Directory Replication: from Gigabit LAN to HF Radio

Steve Kille, Isode

1 Abstract

Directory replication is important for reliability, survivability, performance, and local client access. This paper examines a range of techniques for directory replication, appropriate to different scenarios, providing illustrations of some of the points raised from the Isode product set:

- X.500 DISP (Directory Information and Shadowing Protocol). The open standard for replication; Its functionality; and why it does not have to be complex to deploy.
- The benefits of single master replication, and scenarios where single master is the best solution.
- Dealing with Disaster Recovery: How to avoid a single master being a single point of failure.
- Why Strong Authentication is desirable for replication, and how a PKI based deployment can be straightforward.
- Synchronization: Replication with Active Directory and LDAP servers without standardized replication.
- Dealing with secure gateways and very poor communication links including HF Radio (down to 75 bits/sec); Support for Data Diodes and Radio Silence; Directory Replication by Email.

2 LDAP Directory & Replication

LDAP directories are typically used to store information on "standard" objects (people, accounts, devices; PKI components) using standard schema, often extended to address local requirements. This information is often access by a large number of heterogeneous clients. For all but the smallest deployments, replication is an essential part of the directory capability, providing resilience and local access to different directory users. High reliability directories will often have large numbers of replicated servers at different locations. So replication is a common requirement for LDAP directories.

The key benefit of the hierarchical data model of an LDAP directory is that it allows easy data distribution, with different parts of the Directory Information Tree (DIT) to be held in different servers. A knock-on consequence of this is that it enables replication of data to be partitioned, and so selective parts of the DIT can be replicated.

The LDAP (and X.500) directory model allows for information to be out of date. This allows for replication approaches that do not require complex multi-server locking approaches that are often needed for general purpose database replication. Because of this, directory replication is (relatively) straightforward to achieve. So LDAP is a good choice where there is a need to replicate data and a particularly good choice if high levels of replication are needed.

3 Open Standard Replication: X.500 DISP

The relationship between LDAP and X.500 has evolved:

- Initially, LDAP was designed as a lightweight protocol to access an X.500 directory server.
- Subsequent to this, standardization attempts were made to evolve LDAP into a complete directory service specification, complete with replication and access control, independent of X.500. This standardization attempt failed.
- LDAP has now settled down as a member of the X.500 protocol family and alternative to the DAP protocol for client access. The specification is based on the X.500 models. The X.500 standardization recognizes LDAP, and ongoing X.500 and LDAP standardization is coordinated to ensure that they remain compatible.

An LDAP server will inherently support the X.500 information model, but there is no requirement to support the X.500 protocols or access control.

A consequence of this is that the only open standard replication protocol for use with LDAP Servers is X.500 DISP (Directory Information Shadowing Protocol). DISP is a flexible and efficient directory replication protocol:

