

# Service-Oriented Security

*An Application-Centric Look at Identity  
Management*

*An Oracle White Paper  
April 2008*

# Service-Oriented Security

## *An Application-Centric Look at Identity Management*

Executive Overview.....	4
Introduction.....	4
Externalizing “Identity” .....	5
Application Development Nightmare.....	5
Application Black Box .....	6
User-Centric Identity Management.....	7
Introducing Identity Services.....	7
Authentication Service.....	8
Identity Provider.....	9
Role Provider.....	9
Authorization Service.....	10
Provisioning Service.....	11
Controls & Audit Service .....	11
Service-Oriented Security – An Application-centric Look Of Identity Management.....	12
Design/Development .....	12
Packaging .....	13
Deployment.....	13
Runtime Infrastructure .....	14
Administration .....	14
Policy Management.....	14
User/Role Management.....	15
Governance, Risk and Compliance Administration.....	15
Hot Pluggable.....	16
Current Offerings.....	16
Development and Packaging .....	16
Runtime.....	17
Administration .....	17
Roadmap.....	18
Oracle Access Manager.....	18
Next Generation Single Sign-On (SSO).....	18
Fine Grained Authorization (FGA) .....	18
Oracle Identity Manager (OIM) and Oracle Role Manager (ORM) ...	18

Standards.....	19
Governance, Risk Management and Compliance (GRC).....	19
Conclusion.....	20

# Service-Oriented Security

## *An Application-Centric Look at Identity Management*

### **EXECUTIVE OVERVIEW**

Service-Oriented Architecture (SOA) has become an integral part of enterprise software by providing a framework to efficiently develop software as services that is easily sharable, reusable, and integrated. Nowhere is the need more apparent than in the Identity Management space. Welcome to the age of Service-Oriented Security (SOS).

### **INTRODUCTION**

Today's applications deals with many facets of security – from data encryption to data access, from policy management to user lifecycle, from industry standards to government regulations – and the list goes on. Application vendors and customers have solved many of these problems in their own ways. Such solutions may work perfectly when the application lives in its own silo. But once integrated into an enterprise environment, these individual silos quickly break down. Customers are faced with duplicate and scattered functionalities. Integration with existing customer infrastructure can be challenging and at times impossible.

We have made attempts to bring some of these silos together. For example, system management software exists today spanning across application vendors and their software. LDAP directories have become a common piece in presenting the corporate population in a way that can be consumed by LDAP-enabled applications. Provisioning solutions provide a way to tie in the different user lifecycles needed by individual applications through centralized user management. These bolt-on solutions work to some extent, but are susceptible to changes. With the constant emergence of new standards, new requirements and new applications, it is difficult to keep up.

In taking a holistic view, it becomes obvious that these facets of security problems are in fact common to vendors and customers alike. A set of standard-based

security services must be made available to provide standards, guidelines and the necessary infrastructure to support the entire application lifecycle – and in the process, lowering the overall cost in developing, integrating, administering, monitoring and maintaining these applications and the security infrastructure. That is the ultimate goal of Service-Oriented Security.

## EXTERNALIZING “IDENTITY”

“From an Identity Management perspective, a key step towards SOS is “Identity” Externalization – the externalization of user and security policy data from the applications themselves.”

From an Identity Management perspective, a key step towards SOS is “Identity” Externalization – the externalization of user and security policy data from the applications themselves. Service-Oriented Security enables the creation of an identity layer – providing a platform on which all identity-enabled applications are built. The externalization is needed to solve a number of the underlying problems both customers and vendors are facing with today’s approach when dealing with development, deployment and emerging trends.

### Application Development Nightmare

Consider an application developer and what she has to deal with on a day-to-day basis on identity related issues.

Gone are the days when “identity” in application development simply meant dealing with usernames and passwords, user tables and profile management screens. Today’s application developer is faced with a myriad of issues covering many different aspects of identity.

**Authentication** is no longer about simple username/password based schemes. Instead applications are faced with having to support different types of authentication mechanisms ranging from the simple to the exotic depending on deployment and security needs.

**Authorization** schemes have evolved from the simple ACL based models of yester-year into rich models that rely in complex ways on the very data that they are meant to protect.

**Roles** are now a fundamental part of application security and functionality, and have evolved from the simple group-like structures into complex business objects that rely on both context and relationships.

LDAP provided a means to externalize users and groups for developers to rely on. In pushing the limits of LDAP, many developers have become experts in LDAP, leveraging massive directories with complex user schemas to handle application requirements.

