The Virtual Directory Server: A pragmatic approach to directory deployment

Overview

Directory services are now an established component of the network infrastructure, from the Internet's Domain Naming Service (DNS) to the email systems and OS domains of corporate Intranets. Applications that can leverage the strength of this infrastructure are on the rise and are placing new demands on the directory architecture. Led by the dramatic growth of e-commerce, the new directory-enabled applications are requiring access to a richer set of data than conventional directories provide, and an integrated view of the network and application infrastructure.

For corporate IT staff planning to deploy directories, the process has often proven to be slow and expensive. The Burton Group has estimated that a typical enterprise directory might take a year to deploy and cost up to $2M. There are a variety of reasons for this high cost. Directories suffer from the "yet another database" syndrome. The source of the directory information frequently exists in other parts of the infrastructure and the issues of resolving authoritative ownership of the data, reconciling the different data formats and data models, and synchronizing data from disparate sources into the directory require extensive and careful planning.

Directory deployment does not have to be this complicated or costly. This white paper details a new evolutionary step in the deployment of directory services, promising a simpler, more flexible approach. Based on the observation that the main hurdle in directory deployment stems from a mismatch between the hierarchical data structure of a directory and the more complex data models supported by the databases that house the data needed for the directory; this new approach allows the inclusion of richer, more complex data and data relationships in the directory than has been possible before.

Radiant Logic, Inc. has developed a technology that bridges the database and directory worlds by mapping relational database objects and relationships to directory entries, creating a virtual directory. While the concept is not new, having been detailed in Tim Howes classic Understanding and Deploying LDAP Directory Services, the specific implementation is truly innovative.

Background

Database, Directory which is which?

Discussing databases and directories can often lead debaters into religious territory. Yet the debate is fairly simple. Directories are a special case database and to date their usage has been restricted to a limited type of processing. To say that directories excel in areas where it is obvious that databases do a fine job is misleading. Often sited arguments for directories include their ability to out perform their relational brethren and their specific abilities when data is predominantly read oriented. Neither of these arguments holds up under careful scrutiny. In the world of relational databases performance is the highest priority. Those in any doubt of this fact need only look at the amount of time database vendors spend on TPC benchmarks attempting to woo customers by proving that they have split second differences in performance over the competition. Similarly, the argument for better treatment of read-only data does a disservice to the database vendors. Business-critical

applications deployed in companies around the world rely on sub-second response to read-only database queries and to suggest that a directory could better serve them does not make any sense. If this were the case application architects would have turned to directories many years ago in their quest to constantly provide better performing applications for their end-users.

In reality, the big difference between databases and directories, and the argument that provides a defensible stance against directory detractors, is that directories support a ubiquitous Internet access standard and have the ability to provide a self-disclosing schema. This lookup and discovery specialty may sound minor to database adherents but it provides critical features that cannot be matched by relational databases.

LDAP is King

The support for the Internet standard API, LDAP, gives directories the edge over databases when it comes to ubiquitous lookup over the network. The database access API SQL can provide rich access capabilities when the data is needed locally but it fails when secure access is required over a network. In order to provide network access to database data. application programmers must get involved and they need to use vendor-specific software drivers to give them access. The LDAP API is so ubiquitous, supported by most email clients and web browsers for example, that almost any user connected to the network can guarantee to get directory access, given the appropriate security clearance.

The hierarchical nature of the directory provides the second key differentiator between directories and databases. The directory hierarchy allows users and applications to discover the relationships between directory objects as they delve further into the structure. The architecture of the directory is in fact self-disclosing. Each object clearly shows the relationship between its parents, above in the hierarchy, and its children, below. By comparison the objects in a relational database can have a much more complex web of interactions but they are hidden from view. All relationships in a relational database are implicit and cannot be viewed by those who do not have any previous knowledge of the database schema.

These two criteria make directories an ideal source for data being shared across the Internet. When business partners share data they do not necessarily know the intricacies of each other's database environments and may not have access to the appropriate 3rd party driver software to access a database. Problems arise when the data being shared falls outside of the bounds of what is traditionally considered appropriate for storage in a directory. Up until now directories have been thought of as source for relatively static data. This comes from problems associated with synchronization and replication between the source of the data and the directory. The source data is often stored in the core operational databases used by the enterprise. This data is extracted and copied into the directory using a utility called LDIF. When directories are populated in this way on a nightly, or even weekly basis, the value of the data diminishes the older it gets.

