An Oracle White Paper
April 2009

Developers and Identity Services

- Bridging Usability and Transparency with Role Provider Service

ORACLE

Executive Overview

Service-Oriented Security (SOS) aligns with the overall Application-Centric approach of the entire Oracle Fusion Middeware platform - with the goal of providing a comprehensive, standards-based, developer-friendly platform. By leveraging and sharing many of the common Identity Services, SOS allows developers to spend the effort on where it counts the most – the application logic itself. However, to truly move away from the traditional silo-based approach in dealing with identity management, application developers building identity-enabled applications must look to the benefits of these services and understand when and how to use these services in their application design.

This is the third in a series of "Developers and Identity Services" whitepapers, each focuses on a specific area of Identity Services from a developer standpoint. It builds on the concepts of Identity Services and Identity Externalization defined previously in the paper entitled - *Service-Oriented Security – An Application–Centric Look at Identity Management* - available at:

http://www.oracle.com/technology/products/id_mgmt/pdf/serv_oriented_sec.pdf

In this paper, we will examine the struggle between developers and business users in dealing with users and entitlements - and introduce Role Provider Service, another key component in SOS.

Introduction

In the world of identity management, the word "role" is overloaded with many different meanings. The industry has bombarded us with terms such Role-based Access Control, Role-based authentication, Role-based provisioning, etc. At the heart of it all, roles

provide a way to define an abstract bucket based on the function of a person or a group of individuals. To an application developer, a role may identify a set of application privileges associated with a functional area. An application administrator must find a way to ensure that these application-specific roles across different applications are properly managed and exposed within the enterprise. Business users must be provided with the right tools and the right level of information to manage these roles. A compliance manager or an auditor can define policies around these roles to enforce additional compliance measures to further control the underlying business processes. An efficient runtime role engine must exist for the purpose of role resolution needed by the applications and other runtime components. The solution must be flexible in its ability to implement different types of roles for different users under different contexts and provide the necessary tools for the different management needs. It must provide a high degree of transparency which ultimately defines what a user is authorized to do. Such transparency is needed to properly implement compliance measures – to protect, monitor and audit user activities. The solution - is a Role Provider Service.

"From a business user standpoint, whose tasks may include access assignments, access certification, and policy enforcements, the exposure to low-level entitlement information does not always equate to an increase in clarity."

The Challenging "Role"

From a developer standpoint, exposing finer-grained entitlements may enhance the security of an application at the price of a more complex application deployment and more advanced support in the administrative needs for the application. From a business user standpoint, whose tasks may include access assignments, access certification, and policy enforcements, the exposure to low-level entitlement information does not always equate to an increase in clarity. On the contrary, such details often do not connect with what a business user understands in the language of day-to-day business functions at a business level.

Any identity-enabled application brings along the challenge of proper authorization. A developer should possess a good understanding of the business problem in order to implement the necessary authorization requirements into the application logic. She must also consider the lifecycle of the authorization artifacts - to ensure that they are flexible enough to satisfy the application requirements, to integrate with existing customer environments, and to be manageable for enterprise users from IT administrators to business users. The application must also be able to adapt to ongoing changes and enhancements required by customers. It is a tall order to fulfill.

Granularity versus Usability

When implementing authorization requirements, developers often face the dilemma of granularity versus usability. On the one hand, fine-grained entitlements should be an objective to provide as much control as possible to secure an application. However, as a consequence, this also leads to more levels and varieties of entitlements being defined. Many commercial ERP applications contain thousands and thousands of privileges out of the box. At runtime, when an end user accesses an application, his entitlements are evaluated on the fly by the application to determine whether he is authorized to access a particular resource.

The user-to-entitlement mapping determines the proper authorization. While entitlements can be mapped to users directly, the typical volume of application entitlements means that a single user may be mapped to hundreds, maybe even thousands of fine-grained entitlements at any time. Roles have been introduced to provide a mechanism for developers to group entitlements into logical bundles. We shall refer to these as application roles – as they are tightly tied to the underlying entitlements defined within the application. Rather than granting these entitlements directly to users, application roles can be granted instead where each role contains the needed set of entitlements by users assigned to that role.

From a developer standpoint, application roles balance the struggle between granularity and usability – but only to a certain extent. If the application role grants are managed within the application, then the developer must provide support to manage these grants as part of the application administration. The advantage here is that the roles can be resolved within the application itself as role grants are part of the application data.