For a developer, this presents a dilemma. On the one hand, she needs to address these problems. On the other hand, she has very limited knowledge, if any, of the

“In reality, it is a difficult juggling act for the developer who risks between building too proprietary a solution versus not having all of her requirements satisfied”

customer infrastructure. The ideal solution would allow the application developer to satisfy all her requirements with the flexibility to integrate with and leverage any customers' infrastructure. In reality, it is a difficult juggling act for the developer who risks between building too proprietary a solution versus not having all of her requirements satisfied.

This shortcoming extends to application deployment. The application now lies in the hands of the application administrator. He must ensure that the application satisfies all the intended requirements through integration with his company's existing infrastructure. It can be challenging for the application administrator to fully capture all that is intended by the application vendor. Furthermore, he may have to battle with limitations and additional requirements specific to his enterprise that are not known to the application vendor. These customer issues in turn become challenges that developers and application vendors must now address.

In dealing with the evolution of identity management products, new emerging technologies, standards, corporate policies and government regulations – today's developer is overburdened with responsibilities way beyond that of fulfilling the business requirements as an application developer.

### **Application Black Box**

Many applications have been developed in a silo fashion. Once deployed in an enterprise environment, they can no longer function in a silo manner. Enterprises are increasingly looking at ways to centralize their management and administration, especially in the area of identity management – from identity lifecycle to policy management. This is often hindered by proprietary policies and framework from individual application and vendors.

Furthermore, emerging audit and compliance needs meant that applications could no longer work in a black box mode. It has become essential for auditors to understand what is going on inside an application, so they can understand the application enforcements of the controls and policies (or lack thereof).

The management and configuration of these policies have also ceased being the responsibility of IT professionals, and instead became the domain of business administrators – requiring the policies to be presented in an appropriate business-level context.

This meant that policy logic previously embedded within application code must now be taken out of that code and put into some logical container for administration and audit purpose.

These application black boxes and silos force enterprise to take on a bolt-on approach when it comes to identity management. Applications are not always readily integrated in a heterogeneous fashion with existing customer infrastructure. Similar functionalities exist in each application silo often lead to redundancies in administrative functionalities. Integration among the various applications and identity management products also becomes a nightmare – resulting in identity and

**“Lack of a centralized view for identity and policy information not only affects administrators, but presents an even greater challenge for auditors and security officers.”**

security information often being duplicated and scattered across the enterprise. Lack of a centralized view for such information not only affects administrators, but presents an even greater challenge for auditors and security officers. At the end, a customer is left with a complex and often rigid solution catered to and only to what they have today.

### **User-Centric Identity Management**

One of the newest trends in identity management is User-Centric Identity, a concept that attempts to put the user in the middle of identity related transactions, and provide greater control over the transaction and their own privacy. It relies on a combination of technology and business process to make sure that the user is involved in the exchange of identity data between interested parties.

As more applications are being built to be part of and interact with the wider internet infrastructure, the need to support user-centric identity is becoming a critical requirement for these applications. Applications would like to avoid the headache of dealing with the increasingly stringent audit and privacy requirements facing identity-based applications if they can. It also means that these applications no longer want to be restricted to the identities in their own repositories, and want to be able to work with identities coming in from external sources. All of these means that applications need to be able to adopt some new and emerging technologies into their business processes.

**“Applications would like to avoid the headache of dealing with the increasingly stringent audit and privacy requirements facing identity-based applications if they can.”**

### **INTRODUCING IDENTITY SERVICES**

The concept of Identity Services builds on the basic principles of SOA and forms the building blocks for Service-Oriented Security. It takes all the functionality of an identity management solution that would be bolted onto applications and turns the whole thing inside out, making them available as services in an SOA. Applications following SOA guidelines would be able to leverage these services without worrying about how these services are being provided. It enables enterprises to make identity a transparent, ubiquitous part of their applications, while maintaining consistency in the 4 A's of identity management - Authentication, Authorization, Administration and Auditing.

