). Non-X.500 directories, and metadirectories, may use LDAP for this purpose.

Address look-up is typically the highest-volume application of Directory in an enterprise. For example, in Shell there are tens of thousands of directory accesses per day. These are mostly simple user look-ups, but the Directory is also used increasingly by business applications to look up information. These are a mixture of off-the-shelf and house-written applications.

In a large and distributed organization, information about users may be fragmented, and ownership of it may be unclear. This makes creation and maintenance of the directory difficult, and favors a distributed rather than a centralized directory solution.

More information is needed in a typical enterprise directory than is found in conventional telephone directories: e.g., specialist skills relevant to the organization and other personal skills (such as languages spoken). The Directory System will need to have some of the characteristics of an HR system.

The look-up service is often used from staff's homes as well as office locations, and by users outside as well as inside the organization.

Directories may be used to locate information, as well as people and physical objects. They can in effect play the role of index to large, distributed, and disorganized databases. For example, the UK NHS is considering their use to provide access to electronic patient records.

- Patients can turn up anywhere in the UK, and (by virtue of their medical condition) can not necessarily communicate with medical staff. People can have medical emergencies that require fast access to patient records.
- Over 99% of the population of England is registered with some part of the NHS (= ~40million people).
- Each person may be registered with several different NHS locations in their place of domicile (General Practitioner, one of several hospitals, local authority).
- Existing practice is for a patient's General Practitioner to be told about every treatment applied to that patient. However, although GPs in theory have the means of keeping up-to-date records on all their patients, in practice the quality of information maintenance varies widely. Also there are no facilities in place to transfer records when a patient changes GPs.
- Patients can seek medical advice when visiting other parts of the country. There is thus a need for location and information transfer among GP computer systems.
- Smart cards are under discussion as a possible part of any solution; however:
    - Two different implementations are being considered: one where the card contains all the information, the other where it contains a pointer / URL to where the information is stored.
    - Smart cards are not necessarily acceptable politically.
- There needs to be an audit trail of who has seen, and who has changed, any patient record information.
- Only 2-3% of lookups to patient records would be in a life-critical situation.
- Mobile access is needed also from paramedics and similar people who are mobile by the nature of their job. Also, people in transit will require access to the service. This implies access by public networks, and so validation and authorization are needed.
- Ownership of the different parts of a clinical record is a matter of debate between medical professionals.

Secure E-mail

The security of information being accessed or being transmitted in messages should be protected.

Integrity: No messages should be altered

Confidentiality: As requested by the accessing person or machine, messages should be encrypted

Digital Signature: The parties involved should not be able to deny having received certain instructions.

Timestamping: Some communications are time-critical, so need timestamping.

The requirements for integrity, digital signature and timestamping most commonly apply to information sent in e-mail messages. The requirement for confidentiality often applies both to e-mail and to information held in filestores, databases, etc. Confidentiality of information held in filestores, databases, etc. is addressed under User Access to Information and Applications, below.

Shell's shipping logistics management process provides a number of examples of requirements for integrity and digital signature.

- In a typical shipping movement, a trader – who buys and sells cargo – negotiates with an operator when and where a cargo will be loaded, when and where it will be discharged, and at what cost. The operator issues instructions to a ship's captain to pick up the cargo from one place and take it to another, on specified dates, sailing by a specified route, and using up a specified amount of fuel. The captain supervises loading with the ship's agent in the port, the local inspector, and the port authority.
- Negotiation and transmission of instructions is generally by e-mail. Some messages, such as inspection instructions and results, are safety-critical (it is important to know what has been inspected, where it is, and exactly what it is). Some messages involve high monetary value (for example, notifications to The Revenue that a refinery has paid duty). Integrity protection is needed for such messages.
- Mistakes in instructions to tankers on where to go or how to load can be very expensive. Disasters aside, if a tanker misses its tide, there could typically be additional port fees of $30K.
- In addition, because of the value of the cargo, and the high cost of mistakes, some messages – such as instructions to the captain to discharge the cargo – must be digitally signed.

There is money lost or potentially lost from lack of integrity, but it is often impossible to quantify how much. There are manual procedures to ensure integrity; PKI should remove the need for these, but again the savings can be hard to estimate.

Confidentiality of information is required for various reasons. It may be a matter of employee relations, as with personnel records or salary negotiations. It may be required by law. For example, confidentiality of health records and other personal information is a legal requirement in some countries. It may also be required because of the commercial value of the information.

For example, when a proposal is made to form a consortium to explore and exploit oil and gas resources, there could be a huge investment cost involved: perhaps billions of dollars spread over 20-30 years.

- It will typically be necessary to find sources of funding, find exploitation partners, seek government approvals, and negotiate terms and conditions with lawyers, governments, bankers, etc.
- Typically, there will be competition with other companies to exploit the opportunity. There have been examples of information brokers selling information on negotiating positions.
- All aspects of the negotiation require privacy (and in many cases integrity and non-repudiation also).

Timestamping may be applied as part of non-repudiation or separately. It is a case of knowing you sent it by when or knowing you received it by when, as opposed to just knowing you sent it.

The overall requirements for directory-enabled security services in those of Shell's business processes that were analyzed in preparation of this Scenario (by no means all of their business processes) are illustrated in Figure 3.

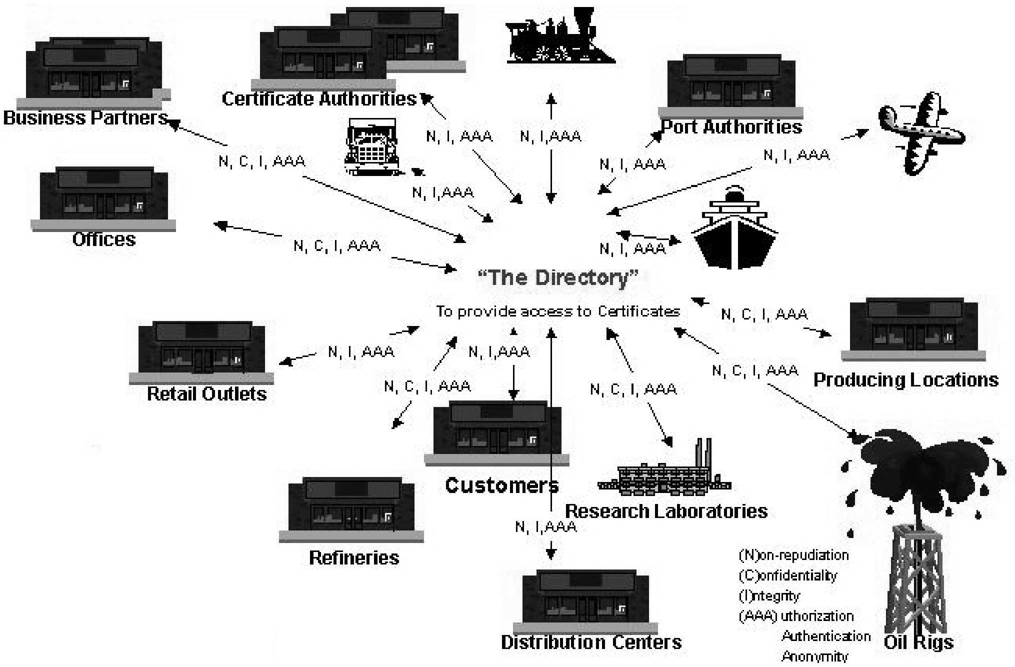


Figure 3: Shell's Requirements for Directory-Enabled Security

Integrity, confidentiality, and digital signature of e-mail can be provided by S/MIME in conjunction with a public-key encryption algorithm. With such an algorithm, a user has a pair of keys: one public, the other private. Information encoded using the public key can not be decoded using that same public key but can be decoded using the private key. And information encoded using the private key can not be decoded using that same private key but can be decoded using the public key.

For integrity checking and digital signature, the sender's e-mail client adds to the message information that has been encoded using the sender's private key. The recipient's e-mail client decodes this information using the sender's public key.

For confidentiality, the sender's e-mail client encodes the message using the recipient's public key. The recipient's e-mail client decodes it using the recipient's private key.

There are various ways in which the recipient can obtain the sender's public key and the sender can obtain the recipient's public key. Public keys need not be kept secret. They can be sent by e-mail. They can be stored in public databases. Often, they are stored in directories, together with other information about their owners.

Obtaining a public key is one thing, trusting it is another. So that their trustworthiness can be verified, public keys are generally made available in *certificates.* A certificate is issued by a *Certificate Authority* (CA) and certifies the ownership of a public key. It is digitally signed by the CA, using the CA's private key. The signature on the certificate can be checked using the CA's public key. If the verifying user does not have or trust the CA's public key, it can obtain another certificate containing the CA's public key, and signed by another CA. In fact, it can obtain a chain of certificates, each verifying the public key of the preceding one. If the chain ends with a CA that the user trusts, the user can also trust the public key at the start of the chain. The certificates that a user needs to build a chain verifying a public key are often stored in directories – in particular, in directories maintained by the CAs concerned.

A CA can revoke a certificate that it has issued. One way in which CAs publish the fact that they have revoked a certificate is to put it in a *Certificate Revocation List* (CRL) stored in a directory. A user verifying a certificate needs to verify not only that it was validly issued, but also that it has not been revoked.

The process of sending and receiving secure e-mail may thus involve searching directories for certificates and CRLs. In theory, such searches should be performed by the e-mail clients. In practice, the user may have to perform much of the verification process manually.

The use of directories to store certificates and CRLs for PKI is described in more detail in Appendix A to this White Paper.

Access to Information and Applications

When users access information and applications across a network – especially when that network is the public Internet – there are often requirements to:

- authenticate those requesting access
- ensure that only authorized users can have access, and
- preserve the integrity and confidentiality of the information as it crosses the network.

For example, Shell publishes specifications of all their products. Some are commodity products, some are speciality products. Associated with each product is a set of materials safety data sheets covering what to do, what not to do, what to do if you do something wrong, etc. Some products have unique characteristics, and Shell may therefore need to implement controlled access to certain parts of the information. The community that has access may or may not be known in advance. For example, they may let ICI see some information but not Exxon, BP but not Elf, etc. Granularity can be important.

Shell cards provide another example. There are 4 million Shell cards. Most belong to road transport fleets. The client businesses give them to their drivers. The drivers can go anywhere in Europe, fill up with fuel, and the client gets a bill at the end of each month. Some fleets have sophisticated control systems, some have none. All clients want to know how to administer the cards, and how to find out what the bills will be. Shell has built a web-based system that gives the clients the information relating to their cards and the use made of them. There is a need to authenticate access to particular records.

Health records provide a further example of information to which access must be carefully restricted to a few authorized people. Here, confidentiality can be a legal requirement, and information accessed across a network must often be protected from snooping.

When the user authenticates (via smart card, or whatever), his environment should be enabled for him, and he should be denied access to everything else. It is necessary to manage the user as a person and also as the filler of a number of roles. A person may fill several roles at the same time.

Specific authentication needs are:

- single sign-on – no-one likes to have to enter multiple passwords, especially when they are different
- platform-independence
- rights management and dynamic resource management — basically ACLs and *good* ACLs, and
- international operation of CAs and RAs.

Increasingly, the Web is the method of choice for providing access to information. The principles are however the same when information is accessed in other ways, such as by client parts of client-server applications, by file transfer, or by “dumb” terminals (or PCs acting as such).

The most common way of authenticating users is probably still by use of passwords. However, transmitting these “in clear” across the Internet is insecure.

The most common way of providing confidentiality is use of the Secure Sockets Layer (SSL) protocol over TCP. With the Web, this is generally invoked by the client requesting a URL with the https (rather than http) protocol.

Usually, this leads to a connection that is secured by the server’s public/private key pair, and the public key is sent to the client in a certificate. The client can verify this certificate by establishing a trust chain. In practice, such verification generally requires all certificates in the chain to be already stored in the client, and revocation of them is not checked, although in theory a client could search directories for certificates and CRLs needed to establish the chain.

SSL contains provision for the client to provide a certificate to the server. In practice, this is rarely implemented. However, establishment of a connection secured by the server's public/private key pair does make it safe for the client to transmit a password for authentication.

The use of directories to store certificates and CRLs for PKI is described in outline in the section on Secure E-mail above, and is described in more detail in Appendix A.

Roaming

When a user is away from his or her normal location, and the resources are available in the location where the user is, he or she should be able to use them there, without connecting back to the normal location.

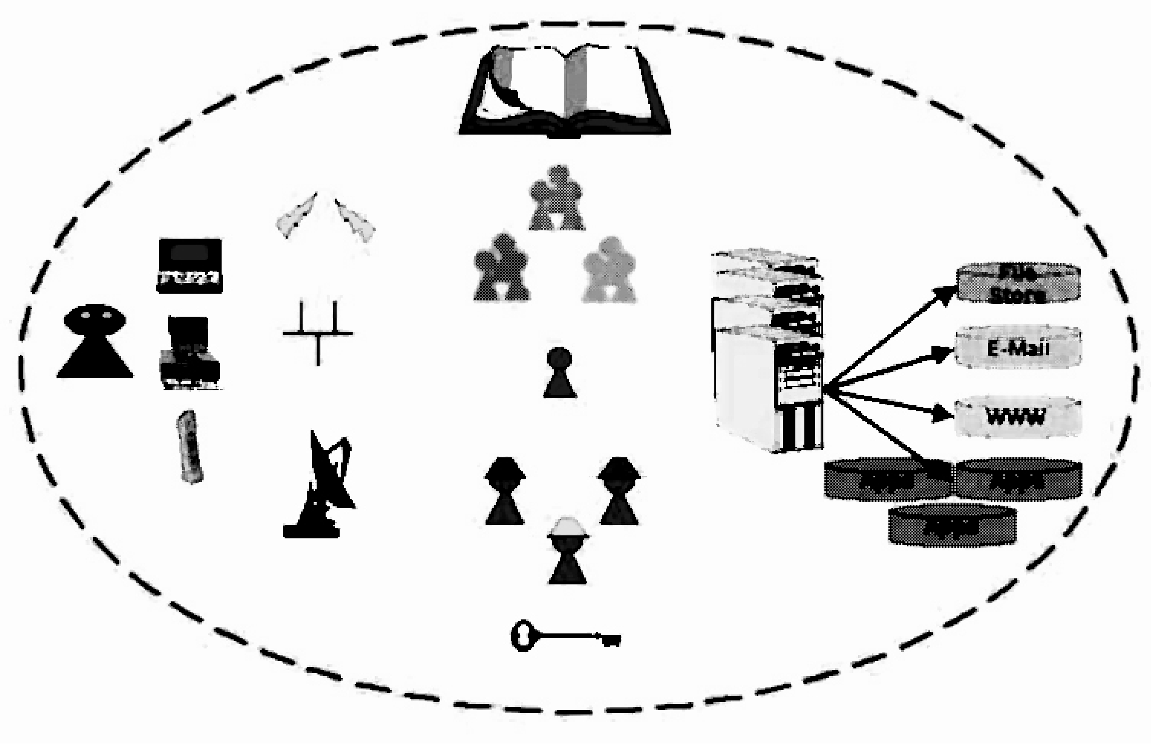


Figure 4: A Local Environment

What services he can use, and how he can use them, may depend on:

- whether he is using a laptop, workstation, PDA, or other kind of terminal device
- what kind of link he is connected by: it could for example be, wireless, 10 MBps LAN, or satellite; it could be symmetric or asymmetric
- what roles he is filling
- what his personal attributes are, and
- what groups he is a member of, and what they allow and enable him to do.

