Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Introduction

Oracle Fusion Applications are Oracle’s next-generation enterprise resource planning applications including Financial Management; Human Capital Management; Customer Relationship Management; Supply Chain Management; Project Portfolio Management; Procurement; Governance, Risk, and Compliance.

Oracle Fusion Applications are designed independently from current Oracle Applications such as Oracle E-Business Suite, Oracle PeopleSoft, Oracle Siebel, or Oracle JD Edwards; Oracle Fusion Applications combine the best of the Oracle business applications Oracle currently provides (and will continue to ship and enhance under the Applications Unlimited program).

Oracle Fusion Applications deliver unsurpassed business performance:

- Productivity - Work smarter with intuitive, intelligent, and collaborative applications: what you need to know; what you need to do; whom you need to know; how to get it done.
• Adaptability - Respond effectively to change with flexible, modular, user-driven business solutions powered by best-in-class business processes built on industry standards.

• Manageability - Deliver faster return on investment with tools for rapid setup and flexible deployment models (in-house or hosted software-as-a-service), and provide a rich environment that empowers end-users to effectively search, analyze, compare, and process enterprise information.

• Security - Provide standards-based, declarative, transparent, portable function and data security policies across all Oracle Fusion applications, defined independently from application code at design time.

Oracle Fusion Applications are built from the ground up on Oracle Fusion Middleware thus creating a unified suite of components based on a Service-Oriented Architecture (SOA).

Oracle Fusion Applications leverage the various foundation capabilities provided by Oracle Fusion Middleware, such as a standards-based application development framework (Oracle ADF), business intelligence, content management, enterprise performance management, SOA and process management, and security and identity management. As a result, the designers of Oracle Fusion Applications focused all their effort on the business value of applications, relying on Oracle Fusion Middleware services for everything else.

This document describes how Oracle Fusion Applications leverage Oracle Identity Management for foundation security services; identity administration (identity life cycle management, self-service account request and password management, enterprise role management); authentication and trust management (single sign-on, identity federation, privacy); access control (risk-based authorization, fine-grained entitlements, web services security); identity and access governance (audit and compliance reporting, segregation of duties, conflict-resolution management, attestation, role mining and engineering, identity and
fraud-prevention analytics); and directory services (persistent storage, identity virtualization, synchronization, and database-user security).

In the first part of this document, we show how the Oracle Fusion Applications designers use Service-Oriented Security (SOS), relying on the Oracle Platform Security Services (OPSS) environment to declaratively apply security to Oracle ADF projects (the core of Fusion Applications).

In the second part of this document, we show how the identity services delivered by the Oracle Identity Management product stack combine with SOS to provide consistent security across Oracle Fusion Applications, seamless integration, high scalability, and centralized administration to multiple instances of Oracle Fusion Applications deployed across the enterprise.

This document is primarily intended as general information for line-of-business managers, development managers, security architects, and identity and access management administrators.
Oracle Fusion Applications' Structure

Oracle Fusion Applications are Java Platform, Enterprise Edition (Java EE) applications. Oracle Fusion Applications are organized in “pillars.” A pillar is a standalone subset of Oracle Fusion Applications, for example Customer Relationship Management (CRM). A pillar can include multiple “families” or sub-groups. The pillar structure allows patching and upgrades to be performed at a granular level without impacting other pillars. For example, customers that want to upgrade their financials and supply chain management applications can do so without impacting the CRM applications.

Technically, a pillar consists of a database, one or more Java EE applications, Oracle ADF Business Services (described in the next section), and SOA composites that connect to the database at runtime. Provision is made for interaction between pillars, such as table replication (Oracle Fusion Applications leverage Oracle Data Integrator to handle transformations if two pillars have different versions of the same table).

Pillars can be co-located, i.e., a customer may choose to install more than one pillar in a single database instance. Oracle Fusion Applications provide essential functionality when a service provided by another pillar becomes unavailable.

Oracle Fusion Applications Development Environment

Oracle Fusion Applications are a particular instance of Fusion applications. Indeed, Oracle Fusion Middleware 11g components such as Oracle WebCenter, Oracle Identity Manager, etc., are also bona fide Fusion applications. Likewise, customers, integrators, and independent software vendors (ISVs) relying on Oracle Fusion Middleware at design time also develop Fusion applications.