The Virtual Directory Server

The RadiantOne Virtual Directory Server (VDS) from Radiant Logic, Inc. represents the next evolutionary step in directory software. The VDS is an innovative implementation of an LDAPv3 compliant directory. The VDS does not store any information itself, unlike other LDAP implementations. Requests received from LDAP client applications are processed by the VDS and passed on to the data source hosting the directory data. Frequently this data source will be a relational database, and more often than not it will be the authoritative source of the directory information, for example, a Peoplesoft HR application database or an Oracle financials database. RadiantOne supports the use of any structured OLEDB compliant database as a source for directory data. The VDS is unique is its ability to map relational database objects into a directory structure.

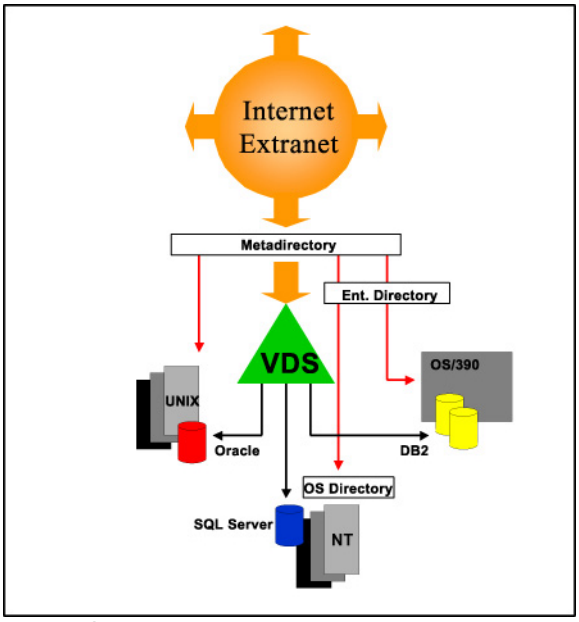

Figure 1. VDS Topology

VDS Benefits for the directory administrator

Eliminating Replication and Synchronization Issues

Traditional LDAP directories require data to be extracted from the authoritative source of the information and transformed into a format matching the LDAP schema of the directory. The data is then loaded into the directory using LDIF on a periodic basis. To maintain current information in the directory this process must be repeated on a regular basis.

The RadiantOne VDS does not hold any data within the directory itself so there is no requirement to replicate or synchronize data. Requests from LDAP clients return live data from the authoritative source. The VDS handles schema transformation automatically.

Dynamic Directory Namespace Configuration

The RadiantOne VDS separates the data structure mapping and LDAP namespace creation into two distinct processes. Relationships in back-end databases are initially mapped into the VDS server using an automated database schema discovery mechanism. LDAP namespace hierarchies are then built on top of this mapping. As new LDAP attributes and objects are required in the namespace they can be added using the RadiantOne DirectoryView application point and click interface. Changes to the directory structure take effect immediately.

Rapid LDAP Deployment

Having mapped one or more relational database structures into the RadiantOne VDS, multiple directory hierarchies can be created based on the same data mapping. This enables the instantaneous deployment of new directory namespace structures, as the needs arise. Unlike traditional LDAP implementations, where a new mapping requires either a redesign of the existing directory or a new directory structure, RadiantOne enables directory administrators to respond immediately to new application requests for directory data.

Unlimited LDAP Extensibility

The RadiantOne VDS provides extensibility to any existing LDAPv3 directory implementation using either LDAPv3 object referral mechanisms or OID (Object Identifier) join tables. Object referral allows one LDAPv3 directory to make reference to another LDAPv3 directory when clients request objects or attributes that are not stored in the primary directory. OID join tables allow one directory tree to be seamlessly connected to another directory tree.

Using object referral and OID join tables the RadiantOne VDS enables the extension of an existing LDAP structure without the pain of a directory redesign. Objects and attributes can be added to an existing directory structure quickly to accommodate the changing needs of the client applications.

VDS Benefits for application developers

Single, Standard API

By providing an LDAP proxy layer to one or more heterogeneous relational databases, the RadiantOne VDS enables application developers to use a single, open standard API to access any relational data source. The VDS provides a self-describing schema eliminating the need for application developers and users to understand the internal organization of each relational database being accessed. As users navigate through successive levels in the directory structure context is retained from one level to the next. This combination of a single API, self-describing schema, and maintaining of context can dramatically simplify database navigation for both application programmers and end users.

Heterogeneous Aggregation

The RadiantOne VDS is a powerful mechanism for navigating across diverse heterogeneous application namespaces. The ability of the VDS to enumerate the relationships implicit within each relational database structure enables the aggregation of like data across previously isolated application boundaries. The VDS, in effect provides a super-catalog of relational database meta-data enabling seamless integration of application structures.