Figure 5 illustrates the information that may be required for correct service provision.

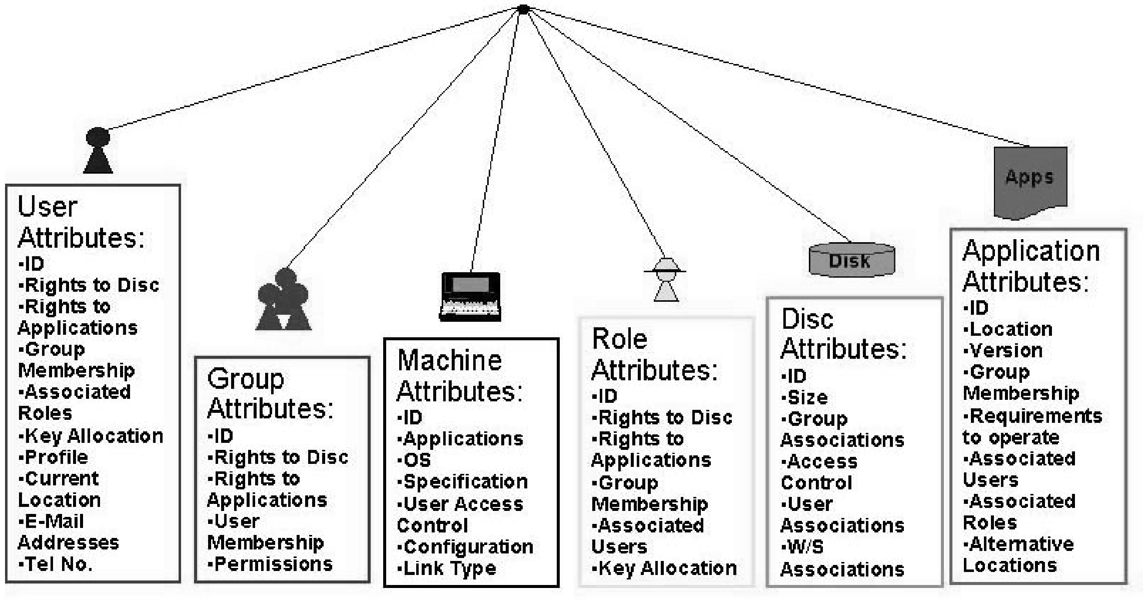


Figure 5: Environment Information

Providing services involves:

- giving the user access to his filestore (wherever it is)
- identifying the nearest mail service, web server, and other applications that can support him and that he can use, and
- determining whether his workstation has the capability of running those applications on his data and, if not, finding alternative means of delivering the capability.

In addition to providing the user with service, it may be desirable to inform other users of his location.

Having established the user within the environment he is visiting, he must be enabled to communicate with other users and access services in other environments. He has authenticated himself within one island; that must be propagated to others.

Directory Update

The volume of changes to a corporate directory may be substantial. For example, currently there are about 5000 changes per week to Shell's central directory. These could be to a person name, company name, departmental name, location, and so on. Central administration of this information can be expensive. An alternative (which Shell have adopted) is to put as much of the maintenance as possible in the hands of the users. This implies assigning ownership of particular entries to particular users, and giving them update capabilities.

However, assigning ownership is not always easy. For example, in the NHS, each Health Authority is typically supported by a single HR function, but that function might be supporting several Health Authorities. Such an HR function would own the information about the staff in the Health Authority offices only (i.e., not the staff in hospitals and trusts in the same Health Community), but it would also have information on all the GPs / Primary Care Groups for which it was responsible. Trusts (hospitals, community trusts, and mental health trusts) have their own HR departments. Trusts mostly own staff information for ordinary Laboratories, while the Department of Health owns it for the Public Health Laboratory Service.

Updates are typically carried out using LDAP (operations add, delete, modify, and modify DN). Bulk updates may be carried out by loading files in LDIF format.

In a distributed directory, updates must often be propagated from one server to another. The X.500 DISP protocol and the (currently being defined) IETF LDUP protocol provide for this.

Directory Federation

Groups from different organizations have a need to work together. Their infrastructures are joined by a WAN and are federated – data-live and sharing basic services.

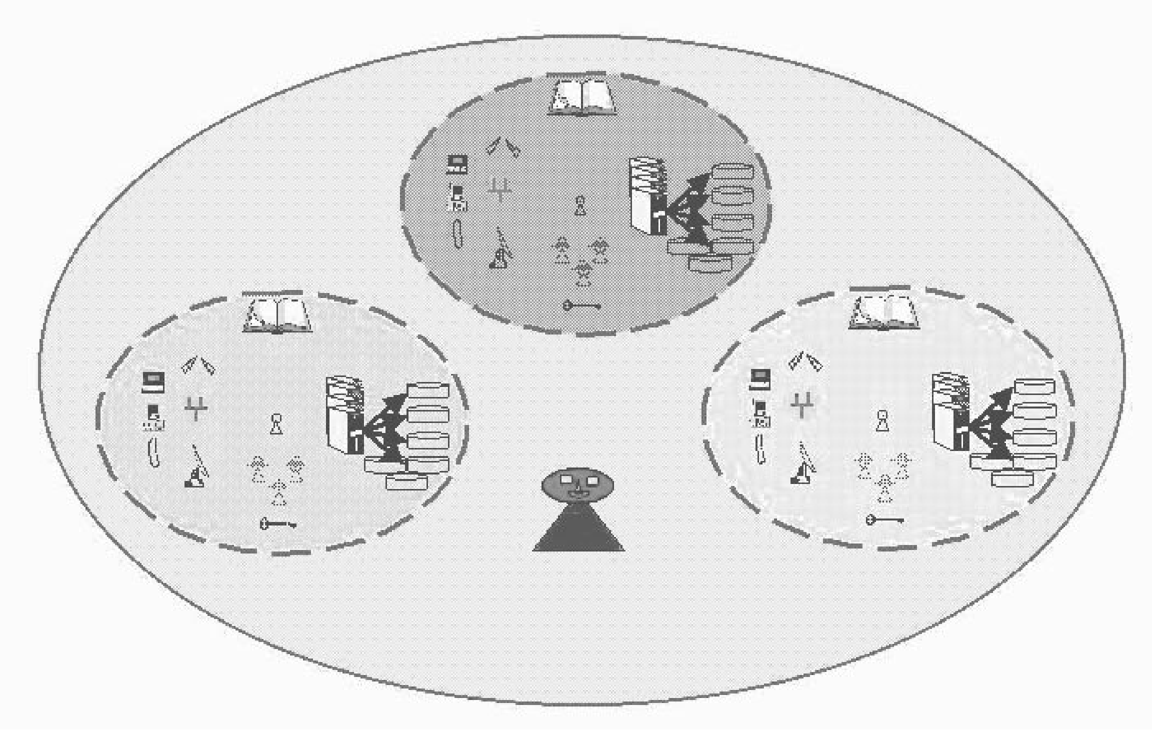


Figure 6: Federation

An example is provided by the formation of a consortium to explore and exploit oil and gas resources, mentioned above. The links between the directories concerned may need to be established quickly. The process of linking the directories should be reversible and non-destructive.

Federation requirements are:

- plug and play – 7 days is too long to get people talking to each other, 72 hours is pushing it
- information available when systems federated should be improved
- common standards-based interfaces, and
- reversibility (most important) — federation must be reversible and non-destructive — if legacy systems are pulled in, it must be possible to pull them back out.

Actors and Their Roles and Responsibilities

Human Actors and Roles

Actor	Role
User	• looks up addresses and other information about people and other entities, either using Directory search engine or using an application (such as an e-mail client) that interrogates the directory • creates, secures, and sends e-mail messages • receives e-mail messages • obtains other users' certificates from the Directory for secure messaging • looks up certificates and CRLs in order to verify other users' certificates for secure messaging • accesses information • uses services and applications • looks up certificates and CRLs in order to verify server's certificate • roams to different local environments • adds, modifies, and deletes "own" directory entries
Administrator	• configures directory schema, etc. • adds, modifies, and deletes entries • defines directory access control • configures replication between directories • configures referrals, chaining, replication, and other server-server communication mechanisms to federate directories

Computer Actors and Roles

Actor	Role
Mainframe	• scientific and business computing
Internet/intranet/ extranet	• connects computing elements
Wireless links	• connects mobile (and some fixed) computing elements to Internet/intranet/extranet
Workstation/ desktop/ portable computer	• hosts client applications • provides user access
Thin client	• provides user access
Firewall	• filters traffic between Internet and intranet • accesses directory for PKI
E-mail client	• user access to messaging • looks up certificates and CRLs for secure messaging
Web client	• user access to information • looks up certificates and CRLs for secure access
Client/Server applications	• user access to information • looks up certificates and CRLs for secure access • looks up information about people and other entities in the Directory • publishes certificates
Directory (internal and/or external)	• contains information about users and other entities • contains certificates and certificate revocation lists (CRLs) used by the PKI
CA (internal and/or external)	• creates certificates (both for users and for CAs) • creates CRLs • stores certificates and CRLs in the directory • updates and manages certificates and CRLs in the Directory
Tools	• information and systems management

Requirements

Directory Requirements

Capacity

The following parameters are important.

Volume: Currently, the highest-volume application of Directory is probably White Pages — simple address look-up. There is currently little use of PKI information in the Directory — but there will be very high-volume use when PKI is used for log-on authentication to services.

Growth Rates: Some organizations require scalability to support tens or even hundreds of million records at a reasonable cost.

Headroom: What will be next barrier to prevent growth? In general it is not possible to control the rate at which traffic will grow (for example, if an organization changes its name overnight, there will be thousands of extra lookups). So there is a need for burst capability, and capacity to cope with the unexpected.

Response Time

For human users, using the Directory must be quicker than picking up the ‘phone and asking the operator. This places an absolute limit of about 5 seconds, regardless of load. Sub-second response is desirable for most operations, longer is tolerable where encryption is involved. Some time-of-day differences may be tolerable.

For some uses by the IT infrastructure and applications, response times should be measured in milliseconds.

Accuracy

100% accuracy of information returned is required.

In a distributed directory, it may take time for an update entered at one point to be propagated throughout the Directory. Time of update propagation should be no more than overnight.

Availability

Any authorized person should be able to access addresses, or other information that they are entitled to access, at any time, anywhere. Authorized computers should be able to access connection information to other computers for electronic data transfer of information any time, anywhere.

Many organizations require their directories to be available 24 hours a day, 7 days a week, 365 days a year.

Some organizations require business continuity “even if the San Andreas fault opens”.

Prioritization

Prioritization of applications and of requests may be required.

Cost

Should be low. Less than 5 cents per access as a maximum for White Pages look-up.

There may sometimes be an internal or external charge, and a need for accounting capabilities.

Ease of Use

Directories and directory-enabled applications are generally used by non-technical people, and must be easy to use.

For example, the Shell Card system had to be designed for use by administrative personnel without technical backgrounds. Such people were in fact able successfully to order cards, change cards, calculate vehicle fuel consumption, etc. for the fleets they administered. The ease-of-use principle is now enshrined by Shell for all extranet applications.

Schema

Co-ordination between different organizations implies a need for common schema and policy object definitions.

Most organizations use off-the-shelf products. Some of these can use directory - but with special schema. A common set of schema attributes is needed at the leaf level if they are to be integrated with other Directory applications.

External communications

Many of the needs for Directory are for internal communications, but more and more are for communications with outside bodies. For example, in Shell, tanker instructions may need to be delivered to shipping agents. The trend is for more change and for more interaction with third parties.

Internationalization

The standards are written by westerners, and have specific ways of expressing names, etc. They don't fit the Dutch, and a number of near and far eastern cultures. For example, far eastern staff have two names: a real Chinese name, and a westernized Chinese name.

There is a need for schemas that can be used across the organization, by CAs and by business partners.

Manageability

The following aspects are important.

Fault-Tolerance: systems should be fault-tolerant.

Self-diagnostics: should be built-in.

System messages: must be understandable by the recipient.

Changes: there must be ability to cope with changes and preserve integrity.

Test system: it must be possible to have one.

PKI Requirements

Introduction

This section describes some general requirements relating to and issues with PKI. A detailed description of how directories are used to store certificate information for PKI, and of some of the problems that can arise, may be found in Appendix A.

What now seems the likely development of PKI is different from what seemed likely two or three years ago. CA services will be adopted, but there are three sets of problems.

- There is a wide variety of potential legislation surrounding digital signatures – governmental and "super-governmental".
- There are problems with interoperability at a policy and at a technical level.
- Working with multiple partners will present problems.

An example of legislative problems is that Malaysian legislation requires use of a Malaysian CA if you want to use digital signatures in Malaysia.

A large organization may operate across many different legal systems. For example, Shell have particular preferences for laws under which contracts are made – UK, Netherlands, US – but that causes problems in other countries.

Industry initiatives may imply particular methods of working. An organization is likely to be a customer of and supplier to partners who participate in different initiatives.

Relationships, for example with banks, can not be avoided. And there will be a need to deal with multiple certificate authorities. But no organization is big enough to determine the form of the relationships with all its partners. As soon as there is more than one partner, there are interoperability conflicts and policy conflicts.

For example, many different smart cards are needed to interoperate with different banks. A treasury clerk will have great problems with this.

These considerations lead to the following requirements.

Liability

Public services should carry some liability for certificate misuse.

There are concerns with CA organizations because they have different policies in different countries. Who do you sue when things go wrong? With Globalsign, for example, sometimes it is Globalsign that has liability, but in some countries it is their partners.

Policy Manageability

Each user organization needs to manage, control, and execute a policy relating to the issue of the liability that a certificate carries. Applications could then allow events, access, etc., in the light of current information and a particular policy. It would help if the Directory contained information that would enable applications to understand when a certificate can and can not be used.

For example, how will BT authenticate someone in Venezuela? It does not have a big presence there. Will it just carry that as a business risk? How should an application behave when presented with a BT certificate for a Venezuelan resident?

Some CA policies say “Our certificate is only valid in country X”. An international organization can not use such certificates. And what if two companies want to do business but there are policy conflicts between their CAs? Is there a “Super-CA”, and if so can it resolve these conflicts? Because of these problems, a user or entity may need several certificates – which must be stored in the directory.

Consistency Across CA Consortia

Two customers may have relationships with different members of a consortium — but those members’ policies may vary. Who manages the conflict?

For example, there is no single CA that matches Shell’s presence across the globe. One user or thing may need to have multiple certificates associated with it. These issues are going to complicate the lives of all multi-country organizations.

Technology Architecture Model

Constraints

There are generally constraints on the definition of any technology architecture. Those applying to Enterprise Directory include:

Quality.

Cost. For example, the NHS directory could be funded by the government Treasury department, which has limited funding and strict rules

Legislation. For example, data privacy legislation

Interworking. The need to work with particular existing or planned elements of the enterprise’s IT infrastructure

Corporate IT rules and policies. For example, Shell has a group communication policy, and has other rules that relate to the directory, including that:

- there will be a single email system
- everyone will appear in the directory
- everyone will have access to the web
- individuals are responsible for protecting company assets — physical and information assets, etc, and

- individuals are accountable for maintenance of own information.

IT Principles

Solutions must be based on standards — which can be international, imposed, or market-driven.