Oracle Application Development Framework (ADF) is the core of Oracle Fusion Applications. All Oracle Fusion Applications (and, for that matter, all of Oracle Fusion Middleware 11g components) follow Oracle ADF’s development and deployment patterns. Oracle ADF is directly supported by the Oracle JDeveloper integrated development environment.

Oracle ADF makes it easy to develop agile applications that expose data as services by coupling a service interface to built-in business services. This separation of business service implementation details is performed in Oracle ADF via metadata (XML files). Use of this metadata-driven architecture enables application developers to focus on the business logic and user experience, rather than the details of how services are accessed and secured.

Oracle ADF implements the Model-View-Controller (MVC) design pattern. The Oracle ADF architecture is based on the following four layers.

- **Business Services**: The Business Services layer provides access to data from various sources and handles business logic. It also manages interaction with a data persistence layer providing such services as object-relational mapping, transaction management, and business logic execution. Oracle ADF’s Business Services can be implemented in different ways: simple Java classes, Enterprise JavaBeans
(EJB), web services, Java Persistence Application Programming Interface (JPA) objects, and Oracle ADF Business Components (ADF BC). Oracle ADF BC is a key element of Oracle Fusion Applications. Oracle ADF BC is based on three main building blocks: Entity Objects (EO) representing rows in a database and acting as an application cache for table rows; View Objects (VO) representing SQL queries; and the Application Module, a container for VO instances that define the data model and transaction for a particular business task.

- **Model**: The Model layer provides an abstraction layer on top of the Business Services layer, enabling the View and Controller layers (described below) to work with different implementations of Business Services in a consistent way. The Model layer connects the business services to the objects that use them in the other layers.

- **View**: The View layer provides the user interface of the application. The View layer can be based on HTML, JavaServer Pages (JSP), JavaServer Faces (JSF), or rich Java components to render the user interface. The View layer can be a web (browser) client, a client-server, Swing-based desktop application, a Microsoft Excel spreadsheet, or a wireless device such as a smartphone.

- **Controller**: The Controller layer provides a mechanism to control the flow of a web application and handle user input. For example, when you click a Search button on a page, the Controller layer determines what action to perform (do a search) and where to navigate (the results page). Using Oracle ADF's Controller layer, you can break your application's flow into smaller, reusable task flows, include non-visual components such as method calls in your flow, and create "page fragment" flows that run inside a region of a single (containing) page.

Oracle JDeveloper wizards provide design-time declarative security for Oracle Fusion Applications’ artifacts such as ADF Business Components and View and Controller objects, based on a service-oriented security architecture (described later in this document).

**Oracle Fusion Applications Java EE Applications**

An Oracle Fusion Applications Java EE application is a standalone unit of deployment. In Oracle Fusion Middleware parlance, all Oracle ADF BC components, e.g., ADF Library Java Archives (JAR) and service client JARs are packaged into an enterprise archive (EAR) file.

An Oracle Fusion Applications Java EE application is deployed against one pillar database and it may have dependencies on other pillars, but at runtime all referenced dependencies are executed within the Java EE application's own pillar. This means that any project containing Oracle ADF BC components referenced from a different pillar must be based on replicated tables, or service-based Entity Objects, or View Objects.

At run time, when an Oracle Fusion Applications Java EE application is deployed, the application’s EAR file contains all the Oracle ADF library and service client interface dependencies, i.e., all the public model and user interface projects referenced from other pillar families, as well as all the security metadata.
Oracle Fusion Applications’ Underpinnings

Oracle Fusion Applications technical underpinnings also include many Oracle Fusion Middleware services such as SOA (composite SOA applications for web services to communicate among themselves and connect to the database), identity management (for security and access control – the focus of this paper), content management, and a pervasive use of business intelligence and data integration. In addition, Oracle Fusion Applications are built on top of a single data model (one data schema for all applications).

Figure 2: Oracle Fusion Applications User Interface (Human Capital Management Example)

Oracle Fusion Applications’ Security

As mentioned earlier, Oracle Fusion Applications designers focused their effort on the business value of each application, leaving the critical security and identity management requirements as well as other logistical needs to be handled by Oracle Fusion Middleware.