A different approach is to have application role grants derived through an external mechanism – for example, through LDAP groups in a corporate LDAP directory. The advantage here is that the lifecycle of a role grant falls outside of the developer's responsibility. More advanced role usages such as role provisioning – the ability to assign roles to users through a request model (manual) or based on rule (automated) – can be supported. Complex role modeling technique from other role provider can also be leveraged through integration. The efficiency and integration involved with the role provider now becomes a crucial factor.

Administration

"… if an enterprise is running multiple financial applications, administrators must understand the different models, nomenclature and idiosyncrasies of these different applications."

Application roles may close the gap between granular privilege control and usability for application development. However, the true audiences are the day-to-day users of the application – most of which are business users. A portion of these business users is tasked with properly authorizing the population by giving them the right level of access to the application and the various business operations and objects it represents. With a role-based model in place, this becomes an exercise of properly defining and assigning roles to each user.

While an application administrator or an IT personnel may be familiar with application roles and entitlements, business users may still see application roles as a low-level abstraction of the actual privilege - making it difficult to grasp their true intent. For example, an application role with the name *SALES_REPOSITORY_VIEWONLY* may be difficult to interpret. A higher-level abstraction is needed to give just the right level of information by defining roles in a business-user-friendly manner. These roles can then be linked to the underlying application roles. A *SALES_ACCOUNTANT* role is a clear definition of a business function. It will give a user the *SALES_REPOSITORY_VIEWONLY* application role. A *SALES_MANAGER* role may be given both the *SALES_REPOSITORY_VIEWONLY* and *SALES_REPOSITORY_EDIT* application roles. In this example, the business user will only be dealing with the *SALES_MANAGER* and *SALES_ACCOUNTANT* roles, which are more business-friendly than the associated application roles.

Another problem faced by many business users today is the need to deal with multiple applications and their different authorization models. For example, if an enterprise is running multiple financial applications, administrators must understand the different models,

nomenclature and idiosyncrasies of these different applications. Similar roles are likely to exist across these applications. "Sales manager" type of business functions may exist across these applications. A higher-level abstraction can also be used as an umbrella role to group and consolidate privileges across applications – and potentially extending to other non-ERP systems such as databases and operating systems. This role umbrella again abstracts out the underlying application details across multiple applications from business users who can then focus on making the right business-level decision.

In more advanced use cases, administrators may want to create their own role hierarchy to streamline their role management. They may want to extend beyond static role grants (explicit assignment of a user to a role) by implementing dynamic roles (implicit assignment with users automatically obtaining role based on some predefined criteria). For example - everyone belonging to "Sales Accountant Organization" gets the *SALES_ACCOUNTANT* role.

Compliance

"A compliance manager needs the ability to figure out the privileges associated with any users at any given time – today, yesterday, last week, at quarter-end, etc."

With compliance being at the forefront of IT focus for enterprises large and small, it is crucial that any processes related to authorization be clearly defined and captured. Tracking entitlement assignments within a single application is a high priority goal. Depending on how the application is implemented, this can be quite a daunting task.

Using roles to organize entitlements at different levels provide a tidier way to handle privileges. When it comes to compliance, transparency is the key. A compliance manager needs the ability to figure out the privileges associated with any users at any given time – today, yesterday, last week, at quarter-end, etc. In the event of any discrepancies or anomalies, she will also need to track the history of all the privilege assignments - Who performed the assignment? Was there a request & approval involved? If so, who requested it? Who approved it?

Any processes involved in user authorization must also comply with any government regulations and corporate policies in place. When implementing authorization through roles, these roles and processes surrounding them are subject to certain compliance requirements. Periodic review on user entitlements by line managers has become a mandatory requirement for many enterprises. In the case of roles, this translates to role grants or role memberships being reviewed periodically. In addition, any role-to-role or role-to-entitlement mappings and changes that could affect the entitlements obtained through the affected roles must be closely monitored and reviewed periodically. Preventive measures such as Segregation of Duty (SOD) controls can be defined around application entitlements as many solutions do today. SOD policies can also be

applied at the role level to prevent toxic combinations of privileges obtained by a user through role assignments.

