Oracle Advanced Security: Enterprise User Management

An Oracle Technical White Paper

November 1999

THE CHALLENGES OF USER MANAGEMENT

Some of the challenges faced by an enterprise today are managing information about users, keeping user information current, and securing access to all the information in an enterprise. These tasks have become complex because of the increased use of technology and high user turnover in enterprises. Each user may have multiple accounts on different databases, requiring him to remember passwords for each of these accounts. This has lead to too many passwords, and too many accounts for administrators to manage. The costs of managing all this information in an enterprise are becoming prohibitive. Furthermore, the lack of centralization is a security risk, because old or unused accounts and privileges can be misused. Ideally if a user leaves a company or changes jobs, his privileges ought to change the same day.

The enterprise user functionality introduced in Oracle Advanced Security release 8.1.6 addresses these user administration and security challenges by centralizing management of user-related information, including authorizations, in a LDAP-based directory service. Now, when an employee changes jobs, the administrator only needs to change information in one location. This lowers the cost of user management.

Figure I on the next page illustrates the challenges of user management within an enterprise. User A has multiple accounts, one on each database he connects to. As shown in the figure, user A is using the same password on each database he connects to. Many people do this because it is hard to remember multiple passwords. Similarly, user B also has multiple accounts, all with the same password. The figure on the right shows how consolidating this information into one user account which is stored in a centralized LDAP directory addresses the challenges created by multiple accounts. In the new model, all the information related to the user (name, address and roles) is stored in a centralized LDAP directory. When the user connects to the database, the database looks up the necessary information about the user in the LDAP directory.

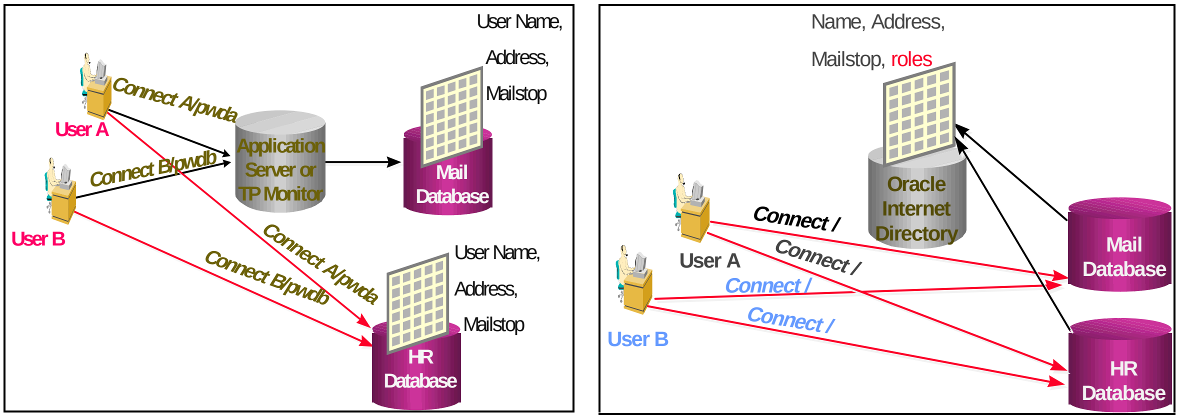


Figure I The User Management Challenge, and Solution provided by Security/Directory Integration

ENTERPRISE USERS, ROLES AND DOMAINS

An *enterprise user* is a user defined and managed in a directory. Each enterprise user has a unique identity across an enterprise. Enterprise users can be assigned enterprise roles, which determine their access privileges in databases.

Enterprise roles are also stored and managed in a directory. An enterprise role consists of one or more *global roles*. A global role is a role administered in a directory, whose privileges are contained within a single database. You can thus think of an enterprise role as a container of global roles. For example, the enterprise role CLERK could contain the global role HRCLERK on the Human Resources database, and the ANALYST role on the Payroll database.

An enterprise role can be granted or revoked to one or more enterprise users, which further leverages the benefits of centralized management. For example, an administrator could grant the enterprise role CLERK to a number of enterprise users who hold the same job. This information about users and roles is protected in the directory. That way, only a privilege administrator can manage users, and grant and revoke roles. In addition to the enterprise roles, the user also gets whatever local roles and local privileges he has been granted in a database.