Technology Architecture Supporting the Process

This section does not attempt to define a standard technology architecture for the Enterprise Directory. In the first place, the requirements are only partially understood. There are several aspects – such as use of the Directory to store information about the enterprise IT infrastructure – that this version of the scenario does not address. In the second place, even if the requirements were understood completely, circumstances differ so widely between different enterprises that it is unlikely that a single architecture could suit them all.

What this section does attempt to do is to outline the concept of the Distributed Directory, and to describe some of the implementation considerations that arose in the discussions with the organizations that have contributed to this scenario.

The Distributed Directory Concept

Most large organizations have many directory servers. (An oft-quoted statistic is that the average Fortune-100 corporation has 181.) Ideally, these servers act collectively as a single, distributed, information store. This store is part of a larger store that includes information held on servers belonging to external organizations also.

The original concept of the X.500 Directory was that there is just one, global, directory, which all the world's directory servers co-operate to provide. This concept is still preserved to some extent in present-day directories.

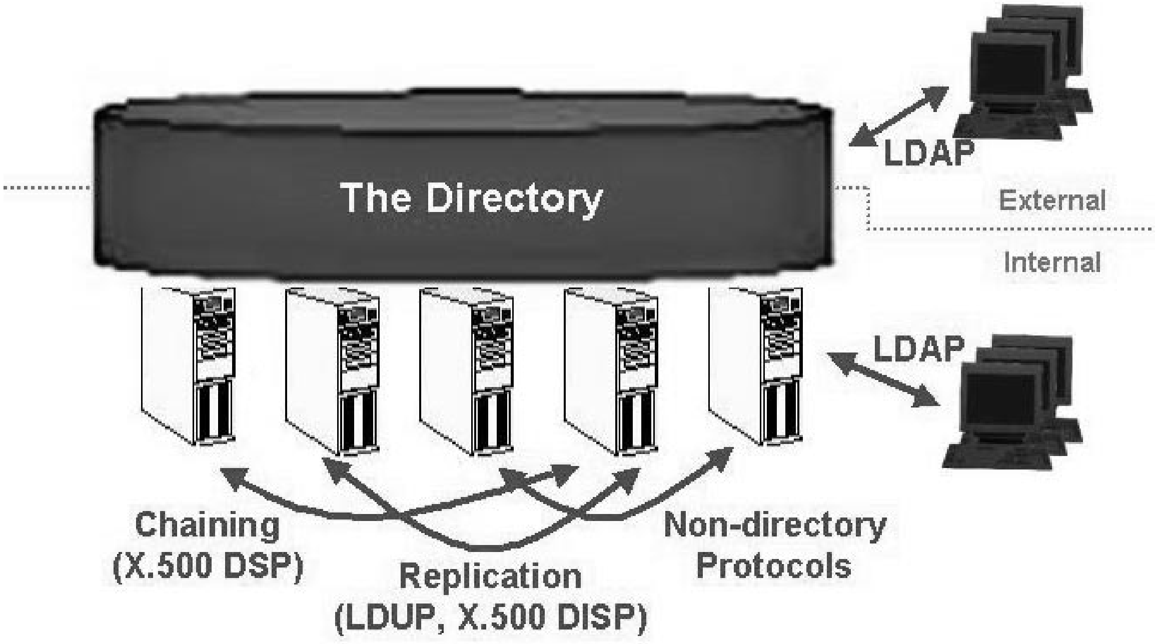


Figure 7: The Distributed Directory Concept

The servers co-operate by:

Referrals and Continuation References: when a server to which a request is made does not have all of the requested information, it may return a *referral* to another server to which the entire request should be directed, or return part of the information together with a *continuation reference* to another server that can provide the rest.

Chaining: when a server to which a request is made does not have all of the requested information, it may obtain some of it from another server.

Replication: information on one server can be copied to others. A particular case of this is *synchronization* of a smaller directory (for example, on a PDA) with a larger one.

The ITU X.500 Recommendations define a Directory Access Protocol (the DAP). This has largely been displaced by LDAP for client access to directories over the Internet. The X.500 Recommendations also define protocols for chaining and replication. The IETF has defined the LDAP protocol for directory access, and is working on protocols for replication and synchronization. LDAP can also be used for a form of chaining. Enterprise directories include X.500 directory products (most of which also support LDAP), non-X.500 directory products supporting LDAP, directories bundled with operating systems and other products, datastores supporting a variety of non-directory protocols, and metadirectories that provide an integrated directory view of disparate storage componants.

NHS Implementation Considerations

The following figure depicts a possible technology view of the architecture of the directory service for the NHS.

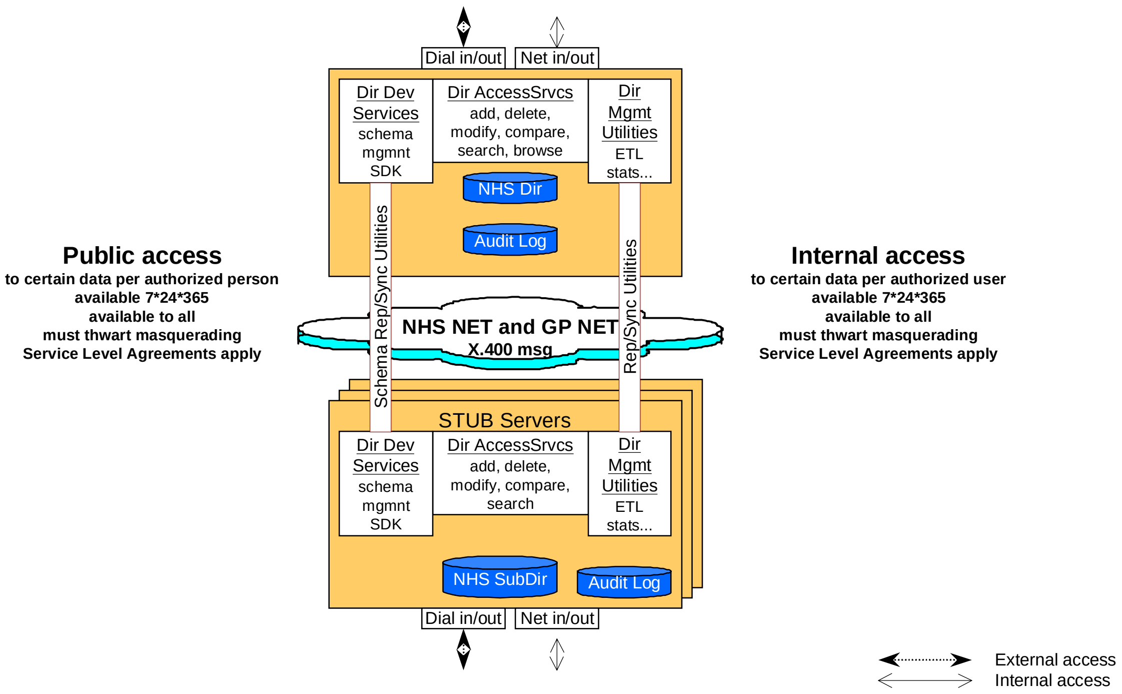


Figure 8: Possible Technology View of NHS Directory Service

Central Server Side

The central server is the record of reference for directory information. It is primarily comprised of the following components:

- Directory Access Services
- Directory Store
- Directory Audit Log
- Directory Management Utilities
- Directory Development Services and Tools.

The central service is likely to be implemented by X.500 directories. It may be outsourced to a service provider.

Stub Servers Side

Stub servers surround the central store and are the first place people will look for information. Though the directory entries are not the record of reference, they do serve the business by providing local directory information that is in synchronization with the record of reference. The synchronization can be set up to support the business needs. For example, if it is critical that directory information be accurate in a real time sense, one may either always go to the record of reference or set up the local directory with a very sensitive synchronization time.

Stub servers are optional. However, if they are used they are primarily comprised of the following components:

- Directory Access Services
- Directory Store
- Directory Audit Log
- Directory Management Utilities
- Directory Development Services and Tools (optional).

Client/Requester Side

The client or requester side has the input and output interfaces. These are assumed to be APIs and/or applications with human interfaces that use the APIs to fulfill a human generated request.

Shell Implementation Considerations

The role of Directory in Shell’s planned technical architecture is illustrated in Figure 9.

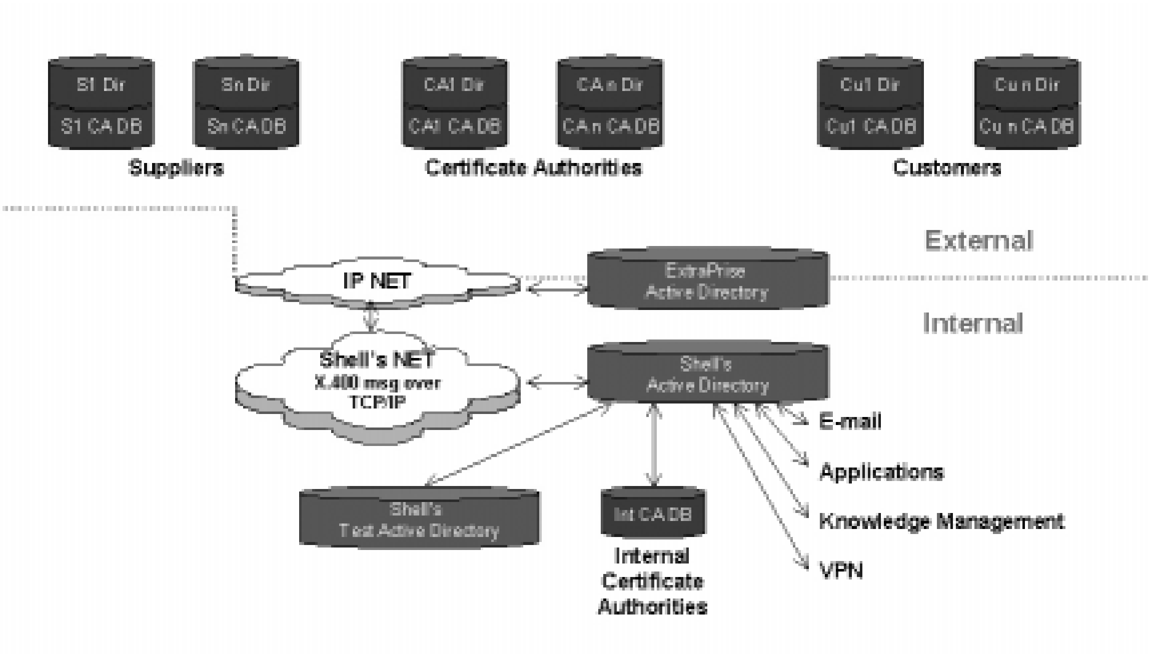


Figure 9: Shell's Planned Directory Architecture

Shell has moved from closed networks to using open networks that they will selectively close. This will be implemented using PKI supported by Directory. Shell wants to move away from USERIDs and ACLs to Certificate Services to authenticate individuals and generate rights to access the systems.

Some of the directory information will be shared with external organizations. This information will be conceptually – and probably physically – stored in an "Extraprise" directory separate from the main corporate directory. But at present this is not done for certificate information, which is retrieved directly from certificate providers' directories.

It is not yet entirely clear how to implement the directory services. Perhaps an independent global metadirectory service provider could play a role. Shell would have to tell them the maximum number of accesses, retrievals, etc. Customer and Supplier administrators will need to maintain parts of the directory information. Shell has an existing corporate X.500 directory, but has now taken an implementation decision to use Active Directory for their directory systems.

Shell will be storing X.509 certificates associated with user records. These will certify things — including mailboxes, function or role-based, that may represent a process — as well as people.

There will be an internal CA for issuing logon authentications and certificates for business partners. Shell is producing policies and procedures for smart cards and certificates.

The Extranet directory will be populated partly with Shell information for external dissemination, and partly external information for Shell dissemination. In particular, external CAs will access the Extranet Active Directory. One concern is replication: at this moment, if you change the value of a field in a schema, that complete record is input to the replication process. Customer and suppliers might well be asked to maintain their own entries in the Extranet directory.

Figure 10 shows the directory components in more detail.

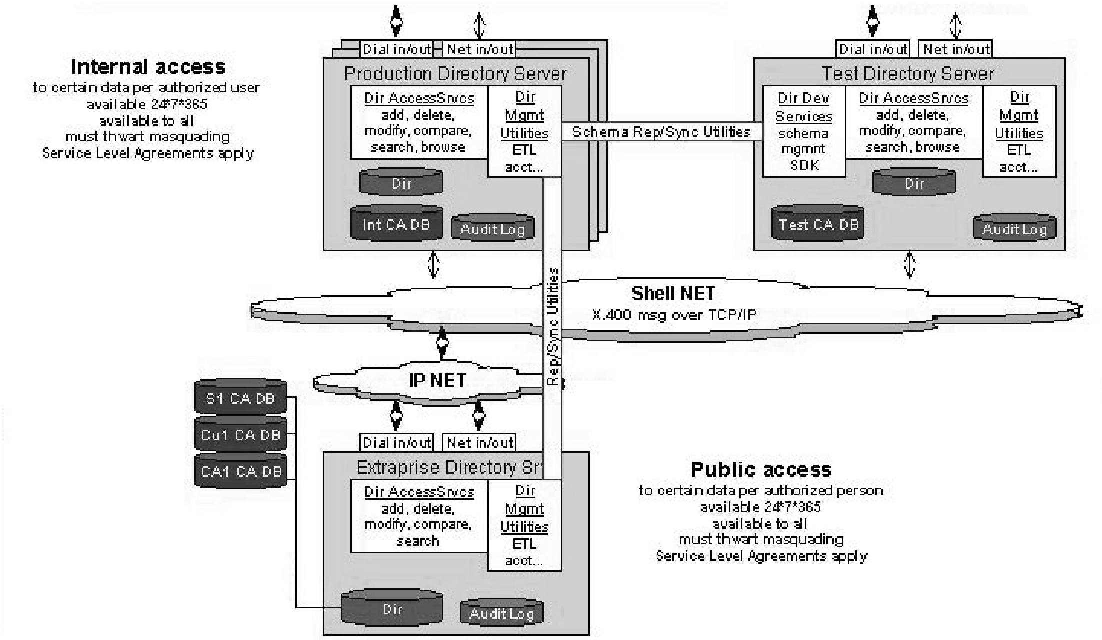


Figure 10: Shell's Planned Directory Architecture – Detail

Kaiser Permanente Implementation Considerations

The Directory employs a "Hub and Spoke" model. There is a corporate metadirectory with distributed repositories and DBMSs. The hub may need to interface to any kind of repository, including (for example) repositories that support the HL7 messaging protocol, and flat files.

Part Two: Understanding Interoperability Requirements

*A Directory is a Directory – **Not!***

Introduction

This part of the White Paper consists of a report produced by Edwards Reed of Reed-Matthews, Inc. for The Open Group that was originally published as a self-standing White Paper in January, 2000.

The Open Group commissioned this report to examine issues that need to be considered in planning for The Directory-Enabled Enterprise.

Program member feedback has been incorporated into this report. Interviews were conducted via phone and email in the 1st three weeks of January 2000. The report was presented at The Open Group meeting in San Diego on 26 January 2000 during the Directory Program session on that day.

The audience of this report is expected to include:

- customers of directory services technologies, particularly those relatively new to the topic who are trying to know if directory services really solve world hunger, the way the analysts and product marketers make it sound, and
- vendors of directory services technologies, who have done a wonderful job of making it seem as though “one size fits all”, when of course it doesn’t.