**“[Identity Services] takes all the functionality of an identity management solution that would be bolted onto applications and turns the whole thing inside out, making them available as services in an SOA.”**

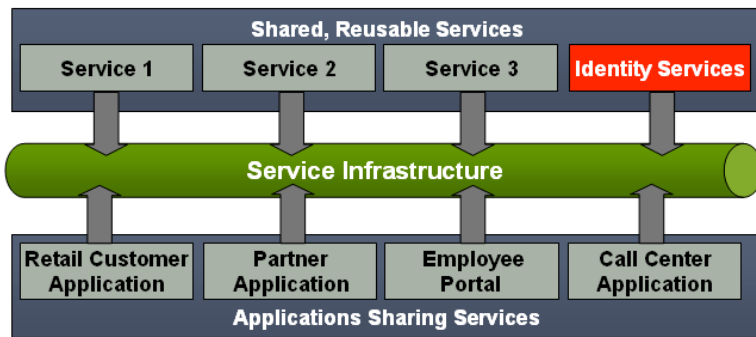


Figure1. Identity Services in SOA.

The focal point of Identity Services is a set of well-defined identity management providers providing a logical set of services across the identity infrastructure.

### Authentication Service

The goal of the Authentication Service is to provide an application the right level of assurance regarding the identity of the interacting user. This is the most commonly externalized identity service today, thanks to the ubiquity of many Single Sign-On solutions, Federation solutions, and the standardized security APIs available in development frameworks (like JAAS in J2EE). Increasingly, applications are accessed across federated domains. Standards such as Security Assertion Markup Language (SAML) and WS-Trust allow Authentication Service to protect applications deployed beyond the intranet boundaries.

But today's authentication service is still stuck thinking of authentication as a binary scheme – as far as an application is concerned, a user is either authenticated or unauthenticated. In reality, the needs of modern applications are well beyond this rudimentary capability.

For example, applications now demand the ability to perform multi-level risk-based authentication. A user may log into an application simply to view their data or manage their profile through password-based authentication. But if she tries to initiate a higher value transaction (such as approving a requisition above a certain pre-defined limit), the application may request the authentication service to further authenticate the user with a stronger token like a biometric token.

Authentication Service must also cater to emerging user-centric technologies like Microsoft CardSpace and OpenID.

Proper authentication is important, but may not be enough in a world where malicious attacks are happening at all times. Authentication Service must extend beyond authentication and provide the capability to detect potential fraud and react to such by enforcing additional-level authentication and providing proper alert if needed.

**“Today’s authentication service is still stuck thinking of authentication as a binary scheme – as far as an application is concerned, a user is either authenticated or unauthenticated.”**

## Identity Provider

The goal of the Identity Provider is to enable the externalization of identity data from the application itself. Identity data is, by its very nature, de-centralized – employee data can be in an HR system, customer data in a CRM system, and there can be an untold number of databases, LDAP systems, even spreadsheets, that hold information about specific identity populations like contractors, vendors, partners, etc. Without a single authoritative source for identity data, applications have to build and maintain user tables to hold the same identity data. In addition, many applications end up building proprietary ways to handle identity lifecycle such as user creation capabilities – causing a huge management overhead.

Centralizing identities into Meta-directory and Provisioning are two of the common approaches used currently by many enterprises. The divergence of data and the synchronization cost in a meta-directory solution is overwhelming. The cost of building and maintaining connectors to all the target systems in a provisioning solution is equally big. Neither solution provides great support to tackle the compliance and privacy issues inherent in any data replication strategy.

The Identity Provider brings order, security and compliance to the identity universe where an application can go to when it wants to retrieve identity data for any identity it cares about.

Data Virtualization presents a rationalized, unified and up-to-date profile of a user to consuming applications independent of where the data resides. This is done through virtualization of data from the underlying authoritative sources - eliminating the need for complicated synchronization. The provider will also enforce minimal disclosure of identity data through a combination of features and controls to satisfy the Principle of Least Knowledge - a key characteristic that enables compliance with security and privacy needs by making identity data available to consumers on a need-to-know basis. The Identity Provider should have the ability to enforce policy-based controls over the identity data.

## Role Provider

The goal of the Role Provider is to enable the externalization of roles from the application itself. Roles have become an integral part of every application's architecture, from being part of their authorization model to being a business construct used in various workflows and task flows. Roles are often used as an abstract container for users to which business objects can be attached, making those users part of some particular application logic or decision flow (like connecting them to an approval workflow, or assigning them certain privileges).

In today's world, LDAP groups are often used as an enterprise role system. While still valuable, over time this model has proven to be too simplistic to support the more advanced role requirements.