An *enterprise domain* is a group of databases and enterprise roles that are managed by an enterprise domain administrator. He allocates enterprise roles to users and manages the enterprise security at the domain level. An example of a domain could be either the engineering division in an enterprise or a small enterprise itself.

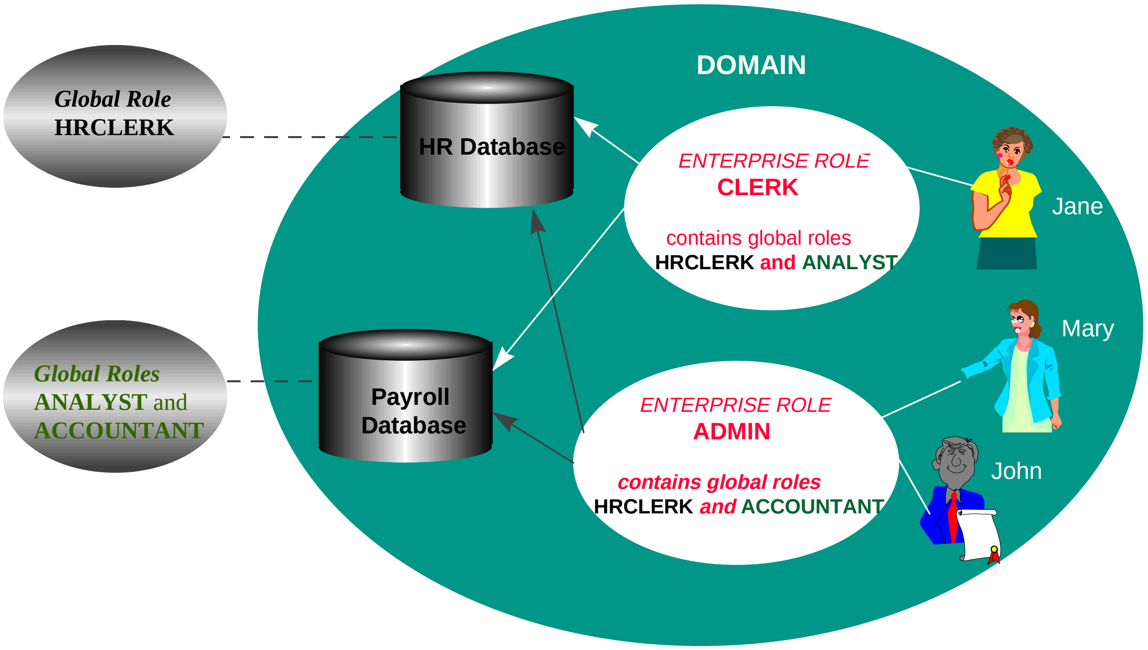


Figure II *Enterprise roles are buckets of global roles within a domain*

USER/SCHEMA SEPARATION

A benefit of managing users in a directory is that it may reduce the number of user accounts needed. In most cases, users do not need their own accounts — or their own schemas — in a database. Typically, users merely need to access an application schema For example, suppose users John, Mary and Jane are all users of the Payroll application, and they need access to the Payroll schema on the Finance database. None of them needs to create their own objects in the database; in fact, they need only access Payroll objects. For most applications, users should be able to share a schema since they don’t need to create their own objects in the database.

Release 8.1.6 supports mapping many enterprise users to the same *shared schema* on an individual database. This separation of users from schemas reduces the cost of user administration. Instead of creating a user account (that is, a user schema) in each database a user needs to access, as well as creating the user in the directory, you can create a user once, in the directory, and “point” the user at a shared schema that many other enterprise users can also access. In the previous example, if John, Mary and Jane all access the Sales database, you need only create a single schema, e.g. ‘SALES_APPLICATION’ which all three users can access, instead of creating an account for each user on the Sales database.