Security is directly provided to Oracle Fusion Applications by a service-oriented security framework.
Service-Oriented Security

Key to Oracle Fusion Middleware is the concept of Service-Oriented Security (SOS). SOS provides a set of security services leveraged by all Oracle Fusion Middleware components and Oracle Fusion Applications.

Oracle’s SOS applies SOA principles to security in order to promote better design (industry-standard security “components”), deployment (appropriate level of security applied where necessary), and management (through a single point of administration). SOS is built upon Oracle Platform Security Services (OPSS), a security development framework described in the following section.

Oracle Platform Security Services

Oracle Fusion Applications designers leverage the Oracle Platform Security Services (OPSS) framework through Oracle JDeveloper security wizards.
Generally speaking, OPSS provides Oracle (as well as non-Oracle) product development teams, systems integrators, and independent software vendors with a standards-based, portable, integrated, enterprise-grade security framework for Java Platform, Standard Edition (Java SE) and Java Platform, Enterprise Edition (Java EE) applications, such as Oracle Fusion Applications.

OPSS insulates developers from the intricacies of tasks not directly related to application development by providing an abstraction layer in the form of standards-based application programming interfaces (API). Thanks to OPSS, Oracle Fusion Applications, in-house-developed applications, third-party applications, and integrated applications benefit from the same, uniform security, identity management, and audit services across the enterprise.

As shown in Figure 4, OPSS is the security foundation for Oracle Fusion Middleware: all Oracle Fusion Middleware components and Oracle Fusion Applications “consume” the OPSS framework’s services.

OPSS is a self-contained, portable environment that runs on an application server such as Oracle WebLogic Server. At development time, OPSS services are directly invoked from the development environment (Oracle JDeveloper) through wizards. When the application is deployed to the runtime environment, systems and security administrators can access OPSS services for configuration purposes through Oracle Enterprise Manager Fusion Middleware (FMW) Control, command line tools such as WebLogic Scripting Tool (WLST), and more specifically Oracle Authorization Policy Manager (APM), described later in this document.

OPSS complies with the following standards: Role Based Access Control (RBAC); Java Platform, Enterprise Edition (Java EE), Java Authorization and Authentication Services (JAAS), and Java Authorization Contract for Containers (JACC).

OPSS includes Oracle WebLogic Server’s internal security services consumed by a Security Services Provider Interface (SSPI), which is also part of OPSS. In addition, OPSS includes Oracle Fusion Middleware’s security framework (formerly referred to as Java Platform Security (JPS) or JAZN).

- SSPI provides Java EE container security in permission-based (JACC) mode and in resource-based (non-JACC) mode. It also provides resource-based authorization for the environment, thus allowing customers to choose their security model. SSPI is a set of APIs designed to implement pluggable security providers in order to support multiple types of security services, such as custom authentication or a particular role mapping.

- JPS was first released with Oracle Application Server 9.0.4 as a JAAS-compatible authentication and authorization service working with XML-based and Oracle Internet Directory providers. In 11g, JPS has been expanded to include the following services (described later in this section): Credential Store Framework (CSF), User and Role API, Oracle Fusion Middleware Common Audit Framework (CAF), and Oracle JDeveloper/ADF integration (application security life cycle support).

In addition, OPSS includes Oracle Security Developer Tools (OSDT), a set of Java-based cryptographic libraries supporting XML signature, XML encryption, XML Key Management Specification (XKMS), Security Assertion Markup Language (SAML), WS-Security, and other non-
XML standards such as Secure / Multipurpose Internet Mail Extensions (S/MIME) and Online Certificate Status Protocol (OCSP).

OSDT is used in many Oracle products including Oracle Fusion Middleware components and Oracle Fusion Applications. OPSS leverages OSDT for SSL configuration and Oracle Wallet (used by Oracle Identity Management products, Oracle Enterprise Manager, and Oracle Database).

OPSS provides out-of-the-box support for (1) applications using WebLogic Server’s internal security and SSPI, and (2) applications using JPS, such as Oracle Fusion Applications, Oracle WebCenter, Oracle SOA, and Oracle Web Services Manager.

Developers can use OPSS APIs to build security features for all types of applications and integrate them with other security artifacts, such as Lightweight Directory Access Protocol (LDAP) servers, database systems, and custom security components. Administrators can use OPSS to deploy large enterprise applications with a small, uniform set of tools and administer all security in them. OPSS simplifies the maintenance of application security because it allows the modification of security configuration without changing the application code.