This report will:

- categorize directory applications according to their uses of directory technology
- discuss the challenges of information sharing and reuse among those applications and their data use patterns (lookup / search / modify)
- resolve consumer anxiety by clarifying how data distribution can be used to support such diverse data use differences, and
- from that perspective, highlight and discuss server interoperation expectations, distributed security objectives and non-objectives, and data interoperability.

Directory Service Categories

There are various ways of characterizing directories:

- centralized vs. distributed
- single- vs. multi-application
- weakly vs. strongly typed, and
- secure vs. untrusted.

Four dimensions may seem too many, but these provide a useful breakdown of the important differentiators. Each directory service can be categorized along each of these dimensions.

There are of course several categories of work that applications do, too, and some of them map well onto one or another combination of the features listed above. For instance:

- Enterprise applications work with information about all the employees or customers or suppliers, for instance, of an organization — which suggests they'll work best against a centralized directory.
- Name services and network operating systems work with geographically distributed users and administrators who need to continue to work in the face of network outages and scheduled downtimes — which suggests they'll work best against a distributed directory.
- Most applications, when deployed, require a predictable, reliable, and accessible infrastructure support — which means they often create the application-specific (stove-pipe) infrastructure they need so they can avoid external dependencies in their project planning and deployment.
- The well-hyped meta-directory marketplace preaches directory consolidation to reduce duplicate data and user administration costs — but the politics of data ownership and external dependencies means there'll never be just one, because not all data *should* be shared with everyone using directories.
- To overcome data owner resistance to sharing infrastructure, assurances about the security and audit-ability of the directory must be provided — but it's widely assumed that LDAP directories are not secure repositories.

Directory-Enabled Networks introduces a whole new collection of applications and data types into the directory, and the work required to represent all the different kinds of policy needed for applications is still unfinished.

In all of the interviews conducted for this report, multi-application directories were seen as the desired outcome of their respective directory deployment efforts. But in many cases, a single application directory was being used to "break the ice" to establish the infrastructure for follow-on applications to use, later. This sets consumers up for additional challenges of migrating from one-directory implementations to another multi-directory implementation, as new application needs drive new directory service requirements, new product versions, new features, and indeed, new vendors. Add the impact of the periodic spate of mergers and acquisitions and directory consolidation begins sounding like more of a lifestyle than a short-term objective.

Data sharing places special requirements on directories and, indeed, on the applications that use them. There are the security consequences of placing application data into a repository used by other (foreign, external, un-trusted, possibly hostile) applications where some application administrators may not share the same sense of high responsibility and moral fitness of others. In addition, there is a whole additional layer of communication and coordination necessary when syntax and semantics of shared data elements have to be understood and correctly used by multiple independently developed applications and their developers.

Data replication in distributed directories introduces additional issues for users and developers when data inconsistencies become noticeable. Certainly, many administration applications don't require "tight" consistency among copies of data. Other applications, like changing passwords, are especially noticeable when replication forces users to wonder which servers "know" about a new password. Finally, applications with genuinely tight consistency requirements, like financial transactions, are simply not suitable for loosely consistent data replication, and should be supported by transactional databases, instead.

Centralized vs. Distributed Directories

X.500 was designed to provide a distributed directory service, and thus provide a single inter-connected directory spanning countries, organizations, and locations. LDAP, the protocol, was designed to provide access to this global directory service. But the global directory has yet to materialize on a large commercial scale.

Instead, commercial X.500 directory products were developed and sold principally as repositories for the specific applications that would justify their cost of deployment — White Pages and address book applications, for the most part. These directories were easily centralized for an organization (which also simplified the politics of financing deployments). The hope was that the projects that funded the deployment, data conversion, and maintenance, would then agree to interconnect with other organizations, and so create the global directory from the "bottom up".

Note, though, that the applications that used these directories might as well have been built using a relational or other database, instead of a directory. In fact, about the only thing that differentiates such applications from traditional SQL applications is the protocol they choose to utter their search requests – LDAP, instead of some ODBC-related SQL transport (like Oracle's SQL*Net).

So, X.500 was designed to be distributed, but has wound up largely being centralized in its deployments. The LDAP market place has tended to be centralized, too.

There are notable exceptions to this picture, though.

Where directories are used to support name services for the network, as with Novell's NDS and Microsoft's ADS, there is a strong bias towards distributed administration and management through an organization. This bias comes from a concrete need for workstations and servers, which are made to depend on configuration and identity information stored in the directory, to continue working in the face of unreliable networks. In other words, branch offices need the ability to continue working even when the network connection to the home office was down. Divisions required the ability to add employees and maintain access control lists for file and print servers, independently of any centralized organization or data repository.

These two fundamentally different types of applications – centralized data repositories vs. distributed name services – have completely different data access patterns, and even different data sources.

The centralized data repository is generally used by humans to locate entries which are only partially remembered by the user – "give me all the people whose last name begins with 'RE' who work in the drafting department", because the user can't remember if Ed spells his name 'Reed', 'Read', or 'Reid'. "Soundex" approximate matches were made to assist this sort of lookup.

The distributed name service is used by the operating system (or some other application software) to retrieve information about an individual, a server, or a service on the network. For instance, a web service may need to know "Is Fred a member of the Administrator Group?" or "What is the S/MIME certificate for Jeanie?" Such queries are not as open-ended or non-specific as the queries humans utter – they begin from a point of information, and tend to use that information to navigate their way through related entries.

In fact, the same design objectives that lead to the design of the Internet Domain Name Service (DNS) apply to distributed name service use of the directory — performance (optimizations for lookup, instead of search), delegation of limited authority (local data administration, but global name space design), and availability (local data access for local operations of the network).

In conclusion, the data access approach – approximate search vs. lookup information about a known thing – is a key identifying characteristic that helps determine whether a centralized or distributed data model is most appropriate for a specific application.

Single- vs. Multi-Application Directories

The second major factor affecting directory architecture is whether a single application will be using data in the directory, or several.

As with any resource, sharing requires more planning, certain rules governing use, and some degree of agreement on how application-private information can be handled. With the directory, there are two kinds of sharing which may go on:

- shared use of the directory, without any data sharing among applications, and
- shared data among applications.

The first of these, sharing the use of the directory without sharing data, is just as easily accomplished with a directory as with any other database product – by creating application-specific areas of the directory, and limiting their use to the application that owns them. This may be a sub-tree in the directory name space, as with an Organizational Unit dedicated to supporting a Certificate Authority using the directory to hold its private information and certificate revocation lists. Or, application-specific attributes may be defined which are unknown to other applications, and so limited in their usefulness to the application that created them.

In many ways, this shared use of a directory is another form of a “stove pipe” solution, isolating the application from interdependencies on other applications. The fact that several such applications can share the same directory may offer some benefits in reduced directory management overhead, but probably has limited benefit as far as reducing data administration costs. Rather, the approach simply allows several applications to each treat the directory as a dedicated, even an embedded, repository of its own application-specific information – a “stove pipe”.

The second of these approaches — actually having applications that share the use of data — is much more interesting. Real benefits can occur when several applications benefit from a consistent, directory-centric administration of application data – all the usual benefits of data normalization (reducing or eliminating duplicate data and duplicate administration) apply here; but note, too, that all the usual political issues associated with data ownership, application interdependency, and “garbage in, garbage out” also apply. So, too, do the risks of “over normalization”, which may adversely impact data access performance for some applications.

In Support of Stove Pipes. Let’s face it – there’s a reason applications get built as “stove pipes”: isolation of interdependencies maximizes the likelihood that each application will be able to meet its own milestones and deadlines. Further, it allows each application to tune the data design to meet its own needs, instead of having to accept a design which may be “a good compromise”, but which is sub-optimal for some (or even all) applications.

This is nothing new. It’s the same quandary application and database developers have faced from the very beginning of recorded time (somewhere back in the ‘50s or ‘60s). In fact, it must have come up the first time someone wanted to use the U.S. Census information (for which the punched card sorters were first deployed in the 1800’s) to do “something just a little different than the designers originally had in mind”.

What is new is that directory technology, which for years enjoyed the quiet solitude of largely single-application objectives, now is tasked with “reducing the number of directories” in an organization from 160-or-more to a manageable number. To do that will require reuse of shared data by all those purchased and home-grown applications, and that will require coordination, cooperation, and not a little bullying from project champions who plan to trim costs when all the work is done.

Can it be done? Surely, for many of those 160-some directory applications do have very similar data requirements and access patterns. And there will surely be enough benefit from being relieved of data administration responsibilities to persuade some data owners to yield responsibility to someone else (especially if such transfer includes the transfer of data entry and/or customer support efforts).

But not all data can, nor probably should, be integrated. Security concerns over who has authority to make or allow changes, or to disallow changes, to data will always impede data normalization nirvana. But, that just means there’ll always be a need to synchronize data between directories, doesn’t it?

Weakly- vs. Strongly-Typed Directories

One consequence of sharing data is that applications are no longer solely responsible for interpreting the data they use. Data formats, or syntax, and data meaning, or semantics, have to be agreed to by everyone who shares use of the data.

Not all application developers will be amused by this new dependency — less so their project managers. Here's the rub: when data syntax or semantics are shared, it either has to be defined once and stay that way, or every application that uses it has to be revised to deal with it when a change occurs. Either way, there's overhead associated with communicating what's going on among all the (known) users of the data.

This is not, yet, a widely recognized issue for shared directory use. But it will be!

The Internet's approach to this has been to build stovepipes, thus avoiding the issue. This comes from years of experience with DNS, IMAP, DHCP, IP, and yes, LDAP.

DNS, if you think about it, is very strongly typed (recompilation and distribution of the changed DNS server code has traditionally been required to add new resource records) – which helps with performance, interoperability, and consistency of use of its data. The lack of extensibility has been seen as a problem, and has resulted in designs for things like the SVR resource record, to handle more information about application services, including things like the UDP or TCP sockets used to talk to them. This seems to be an effort to tear down some of the stovepipe nature of DNS — perhaps just a few bricks.

At the other end of the spectrum are IMAP and the ACAP profile services. ACAP takes the approach that users and applications should make up their own names for the profile and configuration data to be stored. Convenient for the developer of each application, this creates a real problem for administrators and users when multiple applications call the same data different things. This weakly-typed data repository approach is essentially the same approach as that used by ".ini" files for applications in Windows, or ".Xdefault" files in X-Windows.

A similar conflict is brewing in the directory market place, between those vendors and developers who just want to store their information in a convenient place (not a bad objective for the directory to try to support, by the way), and other vendors and administrators who want to enable the maximum data-sharing possible (by shifting some or all of the syntax checking, at least, from the applications to the directories). Directories, like databases, will develop a very practical approach to supporting both communities: if applications can't get along together, then separate them (or rather, their data) so they don't bother each other. If the political winds don't favor multiple databases / directories, and applications are forced to share data, a strongly-typed directory will probably be the eventual winner. Why?

The rationale for using strongly typed directories is the same as for using strongly typed programming languages: when many different users share an interface or data definition, there are likely to be mistakes made in its use. As long as the cost of those mistakes is small, no one is likely to care whether a compiler (or database) enforces the correct syntax usage. But if the cost of mistakes is high, the value in having automated enforcement of correct syntax becomes very cost effective. This rational has held in programming language evolution (from c, to c++, to Java, for instance), and will hold true for shared databases like directory services, too.

Note that semantic consistency is yet another level of agreement needed, though automated enforcement mechanisms are not as well deployed. However, design reviews, good documentation, and persistent project managers have succeeded in the past. Among vendors, these efforts are called “standardization efforts”.

Secure vs. Untrusted Directories

Most directories are built from the premise that “application data which needs protection should be protected by the application, and not by the directory”. In other words, the assumption is made that the directory is an “un-trusted repository”.

This assumption begins breaking down as data-sharing increases among applications. If data need to be protected from prying eyes to and from the directory, it’s not “data sharing friendly” for applications to encrypt the data before it’s stored in the directory. Besides eliminating the ability to search for the data by even the application that created (and encrypted) it, such “protection” also effectively eliminates the data usefulness to other applications (or, alternatively, shifts the problem back to the applications by making them figure out some way to share a common encryption key).

Access control, currently being introduced into LDAP, and available for a long time in X.500 and commercial directory products, begins with the assumption that the directory service, itself, will prevent access to data by unauthorized users. Confidentiality in transit, then, can be accomplished via a transport mechanism, like SSL or TLS, and not at the application layer, itself.

This is an important requirement for data sharing – data entrusted by an application to the directory must be protected according to the wishes of the data owner. Sharing may only be allowed insofar as the data owner approves of the sharing. If application data owners don’t trust other data owners, chances are that they won’t trust each other’s administrators, either. Whether directories will protect data from un-trusted administrators, or just from un-trusted application users, is a distinction that must be addressed as part of the interoperability question.

The availability of a directory infrastructure trusted to enforce data owners’ policies on data use and modification will be essential to achieve the inter-departmental, inter-ministry, and inter-company cooperation necessary to build the global directory. Certainly the un-trusted directory approach has not taken us far in that direction.

Application Data Access Patterns

Name service applications, data mining and reporting applications, narrow scope searches, unconstrained searches, data synthesis and correlation (meta-join) applications; each of these has quite different data access expectations, timeliness of data requirements, and consequently different performance tuning requirements.

Lookup

Name services, including retrieval of email addresses, network addresses, S/MIME certificates, CRLs, and group membership values, all proceed from a known piece of information – the name of the thing about which information is required. Hash tables work well to accelerate retrieval times in the directory (and are used by DNS, as well as older (pre NDS-8) versions of Novell’s Directory Service, etc).

Because the name of the entry is generally “known”, and is provided as part of the query, a lookup behaves like an “indexed read”, in which the entry is first located, and then the data requested is read and returned to the application.

Like DNS, an interconnected directory service can use the namespace to delegate administrative and management authority, and can distribute data to servers that each hold the entries to which they’re entrusted. This distribution of data greatly facilitates performance (data can be held locally, near its frequent use), administration (data can be created and changed by those who use it most often, meaning it will be more likely to be accurate, or fresh), and availability (locally held data supports local operations, even in the face of network outages).

This is not to say that lookup services cannot be centralized – of course they can. But lookup services are the most easily distributed directory applications, because the distribution of data closely matches the distribution of work and lines of authority in many organizations.

Applications that use lookup services are also generally less susceptible to requiring tight consistency of data among all their many replicas. First, because the data is generally located near the most frequent users, and second because the kinds of data used by lookup-style applications tends to be configuration- or identity-related. Probably the best counter-example of this is passwords, and their synchronization among replicas — people expect that when they change their password, that it is changed everywhere at the same time — and the propagation delay associated with replicating the change among replicas can be noticeable, in this instance.

The Directory-Enabled Network (DEN) initiative uses the directory to hold identity, service address, resource inventory, configuration, and policy information for Quality of Service offerings, among other things. Most of the data accesses “in the network” will follow the Name Service Lookup pattern. Whether they work better with distributed or centralized repositories remains to be seen.

Narrow Search

We use the term “narrow search” here to mean that some information about the desired directory entries is known, but not the name of the entry itself. In relational database terms, a lookup uses a “candidate key” to return the specific record (or entry) desired, whereas a narrow search returns a few records (or entries), but not a huge number. The known information can help narrow the scope of the search by indicating where the data is likely to be stored (to which server should the query be directed), and to find a handful of possible matches which can be delivered to the user so a specific entry can be selected for further use (via a lookup mechanism, like those described above).