Example of creating a shared schema

The DBA for the HR database creates a shared schema called SALES_APPLICATION as follows:

```
CREATE USER sales_application IDENTIFIED GLOBALLY AS ‘’;
```

Mapping an enterprise user to a shared schema

The mapping between enterprise users and a schema is done in the directory by means of one or more *mapping objects*. A mapping object is used to map the Distinguished Name (DN) of a user, contained in a user's X.509 certificate, to a database schema that the user will access. This mapping can be done in one of two ways.

- A *full DN mapping* maps the DN of a single directory user to a database schema, thus associating this user with a particular schema on a database. This mapping results in one mapping entry per user. There can be many full DN mappings that map multiple users to the same schema. Thus, when using full DN mapping, each enterprise user can either be mapped to a unique schema, or to a shared schema.
- Another way of mapping enterprise users to schemas is by means of a *partial DN mapping*. A partial DN mapping is useful if multiple enterprise users that have something in common are already grouped under some common root in the directory tree. The subtree corresponding to these users can be mapped to a shared schema on a database. For example, all enterprise users in the directory subtree corresponding to the engineering division can be mapped to one shared schema on the bug database. In this way, multiple enterprise users sharing part of their DN can access the same shared schema.

When the database tries to determine the enterprise user's schema in the directory (that is, the schema to which the database will connect the user) it first looks for a full DN mapping. If it does not find a full DN mapping, then it looks for a partial one. A full DN mapping thus takes precedence over a partial one.

If there are some privileges that need to be granted to a group of users, this can be done by granting roles and privileges to a database schema. Everyone sharing such a schema inherits these local roles and local privileges in addition to their personal enterprise roles. But one needs to exercise caution when doing this, because everyone who points to this shared schema now has the privileges assigned to the schema.

To reiterate, user-schema separation eliminates the need to have a dedicated database schema on each database for every enterprise user. Each enterprise user can be mapped to a shared schema on each database he needs to access. An enterprise user is pointed to the shared schema when he connects to a database. This user-schema separation lowers the cost of managing users in an enterprise.

OVERVIEW OF THE ENTERPRISE USER MANAGEMENT TOOLS

Oracle provides several tools to manage this functionality, both for end users and for administrators. Figure II below shows the overall directory and Public Key Infrastructure (PKI) in which these tools are deployed. At the center of the infrastructure is Oracle Internet Directory (OID). *Oracle Wallet Manager* and *Oracle Enterprise Login Assistant* are client tools which the end user can deploy for managing his wallet and enabling single sign-on (auto-login). *Oracle Enterprise Security Manager* is a tool used by an administrator to manage the enterprise user functionality including enterprise users and enterprise roles. The *Certificate Authority* (CA) shown in Figure II is a third party CA which provides the digital certificates necessary for authenticating each component in the enterprise.