**“Without a single authoritative source for identity data, applications have to build and maintain user tables to hold the same identity data. In addition, many applications end up building proprietary ways to handle identity lifecycle such as user creation capabilities – causing a huge management overhead.”**

**“The Role Provider can also support more exotic concepts that LDAP groups are simply not capable of handling...”**

The Role Provider is a centralized system that supports both enterprise and application roles. Role Management happens at the enterprise role level, which is a simpler, more understandable role structure. In turn, these enterprise roles impact applications in very granular, very specific ways as intended in the application design through the relationship between the enterprise and application roles. Application roles can also be shared with other applications, allowing for easy integration and functional continuity across applications in the enterprise.

The Role Provider can also support more exotic concepts that LDAP groups are simply not capable of handling – such as relationship-based role where role membership is based on relationship with a business entity; or a session role 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.

A major goal for a centralized Role Provider is to allow an enterprise to put in place the right controls to ensure integrity of the system by enabling the enforcement of Segregation of Duty (SoD) rules not just within an application, but across related applications as well. It also allows for approval controls over role assignments related to sensitive privileges. As a result, the overall compliance is improved.

### **Authorization Service**

**“The system is no longer constrained by what the application developer was able to support in terms of policy capabilities. The authorization policies can now be as complex as needed, since the authorization service has far greater capabilities than any individual application.”**

The goal of the Authorization Service is to enable the externalization of authorization checks and decisions from the application itself into a centralized framework. Being able to decouple authorization from the core application logic allows the application developer to concentrate on their application logic, and frees them from having to rewrite their application every time the authorization needs change. The application developer simply defines the permission checks or entitlements that they care about, and publishes these entitlements to the external service. Authorization policies can now be deployed in the externalized service, with the application simply asking for the appropriate permission check. The system is no longer constrained by what the application developer was able to support in terms of policy capabilities. The authorization policies can now be as complex as needed, since the authorization service has far greater capabilities than any individual application.

This externalized Authorization Service supports both entitlement modeling and fine-grained authorization. The emergence of the eXtensible Access Control Markup Language (XACML) standards allows entitlements to be easily defined in application terms, and complex policy criteria to be defined that rely on both identity and application data. These policy definitions can further rely on other components of the Identity Services, especially the Role Provider. Roles are a key part of authorization policy definitions, and having access to the powerful role concepts in the Role Provider service means that authorization decisions can also be more granular and meet the increasingly complex business needs.

The centralization of authorization decisions allows for better compliance by providing control points in the application environment where SoD checks, auditing and enforcement of policies can be handled in a uniform manner.

### **Provisioning Service**

The goal of the Provisioning Service is to enable applications to become collaborators in the identity process instead of simply being consumers of identity data. It exposes various services that allow applications to also be involved in the administration of the IAM context.

The Provisioning Service in the Identity Services layer keeps in place those same business controls, even as the other services eliminate the need for data flow. This service provides a centralized administration framework by enabling delegated administration and approval-based administration. It combines the administrative needs of the other Identity services (like identity and role creation, role membership requests, etc) and pushes them through a centralized approval, audit and attestation environment. It ensures that those other services have all the data they need, while enabling full enforcement of SoD checks, audit and regulatory policies. Thus,

- Users of an application can request a particular role or entitlement.
- An application can expose self-registration features that add identities to the enterprise environment.
- An auditor can view and attest to a person's access across the entire enterprise.

In some sense, the Provisioning Service acts as the *bus* for the connected Identity Lifecycle process.

### **Controls & Audit Service**

The goal of the Controls & Audit Service, as the name implies, is twofold. Segregation of Duty has shown up numerous times in the services mentioned before. Due to the increasing focus on regulatory requirements and corporate security, internal controls have become a key concern in the enterprise. The Controls Service provides the ability to centrally manage and enforce internal controls and other compliance related activities. When integrated with other Identity Services, Controls Service provides the support to enforce, for example, SoD rules on security policies, to protect access to the application and other crucial data.

The criticality of internal controls means that the management of such controls must be carefully monitored. A comprehensive Audit Service must be available to provide a common service through which to audit the events that are happening within the application. The service can then (at deployment time) be hooked up to a centralized or distributed audit repository as necessary. The service can de-normalize and correlate audit data, providing the enterprise such key features as *Event Correlation*, *Tamper-Proof Audit Trails*, *Activity Monitoring* and *Fraud Detection*.