OPSS’s functional layers include:

**Authentication**: OPSS uses WebLogic Server authentication providers, components that validate user credentials or system processes based on a user name-password combination or a digital certificate. Authentication providers include the Default Authenticator, external LDAP stores, and database systems to host data for enterprise applications.

**Identity Assertion**: The WebLogic Identity Assertion providers support certificate authentication using X.509 certificates, Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO) tokens, SAML assertions, and CORBA Common Secure Interoperability version 2 (CSIv2) identity assertion.

---

![Figure 5: Oracle Platform Security Services Architecture](image-url)
Single sign-on (SSO): Authentication providers can use different types of systems to store security data. The Authentication provider that WebLogic Server installs uses an embedded LDAP server. Oracle Fusion Middleware 11g also supports perimeter authentication and SSO through Oracle Access Manager (OAM), described later in this document. For small environments that don’t need to be integrated with an enterprise SSO solution such as OAM, lightweight SSO is provided by a SAML-based solution using WebLogic Server's SAML Credential Mapping Provider.

Authorization: OPSS provides a Java policy provider that supports code-based and subject-based authorization.

Note: A (Java) subject is a grouping of related security information that includes a collection of principals such as a name (“John Doe”), an email address (“jd@oracle.com”), together with (optional) security-related attributes (credentials) such as passwords or cryptographic keys. The Java class javax.security.auth.Subject represents a subject and an instance of this class is created and populated with principals when authentication succeeds. OPSS authentication providers enable identity propagation across multiple components in a domain through subjects.

OPSS supports application roles (logical roles specific to an application). Unlike Java EE’s logical roles, OPSS supports role hierarchy. OPSS also provides an advanced policy model that includes elements such as resource types (e.g., an Oracle ADF task flow) and entitlement sets (authorized actions on a given resource instance) allowing complex authorization policies to be conveniently defined and managed. Using Oracle Enterprise Manager FMW Control or WLST, the administrator can manage an application’s authorization policies, including mapping application roles to enterprise groups and users, or editing the permissions granted to an application role. OPSS also provides a policy management API allowing programmatic control over authorization policies.

User and role: OPSS’s User and Role API framework allows applications to access identity information (users and roles) in a uniform and portable manner regardless of the particular underlying identity repository. The User and Role API frees the application developer from the intricacies of particular identity sources.

Role mapping: OPSS supports the mapping of application roles to enterprise groups in the domain Policy Store, no matter the kind of domain policy repository employed (file-based or LDAP-based). This mechanism allows users in enterprise groups to access application resources as specified by application roles.

 Security stores: The Identity Store is the repository of enterprise users and groups. The Policy Store is the repository of application and system policies. The Credential Store is the repository of domain credentials. Credentials are used during authentication when principals are populated in subjects, and during authorization when determining what actions the subject can perform. OPSS provides the Credential Store Framework (CSF), a set of APIs that applications can use to create, read, update, and manage credentials securely. OPSS uses one logical store to keep both policies and credentials. OPSS’s security stores are virtualized through Oracle Virtual Directory (OVD).

Audit: OPSS provides a common audit framework (CAF) for Oracle Fusion Middleware products. Customers using OPSS automatically get the benefit of audit without writing a single line of audit-related code. CAF provides out-of-the-box customizable analytical reporting capabilities within Oracle Business Intelligence Publisher; data can be analyzed on multiple dimensions (e.g., Execution Context
Identifier (ECID) or user ID) across multiple components or products involved in a single transaction. ECID is basically used to track the flow of a particular request through the various layers of the product stack. The ECID value for a particular request is generated at the first layer and is passed down to the subsequent layers. The ECID value is logged (and auditable) in each product involved in the transaction. ECID allows an administrator to track the end-to-end flow of a particular request across the product stack. The administrator can search the logs using a particular ECID value from Oracle Enterprise Manager FMW Control or using WLST. Audit logs can be queried for a given ECID through Oracle BI Publisher’s audit reports. For example, if you send an LDAP request to Oracle Virtual Directory front-ending Oracle Internet Directory, an ECID associated with the LDAP request is present in the OVD diagnostic logs and audit logs; similarly, when the query reaches OID, OID includes the same ECID in its diagnostic logs and audit reports. If the OPSS layer is also involved in the flow, the same ECID is passed on to the OPSS audit reports and diagnostics logs.