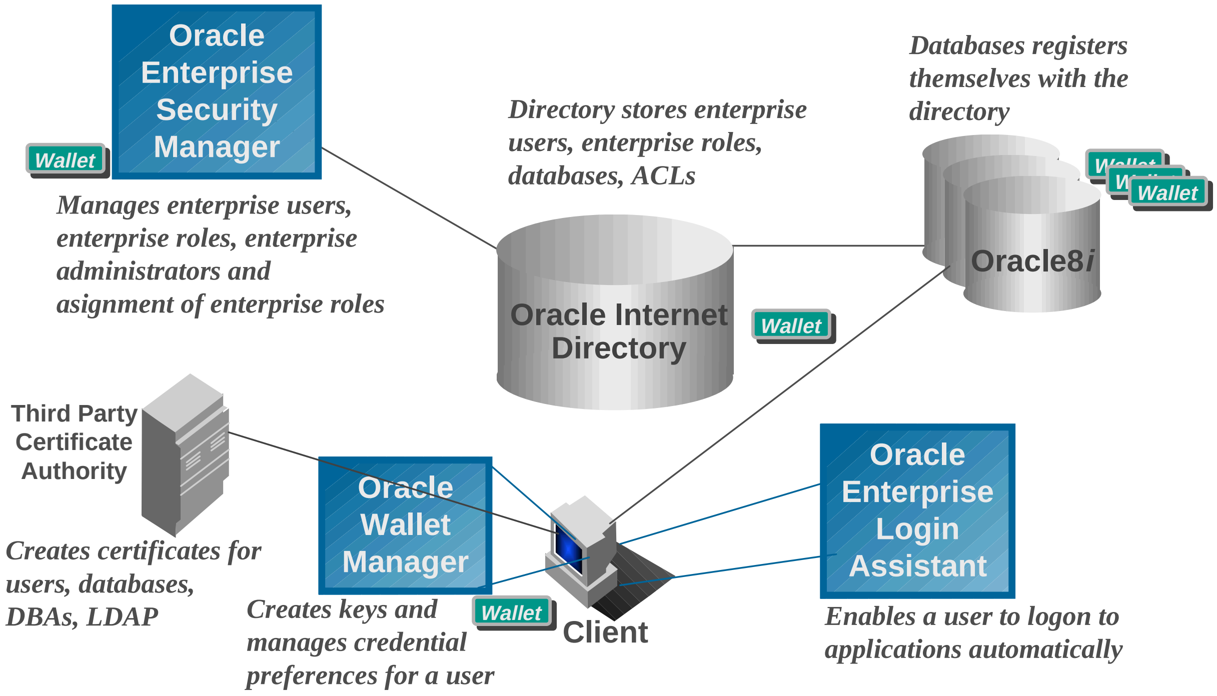

Oracle Enterprise Security Manager
Wallet
Manages enterprise users, enterprise roles, enterprise administrators and asignment of enterprise roles
Directory stores enterprise users, enterprise roles, databases, ACLs
Databases registers themselves with the directory
Wallet
Wallet
Wallet
Oracle8i
Oracle Internet Directory
Wallet
Third Party Certificate Authority
Creates certificates for users, databases, DBAs, LDAP
Oracle Wallet Manager
Creates keys and manages credential preferences for a user
Wallet
Client
Oracle Enterprise Login Assistant
Enables a user to logon to applications automatically

Figure III Enterprise User Management Infrastructure

Oracle Wallet Manager

Oracle Wallet Manager provides an easy way to manage (request, store, and view) wallets. It creates keys and manages credential preferences for a user. A wallet contains a certificate, encrypted private key, and trust points for the user. The entire wallet is also encrypted.

Oracle Enterprise Login Assistant

Oracle Enterprise Login Assistant is a tool for enabling single sign-on. It enables a user to log on to applications automatically. This tool provides the subset of the wallet manager functionality necessary for opening a user wallet and enabling applications to use the open wallet to automatically authenticate a user.

Oracle Enterprise Security Manager

Oracle Enterprise Security Manager is an administration tool that provides a Graphical User Interface (GUI) to create and manage enterprise users in the directory and create and assign enterprise roles. It also provides functionality to administer enterprise domains of databases and to administer enterprise administrators. This tool simplifies and centralizes the management of enterprise users by enabling single station administration (SSA) across an entire enterprise.

The four administrator categories supported by this tool are:

1. DB Security Administrators

These are a group of highly trusted administrators in an enterprise. They have create, read and modify access for enterprise user security. They have permissions of all domains in the enterprise and are responsible for:

- Administering the groups corresponding to the DB Security Administrators and the Database Administrators
- Creating new enterprise domains
- Moving databases from one domain to another within the enterprise

2. Domain Administrators

These administrators have complete privileges over a single domain, but have no ability to add new domains.

3. DB Installation Administrators

These administrators are in charge of creating new databases in the enterprise and registering each database in the directory by using Oracle Database Configuration Assistant. They have enough permissions in the directory to register the database. They have create and modify access to database objects and attributes. They can also modify the default domain. They can do so by adding a new database to the default domain and modifying the Access Control List (ACL) on the default domain to enable the new database to access the domain information. This is a fairly powerful set of permissions. So a security-conscious organization might want to put people in this group only temporarily when they really need to create a new database.

4. Database Administrators

Database Administrators have complete privileges over a single database (DB object) and everything underneath it (for example, Net8 objects, etc.), but have no ability to add new databases or affect the default domain.