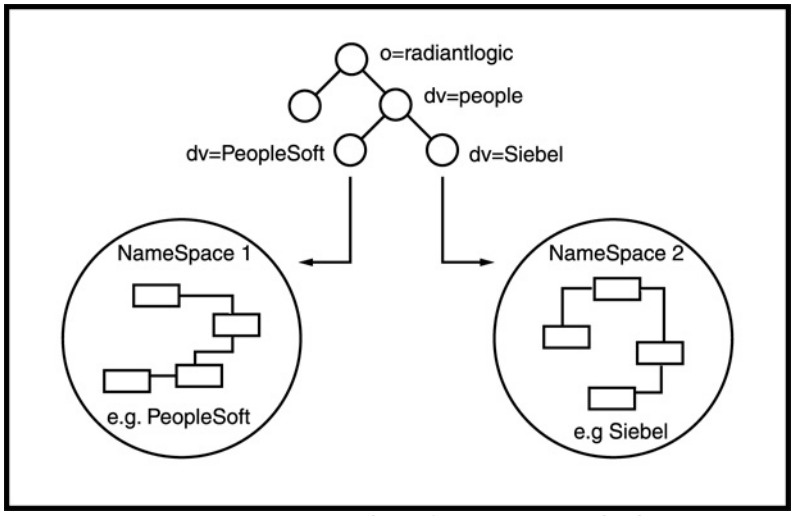

Figure 2. Heterogeneous aggregation of namespaces in the VDS

B2B Collaboration

The RadiantOne DirectoryView application is used to construct customized views of data in the corporate relational database farm. The deployment of customized views is fast and simple and does not require a great deal of technical sophistication. This means that line of business users can utilize the DirectoryView application to deploy customized views of real-time operational data as the needs of business partners arise. Roles-based security provides for granular authorization to view objects, assuring complete confidentiality to business partners accessing data over the Internet or Extranet. Business partners have the flexibility to use customized LDAP applications or the RadiantOne SmartBrowser application plug-in to Internet Explorer or Netscape Navigator.

Move Business Processes into the Network

The relationships between tables in a relational database system enumerate the business processes acting upon the corporate data and together build an inter-related sequence of hierarchical connections. These hierarchical connections represent how the work of the business is done. The RadiantOne VDS enables the enumeration of these business processes to be moved out of the proprietary bounds of each unique database management system and into the network where they can be operated upon by the individuals and applications that can make best use of them. These could be employees, customers, business partners and suppliers.

Deploying the VDS

Assessing your current environment

The RadiantOne VDS is an innovative solution to accessing directory data that resides in relational databases but it is not intended to supplant enterprise directories. Each directory solution has its unique benefits and the ideal implementation will most likely involve a combination of best-of-breed products.

When to use VDS vs. Enterprise Directory

RadiantOne does not replace the enterprise directory.

Enterprise directories are an integral part of any network infrastructure. The RadiantOne VDS inter-operates with the enterprise directory to provide even more functionality to directory-enabled applications. Enterprise directories store information from a wide array of sources including the NOS, and they are suitable for hosting this level of data. The RadiantOne VDS excels in its ability to provide data housed in relational databases to LDAP-enabled applications and users.

When to use VDS vs. Meta-Directory

RadiantOne does not replace the meta-directory.

Metadirectories that consolidate the management of multiple application and NOS directories are a valuable component of many network infrastructures. The RadiantOne VDS specializes in providing an LDAP interface to data that already exists in the relational database infrastructure of a corporation. Combining the RadiantOne VDS with a corporate metadirectory will result in a faster directory infrastructure implementation and a more flexible directory design.

Plug-ins, OID's and the join table

The RadiantOne VDS can seamlessly integrate with existing LDAP directories that have deployed the SLAPD pre/post-processing plug-in extension. Using a mechanism called a join table, the VDS is able to transparently intercept LDAP requests bound for objects in the VDS structure and pass these to the VDS for processing. Other LDAP requests will be passed to the original LDAP directory. The VDS maps OID's to database keys through a virtual directory ID. This enables additional objects and attributes to be mapped into the directory from source relational database tables.