**“[Provisioning Service] combines the administrative needs of the other Identity services (like identity and role creation, role membership requests, etc) and pushes them through a centralized approval, audit and attestation environment.”**

**“When integrated with other Identity Services, Controls Service provides the support to enforce, for example, SoD rules on security policies, to protect access to the application and other crucial data..”**

## SERVICE-ORIENTED SECURITY – AN APPLICATION-CENTRIC LOOK OF IDENTITY MANAGEMENT

As a key concept to SOS, these Identity Services take us towards our vision of Application-Centric Identity Management. Starting from development, through deployment, to end user access, administration and maintenance, the application lifecycle exercises many different aspects of these services. From an application-centric view, the success of Identity Services lies upon its ability to tackle the requirements at each stage of the application lifecycle.

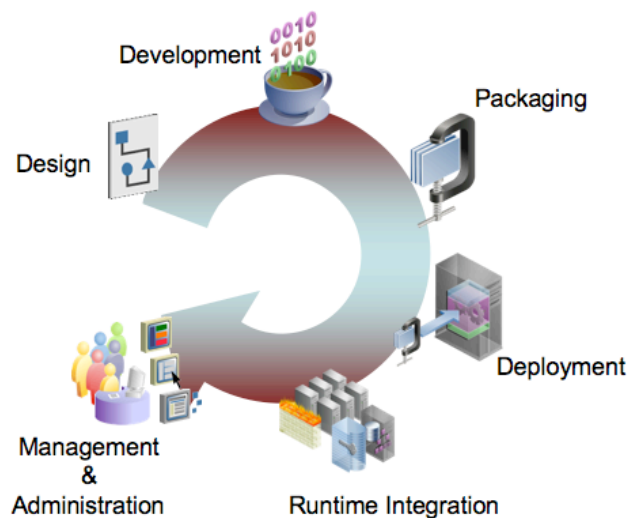


Figure 2. Application Lifecycle

### Design/Development

The role of an application designer is to outline the business functionality that the application intends to provide and conceptualize the interfaces through which these features are implemented and exposed to the various types of users of the application. These interfaces include UIs, page-flows, tabs, buttons, APIs, web services, etc. From a security standpoint, the designer should carve out a security model to protect these interfaces through a common authorization framework.

The developer is expected to function in an Integrated Development Environment (IDE). The developer's role is to incorporate the authorization policies defined in the design phase to the actual code. She should also be able to further refine the existing policies. Lastly, the application must be tested in the IDE environment by simulating real world usages.

**“Identity Services must define a standard format to capture the authorization model, a model that is used, not only at the design and development phase, but through the remainder of the application lifecycle.”**

Identity Services must define a standard format to capture the authorization model - a model that is used not only at the design and development phase, but through the remainder of the application lifecycle. Appropriate tools and integration with IDEs must provide tools for designers and developers to articulate the security policies and their association with the actual code and to allow sharing of the authorization policies through appropriate import/export facilities. And finally, the IDE must provide, in conjunction with Identity Services, the ability to test these security policies in the application before it is packaged.

## Packaging

A typical IDE provides the ability to package an application for deployment purpose. In the J2EE world, applications can be packaged as different types of archives (JAR, WAR, EAR, etc). At this point, the application contains not only the code artifacts but also the entire footprint of the security artifacts and its association with the code artifacts as implemented by the developers.

Identity Services must define a standard format to package the security artifacts into an application archive. The standard must capture all the security requirements during the design and development stage. Furthermore, such a standard must be understood by the deployment mechanism to carry these security artifacts into the runtime environment when the application is finally deployed. Using this standard, IDE can package these security artifacts into the application archive in a format that can be consumed at deployment time. The set of security artifacts will essentially be the out-of-the-box security seed data when the application is first deployed.

## Deployment

Once packaged, an application may be deployed in multiple ways. A release product may be deployed through an installer. A customer application may be deployed through the middleware framework. For any security-aware applications, deployment means more than having the bits up and running. Identity Services must ensure that the security artifacts are correctly deployed and wired in the runtime environment.

Identity Services must define a standard format that is understood by the deployment mechanism to deploy the security artifacts during application deployment. Such format provides the bridge carrying the security artifacts from the design and development phase to the runtime environment. The standard provides the handshake between the IDEs and the different runtime container vendors in a well-known format that adequately captures the security artifacts on both sides.

A runtime environment can be a customer development environment, a staging environment, a QA environment or an actual production environment. A deployed application must be configured to leverage the available identity services before it can be fully functional. Identity Services must provide tools to configure the

**“In addition, Identity Services should provide mechanisms to integrate with external components such as a 3<sup>rd</sup> party Single Sign-On products, an external policy engine such as an XACML engine or an external SoD engine..”**

deployment to complete this runtime identity infrastructure wiring. In addition, Identity Services should provide mechanisms to integrate with external components such as a 3<sup>rd</sup> party Single Sign-On products, an external policy engine such as an XACML engine or an external SoD engine.