Domain Administrators and Database Administrators are implemented in the directory using Access Control Lists (ACLs), while the others two categories are managed by using simple groups.

A USAGE SCENARIO

The local DBA or a privileged user of a database needs to create at least one database schema in the database. Enterprise users can be mapped to this schema. For example, suppose a security officer creates a shared schema called EMPLOYEE on the HR database.

The enterprise security administrator uses Oracle Enterprise Security Manager to create and manage enterprise users and enterprise roles. He assigns enterprise roles to enterprise users, that determine the level of privileges the user will have in individual databases. For example, the enterprise security administrator creates enterprise user Jane and an enterprise role named MANAGER. He now assigns Jane the enterprise role MANAGER. Then he maps Jane to the shared schema EMPLOYEE on the HR database. Now when Jane connects to the database, she is automatically connected to the EMPLOYEE schema. Multiple enterprise users can be mapped to the same shared schema. For example, the security administrator can create another enterprise user Scott and map Scott to the EMPLOYEE schema. From this point on, both Jane and Scott will automatically use the EMPLOYEE schema when connecting to the HR database.

The enterprise user and enterprise role functionality is tied to the use of SSL for authentication to the database because Oracle8*i* needs a distinguished name (DN) to look up the user. The steps below describe the sequence of actions that need to take place for a user to connect to a database.

1) The user opens her wallet

When an enterprise user wants to connect to a database, she opens her wallet by providing a password. Then Oracle Advanced Security performs an SSL handshake with the database, in which it passes the user's certificate to the server. The database authenticates her identity.

2) The database base maps the user to a schema

The database extracts the DN of the user from her certificate. Then it looks up the DN in the database. If the DN is found, then the database logs in the user-name in the associated database.

If the database does not find the DN locally, it tries to look up the appropriate DN mapping in the directory. The directory maps a user to a database schema. The database may find either a full DN mapping, a partial DN mapping to a shared user schema on the relevant database, or no mapping at all. If no mapping is found, the user cannot connect to the database.

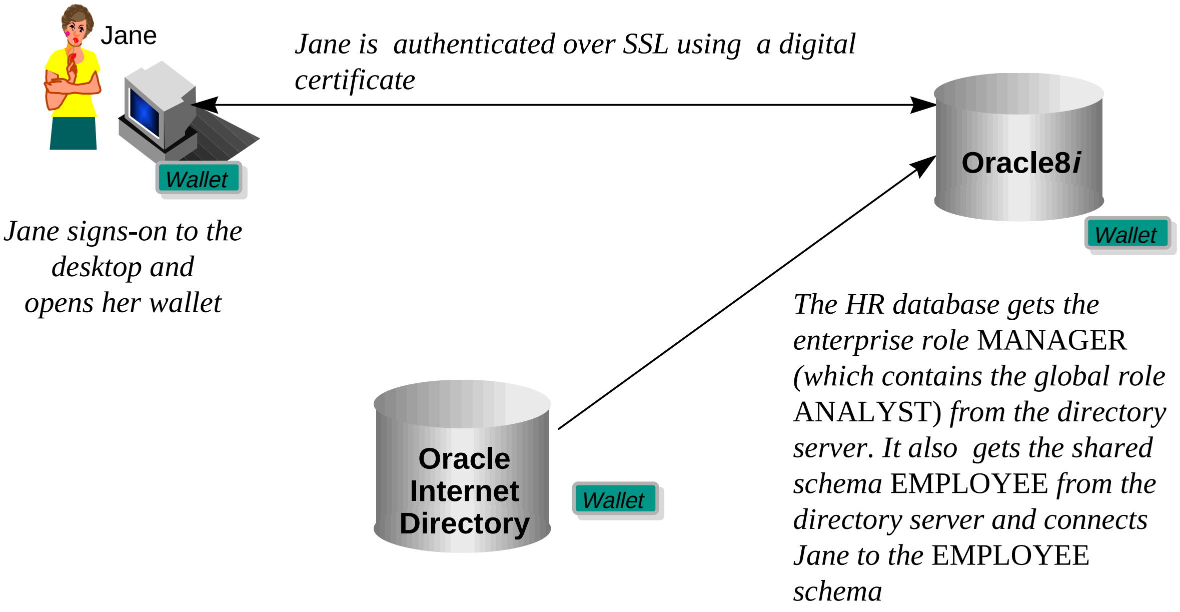
Suppose that Jane is trying to connect to the HR database. The HR database does not find Jane's DN so it looks up the DN in the directory. The directory has a mapping of Jane to the shared schema EMPLOYEE and returns this schema. The database now logs Jane in and points her to the EMPLOYEE schema.