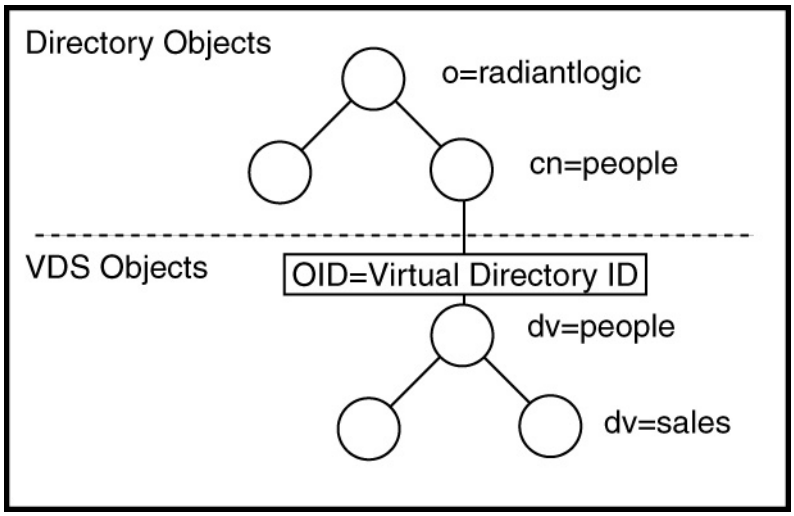

Figure 3. Join table mapping

Security

RadiantOne uses a roles-based security model with support for third party security mechanisms from Netscape, IBM, Microsoft and Oracle. Access to the directory contents is based on a combination of group membership and individual authorization. The tasks being performed by the individuals determine the level of authority that the individual has over the data. The administrator implementing the directory service will have more authority than the end user for example.

Owner of the data

The individual performing the relational database mapping task will have a high level of authority over the source databases and will probably be considered the owner of the data. Responsibility for this task often falls to someone who is very familiar with the database layout such as a senior application analyst or a database administrator. The owner of the data will map only those components of the database that are required for the directory structure. Authority over these data objects and elements will then be granted to the RadiantOne Server administrator.

RadiantOne Server Administrator

The RadiantOne Server administrator has authority over all data elements and objects mapped into the RadiantOne server. The person can create security groups based on the specific application needs of the LDAP clients and associate users with these groups. Each group can have a view administrator.

View Administrator

The view administrator has responsibility for generating customized directory views using the DirectoryView application. This person also grants user access to the views either on an individual or group basis. For many implementations the RadiantOne server administrator and view administrator will be the same person.

End User

End user access to data views is determined by the view administrator. Access can be granted based on group or individual.

Standards Compliance

LDAP

RadiantOne is an LDAPv3 compliant directory service and can be used in a stand-alone configuration or in tandem with another LDAPv3 compliant product.

ADSI

Active Directory Service Interface (ADSI) is a Microsoft standard API designed to allow applications to interface with any directory service. ADSI can insulate programmers from the need to understand directory protocols. ADSI supports Windows NT directories, Windows 2000 AD, Novell Netware directories, and LDAP directories.

RadiantOne supports ADSI directly, allowing Microsoft applications like Windows Explorer to interface with the VDS.

XML

RadiantOne provides the ability to return result-sets from a directory query in multiple formats. The application can specify whether to return the data as LDAP, XML or ADO formatted result set. In addition, the LDAP query can return the SQL syntax necessary to generate the result-set through an ODBC or OLEDB provider.

SQL

The RadiantOne VDS communicates with the back-end relational databases hosting the directory data using OLEDB. SQL commands are generated by the server to request the attributes specified for a particular directory object. RadiantOne supports the common SQL92 standard used by all major database vendors.

Summary

The growth in directory-enabled applications is prompting more directory administrators and application architects to examine what data can be stored in the enterprise directory infrastructure. Closer analysis leads to the inevitable question of where the enterprise directory ends and the enterprise relational database begins. What directory administrators and applications architects need is a solution that allows the simplicity and ubiquity of LDAP to be combined with the real-time back-end data in the enterprise relational databases. The RadiantOne Virtual Directory Server is the only product that enables RDBMS data and relationships to be seamlessly mapped to an LDAP hierarchical structure. The VDS eliminates the problems of replication and synchronization saving the directory administrator time and giving the application architect access to the rich source of corporate back-end data. The innovative mapping of RDBMS relationships to a hierarchical structure provides a simple, fast and flexible LDAP implementation.

For more information about solutions from Radiant Logic please contact us at:

Radiant Logic Inc.
1682 Novato Blvd,
Novato,
CA 94947
Phone: 415.209.6800
Fax : 415.892.7085
Emails: Sales@radiantlogic.com

http://www.radiantlogic.com