Identity Services must also handle other application lifecycle activities such as patching, upgrade and migration by ensuring that security artifacts are correctly patched and that any extensions or customizations post-deployment are preserved.

### **Runtime Infrastructure**

The runtime infrastructure for an application stretches from interfaces exposed to end users down to the database tier. Identity Services provides, not only the services themselves, but a framework to integrate an application with these services in a provider-agnostic manner.

Wherever a service provider acts as the authoritative source (eg. authoritative identity repository or role repository), Identity Services must ensure that the service provides a standard mechanism to expose its data for runtime consumption by other applications in a secured manner through APIs, Web Services, etc.

These service providers must also focus on scalability, performance, high availability, back-up and recovery mechanisms – both for the backend storage and the corresponding services provided for accessing this data.

### **Administration**

The administration of an enterprise application is handled by many different types of users. A security administrator may change the underlying security artifacts to alter the behaviour of an application. A delegated administrator may revoke somebody access to certain parts of the application by changing his roles. An end user may reset her password through self-service. An auditor may want to review evidence of certain policy enforcements protecting sensitive data in the enterprise. Providing such a platform for administration is a key goal in Identity Services.

### **Policy Management**

Policy Management deals with the authorization policies themselves. Immediately after deployment of the application, the seeded security artifacts packaged with the applications becomes available in the policy repository. Before these policies are associated the actual end users, the applications driven by these authorization policies will not function properly.

Identity Services must define a standard format to represent security policies in a runtime environment. The standard captures the policies and how these policies are associated with end users during runtime. For example, a role in the application might be mapped to an LDAP group or to a business role in the enterprise role management system.

**“The administration of an enterprise application is handled by many different types of users ... Providing such a platform for administration is a key goal in Identity Services.”**

The Authorization Service provides a central policy repository along with a centralized framework for policy management to allow administrators the ability to view, modify and create new security policies. It also provides support for audit and compliance such as SoD support where appropriate. The definition of roles and role hierarchy will require this support as it alters the relationship between users and permissions.

### **User/Role Management**

In an application-centric view, user management primarily deals with lifecycle of an identity. The Identity Provider provides a central authority on identity data. If multiple user repositories are required, the Identity Provider provides the necessary virtualization to abstract the details from the consuming applications.

Identity Services must define a standard format to represent and to access identity information. The standard should allow an application to specify its own identity requirements and be understood by the Identity Provider to expose the appropriate identity data.

**“Identity Services must define a standard format to represent and to access identity information. The standard should allow an application to specify its own identity requirements and be understood by the Identity Provider to expose the appropriate identity data.”**

Identity Services also provide the underlying support for user creation, self-registration, self-service, delegated administration, proxy user support, etc. along with the necessary support for compliance such as SoD support and Restricted Party Screening, etc. along with thorough auditing and reporting capabilities in this area. Where provisioning is required, the Provisioning Provider provides support to and from various targets.

Similar to the Identity Provider, the Role Provider provides a central role repository and a framework for role management, role request, role assignments, role catalogues, etc. As roles become an integral part of the authorization framework, advanced features in the Role Provider provides administrators with greater flexibility and control. As important as it is with user management, the Role Provider must provide support for compliance such as SoD support during any request, approval and auto-provisioning of role assignments along with auditing and reporting capabilities on role assignments or role changes.

### **Governance, Risk and Compliance Administration**

The need for SoD is just a steppingstone into the broad arena of IT Governance, Risk and Compliance. The ability to enforce security policies within an application is obviously crucial. But more importantly, Identity Services must provide the ability to properly manage and monitor the lifecycle of these security policies and other activities that may alter an individual's access. Controls and Audit Service provides not only the underlying infrastructure for applications and identity services to plug into, but also the administrative framework to manage policies and controls. This allows the administrator to control and track critical changes to policies and controls; to administer and monitor enforcement of crucial policies such as SoD; to detect and be alerted of any suspicious activities or policy violation. Such reporting

**“But more importantly, Identity Services must provide the ability to properly manage and monitor the lifecycle of these security policies and other activities that may alter an individual's access.”**

and alerting capability provides the necessary input for auditors and security officers alike.