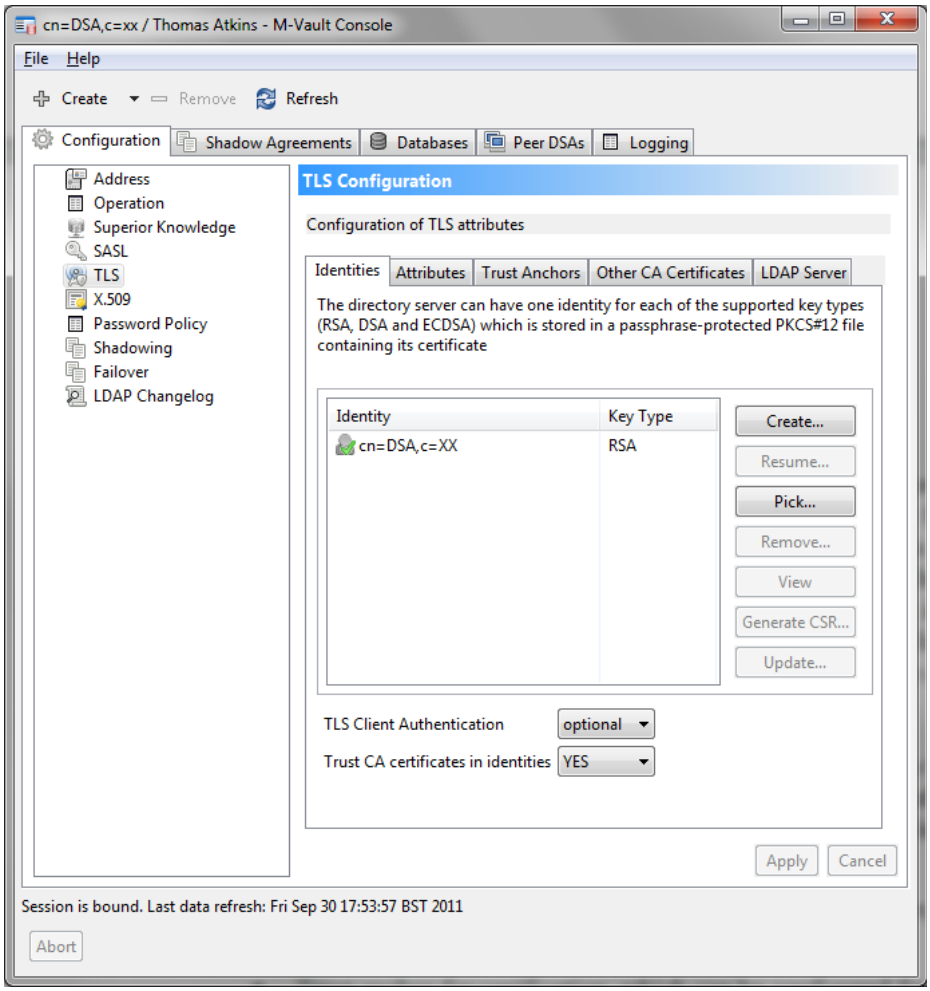
- Incremental replication, to optimize network and resource usage.
- Scheduled and on demand updates.
- Manager controlled updates.
- Supplier and Consumer initiated updates.
- Hierarchical shadowing.
- Flexible definition of data to be shadowed.
- Total updates, which allow for setup and reset.
- Flexible data filtering.
- Security features, including digital signature of associations and digital signatures of replicated data.

If you need to build a multi-vendor replicated directory, use of X.500 DISP is ideal.

4 Strong Authentication

Replication of data needs to be secure. While shared secret (password) based authentication can work well for client access in many situations, it is a poor choice for server to server authentication, as it is difficult to manage securely and passwords need to be set up and managed for each pair of servers. Strong Authentication is often a good choice for client authentication,

but is almost invariably the best choice for server authentication, including server to server authentication for replication. See "Why Strong Authentication for Directory?" http://www.isode.com/whitepapers/strong-auth-dir.html for more discussion.



A key benefit of strong authentication for server authentication is that setup is once per server. Once this is done, any pair of servers so set up can use this for authentication. It avoids the n-squared management problem of separately authenticating each link. This is very important for a large deployment. Configuration for each server needs:

- Private key and associated certificate. This can be soft (e.g.. PKCS#12 file), smart card, or API (e.g., CAPI or PKCS#15) that can access either.
- Trust anchor for verification, which can be configured for the directory or taken from the system.

Setup of these can be very straightforward, as shown in the above screenshot of M-Vault. Isode strongly recommends the use of strong authentication for all directory replication.

5 Single Master vs Multi-Master

There are two basic strategies to dealing with write access (data modification) in a directory:

- Single master. Where all changes are made to a single server (the master).
- Multi-master. Where changes may be made to one of several servers.

There are scenarios where multi-master is preferable, and others where single master is best. It is mistake to consider that multi-master is better, simply because it is more complex. The respective benefits of the two approaches are considered:

Advantages of a multi-master approach:

- Where there is a need to perform updates at multiple locations, and there is poor network connectivity between the locations. In a setup with poor network connectivity, multi-master becomes important if high availability of directory update is required from many locations.
- Automatic disaster recovery is possible.

Advantages of single master replication:

- There is definitive state. The data in the master server is authoritative, whereas in multi-master, the state depends on multiple servers, and cannot generally be determined.
- Client update is definitive. A related characteristic is that when a directory client makes a change to the directory, it knows that it has been made. For multi-master, a change may be overridden by a change made in another server. This can have bad effects for some data, such as system configuration.
- Complexities of conflict resolution avoided. Dealing with conflicting directory updates can be complex, and there needs to be an approach for dealing with directory updates that are rejected. The consequences of rejects can be awkward, particularly where multiple related changes are made, and one gets failed due to conflict.
- Efficient network use. Single master replication is much simpler and more robust than multi-master. This can be particularly important when operating over constrained networks.
- Efficient update. Replication gives natural performance gains for read and search. If updates are made on multiple servers, the individual updates still need to be made on each server and conflict resolution needs to be done in addition. This means that for large directories, multi-master systems will generally be operated as if they were single master, so that most updates are applied to one server and conflict resolution is minimized.
- X.500 DISP can be used. The only open standard for directory replication is X.500 DISP (Directory Information Shadowing Protocol). All multi-master protocols are proprietary. So single master is essential if you want to have open standard multi-vendor replication.