Common experience with the ubiquitous web search engines is instructive here. If a user enters a search with a few specific key words, the search engine may be able to narrow the result set to a few dozen matches, from which the user can then "click through" to get more information. On the other hand, a poorly worded search will return thousands of matches, which may be useless to the user.

Narrow searches benefit from having all the data that match on some criteria locally available to the directory service, so that indexes can be constructed to speed things up. Examples of the criteria that can help "localize" entries include their parent organization names, the country in which their parent organization may be found, department, etc. These are the commonly used geographical and organizational labels for constructing LDAP distinguished names, Internet DNS names, telephone numbers, and postal addresses.

Narrow searches further presume, in addition to being able to help the application "guess" which server to ask, that there is sufficient information about the entry to narrow the list of candidates to a few. Basically, this is so that the server that gets the request has some hope of selecting an index (of the data it holds in its local database) to scan for candidate matches, instead of having to look at every entry it holds (a process called a "full table scan", which usually takes much, much longer than reading an index).

So, there are two aspects of narrowing the search – first, to know what server(s) to ask (out of the millions of servers in the known universe), and second, to limit the number of matches by using what is known about the desired entries (hopefully achieving better performance by using pre-constructed indexes, too). Without knowing which server to direct the query to, though, the search becomes essentially impossible to complete.

Data Mining

Data mining, by which we mean reporting, correlation, summarizing, and categorizing data, requires data to be centralized for analysis. Latencies associated with trying to perform data-intensive operations (including joins, counts, and arbitrary sorts) would make trying to do much data mining with a widely distributed directory too expensive in time and bandwidth to be practical.

But there are people wanting to do very sophisticated data mining against directory information – particularly user profile information associated with web sites and e-commerce.

Data mining is not an appropriate use of directory technology, though. SQL, or other data analysis languages are much, much better than LDAP Search for expressing and manipulating data for reporting and analysis. Far better, then, to consider using the directory to provide distributed administration and replication of user profile information, and then synchronize the directory contents with a relational database for the heavy-duty data mining. Of course, if all the user profile information is centralized already, and all the administration of the content is already directed to the central repository, there may be little need to provide a directory interface to the profile information at all, unless there's a desire to access the profile information via LDAP by users for lookup or narrow search applications.

Data mining applications, and customers for whom data mining is the principle use of directory information, should seriously consider using relational database replicas of their directory information.

Modification

There are two types of modification of directory information: single entry and batch.

Single entry modifications use the normal MODIFY LDAP operation to change a named entry. Note that the entry to be changed is known before the modification is made, so this sort of modification works well with name service-style directories, as described above, whether centralized or distributed.

Batch modifications operate against whole groups of entries, frequently as a result of an import or replication operation. This sort of batch transaction can also be handled well by name service-style directories, although the cost of directing updates individually against one or several servers is substantial. Practice has shown that when the number of operations is large, the costs of network latency and disk I/O performance are enormous. In such cases, it is much faster to send a group of operations to the server in a single network packet or stream, and to have the directory process them all as a single transaction, rather than to process them each individually. Experimentation with such bulk modifications is being done now, and may prove to be very useful in the near future.

There is another kind of batch modification for which LDAP is not well suited: when a common, algorithmic change needs to be made to more than one entry. For instance, when a telephone area code changes, there may be an algorithmic way to describe which numbers for which entries need to be changed, and to describe the appropriate change. As things stand today, though, the only way to do this with LDAP is to create an LDIF batch file with the changes enumerated for each entry, explicitly, and then to run the LDIF file against the directory, changing each entry, in turn. At some point in the future, something comparable to the way SQL applies update logic to selected data sets might be useful.

Interoperability Considerations

So, now the question is, how do these characteristics of directories and their applications affect the ability of customers to field directories from multiple vendors? Is there any hope that they will interoperate, and what must be done to ensure that they will?

More importantly, knowing that there are different kinds of directories, and that some work better for certain types of applications and not for others, how can directories of different types work well together to meet all the needs of the enterprise, and for e-commerce applications, in particular?

Management

Managing a community of directory servers should be easier than managing each of them individually. Whether products lend themselves to such community management or not is the issue, and whether standards are being developed to encourage products to be jointly managed is the issue.

Configuration – no, vendors won't standardize on configuration tools, but there should be standard ways to read a directory server configuration, and so that vendor-provided tools can at least display common options. The RootDSE attributes are a beginning in this direction, but need to go much further. Expanded profiles of common server configuration options need to be established and supported by vendors.

Policy Establishment & Enforcement – the industry needs to be told by customers what kinds of directory-enforced policies are generally required, and whether they ought to be able to be turned off. Examples include schema checking, access control enforcement, naming rules governing namespace containment, etc. Customers must realize that if some applications require enforced policies, that other applications must be designed to work with the policies turned on, if the directory is to be shared by those applications.

Performance Management — the benchmark wars have barely begun. All the more disappointing is that right now, they compare apples and oranges. The industry (meaning customers demand and vendors provide) must define benchmarks that include name service applications performance, as well as narrow search performance. The introduction of Microsoft's ADS product as another name service directory, in addition to Novell's Directory Service, should facilitate such a development.

Operations

Import/Export — good progress is being made on standardizing around LDIF as the import/export batch file format. Extensions to LDIF will be required to support new operational data being introduced by LDUP.

Replication Among Directories — the IETF is making progress, but more importantly, there are rumors that development has actually begun on LDUP implementations. Year 2000 will still have much work by all to be done, though. And in the mean time, a client-driven "dredge" of the directory for changes will emerge as a lighter-weight solution for remote users and casual directory interconnection.

External Data Source Synchronization — industrial strength infrastructures for designing mappings between data sources, and actually pushing and pulling data changes between them, are coming to market, and will not, for the foreseeable future, have anything to do with standardization efforts. But these meta-directory solutions will be important integration tools for many more years, and will never disappear. We can hope, though, that fewer custom protocols and solutions will be required, over time.

Schema

Strong vs. Weak Data Typing — follow the principle of the Internet – applications should be strict in what they write, and generous in what they accept. That means that applications which write with no discipline (i.e., assume weak data typing) are simply not going to interoperate with anyone other than themselves. For the sake of interoperability, customers should avoid them, the same way they avoid applications that play fast and loose with the IP protocol packet header contents.

Extensibility and Sufficient Data Type Support — directory vendors could do much good if they provide sufficiently rich data types and adequate schema extension options that developers will be less inclined to develop “sloppy applications” (see “Strong vs. Weak Data Typing”, above.

Schema Naming and Conflict Management — this is a bomb waiting to go off. Right now, there are only a few companies aggressively extending their schemas beyond what the standards require. Microsoft and Novell must be compelled by customers to ensure their schemas will facilitate interchange of data, if not full bi-directional interoperability of their directory services.

Multi-Byte Character Support for Names — UTF-8 and UNICODE support is essential for shared-use entries in the international community. Universal support for them by applications as well as directories is essential to avoid needing multiple entries for people with extended Latin and non-Latin names. This means putting an end to old LDAPv2 and X.500 client applications and directories with their use of T-61 and Printable Strings.

Security

Authentication — good progress is being made towards arriving at a set of standards everyone will live with. Details remain to be worked out, though… SSLv3/TLS X.509v3 certificate-based mutual authentication has the most momentum in the Internet at the moment, but Kerberos (in one of its variations) will certainly be an important mechanism, as well.

Authorization — Year 2000 will be nearly over before we see whether the proposed LDAP ACL model and design will be generally accepted. Until then (and for a few years thereafter) access controls will continue to be vendor- and application-specific. Some users (especially governmental customers) are advocating use of X.500 (97) Rule-based access controls as an important enabler for interconnection of multi-organization directories. However, there’s not much product support for it yet.

Privacy Regulations — the European Union Privacy regulations governing data and its export highlights the need for adequate policy enforcement by directories of access controls and data filtering requirements. Multi-national corporations are already faced with the challenge of supporting the regulations within their own corporate boundaries. Internet interconnection of companies, suppliers, consumers, government agencies, and the public at large requires much more attention to these issues than many vendors are giving it.

Part 3: Delivering Assurance of Interoperability

Introduction

Delivering assurance of interoperability is not easy. Two products can independently pass protocol conformance tests, and yet fail to interoperate. And when they are supplied by different vendors, responsibility for the failure can be difficult to determine.

At the July 99 Open Group Conference in Montreal, members of the Directory Interoperability Forum (DIF) put forward a strong and clear requirement for The Open Group to design and manage certification schemes for Directories and for Directory-Enabled Applications. The Open Group took up the challenge of providing interoperability certification for LDAP, and accepted the requirement.

How to satisfy the requirement was the subject of discussion by The Open Group Directory Program, the DIF, and The Open Group department for Testing and Certification. The result is the certification program described here.

Overview of the Approach

The approach adopted is to certify:

- conformance of servers to the protocol standards, and
- interoperability of applications with conformant servers.

What the server has to do is tightly defined and limited in scope. This makes it possible to test it. The Open Group is developing a test suite – VSLDAP – that will verify conformance of servers to the LDAP Version 3 protocol standards. This test suite will underpin the server conformance certificate – the Open Brand for LDAP 2000.

What the application has to do is broader in scope and more loosely defined. The application must interoperate with servers. This means more than just driving the protocol correctly. An application that tries to add an entry in a way not allowed by the schema on a particular server may be driving the protocol correctly, but it is not – as far as the user is concerned – interoperating.

This illustrates one of the difficulties of certifying application interoperability. It may depend not only on the design of the application, but also on aspects of the server that are not regulated by the standards, or even on configuration settings that are under the customer's control.

Because of these considerations, application certification has to be radically different in concept from server certification. "Works With LDAP" certification is based on a vendor guarantee – and a guarantee of interoperability in the fullest sense – but the vendor qualifies that guarantee by stating the conditions under which it holds.

This means that:

- a customer who buys a certified server and a certified application has a guarantee that they will work together
- the responsibilities of the server vendor and the application vendor in providing that interoperability are clear
- no-one has to guarantee anything that is not reasonable, and
- the customers must ensure that the conditions for interoperability are met – but have the information to enable them to do so.

The Open Brand for LDAP 2000

The Open Brand for LDAP 2000 provides the guarantee of conformance of servers to the protocol standards. Its definition was completed in December 1999, and it is now ready to launch.

The Open Brand

Open Brand certification provides the means by which vendors guarantee to buyers that their products do conform to required specifications. That buyers find this valuable is attested to by the fact that they have mandated more than $65 billion worth of branded products in their procurements. The full Open Brand program currently includes 25 suppliers registering over 1,500 products.

The Open Brand program is built around the right to use certain trademarks, on and in connection with certified products. Once a vendor has executed the Open Brand Trademark License Agreement, their product complies with the relevant specifications, and they have successfully registered the product, thereby guaranteeing conformance, they are entitled to use the trademarks in relation to the registered product.

LDAP Certification

The Open Brand for LDAP 2000 is a member of the overall Open Brand program. It signifies guaranteed conformance of directory servers to the IETF Lightweight Directory Access Protocol (LDAP) version 3.

The requirements are fully described in the Product Standard at **http://www.opengroup.org/regproducts/dim0.htm**. In summary, a server must:

- meet the mandatory requirements for a server of IETF RFC 2251 (Lightweight Directory Access Protocol Version 3), of RFC 2252 (Attribute Syntax Definitions), and of RFC 2253 (UTF-8 String Representation of Distinguished Names)
- return referrals and continuation references as described in RFC 2251, and conform to the mandatory requirements of RFC 2254 (The String Representation of LDAP Search Filters) and IETF RFC 2255 (The LDAP URL Format) when returning LDAP URLs in referrals and continuation references

- implement the mapping of LDAP over TCP described in Section 5.2.1 of RFC 2251 in which the LDAP messages are mapped directly onto a TCP byte stream
- conform to the mandatory requirements for a server of the Secure Sockets Layer Protocol (SSL), Version 3, and
- implement a mapping of LDAP over TCP in which the LDAP messages are mapped directly onto an SSL byte stream.

A branded server can also declare support for any or all of the following optional features of LDAP version 3:

- extensible match
- notice of disconnection
- client modification of subschema entries
- validation of client SSL certificates, and
- access to SSL credentials via SASL EXTERNAL.

The VSLDAP Test Suite

Like most members of the Open Brand program, the Open Brand for LDAP 2000 is underpinned by a test suite, VSLDAP. Passing the test suite does not automatically qualify a product for the brand. The vendor must guarantee conformance to the standards. Such a guarantee is a serious matter, and passing the test suite provides the vendor with a reasonable indication that the guarantee can safely be given, but ultimately it is the vendors’ responsibility to satisfy themselves that their products really do conform.

Of course, if the product does not pass the test suite, then it does not conform, and it can not receive the Open Brand.

VSLDAP is expected to be released in October of 2000. Before then, vendors can obtain the Open Brand for LDAP 2000 provided that they make the guarantee of conformance. They must also pass the test suite within 90 days of its release.

“Works With LDAP” Certification

The “Works With LDAP” certificate provides the guarantee of interoperability of applications with conformant servers. Its definition is agreed in principle, but may be subject to further modification. It is intended to be launched in October 2000.

Outline Definition

To obtain the “Works With LDAP” certificate for a product, an application vendor must say:

- which of the product functions use LDAP
- under what circumstances they work with LDAP, and
- what tests have been run to verify that they work with LDAP.

There will be a web application form incorporating a checklist to help vendors make these statements. It will ensure that they consider common circumstances under which products may or may not work with LDAP, and will ensure that the tests are reasonably thorough. There is a draft of this form in Appendix B.

The certificate will be awarded if the vendor submits the application as required by the web form and warrants that the product works with any standard LDAP server. The vendor's warranty will be the essence of the certificate. The tests will be supporting evidence. The test results will not be audited as part of the application process.

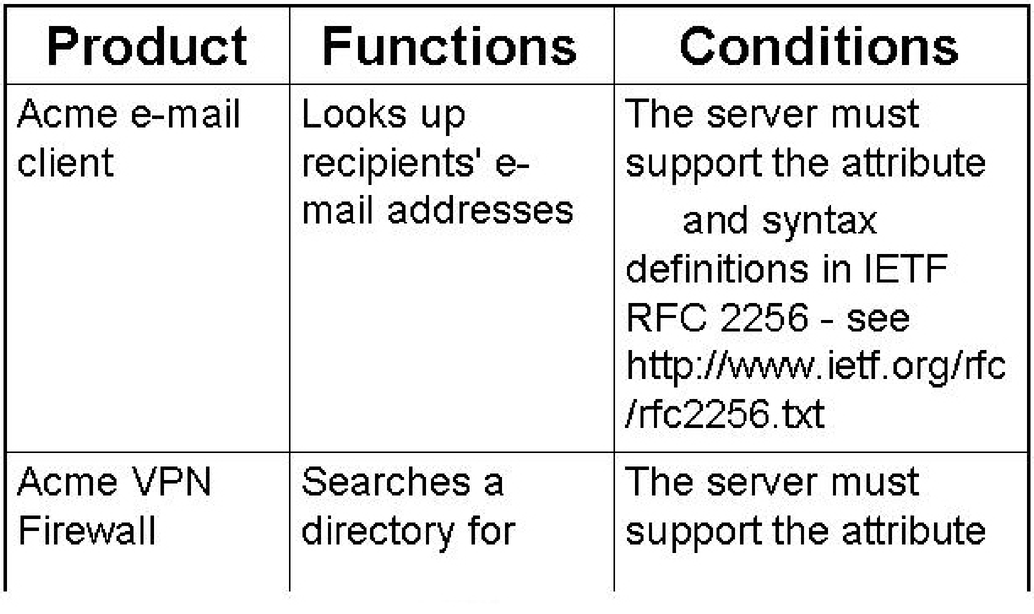
If a vendor's claim is found to be incorrect, the vendor will have to fix the product or fix the claim (for example, by changing the statement of the conditions under which the product works). If the vendor does not do so the certificate will be withdrawn.