Application life cycle support: OPSS provides support for all the phases of an application’s life cycle. OPSS is integrated with Oracle JDeveloper, which allows an Oracle Fusion Applications or a custom application designer to model security into the application when building Oracle ADF task flows. Oracle JDeveloper also provides an authorization editor that allows developers to create authorization policies for ADF task flows and pages without writing a single line of code. Typically a developer deploys her application to a WebLogic Server domain embedded in JDeveloper. The developer can then deploy the application to a remote WebLogic Server domain using Oracle Enterprise Manager FMW Control. OPSS is integrated with FMWControl to allow application security policies and credentials migration to be configured during application deployment. Post deployment, an administrator uses FMWControl to manage the application’s security policies, e.g., edit authorization policies, or change audit policies. All such changes are transparent to the application and do not require any application code change. In any non-trivial application scenario, an application normally goes from development to a staging (or test) environment before being put in full-blown production. OPSS supports this model by providing migration tools that move security policies from a test domain into a production domain. For example, audit policies configured in a test domain can be exported into the target production domain.

Oracle Authorization Policy Manager

Oracle Authorization Policy Manager (APM) is a graphical user-interface console for managing OPSS-based authorization policies. APM was specifically designed to support Oracle Fusion Applications’ security policies using a centrally managed approach (see Figure 6).

APM is designed for customers relying on Oracle Fusion Middleware products consuming OPSS services, such as Oracle Fusion Applications, or OPSS used by in-house custom applications.

APM is a standards-based environment (JAAS permissions and enterprise Role Based Access Control) that supports delegated administration, advanced life cycle management, and identity store access through Identity Governance Framework (IGF) / ArisID (described later in this document).
APM-Administered Artifacts

APM administers both global and application-specific artifacts. Global artifacts include users, external roles, and system policies.

Global artifacts apply to all application stripes (an application stripe is a logical subset of the domain Policy Store where the application policies are kept).

Application-specific artifacts include the resource catalog, application policies, application roles, and role categories. Application-specific artifacts apply to a single application stripe.

Figure 6: Oracle Authorization Policy Manager for Oracle Fusion Applications

OPSS Authorization Policy Model Concepts

**Resource Type**: A template of secured artifacts is represented as a Resource Type. An Oracle ADF task flow is a good example of a Resource Type.

**Resource Instance**: Each secured resource of a given type is represented as a Resource Instance (e.g., orderEntryTaskflow) and points to a physical resource.

**Entitlement**: Aggregates resources and allowable actions, and encapsulates privileges sufficient for a task (e.g., CreatePOTaskFlow).

**External Role**: A collection of users and other groups, synonymous with enterprise role or enterprise group, typically implemented as an LDAP group in the Identity Store.

**Application Role**: A logical and hierarchical role that exists in the Policy Store. An Application Role is tagged via the Role Catalog.

**Application Policy**: A collection of entitlement and resource permissions granted to a principal (an Application Role or an External Role are examples of principals).

**System Policy**: A global policy that grants an application access to OPSS’s APIs.
**Role Mapping**: Role mapping allows users to access protected application resources. Application Roles are mapped to External Roles.

**Oracle Fusion Applications’ Security Reference Implementation**

From an administrator’s point of view, Oracle Fusion Applications includes multiple roles in *function security* and *data security* categories.

Function security privileges are used to control access to a page or specific functionality within a page. Data security includes privileges conditionally granted to a role. Data security privileges are used to control access to data.

In accordance to RBAC standard principles, roles are designed based on jobs, and permissions to resources are associated to roles. Users are assigned roles based on the duties they need to perform.

Within Oracle Fusion Applications the role, role memberships, and privilege and data security policies are authored at design time. The seeded content for function and data security can subsequently be changed by customers.

Oracle Fusion Applications’ RBAC roles are as follows:

- **Job Role**: Defines the job that a person is hired to do, for example Buyer, Lawyer, etc.; Job roles are decomposed into their constituent duties and are provisioned to a user.
Fusion data roles are authored by the customer that grants conditional access to application data (during customer implementation). A data role grants one or more data security privileges on an object or attribute group for a condition. These grants then enumerate the data privileges that are enabled for the provisioning of that data role.