3) The database retrieves the user's roles

When an enterprise user connects to a particular database, the enterprise roles assigned to the user determine what information the user can access. The database retrieves enterprise roles and associated global roles for that database from the directory for this user. The user also gets any local roles and privileges associated with the database schema to which he/she was mapped. The database uses both the global roles and the local roles to determine what information the user can access.

There is a way to configure the database to force it to only use local roles and not look up global roles in the directory. (See appendix for the exact configuration.) If this configuration is set, then the database does not look up the user's roles in the directory, but retrieves local roles only and uses those to determine what the user can access. Note that this configuration does not work with mapped users and shared schemas. The advantage of this is that customers who wish to use SSL for client authentication but do not have a desire to manage user privileges centrally are able to do this.

In our example, suppose the enterprise role MANAGER contains the global roles CLERK on the Payroll database and ANALYST on the HR database. When Jane (who has the enterprise role MANAGER) connects to the HR database, she is connected to the schema EMPLOYEE on that database. The privileges she gets on the HR database are determined by the global role ANALYST and any local roles and privileges associated with the EMPLOYEE schema. Figure III below, demonstrates the sequence of actions that need to happen before Jane logs in to the database.



Jane is authenticated over SSL using a digital certificate

Jane signs-on to the desktop and opens her wallet

The HR database gets the enterprise role MANAGER *(which contains the global role* ANALYST) *from the directory server. It also gets the shared schema* EMPLOYEE *from the directory server and connects Jane to the* EMPLOYEE *schema*

Figure IV Enterprise User Login

4) User is logged in to the database

Now Jane is all set. She is logged in to the HR database, pointed to the EMPLOYEE schema and has the necessary roles (ANALYST and the local roles assigned to the EMPLOYEE schema) to determine her access privileges.

CURRENT USER DATABASE LINKS

Oracle8*i* supports a particular type of database link called a *current user database link* which allows you to connect to a second database as another user, with that user’s privileges, without that user’s credentials being stored in the database link definition. For example, Jane, an Accounts Payable user, is able to access the Human Resources database by executing a procedure which connects her to HR as Scott, using Scott’s credentials.

For Jane to access a current user database link to connect as Scott, Scott must be a global user in both databases. This means that Scott was created by using the command CREATE USER Scott IDENTIFIED GLOBALLY AS ’CN=Scott,OU=ST,O=Oracle,C=US’ in both the Accounts Payable and Human Resources databases. Jane, however does not need to be a global user. Note that in order for the current user database link to work, the schema created for the global user Scott by the command above, is not shared with other users. The syntax for creating this kind of global user (in which the schema is Scott’s alone) is slightly different that the syntax for creating a shared schema.

Current user database links are also an Oracle Advanced Security feature.

Appendix A

CONFIGURATION FOR FORCING THE DATABASE TO ONLY USE LOCAL ROLES

If the parameter *rdbms_server_dn* exists in the init.ora of the database, then the global roles for the user are looked up in the directory. Any global roles that are found for that user are used in addition to the local roles when determining the access privileges of the user.

If the init.ora does not contain the above mentioned parameter, then global roles for the user are not looked up in the directory. Only local database roles are used to determine the access privileges of the user. This configuration assumes that the DBA who set up the database configuration, only wants to use local roles and privileges when authorizing users.

If the parameter is set, i.e. the DBA supposedly intended to use a directory to look up global roles, and the database cannot access a directory at run time, then the user gets a warning when he tries to connect.

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
+1.650.506.7000
Fax +1.650.506.7200
http://www.oracle.com/