Operation of the certification process will be funded by registration fees paid by vendors. The process is sufficiently lightweight that these fees can be kept low.

The rationale for the particular definition chosen can be found in Appendix C to this White Paper.

Customers' Perspective

The “Works With LDAP” web site will list all the products that work with LDAP. For each product, there will be a description of how it uses LDAP, and under what conditions. How the “Works With LDAP” web site might look is illustrated in Figure 11.



Product	Functions	Conditions
Acme e-mail client	Looks up recipients' e-mail addresses	The server must support the attribute and syntax definitions in IETF RFC 2256 - see http://www.ietf.org/rfc/rfc2256.txt
Acme VPN Firewall	Searches a directory for	The server must support the attribute

Figure 11: How the Works With LDAP Web Site Might Look

The description of the conditions will help customers to understand how easy it will be to integrate the product into their environments. For example, it will say whether there are any access control requirements that their servers might not be able to meet. And it will say whether there are any schema requirements that might conflict with those of other applications.

The certificate is a certificate of interoperability, not of scalability or performance. As confusion could easily arise about this, the description of the “Works With LDAP” certificate, and the “Works With LDAP” web pages, must make it clear that scalability and performance are not covered.

The statement of what product functions use LDAP and the description of the circumstances under which they work with LDAP will appear on the “Works With LDAP” web page. The description of the tests performed by the vendor will not be shown on that page, but will be available on the web for inspection by customers that need or wish to investigate at that level of detail.

There will be a logo associated with the certificate. Vendors that have the certificate will be able to use the logo on their registered products.

The use of the logo on a product, and the appearance of the product on the “Works With LDAP” web site, will mean that the customer knows that the product will work with standard LDAP servers.

Vendors' Perspective

There will be a web form that vendors will use to apply for certification. Using the form, they will enter the information about their products that will be displayed to customers on the “Works With LDAP” web site, and will complete a checklist to support their application.

The information that will be displayed to customers on the main “Works With LDAP” page is:

- product name
- functions that use LDAP, and
- conditions under which the product works with LDAP.

The form will contain questions and instructions designed to help the vendor provide accurate and useful information to be displayed to customers. It will also require the vendor to say how the product has been tested to verify that it works with LDAP servers.

A draft of the “Works With LDAP” application form is contained in Appendix B to this White Paper.

The level of testing that will be expected will be no more than what any prudent vendor would carry out as a minimum as part of product development. For example, testing of a similar nature to that of BLITS will be perfectly acceptable.

The form will include a checklist of LDAP features. The vendor will use the checklist to indicate (a) which features are used by the product to perform the functions and (b) which features have been tested. Where features are used by the product but have not been tested, the vendor must say why.

The information about testing will not be displayed on the main “Works With LDAP” web page, but will be displayed elsewhere on the web and available for customers to inspect.

The vendor will complete the form and submit it. The Open Group will review the submitted application, and will discuss any incomplete or unsatisfactory answers with the vendor. The Open Group will not perform testing or auditing of test results.

The Open Group will prepare a legal agreement using the submitted information, and send it to the vendor for signature. When the agreement has been signed, and a registration fee paid, The Open Group will countersign it and place the product information on the “Works With LDAP” web site. The vendor will then be entitled to use the logo in conjunction with the product.

The registration fee will be low, just sufficient to cover administration costs. The target fee is $500.

The main expense for the vendor will be in providing the information and in doing the testing. However, generating this information, and performing the tests, is something that vendors need to do to develop and market products that work with LDAP, regardless of the existence of the certificate. Completing the checklist should in itself be valuable for them.

The agreement may be terminated by the vendor or by The Open Group at any time. The Open Group will do this if, and only if, it is clear that the product in question does not work with LDAP as claimed.

Legal Framework

“Works With LDAP” will be one of a range of Open Group “Works With” certificates.

The legal basis for these certificates is trademark law. The vendor enters into an agreement with The Open Group. Under this agreement they are licensed to use an Open Group trademark in conjunction with their products. As part of the agreement, the licensee represents and warrants that the product meets the applicable Quality Standards set forth in a schedule to the agreement.

That schedule is specific to the particular certificate concerned. The Quality Standards schedule for “Works With LDAP” will be as follows.

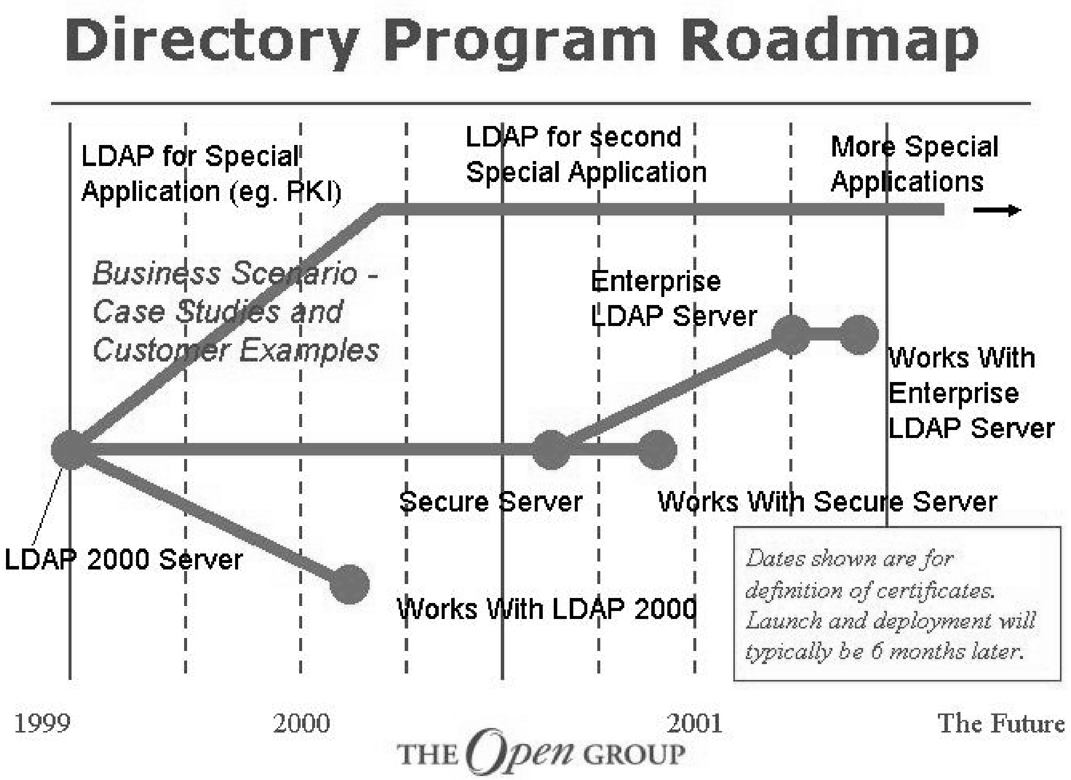
> The Products interoperate with LDAP servers in order to perform the functions set forth in Schedule X. Each of The Products will correctly perform these functions when interoperating with any LDAP Server that conforms to The Open Group LDAP 2000 Product Standard provided that the environmental conditions set forth in Schedule Y are satisfied.

Schedules X and Y are specific to the products in question. The contents of them are supplied by the vendor on the web form when applying for certification.

Roadmap for Evolution

Introduction

Evolution of the Open Brand for LDAP 2000 and of the “Works With LDAP” certificate will be the responsibility of The Open Group Directory Program. Figure 12 shows the Directory Program Roadmap.


Directory Program Roadmap
LDAP for Special Application (eg. PKI)
LDAP for second Special Application
More Special Applications
Business Scenario - Case Studies and Customer Examples
Enterprise LDAP Server
Works With Enterprise LDAP Server
Secure Server
Works With Secure Server
LDAP 2000 Server
Works With LDAP 2000
1999
2000
2001
The Future
THE Open GROUP

Dates shown are for definition of certificates. Launch and deployment will typically be 6 months later.

Figure 12: The Directory Program Roadmap

Where We Are Now

The Product Standard for the Open Brand for LDAP 2000 was agreed by the Directory Program Group and approved by the Architecture Board of The Open Group in December 1999. The requirements of the Product Standard are summarized in the section *The Open Brand for LDAP 2000* above.

However, definition of a brand is not in itself a meaningful event. What is needed to make it meaningful is commitment of vendors to register products conformant to the brand. That commitment was reached in the second quarter of 2000. The Open Group has commenced development of the VSLDAP test suite, and there are vendors working towards product registration. This enables the formal launch to take place at the end of June.

Six months is a reasonable length of time to launch a certificate, following its definition. We can expect that launch of the other certificates shown on the roadmap will follow at least six months after definition of those certificates has been completed.

The Next Step – "Works With LDAP"

The next step is to complete the delivery of assurance of directory interoperability at the basic level by defining, and then launching, the "Works With LDAP" program.

The "Works With LDAP" definition was agreed in outline by the Directory Program Group at the April Open Group conference. "Works With LDAP" will be one of a range of Open Group "Works With" programs. These programs will have a common legal framework and registration infrastructure. The definition of this common structure must be completed, and the aspects specific to "Works With LDAP" must be finally agreed by the Directory Program Group.

One particular aspect – links to server vendors' application certificate programs – must be addressed both at the general and at the LDAP-specific level.

The most important LDAP-specific aspect is the definition of the *Quality Standard* – what the vendor must guarantee. That definition is agreed, and is quoted in this White Paper under *"Works-With LDAP" Certification – Legal Framework* above.

The other LDAP-specific aspects are largely covered by the definition of the "Works With LDAP" application form and checklist. A draft that incorporates comments received to date is given in Appendix B to this White Paper. The Directory Program Group will work on this draft, with the aim of reaching agreement by its meeting in October 2000.

Extending The Open Brand for LDAP 2000

In defining the LDAP 2000 Product Standard, the Directory Program Group recognized that the server functionality that it covers does not meet the full requirements for an Enterprise Directory Server. The Open Group does not define the Directory standards; it references standards defined by other bodies, particularly the IETF. Those standards are not at this point sufficiently mature. Nevertheless, the Open Brand for LDAP 2000, which covers those basic areas where standards are mature, is still valuable. Its principal shortcomings are in the areas of security and server-server interoperation.

There has been some progress on security. In February 2000, the IETF IESG approved the "Authentication Methods for LDAP", "LDAP v3 Extension for Transport Layer Security", and "Using Digest Authentication as a SASL Mechanism" Internet Drafts as Proposed Standard RFCs, and approved the "Access Control Requirements for LDAP" Internet Draft as an Informational RFC.

Within the IETF, server-server interoperation is addressed by the ldup working group. The original terms of reference of that group largely centered on directory replication (see the group charter at http://www.ietf.org/html.charters/ldup-charter.html). It now looks as though Directory Synchronization will be added to their remit.

The knowledge references to support server-server co-operation through referrals and continuation references are, however, being defined by the IETF ldapext working group. (This group was also responsible for the security RFCs mentioned above. See its charter at http://www.ietf.org/html.charters/ldup-charter.html).

The ldup working group has produced a Requirement Statement, an Architecture Definition, and an Information Model for replication, as Internet Drafts.

At its April 2000 meeting, The Open Group Directory Program considered whether the time was right to start work on LDAP Product Standards including security or server-server interoperation, and decided that the standards work was not yet sufficiently mature. The Program Group will return to this question in future meetings.

Certification for Directory Applications

Extensions to the Open Brand for LDAP 2000 will be mirrored by corresponding extensions to "Works With LDAP". The ultimate aim is to have an Enterprise LDAP Server brand and corresponding "Works With Enterprise LDAP Server" program that will deliver assurance of interoperability between servers that meet the full range of enterprise server requirements, and the applications that use them.

The Open Brand for LDAP 2000 and "Works With LDAP" program as currently proposed are not in any way specific to particular applications. It may be that there will be value in defining certificates that are specific to particular applications.

For example, the need to support PKI places certain requirements on the Directory, such as the need to support common PKI-related schema definitions. There might be value in making support for these schema an identified option of the LDAP Server Brand, and defining an application certificate for applications that can work with such directories to provide PKI functions.

At present, we do not understand the application space sufficiently well to know whether it will be valuable to define specific brands for particular applications and, if so, for which applications.

Our understanding of directory applications and their deployment in The Directory-Enabled Enterprise will be developed by working on case studies and customer examples. This will be done within the Business Scenario framework. The initial – and not yet complete – version of the Business Scenario for The Directory-Enabled Enterprise is contained in the first part of this White Paper. This scenario will be broadened and deepened by exploring case studies and customer examples in special application areas, such as PKI.

The development of the Business Scenario will thus not only communicate the Program Group's understanding of The Directory-Enabled Enterprise but will also determine the requirements for future extension of the certification and testing programs that deliver assurance of interoperability.

Appendixes

Appendix A: Certificates and the Corporate PKI Directory

Introduction

This Appendix describes how certificate information is held in a directory for a corporate PKI, and identifies some of the problems that can occur.

A PKI Repository

Definition of the Repository

PKIX definition in RFC 2459: a system or collection of distributed systems that stores certificates and CRLs and serves as a means of distributing these certificates and CRLs to end entities.

(Note that the official wording is *repository* and not *directory*; a repository can be a directory, but can also use other technologies.)

CRL: Certificate Revocation List (a list of revoked certificates)

End entity: User of PKI certificates or end-user / system that is the subject of a certificate

Information Stored in the Repository

Two kinds of information will be stored in the repository:

1. certificates, and
2. information about revocation of certificates, which means a list or certificates which are no longer valid for several reasons.

It is very important to see clearly:

- who / which instance creates / updates this information,
- with whom / which instance this information is associated or to whom / which instance this information "belongs", and
- who / which instance reads and uses this information.

Instances and Roles

In order to understand this better, we will extract the three important instances of a PKI which are relevant for the simplest scenario:

1. the CA, certificate authority

2. one end entity — who we will call Bob, for whom the CA creates a certificate, and

3. another end entity — Alice, who wants to use Bob´s certificate for her secure communication with Bob.

Note : The simplest scenario involves only one CA. A complete scenario should involve several CAs which work together (cross-certification).

Basically the roles are:

- Bob wants to have a certificate and expresses this request.
- Only the CA creates the certificate.
- Only the CA also updates the certificate if necessary.
- Only the CA creates and updates CRLs.
- Alice only retrieves the certificate and revocation information in order to validate the certificate when she wants to communicate securely with Bob. She will need some means in order to find the right certificate and the complete revocation information which is necessary for validation.

Ownership of Information

How is the information associated with the PKI instances and stored in the repository? Basically, several objects are defined in the repository and the information to store is considered as a property of the object:

- Bob´s certificate is associated with the object Bob.
- Revocation information is associated with the object CA.
- The CA itself needs to have its own certificate; this *CA certificate* is also associated with the object CA.

The Directory Schema

If the repository is a directory, the directory concepts of *Entry* and *Attributes* can be used. The following entries will be used in the directory:

- the CA´s entry, and
- Bob´s entry.