In Isode's experience, single master is a good choice for most directory deployments, and strongly preferable for many.

6 Disaster Recovery

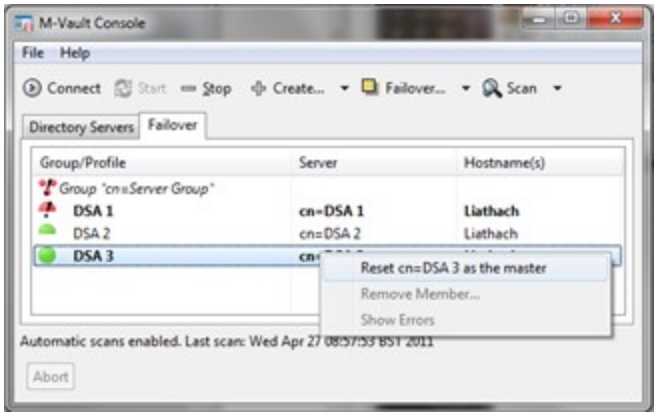
A key question for a single master architecture is "what if the master fails?". For many directory deployments, including those with extremely high requirements for read/search availability, requirements for availability to make updates (write availability) are lower. It is possible to make the master directory of a single master directory server have very high availability, using for example:

- RAID disks.
- Clustered server (to deal with server or processor failure).
- Independent power supplies and network connections.

With modern hardware, very high availability can be achieved, with modest scheduled downtime for software and hardware updates. It is usually straightforward to provide the level of write availability needed for most directory services.

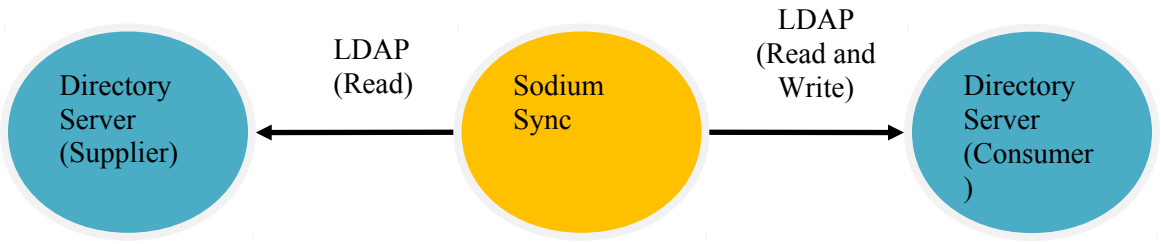
The scenario that this cannot address is catastrophic system failure and in particular site destruction (9/11 scenario). While such situations may be very unlikely, they must be considered in mission critical deployments.

Single master can handle disaster recovery by use of mirror servers, which are updated by the master, and can be switched into action in the event of catastrophic failure of the original master.



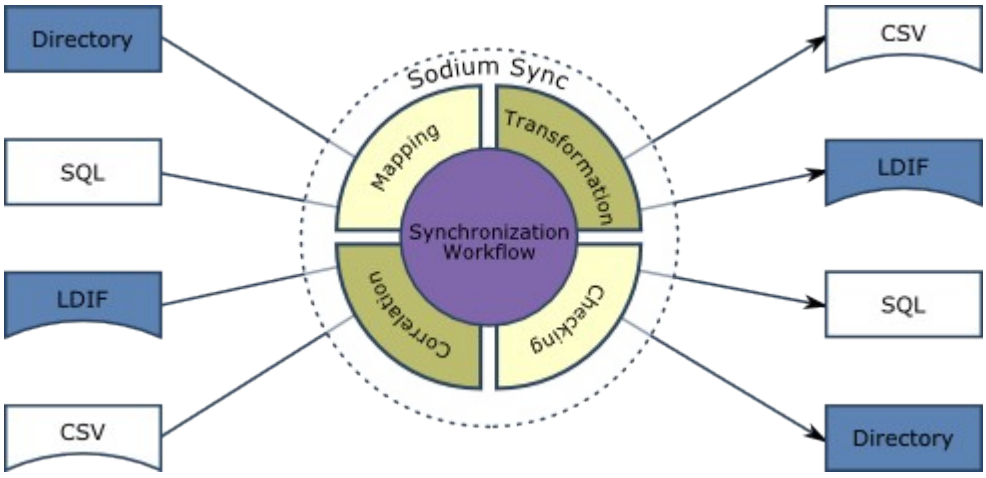
Isode's approach is to have operator controlled failover in the event of a disaster. More details of this, including handling of secondary shadowing is given in "M-Vault Failover and Disaster Recovery" http://www.isode.com/whitepapers/mvault-dr.html