As an application developer, these are difficult problems to tackle. Firstly, they diverge away from the core application logic itself into security specific subject areas that the developer may not be familiar with. Secondly, the nature of the problem is often customer or deployment specific, making it even harder to solve at the application design time. Business users are also dealing with more than one application at a time. The ideal solution is something that spans across multiple enterprise applications. An application developer is simply not coped to deal with all the requirements and deliver them within the context of the application itself.

Role Provider Service – The Bridge

It is apparent that roles can be extremely beneficial at all stages of the application lifecycle. A centralized and standardized Role Provider Service is needed to facilitate the usages of roles by providing a set of core functionalities needed by the full spectrum of users – from application developers, application & IT administrators, identity administrators – to application business users, compliance managers and auditors.

Role Types

"The Role Provider Service must support more exotic roles to meet today's demand."

As seen in the previous section, different types of roles are needed at different stages of the application lifecycle with different intents.

During the development phase, the Role Provider Service should define the methodology for developers to create application roles and to associate these application roles with application entitlements. In addition, the Role Provider Service should provide a mechanism to package these application roles and entitlements as part of the application archive, which will later on be deployed and integrated with a runtime role provider service as part of the application deployment.

Once deployed, an application must be integrated in the runtime environment. From a role perspective, a runtime Role Provider Service must be available to provide a business-level view of roles by allowing the definitions of high-level roles intended for business users. We will refer to these as enterprise roles. Enterprise roles should support static role grants where users are explicitly assigned to a role – similar to the traditional usage of LDAP groups. However, the Role Provider Service must support more exotic roles to meet today's demand. Dynamic roles have been mentioned earlier - where users obtain a role by virtue of satisfying a criteria typically

based on user attributes. Relationship-based role is defined based on relationship with a business entity. For example, a broker with the ACCOUNT_MANAGER role can see his clients' account information. However, the ACCOUNT_MANAGER role will not allow any broker to view any client information outside of the clients whose account he manages. The role here is based on a relationship between the broker and his client. Session role is another advanced role construct where role membership is based on the context of a user session, such as a user being within the firewall, or accessing the system during certain hours.

Role Administration

A centralized Role Provider Service demands proper administration support starting in the development phase. Developers must be provided with the proper tools integrated into an Integrated Development Environment (IDE) where application entitlements and roles can be defined and associated with each other. Once deployed, these application roles and entitlements, along with their mappings, must be made available to IT and business administrators. Depending on the nature of the application, new application entitlements and roles may be created during general usage or customization of the application. Role Provider Service must provide standalone and embeddable administration tools for such tasks

The resolution of any application roles ultimately lies in their mappings with enterprise roles available in the runtime environment. The Role Provider Service must provide comprehensive support for application-to-enterprise role mappings, enterprise role definitions and assignments. At this stage of the application lifecycle, the audience becomes more business-user centric. Support for request and approval workflows should be used for any sensitive operations that would alter the privileges of any users. The interface should also support the creation and management of complex role constructs such as role hierarchies, relationship-based roles, session roles, etc.

Role Engine

"The runtime role engine must ensure performance and scalability during both role resolution and role administration – since it has now become a crucial piece of the runtime logic."

In addition to administrative interfaces, the runtime Role Provider Service must a runtime role engine for the resolution of application roles through mappings with the associated enterprise roles, where role grants or role-to-user mappings occur. From the application standpoint, since it is only aware of the application roles, the role grant must be resolved at that level. The runtime role engine externalizes the role resolution logic from the application – allowing applications to benefit from advanced role features, complex role constructs and resolution logic from the Role Provider Service without worrying about the implementation as such.

In addition to correctly resolving roles, the runtime role engine must ensure performance and scalability during both role resolution and role administration – since it has now become a crucial piece of the runtime logic. It must provide the appropriate interfaces for any role engine client to pass in additional information if required. An example of such is where application context is involved in determining the appropriate role grants. In this case, the application may be required pass in session or application-specific information to the role engine for role resolution.

Role Compliance

From a compliance angle, transparency in understanding user authorization is the key requirement. Compliance managers and auditors will rely on the Role Provider Service to provide a clean trail of current and historical data including role grants, role mappings, role definitions, and any request and approval workflow information, etc. It also provides support for periodic review of role definitions, role grants and role mappings to satisfy any customer and regulatory attestation requirements. In addition, the Role Provider Service must also possess the necessary infrastructure for enforcement of compliance related policies such as Segregation of Duties policies and remediation of any violations or exceptions.