The information associated with these objects will be mapped to attributes of these entries:

- Bob´s entry will have the attribute “Certificate”, and
- the CA´s entry will need the attributes “Certificate Revocation List” and “Certificate”.

Now the directory has additionally the concept of *object class.* An object class allows control over which attributes can be placed in which entries.

All objects belonging to the same object class have the same characteristics. The definition of an object class prescribes which attributes are allowed for an entry if this entry has just that object class:

- Some attributes are *mandatory,* which means that they have to be present if an entry has this object class.
- Other attributes are *optional*, which means that they are allowed for the entry, but need not always be present.
- Attributes that are not contained in the definition of the object class are not allowed in the entry.

Every entry has one or more object classes.

All these means of controlling the information in the directory, including additional aspects such as syntax of attributes, are collectively called *directory schema*.

This is not only a characteristic of directories. All good databases have schema: this is the set of rules that control all aspects of what can be put into the database.

When an entry has to be created or modified, the server will check if the request for creating / modifying the entry is compliant with the rules of the schema in force.

And all good databases will reject requests that are not compliant with the defined schema.

Schema Problems

Problems arise because:

- The Directory server is not administered correctly and does not support the necessary schema elements.
- Some PKI implementations do not understand the X.500 schema concept.
- Some LDAP servers have too few or no schema control at all.
- The schema defined for PKI in the second (1993) and third edition (1997) of X.500 was too inflexible and made too much use of mandatory attributes.

Handling of Certificates

Schema Elements for Certificates

Please note that the schema information contained in this illustration is not complete; it is limited to the needs of the scenario and its comprehension.

Schema Elements in Second and Third Edition of X.500 (X.520, X.521)

For Bob´s entry:

Object class **Strong Authentication User**
Mandatory attribute : User Certificate

Attribute : **User Certificate** with Syntax Certificate, attribute is multi-valued

For the CA´s entry:

Object class **Certification Authority**

Mandatory attributes: CA certificate
Certificate Revocation List

Optional attribute: Cross certificate pair *(not discussed here)*

Attribute : **CA Certificate** with Syntax Certificate, attribute is multi-valued

Schema elements in fourth edition 2000 of X.500

For Bob´s entry:

Object class Strong Authentication User *is obsolete*

Object class **PKI User**

Optional attribute : User Certificate

Attribute : **User Certificate** with Syntax Certificate, attribute is multi-valued

For the CA´s entry:

Object class Certification Authority *is obsolete*

Object class **PKI CA**

Optional attributes: CA certificate
Certificate Revocation List
Cross certificate pair *(not discussed here)*

Attribute : **CA Certificate** with Syntax Certificate, attribute is multi-valued

Schema Elements in PKIX Documents

In RFC 2587: Internet X.509 Public Key Infrastructure - LDAP V2 Schema, only the object classes defined in the fourth edition, PKI User and PKI CA are used.

Schema Elements Actually Used in PKI Implementations

The latest integration tests with several PKI implementations showed that the schema elements defined in the second and third edition of X.500 are still used. This caused several problems. Using the new object classes PKI User and PKI CA, which only have optional attributes, will considerably reduce these problems.

What the Directory Administrator Has To Do

The schema elements used by the PKI implementation have to be supported and administered in the directory server that is used.

Normally a directory product is delivered such that every DSA already supports a number of standard schema elements and such that the directory administrator disposes of tools allowing him to extend the schema.

Before running the PKI application, the administrator has to make sure that the necessary schema elements are available in the directory server.

If it is not the case, the server will normally not allow him to add the certificate attribute to an entry.

What the CA Does

After having created a certificate for a user or for the CA itself, the CA will want to publish this certificate. This means that the CA will try to add the attribute “user certificate” or “CA certificate” to the corresponding entry. It will first try to find if this entry exists, and then behave differently if the entry exists or not.

The operational protocol recommended by PKIX with a directory is LDAP. Nevertheless it is possible to use the X.500 DAP, and some PKI implementers support both protocols. In both cases the strategy is the same.

Searching an Existing Entry

The CA will use the directory operation “search” to see if the entry exists.

The search request normally contains a base object, often the root of the tree where the entry should be, and a filter containing characteristics of this entry. This is normally the value of one or more existing attributes of this entry: for instance, the e-mail address.

Note: The behavior of the CA could be different for a user or for its own CA entry.

Entry Exists

If the entry exists, then the CA will use the directory operation “modify entry” to add the attribute “certificate” to the entry, or add a value to this attribute if it already exists.

If the attribute already exists, the steps are the same for the old and the new object classes.

If the attribute does not exist, adding this attribute has to be done in different ways, due to the fact that the old object classes have mandatory attributes.

Use of object classes “Strong Authentication User” and “Certification Authority” with their mandatory attributes.

If it is Bob´s entry, a user entry, and the entry has no attribute “User certificate”, then this operation must simultaneously:

- add the object class “strong authentication user”, and
- add the attribute “user certificate” with a valid value.

It is **not** possible to use two separate operations to perform these steps.

If it is the CA´s entry, and the entry has no attribute “CA certificate”, then this operation must simultaneously

- add the object class “strong authentication user”, and
- add the attributes “CA certificate” and “Certificate Revocation List”, each with a valid value. (The value for the Certificate Revocation List can be empty.)

It is **not** possible to use separate operations to perform these steps.

Use of object classes “PKI User” and “PKI CA” with their optional attributes.

In this case, due to the absence of mandatory attributes, there are several possibilities for performing the necessary steps.

If it is Bob´s entry, a user entry, and the entry has no attribute “User certificate”, then this operation can simultaneously:

- add the object class “strong authentication user”, and
- add the attribute “user certificate” with a valid value.

It is also possible to use two separate operations to perform both steps.

If it is the CA´s entry, and the entry has no attribute “CA certificate”, then this operation can simultaneously:

- add the object class “strong authentication user”, and
- add the attributes “CA certificate” and “Certificate Revocation List”, each with a valid value. (The value for the Certificate Revocation List can be empty.)

It is also possible to use separate operations to perform these steps.

Entry Does Not Exist

If the entry does not exist, then the CA has to use the directory operation “add entry” to create this new entry.

In this case the handling of mandatory attributes has to be done the same way as for the “modify entry” operation: object class and mandatory attributes have to be contained in the “add entry” operation.

In addition, the name of the entry has to be clear to the CA, and the form of the name has to be allowed by the server.

Problems That Can Occur

Schema Elements are Not Available in Server

In this case, the server will reject every “add” or “modify entry” operation.

Adding an Attribute Without Having Added the Object Class

When the new object classes are used, it is possible to add the object class and the attributes in several steps. But the object class has to be added before the attributes. If it is not the case, the attribute is not allowed for this entry and the server will reject the request.

Note: The X.500 “content rules” mechanism allows another way of administration here, but is not recommended.

Handling of Mandatory Attributes

If the steps described under *Use of object classes “Strong Authentication User” and “Certification Authority” with their mandatory attributes* are not done exactly as defined, the server will reject the request with an error such as “object class violation”.

Name Form Not Allowed

The directory administrator has always to define the structure of the tree. One part of this is which components may be used in order to build the name of an entry. If the CA does not use a correct name in the “add entry” operation, the server will reject the operation with an error such as “naming violation”.

CA Has Insufficient Access Rights

All directory servers have ways to administer access control rights to the information contained in the database. X.500 has its own complete access control concept.

In order to be able to perform “add” or “modify entry” operations, the administrator must give the CA the corresponding rights. The CA has then to bind to the server with the corresponding credentials to make sure that it gets the necessary rights.

If the CA does not have the rights, an “add” or a “modify” operation will be rejected with an error such as “insufficient access rights”.

Adding Several Certificates to an Entry

If the entry has already one certificate attribute, the CA should have the possibility to add a new value to this attribute. Two problems could then arise:

1. *administration error*: the attribute is administered as single-valued attribute and not as multi-valued attribute.

   In this case the server will reject the request with an error such as “constraint violation”.

2. *application error*: the CA uses with the “modify entry” operation the functionality “add attribute” and not the functionality “add value”.

   In this case the server will reject the request with an error such as “attribute already exists”.

Appendix B: Draft "Works With LDAP" Application Form

Introduction

This is a draft of the form to be used by vendors applying for "Works With LDAP" certification.

The Products

The information that you enter in this section will be displayed as part of your entry in the register of products certified to Work With LDAP.

We apply for the "Works With LDAP" certificate for the following products.

The version of each product should be identified. Different versions of a product should be listed separately.

More than one product, and more than one version of each product, can be covered by a single application. Different families of products should be covered by different applications. It is up to you to decide what constitutes a family.

As a practical guide, if you find that many of the answers to the questions take the form "For product A so-and-so and for product B such-and-such" then you should be putting in separate applications for products A and B.

Product Functions That Use LDAP

The information that you enter in this section will be displayed as part of your entry in the register of products certified to Work With LDAP, and will form part of your legal agreement with The Open Group.

The following functions of the products use LDAP:

You should list all of the product functions that use LDAP.

Functions should be described at an outline level. Examples are:

- *for an e-mail client — “Looks up recipients' e-mail addresses”, and*
- *for a VPN Firewall — “Searches a directory for cross-certificates in order to authenticate certificates presented by parties requesting access to the virtual network”.*

Circumstances Under Which the Products Work With LDAP

The information that you enter in this section will be displayed as part of your entry in the register of products certified to Work With LDAP, and will form part of your legal agreement with The Open Group.

Does the product require the server to support any optional features of the LDAP 2000 Product Standard?

□ Extensible Match

□ Notice of Disconnection

□ Client Modification of Subschema Entries

□ Validation of Client SSL Certificates

□ Access to SSL Credentials via SASL EXTERNAL .

See the LDAP 2000 Product Standard for definitions of these features.

Which of the following naming schemes does the product support?

□ X.500 Naming

□ Domain-Component-Based Naming (see RFC 2247).

What schema definitions does the product require the server to support?

Any such definitions should be identified, and URLs for them should be provided. For example, a statement might be: “The server must support the attribute and syntax definitions in IETF RFC 2256 — see http://www.ietf.org/rfc/rfc2256.txt.”

The definitions do not have to be official standards. They can be company- or product-specific. But they must be public and stable. Version numbers and issue dates should be given where applicable.

Draft “Works With LDAP” Application Form

Does the product require authorized users to have directory entries?

□ Authorized users must have directory entries.

If you checked the box above, where must such entries be in the DIT?

Does the product rely on any particular form of access control?

□ The server must have access control.

For example, does it assume that some entries or attributes are read-only? Does it assume group-level access control?

If you checked the box above, what access control is required?

Are there any extensions to basic LDAP functionality that the product can use?

The product must not require servers to support any functionality beyond that defined in the LDAP 2000 Product Standard. But if there are extensions that improve its operation, they can be listed.

Is there anything else that the product requires servers to support?

Customers should be certain that they can use the product provided that their servers conform to the LDAP 2000 Product Standard and provided that they satisfy any requirements stated in this section of the “Works With LDAP” application form. Any requirements not covered by the previous questions should be listed.

Tests Performed

The information that you enter in this section will not form part of your legal agreement with The Open Group. It will be made publicly available on the web for the benefit of customer organizations that wish to view it. It is kept as a record of the fact that you have done adequate testing, and of what that testing was. If necessary, the tests may be repeated if there is a question raised about the working of your products.

The testing assumed by the questions in this section is the minimum that a responsible vendor will carry out as part of product development.

The tests should be carried out against several different standard servers. There are no minimum requirements for this, but it is clearly something that application vendors will want to do to assure themselves that their products really Work With LDAP.

What tests have been done to ensure that the product works with standard LDAP servers?

A set of publicly available tests, such as BLITS, is desirable. In-house tests are acceptable, provided the vendor deposits a test suite specification with The Open Group and is prepared for The Open Group to disclose it to third parties.

What Protocol Features areTested?

Protocol Feature	Used by Products to perform functions listed in this application	Covered by Tests
Operation over TCP	□	□
Operation over SSL	□	□
Anonymous bind	□	□
Bind with simple password	□	□
Bind using SASL EXTERNAL	□	□
Unbind	□	□

Draft “Works With LDAP” Application Form

Search with simple (single-term AVA) filters	□	□
Search with complex filters	□	□
Search using 3-valued logic	□	□
Search retrieving operational attributes	□	□
Search requiring alias de-referencing	□	□
Modify - add	□	□
Modify - delete	□	□
Modify - replace	□	□
Add entry	□	□
Delete entry	□	□
Modify DN - rename leaf	□	□
Modify DN - move leaf to new parent	□	□
Modify DN - rename subtree	□	□
Modify DN - move subtree to new parent	□	□
Compare	□	□
UTF-8 Characters (including non-ASCII)	□	□
Referrals	□	□
Continuation References	□	□

Draft “Works With LDAP” Application Form

Do the tests cover all of the Circumstances Under Which the Products Work With LDAP that you have described above?

● Yes ○ No

If the answer to the last question was “No”, what circumstances were not covered, and why?

What is the test coverage?

□ All success cases are tested.

For each function listed under “Product Functions that Work With LDAP”, successful operation of that function should be covered by the tests.

□ All important error cases are tested.

Tests should cover not only successful operation but also all commonly arising error situations.

□ All required LDAP operations are tested.

Each LDAP operation — bind, search, etc. — that is used by the product should be covered by the tests.

□ All schema branches are tested.

The tests should exercise every branch of the directory schema that the product uses.

If you did not check all of the boxes in the previous question, what is the reason for that lack of test coverage?

Is there anything else you would like to say about the tests performed?

If there is anything else that you think people should know about your testing, please say it here.

Submission

When you are satisfied with your answers to the questions, click on the “Submit” button.

The answers in the sections *Product Functions that Work With LDAP*, and *Circumstances Under Which the Products Work With LDAP* will become schedules to a legal agreement with The Open Group which, when signed and countersigned, will entitle you to claim “Works With LDAP” certification for the products that you listed in the section *The Products*.

Submit Reset

Appendix C: Rationale for the “Works With LDAP” Definition

Background

At the July 1999 Open Group Conference in Montreal, members of the Directory Interoperability Forum (DIF) put forward a strong and clear requirement for The Open Group to design and manage certification schemes for Directories and for Directory-Enabled Applications.

The Open Group responded by defining the LDAP 2000 Open Brand for Directories. Products that conform to the LDAP 2000 Product Standard are eligible for certification under this scheme.

The companion scheme for Directory-Enabled Applications — the “Works With LDAP” scheme — took longer to define because there was no widely accepted pre-existing model. At the January 2000 Open Group Conference in San Diego, The Open Group presented a framework for this scheme, and the discussion of how to apply that framework was started.

Following the San Diego meeting, The Open Group circulated two alternative proposals, which were discussed by a teleconference of interested parties. The second of those alternatives was chosen at the teleconference.

That alternative was presented at the London meeting of the Open Group Directory Program in April 2000, and was there agreed in principle by the Program Group.

Value of the Scheme

The value of the scheme is that it:

- meets the needs of Customers for proof that the products they buy will interoperate
- meets the needs of Vendors to provide that proof to Customers, and
- by increasing Customer confidence, enables Vendors to sell more products, earlier.

Requirements

The principal requirements are:

- Customers must find certification **easy to understand**.
- Vendors that deserve certification must find it **easy to acquire**.
- It must be **inexpensive**. The costs will be covered by application vendors' registration fees. These must be kept low.
- It should have **links** to server vendors' programs for certifying applications.