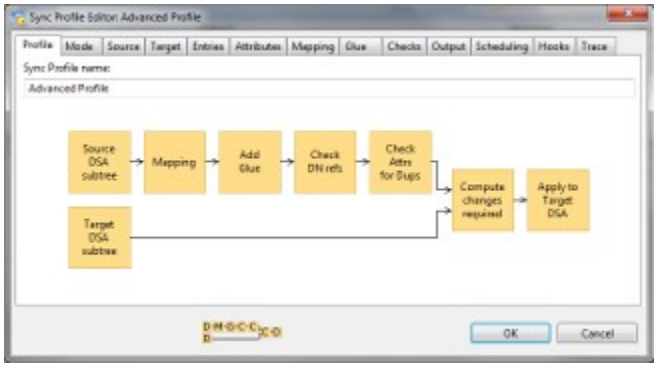
7 Synchronization: Replication with LDAP



An implicit assumption of the paper so far is that replication of data between a pair of directory servers needs a special protocol. There are substantial merits to use of a special protocol, but a key observation is that LDAP can both read and write data, and so can be used to move data between servers. The basic model is that a synchronization process reads data from one directory and writes it into another.

This basic architecture can be taken in two directions. First, it can be built on to provide value added services. Isode's Sodium Sync Product is a good example of this,





Sodium Sync enables mapping and merging between directory and directory-like data sources and targets. When directories are decoupled, there is often a requirement for complex transformation in addition to the basic replication. Details are given on http://www.isode.com/products/directory-sync.html

The model can also be used to build a system optimized for replication. If the supplying directory publishes information on changes, then data transfers can be optimized. Many directory servers do this, some based on the changelog specification. If the reading is done by the consuming directory (rather than an intermediate process), then system attributes (such as entry creation date) can also be replicated. There is no standard to do this, although a number of directory servers (including M-Vault) support a mechanism based on the draft changelog specification. As a replication mechanism, this is inadequate for a number of reasons:

- It is a polled mechanism, so there must be some delay in replication. Where changes are pushed, updates can be sub-second, which will rarely be noticed by the user.
- In practice, it is a single product approach.
- The access control model is messy, and the approach would be awkward for a large number of servers.

DISP is a better approach for "standard" server to server replication. LDAP synchronization is more usefully reserved for more heterogeneous deployments where data transformation is needed in addition and not just basic replication.

8 Filtered Replication

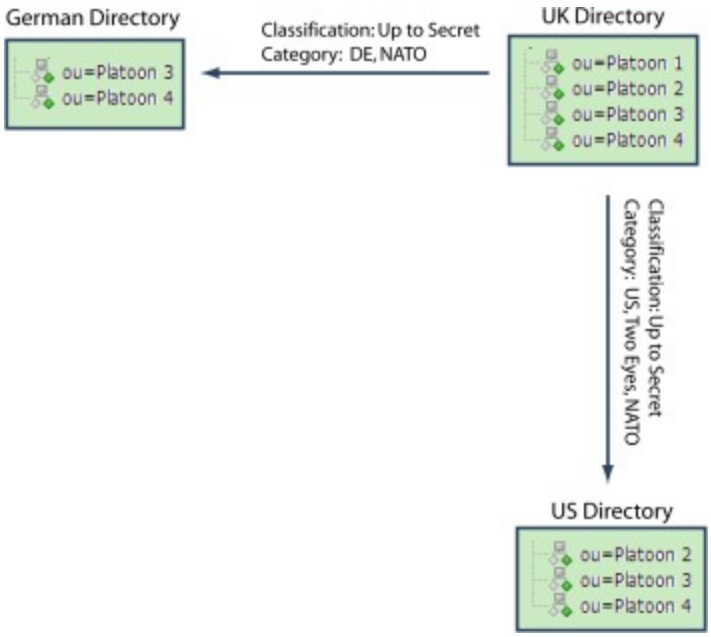
Directory replication is often used to share information with external parties, for example to share directory information with a partner organization. It is often desirable to share a subset of the full information and so it makes sense to filter the information. Isode provides two ways to achieve this:

- X.500 DISP provides a basic filtering capability, so that certain types of object (object classes), certain attribute types, or whole sub-trees can be excluded. This is a good approach for simple filtering.
- Sodium Sync provides a more generic and extensible capability to filtering and transformation, which is useful in more complex scenarios and where DISP is not available,

However, neither of these approaches works very well when the desire is to replicate selected information at the entry level (for example, an arbitrary subset of users in the directory). To achieve this with mechanical filtering requires too much information in the filter. An approach is needed to configure what is replicated along with the data.

A contrived approach to doing this would be to store information on propagation in each entry using a custom attribute that would be picked up by Sodium Sync. A more elegant approach would be to use access control on the entries. And then using Sodium Sync binding to the directory with a special user. This could be achieved using identity based access control, by giving the special user access only to the entries that can be replicated. This would be awkward to manage, as the access control would typically only be visible to a system administrator.

A cleaner approach would use Rule Based Access Control using Security Labels. Consider the example setup.



Security Labels associated with directory data can be used to control replication. This is illustrated above, with a UK military directory (server) replicating data to a US and German directory servers. The Security Labels (using Security Categories) and consequences for replication are explained in the table below:

Role	Security Label		Replicate to	
	Security Classification	Security Category	US	Germany
Platoon 1	Secret	UK	No	No
Platoon 2	Secret	Two Eyes	Yes	No

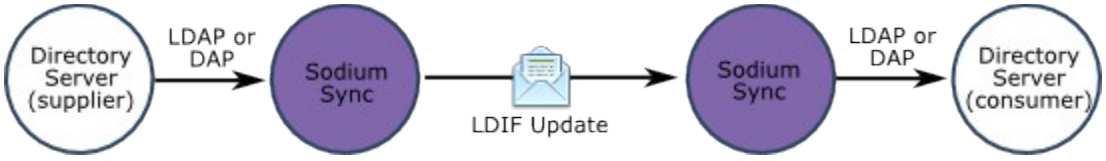
Platoon 3	Secret	None	Yes	Yes
Platoon 4	Secret	NATO	Yes	Yes

It can be seen that all of the data in the directory is at a single security classification, but that the Security Labels define different categories that control access to the data and are also used here to control where the data is replicated to. To implement the replication control, there is a Security Clearance associated with the replication channel from the UK directory to the partner directory, and only data with Security Labels that match the Security Clearance are replicated. Security Clearance for the two links from the UK directory in the example above are shown in the table below.

Replication Link (from UK directory)	**Security Clearance of Link**	
	Security Classification	**Security Categories**
To US Directory	Up to Secret	US; Two Eyes; NATO
To German Directory	Up to Secret	DE; NATO

For further information see "Using Security Labels for Directory Access Control & Replication Control" http://www.isode.com/whitepapers/security-labels-directory.html .

9 Directory Replication by Email



A final directory replication approach is to use email. Isode supports this in its Sodium Sync product, which can:

- Monitor a directory and generate LDIF deltas (or total updates) and send them off by email.
- Receive LDIF updates over email, and use these to update a directory, ensuring that updates are applied in order and that no updates are missing.

At first sight, this approach may seem extraordinarily complex, when a simple directory replication approach such as DISP could be used. It turns out that there are a number of scenarios where this approach is very useful.

1. For some environments, email gateways are the only way in/out (or at least enabling other mechanism is seen as very undesirable).

2. In secure environments, “data diode” lead to one way flow of data over boundaries. Although special directory replication protocols could be developed, email protocols to operate over data diode are in place http://www.isode.com/whitepapers/data-diode.html . These protocols give reliable message transfer over the one way data connection. Directory replication by email utilizes existing infrastructure.
3. Operation over constrained networks with low bandwidth (down to 75 bits per second for HF) high latency and low reliability. Email protocols have been developed to do this, providing reliable multicast, compression, and operation in radio silence (EMCON). Rather than develop special directory protocols, there is substantial advantage to using the protocols that are in place, and do directory replication by email.

For further information see “Directory Replication by Email and over 'Air Gap'”

http://www.isode.com/whitepapers/email-directory-replication.html

10 Conclusions

This paper has explored a number of directory replication options. It is clear that there is no “one size fits all” solution, and that different replication techniques are appropriate to different situations.