### Hot Pluggable

One of the goals for Identity Services is to simplify and reduce the number of moving parts in the identity infrastructure. That said, the complex nature of the identity infrastructure makes this a challenging problem. With the heterogeneous nature of today's enterprise deployment, a Hot-Pluggable framework at every stage of the application lifecycle is essential for Identity Services to succeed.

**“With the heterogeneous nature of today's enterprise deployment, a Hot-Pluggable framework at every stage of the application lifecycle is essential for Identity Services to succeed.**

A developer must be able to use the IDE of her choice. The development framework defined in Identity Services must be portable to all the common IDEs.

Identity Services will also not assume a particular runtime environment for the application. In the J2EE world, this implies the ability to run in any vendors' containers.

In the heterogeneous world, Identity Services must allow customers the flexibility to use any identity management components of their choice where appropriate, provided that they satisfy the provider functionalities as required by the services themselves.

## CURRENT OFFERINGS

Oracle's current offerings are aligned with our direction towards our vision of Service-Oriented Security. **Oracle Fusion Middleware** offers a comprehensive stack of products through **Oracle Application Server**, **Oracle SOA suite** and **Oracle Identity Management Suite** - addressing many of the key areas touched upon in this document.

### Development and Packaging

**Oracle JDeveloper**, part of the Oracle SOA Suite, provides an IDE of choice for developing J2EE applications. From the identity management perspective, it allows designers and developers to define the various aspects of the authorization policies through Oracle's implementation of Java Authentication and Authorization Service (JAAS). Oracle JDeveloper also contains an embedded application container allowing a developer to test her application and the authorization policies within the IDE.

10gR3 also introduces **Oracle Application Development Framework (ADF)** with rich security support allowing developers to define fine-grained access control to iterators, attributes, and methods exposed by a business services. JDeveloper has implemented many ADF security features such as ADF Security Wizards to help developers in authoring security policies during the development phase.

From a packaging standpoint, Oracle JDeveloper packages all the security artifacts as part of the application archive, carrying them to the runtime deployment environment.

## Runtime

**Oracle Application Server** provides the J2EE runtime environment. Authorization policies defined during the development phase are made available to the runtime container, **Oracle Container 4 Java (OC4J)** during deployment. The policy data can be directly deployed or migrated into LDAP to benefit from the performance, scalability and high-availability benefits of an enterprise directory.

OC4J is also able to leverage many other Identity Management infrastructure components. For single sign-on, OC4J can be configured to use **Oracle Access Manager** as well as other 3<sup>rd</sup> party single sign-on solutions. Enterprise Identity Stores in the form of enterprise LDAP, including **Oracle Internet Directory** and **Oracle Virtual Directory** and other 3<sup>rd</sup> party LDAP such as Microsoft Active Directory, can be used as the identity store for authentication and authorization purpose. They can also be used as an LDAP-based policy store by OC4J.

On the authentication front, **Oracle Enterprise Single Sign-On** provides support for desktop single sign-on. **Oracle Identity Federation** delivers a comprehensive multi-protocol federation solution for cross-domain single sign-on. **Oracle Adaptive Access Manager** brings in strong multi-factor and mutual authentication capability. In addition, it provides a risk management capability to analyze real-time data and detect potential fraud.

## Administration

On the user management front, **Oracle Identity Manager (OIM)** brings a well-defined set of identity management and enterprise provisioning functionalities such as self-registration, self-service, delegated administration. OIM provides ability to provision to various backend targets such as RDBMS, LDAP servers, Operating Systems, and applications such as SAP, PeopleSoft and our own EBusiness Suite R11/R12. Custom connectors can also be built to cater other provisioning needs for any custom applications.

**Oracle Role Manager** provides an enterprise role lifecycle management solution that can act as the authoritative source for the relationships between business users, organizations, and entitlements, thus enabling automation of role based provisioning and access control across the IT infrastructure. The integration between Oracle Role Manager and Oracle Identity Manager marries the role management and provisioning solutions, creating a powerful combination in driving automation and compliance need of any applications in an enterprise environment.

In 10gR3, the JAAS security policies can be managed through **Oracle Enterprise Manager** providing the necessary means to manage JAAS policies without directly

modifying the policy store, which would be LDAP in many production environments.