There are a host of applications that can benefit from being directory-enabled. They include e-mail, virtual private networks (VPNs), distributed systems management, human resources management, and others. From the point of view of these applications, directory is not the main purpose — it is a useful tool that helps that purpose to be attained more easily. Certification must be usable and valuable in this context.

Implementation Framework

A framework within which the scheme should be implemented was agreed at the San Diego Open Group Conference in January 2000. This framework covers delivery and legal aspects of the scheme.

The principal means of delivery will be the World-Wide Web. This will provide:

- a repository for documents, logos, etc.
- a register of certified products
- a list of Vendors that are in the program, and
- information for Customers.

The legal basis of the scheme will be as follows. There will be a logo associated with the scheme. The Open Group will own the logo, and will register it as a trademark as widely as is practical. Vendors will be licensed to use the logo on products that are certified under the scheme. The license will require them to "warrant and represent" that the products conform to certain quality standards. The license will provide for the right to use the logo to be withdrawn if a product does not conform.

Quality Standards

The quality standards state what precisely a product must do to be certified. (The quality standards chosen for Works With LDAP are presented in the *Legal Framework* section of Part 3 of this White Paper.)

There are three aspects to the Quality Standards:

- What must the Application Vendors state about their products?
- How do the Application Vendors satisfy themselves that their products do what they say?
- How do the Application Vendors satisfy The Open Group that their products do what they say?

Criteria for Selecting the Quality Standards

The following criteria, in order of importance, should be used to define the Quality Standards:

1. *Achievability* — the certification must be realistic.

2. *Effort from server vendors* — certification should not rely on effort from server vendors.

3. *Value to Customers* — the more valuable it is to Customers of Directory products, applications and services, the better.

4. *Administration cost* — must be covered by the fees charged, and the lower the better.

5. *Effort from application vendors* — the less required, the better.

What the Vendor Guarantees

With traditional Open Group certification, including the Open Brand for LDAP 2000 brand for servers, the vendor guarantees that the product conforms to an independent standard. (In the case of LDAP 2000, this is the LDAP 2000 Product Standard, which incorporates by reference IETF RFC 2251 and other published standards.)

The problem with applying this principle to “Works With LDAP” is that there are currently no standards for Directory applications, other than the LDAP protocol standards. The Application Vendor should of course guarantee that the product conforms to these. However, this guarantee will not in practice provide a high degree of assurance that the products will really interoperate with LDAP servers. The reasons for this include:

- An application may operate the standard LDAP protocol correctly but also rely on non-standard features.
- An application may have particular schema requirements that some servers can not meet.
- An application may assume a particular access-control model that some servers do not support.

The options for defining what the vendor guarantees beyond basic LDAP conformance are as follows.

Independent Standard

Standards for application behavior could be created. This might be possible for a few specific applications (e.g., “E-mail client that Works With LDAP”) but is not realistic for generic “Works With LDAP” certification.

Vendor-Defined Specification

Each participating vendor could state how their products Work With LDAP. This would require effort on the vendors’ part, but would not necessarily mean significantly higher administration costs. The Vendor statement could include information such as how functionally they use LDAP, what optional features of the LDAP 2000 Brand they rely on, what protocol features not included in LDAP 2000 they use, what schema they require, and what access control model they assume.

Basic LDAP Conformance

The scheme could simply require conformance to the basic LDAP protocol requirements. This would be the easiest option. The question is whether it would give the scheme sufficient value in the eyes of Customers.

The principle of the **Vendor-Defined Specification** was selected because it gives the certificate more value than Basic LDAP Conformance and is realistic.

Rationale for the “Works With LDAP” Definition

How the Vendors Know They Work With LDAP

In traditional Open Group certification, there is normally a test suite. The stated purpose of this is to be an indicator of conformance: that is, to be part of what tells Customers and others that the product conforms. However the test suite also provides an equally valuable — perhaps more valuable —function. It gives the vendor assurance that the product conforms, and that the guarantee that certification requires can safely be given.

The following options might be used to provide this function for “Works With LDAP”.

Independent Test Suite

Defining a test suite for applications in general — even to cover only basic protocol requirements — is not possible. Creating independent application-specific tests for all applications that vendors might wish to certify would be impractical.

Vendor-Defined Tests

Each participating vendor could define their own tests to prove that their products Work With LDAP. This would require effort on the vendors' part, although they would presumably define tests for their products even if not participating in the scheme, and the requirement to document such tests for the scheme need not be too heavy an additional burden. The tests they use could be publicly available via the “Works With LDAP” scheme Web pages. This option would not necessarily mean significantly higher administration costs.

Interoperability Test Bed

Server Vendors could make LDAP-2000-conformant servers available for interoperability testing by application vendors. This would be a valuable means of assurance for the application vendors. It would also be a valuable addition to the *Vendor-Defined Tests* option, as it would provide a neutral platform to run such tests on.

DirConnects

While not maintaining a permanent test bed, Server Vendors could offer Application Vendors the opportunity to test against their products at regular DirConnect interoperability testing events. Again, such testing should not be required, but should be encouraged.

Use of a Standard SDK

It would be possible to define standard tests — and even a certification scheme — for LDAP Client Software Development Kits (SDKs). Use of a certified or tested SDK would at least give the Application Vendor assurance that the product conformed to basic protocol requirements. This would be a simple option but would exclude vendors that write their own LDAP-handling software. It would not require high administration costs per se, but would mean establishing a separate scheme for SDKs, thus introducing an extra dimension of complexity.

Source Checker

If use of a standard SDK is a criterion, The Open Group Source Checker might be used to verify that the SDK is used correctly and that no APIs outside the SDK are used by the application to communicate with the LDAP server.

No Assurance

It would be possible to operate the scheme even if it did not include any way by which vendors could check product conformance. This would be the easiest option. It would however result in lower numbers of certified products or lower quality of certified products (or perhaps both).

Defining an independent test suite was rejected as impractical. Making an Interoperability Test Bed or DirConnects a formal part of the scheme would require significant effort from the server vendors, and probably a legal agreement between them and The Open Group, and were both therefore rejected. (As a separate exercise, vendors should be encouraged to participate in such a test bed and in DirConnects.) Making the Source Checker usable for all applications (Java as well as C, for example) would not be practical.

The principle of **Vendor-Defined Tests** was selected because, of the remaining options, it gives the highest value while being practical to implement.

How The Open Group Knows the Vendors Work With LDAP

Evidence of the application vendors' claims could be given in various ways.

Independent Testing

Testing could be performed by The Open Group or by an independent third party. This would provide a high degree of assurance but would have very high administration costs.

Independent Audit

Testing or other procedures performed by the Application Vendors to provide assurance that their products Work With LDAP could be audited by The Open Group or by an independent third party. This would be less expensive than independent testing and could provide almost the same degree of assurance. It would however impact the registration fee to some extent — perhaps by around $1K per registration.

Spot Checks

Rather than auditing every registration, there could be spot-checks on (say) one in ten, chosen at random — one tenth of the cost of auditing, but a lower degree of assurance.

Vendor Statement

The Application Vendors could provide statements of what tests or other procedures they had carried out. This would mean some work for the Application Vendors, but no impact on the registration fee. The quality of these statements would probably be highly variable, but could be made clearly visible to the Customers.

Vendor Assurance

The Application Vendors could simply give the assurance that their products Work With LDAP, without giving any further information to back this claim. This would be simplest and cheapest but would give the scheme little added value — most vendors will presumably make such claims in any case.

Independent testing and independent audit were both rejected as giving too high a cost of administration. Vendor assurance does not give sufficient value.

The **Vendor Statement** principle was selected.

It would be possible to add Spot Checks to this, but it is doubtful whether the additional assurance gained would be worth the added cost and complication.

Glossary

ACL	Access Control List
API	Application Program Interface
BLITS	Basic LDAP Interoperability Test Suite
CA	Certificate Authority (in a *PKI*)
Chaining	A mode of interaction that may be used by a directory server that can not perform an operation itself. The server chains by invoking an operation of another server and relaying the outcome to the original requestor.
Continuation Reference	A continuation reference describes how the performance of all or part of an operation requested of a server can be continued at different servers. See also *referral*.
CRL	Certificate Revocation List
DAP	The X.500 Directory Access Protocol
DBMS	DataBase Management System
DEN	Directory-Enabled Networks
DHCP	Dynamic Host Configuration Protocol (of the Internet). See IETF RFC 2131.
DIF	Directory Interoperability Forum
DIT	Directory Information Tree. The set of entries in a directory form a tree. Each entry (except the root entry) has a single superior entry and may have one or more subordinate entries. See also *RDN*, *Leaf Entry*.
DN	Distinguished Name. In a directory, the unique name of an entry formed by the concatenation of the *RDN*s of the entry and each of its superior entries.
DNS	Domain Name Service (of the Internet). See IETF RFC 1035.
End Entity	User of PKI certificates, or end-user / system that is the subject of a certificate.
Extranet	Extension of an organization's *intranet* that includes its business partners and other correspondents.
Firewall	Device or set of devices that protects an organization's network by filtering traffic between it and the global Internet.
GP	General (medical) Practitioner, in the UK
HR	Human Resources
HTTP	The HyperText Transfer Protocol. See IETF RFC 1945.

Glossary

HTTPS Secure HTTP — obtained by running *HTTP* over *SSL*.

IESG The Internet Engineering Steering Group

IETF The Internet Engineering Task Force

IMAP Internet Mail Access Protocol. See RFC 2060.

Intranet The systems and networks that operate the Internet Protocol and belong to a single organization. They are often separated from the global Internet by a *firewall*.

IP The Internet Protocol defined in IETF RFC 791.

IT Information Technology

ITU-T Telecommunications Standardization Section of the International Telecommunications Union

LAN Local-Area Network

LDAP Lightweight Directory Access Protocol. See IETF RFC 2251.

LDIF The LDAP Data Interchange Format. A data format specification for directory contents, currently at draft status within the IETF.

LDUP LDAP Duplication / Replication / Update Protocols

Leaf Entry A directory entry that has no subordinate entries in the *DIT*.

Metadirectory Product that provides a uniform (LDAP) directory interface to collections of directory and other data storage products.

NHS (UK) National Health Service

ODBC Open DataBase Connectivity

PC Personal Computer

PDA Personal Digital Assistant

PKI Public Key Infrastructure

PKIX Public-Key Infrastructure (X.509) working group of the *IETF*

RA Registration Authority (of certificates, in a *PKI*)

RDN Relative Distinguished Name. In a directory, a name that uniquely distinguishes an entry from other entries with the same superior entry in the *DIT*.

Referral An outcome which can be returned by a server which cannot perform an operation itself, and which identifies one or more other servers more able to perform the operation

Replication The process by which copies of entries are made and maintained between servers.

RFC Request For Comment – generic name given to standards and other publications of the *IETF*.

Glossary

SASL Simple Authentication and Security Layer. See IETF RFC 2222.

SDK Software Development Kit – especially one that includes a client implementation of LDAP and provides LDAP functionality to applications via an API.

S/MIME Secure / Multipurpose Internet Mail Extensions. See IETF RFC 2633.

SQL Structured Query Language

SRV *DNS* Service Location Record. See IETF RFC 2782.

SSL Secure Sockets Layer protocol

T.61 A recommendation of the *ITU-T* that includes the definition of a particular character set used in telematic services.

TCP Transmission Control Protocol of the Internet, defined in IETF RFC 793.

TLS Transport Layer Security. An IETF-defined protocol based on *SSL*. See IETF RFC 2246.

UDP User Datagram Protocol of the Internet, defined in RFC 768.

UNICODE The Universal Character-Set Encoding defined by the UNICODE consortium, and a subset of the encoding defined in ISO 10646.

URL Uniform Resource Locator — colloquially, a *WWW* address. See IETF RFC 1738.

USERID User Identity

UTF-8 The File System Safe Universal Character Set Transformation Format defined in Open Group Technical Standard C501. A representation of the *UNICODE* character set.

VPN Virtual Private Network. See also *intranet*.

VSLDAP *LDAP* Verification Test Suite produced by The Open Group.

WAN Wide-Area Network

WAP The Wireless Applications Protocol

WWW The World-Wide Web (of *HTTP* servers on the Internet)

X.500 Series of recommendations for Directory Services defined by the *ITU-T*. Different versions of these recommendation have been produced in different years.

X.509 One of the X.500 recommendations – “X.509: The Directory: Authentication Framework”. Includes a data format specification for certificates.

Bibliography

Practical Guide to the Open Brand, January 1998, Open Group Document X981.

The Open Brand Trademark License Agreement (TMLA), January 1998, Open Group Document X982.

LDAP 2000 Product Standard, December 1999, Open Group Document X99DI.

IETF RFC 2251, Lightweight Directory Access Protocol (Version 3), December 1997.

IETF RFC 2252, Lightweight Directory Access Protocol (Version 3): Attribute Syntax Definitions, December 1997.

IETF RFC 2253, Lightweight Directory Access Protocol (Version 3): UTF-8 String Representation of Distinguished Names, December 1997.

IETF RFC 2254, The String Representation of LDAP Search Filters, December 1997.

IETF RFC 2255, The LDAP URL Format, December 1997.

IETF RFC 2459, Internet X.509 Public Key Infrastructure Certificate and CRL Profile, January 1999.

The SSL Protocol, Version 3.0, Netscape Communications Corporation, March 1996.

ITU-T Recommendation X.500, Information technology — Open Systems Interconnection — The Directory: Overview of concepts, models and services, August 1997.

Understanding and Deploying LDAP Directory Services, T. Howes, M. Smith, and G. Good, MacMillan, December 1998.

Directory-Enabled Networking, J. Strassner, MacMillan, October 1999.

Directory-Enabled Computing: The Directory's Expanding Role, Network Strategy Overview, The Burton Group, December 1999.

The Enterprise Directory Value Proposition, Network Strategy Overview, The Burton Group, February 1999.

Acknowledgements

Part 1: The Business Scenario

The scenario in its current state incorporates the work of a number of people, both within and outside the Directory Program Group. Particular thanks are due to the following:

Globalsign — Wim Hendrickx for his input.

Kaiser Permanente — Doug Shelton for his input.

Shell —Nick Mansfield for his support, Pauwl Lunow for his input, and especially Peter Harris for his patience in explaining Shell’s business processes and their use of Directory to support PKI.

Siemens — Patrick Fantou, for his input and for the contents of Appendix A, which is his work.

The UK NHS — Hugh Fisher for his input and support, and Ron Bissell and Andy Garcarz for their patience in explaining NHS requirements and thinking.

Finally, a special acknowledgement is due to Terry Blevins of NCR, who introduced the concept of the Business Scenario, led the workshops with the NHS and Shell, and developed specific scenarios from them. Without his help, this Business Scenario for The Directory-Enabled Enterprise would not have been possible.

Part 2: Understanding Interoperability Requirements

This part was originally prepared as a self-standing White Paper by Edwards Reed of Reed-Matthews Inc. The Open Group is pleased to acknowledge his authorship of it, and his contribution to our understanding. The author would like to express appreciation to the reviewers of a rough draft, for their insight and thoughtful comments. Written and oral feedback came from:

David Blair, of the UK Department of Social Services

Viv Danks, of the UK Defence Evaluation and Research Agency

Pauwl Lunow, of Shell

Bryan Littlefield, recently of NASA JPL, now at Oblix

Richard Paine, Boeing

Terry McFadden, Proctor and Gamble