Role Provider Service forms a core component of identity administration in Service-Oriented security. By externalizing and consolidating user and entitlement mappings we now have a centralized service that not only provides a better methodology for developers and application users to deal with users and entitlements. It also provides the necessary administration and compliance support to satisfy today's requirements and adapt to future needs.

Roles & Application Lifecycle with Oracle Platform Security Service

"Oracle Platform Security Services aligns with the concept of a Role Provider Service by bridging the developers and business users with the needed role types along with the administration and runtime support to secure an application in Oracle Fusion Middleware."

Oracle Platform Security Services (OPSS) is a core component of **Oracle Fusion Middleware**. Most application developers are not experts in the area of security and identity management. And yet, they are frequently confronted with requirements in these areas. OPSS provides an abstraction layer for many security services such as authentication, authorization, audit, identity assertion, etc. Integrated into **Oracle JDeveloper**, Oracle's Integrated Development Environment, OPSS allows developer to define application permissions and organize them through application roles. It supports a Role-Based Access Control (RBAC) model through these application roles and the corresponding role hierarchy

Oracle JDeveloper packages these security artifacts as part of the application archive. When deployed, the application permissions and roles are also deployed into the runtime OPSS environment in Oracle Fusion Middleware, which provides the necessary administration tools for application role to enterprise role mappings. For example, application roles can be mapped to LDAP groups in a corporate LDAP directory representing enterprise roles. This way, role grants are driven independently through LDAP group memberships. With this as a foundation, OPSS aligns with the concept of a Role Provider Service by bridging the developers and business users with the needed role types along with the administration and runtime support to secure an application in Oracle Fusion Middleware. It will also allow OPSS to leverage more advanced role provider technology in the future to enhance security of all the applications built with Oracle Fusion Middleware.

Enterprise Role Management with Oracle Role Manager

"Oracle Role Manager (ORM) integrates with Oracle Identity Manager (OIM) to control access to ERPs such as Oracle E-Business Suite, PeopleSoft, Siebel, and to a variety of other resources such as LDAP directories, operating systems and databases."

User Entitlements extend beyond a single application. Typical enterprises deal with multiple applications and other systems such as databases, directories, operating systems, etc. – each of which deals with user entitlements differently. **Oracle Role Manager** (ORM) provides a holistic view of business users, job functions, and associated entitlements across the enterprise through its enterprise class role lifecycle management capabilities.

Application access starts with account provisioning followed by entitlement/access provisioning. ORM presents the authoritative source of roles for business users by providing the tools to manage relationships between business users and their entitlements. In turn, this information can be used to automate role-based provisioning and access control across the enterprise. Today, ORM integrates with **Oracle Identity Manager** (OIM) to control access to ERPs such as **Oracle E-Business Suite**, **PeopleSoft**, Siebel, and to a variety of other resources such as LDAP directories, operating systems and databases.

ORM also enhances application security across enterprise applications by providing a single authoritative source of users and entitlements in ORM. With self-service support, approval workflows and additional compliance controls, ORM delivers many of the much sought-after requirement that application developers can rely on. The single source of information also enhances the quality of reporting, auditing and attestation on user entitlement information by providing compliance managers and auditors the right level of details in a centralized manner.

Conclusion

Today, compliance objectives have become the No. 1 requirement for many enterprises that are implementing, deploying and managing identity-enabled applications. The age-old dilemma lies in the balance between transparency and usability of user entitlement information. From a developer standpoint, exposing finer-grained entitlements may enhance the security of an application at the price of a more complex application deployment and more advanced support in the administrative needs for the application. Similarly, from a business user standpoint, whose tasks may include access assignments, access certification, and policy enforcements, the exposure to low-level entitlement information often takes away the usability of that information. A well-defined Role Provider Service bridges these gaps by defining a methodology in dealing with users and entitlements across the entire application lifecycle – and providing all the necessary tools for administration and compliance support. Aligning with the key concept of Service-Oriented Security, the Role Provider Service externalizes the role management problem from the developers – and in doing so, providing a more comprehensive solution for application-centric role management.

Develpers and Identity Services – Bridging Usability and Transparency with Role Provider Service
April 2009
Author: Stephen Lee

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

0109