In addition, Oracle Enterprise Manager provides system management and performance monitoring capability across the board for system and application administrators. **Oracle Management Pack for Identity Management** streamlines the management and monitoring for Oracle Identity Management to improve service levels and ensure high availability. It provides a single console to manage systems spanning directories, firewalls, applications servers, business applications for Oracle and many non-Oracle systems – providing automated configuration management, fault isolation and diagnosis, and one-step discovery of systems.

## **ROADMAP**

As our components continue to evolve, there are opportunities for improvements – to move us closer to our vision. However, the message has to be well understood and carried across to all the components supporting Service-Oriented Security. Many of our components are already taking their next steps to move towards the standards-based, well-integrated and packaged application-centric identity management. Here are a few highlights:

### **Oracle Access Manager**

#### **Next Generation Single Sign-On (SSO)**

The next generation SSO product will consolidate and enhance the existing authentication mechanisms provided by our current single sign-on products. From the SOS perspective, the next generation plans to provide more standard-based solutions and a pluggable architecture to integrate with other authentication mechanisms from Oracle and other 3<sup>rd</sup> party or customized solutions.

#### **Fine Grained Authorization (FGA)**

FGA is targeting to extend the existing authorization engine within Fusion Middleware. This allows the security model to extend from RBAC model to other security principles such as Attribute Base Access Control (ABAC), providing additional functionalities in defining finer grained policies. Access control based on temporality (such as start date/end date) or IP-based can now be centralized into a single authorization model with centralized storage and administration of the policies.

### **Oracle Identity Manager (OIM) and Oracle Role Manager (ORM)**

As seen from this document, the user and role management aspect is crucial in application-centric identity management. The requirements from Identity Services

for OIM and ORM are similar in nature to those for an enterprise provisioning and role management system. The SOS vision requires OIM and ORM to be more application-centric in order to satisfy the requirements of being an Identity, Provisioning, and Role provider. The two products also share similar functionalities in areas such as user and role managements, approval workflows, notification, etc. One of the goals is to centralize such shared services across, not only OIM and ORM, but across the entire Oracle Identity Management to better the integration and administrative experience.

## **Standards**

One key aspect in the application lifecycle is the ability to smoothly carry security information from one stage to the other. The problem is neither component specific nor vendor specific. Yet standards-based formats are lacking in many of these areas where standards should be defined.

On the authorization front, eXtensible Access Control Markup Language (XACML) is a starting point in capturing the authorization model. It provides a policy language to define access control. Much work is being done to figure out how XACML can be extended to other areas of the application lifecycle – for example, how to represent XACML policies during development and deployment; how to efficiently provide access to XACML policies during runtime.

On the identity front, Oracle's **Identity Governance Framework** is leading the way in defining the standards.

From the application end, **Client Attribute Requirements Markup Language (CARML)** presents an interesting option in dealing with application user attribute mapping. It provides a declarative way for designers and developers to communicate their "identity requirements" to deployment administrators – paving the way for identity virtualization. In addition, a secondary goal with CARML is to support expression of privacy constraints for "identity data".

On the identity provider end, **Attribute Authority Policy Markup Language (AAPML)** allows identity sources to specify constraints on how information can be used by applications.

Together, the two standards define the handshake between applications and the identity providers, providing a way to govern and protect the flow of user data and identity information.

## **Governance, Risk Management and Compliance (GRC)**

GRC is an important area in user/role provisioning as well as in the overall authorization policy management where permissions/privileges and role hierarchy/inheritance can be modified. The short-term roadmap involves integration with **Oracle Application Access Control Governor**, part of the **Oracle Governance, Risk, and Compliance Suite**, to provide Segregation of Duty support in products such as OIM and ORM for enforcement and audit.

## CONCLUSION

Service-Oriented Security presents a unique set of challenges for many aspects of the identity management space. The introduction of Identity Services brings real value in addressing many of the shortcomings in today's solution. From an application-centric perspective, the ultimate goal is to arrive at a standards-based application lifecycle that addresses all the identity management needs by providing a framework for development, deployment and runtime support – a model that is flexible to support heterogeneity, to support other existing or emerging industry standards – a model that is embraced, not only by Oracle, but by other software vendors – and more importantly, by any customers wishing to develop applications in an application-centric identity management environment.

*For more information on Oracle's security technology,*

*Go to <http://www.oracle.com/security>*



Service-Oriented Security – An Application-Centric Look at Identity Management

April 2008

Author: Stephen Lee

Contributing Author: Nishant Kaushik

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](http://oracle.com)

Copyright © 2008, Oracle Corporation and/or its affiliates. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.