Identity Governance Framework and ArisID

As previously mentioned, OPSS leverages its User and Role API to provide developers with simple methods to manage identities. However, developers will still be inclined to map the User and Role API to business objects. To simplify this development process, Oracle has created the Identity Governance Framework (IGF) project, now hosted by the Liberty Alliance (www.projectliberty.org).

IGF is designed to help enterprises control how identity-related information (e.g., attributes and entitlements) is used, stored, and propagated between applications.

ArisID (the open source implementation of the IGF standard, described later in this section) allows developers to build Oracle Fusion Applications (as well as in-house applications) that access identity-related data from a wide range of sources, and administrators and deployers to define, enforce, and audit policies concerning the use of identity-related data.

The IGF specification’s functional layers include:

- **Client Attributes Markup Language (CARML):** A specification built by the developer during the development process. CARML indicates the required and optional attributes, operations, and indexes the application will use when deployed (CARML is to an application what WSDL is to a web service). The application developer uses the CARML API to both declare the attribute data needed for the application and the operations needed to support the application (the CARML API uses SAML and SOAP-based protocols to communicate with attribute services).

- **Attribute Authority Policy Markup Language (AAPML):** An Extensible Access Control Markup Language (XACML) profile designed to allow attribute authorities to specify conditions under which information under management may be used (and possibly modified) by other applications.

- **Attribute service:** A web service that reads the CARML file in order to configure “views” of one or more attribute authorities that meet the requested data requirements of the application specified in the CARML document (ArisID is a typical example of such a service).

ArisID Identity Beans (ArisID for short) is an Apache-licensed project hosted by the Kantara Initiative at http://openliberty.org/wiki/index.php/ProjectAris. ArisID is a genuine open source project where anyone is welcome to participate in and contribute to the development of this new technology (you can download ArisID from http://arisid.sourceforge.net/).
ArisID is designed for developers to access identity information using a single API. ArisID enables access and management of identity information stored in different types of repositories accessed using different protocols. ArisID enables developers to create their own virtual identity database while retaining the ability to interconnect with enterprise identity services (more on this later in this document).

ArisID uses a declarative, multifunction API that depends on provider services to do the work of data mapping, protocol transformation, and connectivity. The Oracle Virtual Directory (OVD) Provider for ArisID is an example of an ArisID provider service. The OVD Provider for ArisID is a library that enables OVD to provide identity services to an application using the ArisID API. In this way, OVD plus the OVD Provider library for ArisID and the ArisID API library comprise a complete set of libraries that can be used by applications to access identity services.

In the current release of ArisID, Oracle provides a set of beans known as UserRole Beans (used by Oracle Fusion Applications) replacing the User and Role API mentioned earlier in this document. Not only are ArisID-based beans 100% open source, they offer true de-coupling of client beans from physical infrastructure dependencies and physical data models. For customers, this means that Oracle Fusion products (including Oracle Fusion Applications) have vastly improved flexibility to deploy in varied computing infrastructure environments. For example, whether an enterprise is using Microsoft Active Directory or Oracle Directory Server Enterprise Edition (DSEE), the functionality is the same, despite a different underlying data model for these directory products. The UserRole Bean API also offers relationship functionality that makes it easy to pull information based on identity relationships, e.g., pull the User bean for the manager of the currently authenticated user.

Oracle Fusion Applications share a common "User" profile bean as defined by OPSS (Note: user information is authored from within the Human Capital Management (HCM) systems and is made available to OPSS). This is used to define both a credential and typical user profile using more than 70 standardized attributes across all of Oracle Fusion Applications. In this way, Oracle Fusion Applications continue to have separation between application data and user credential information (as expressed in the User bean).

In the future, ArisID’s Java language binding will be enhanced. For example, Oracle Fusion Applications designers will be able to create annotated Java objects that are managed using injection and/or entity managers. As this happens, the use of CARML by Oracle Fusion Applications will become more and more specific to each individual product, providing the following advantages:

- Identity programming will be similar to typical handling of objects persisted in a database, providing for wider development tool support, and reducing or completely eliminating the special knowledge that developers currently need to have in order to build applications that integrate well with Oracle’s identity management services.
- Each product will have its own CARML manifest, which can be used by customer privacy or regulatory officers to understand how personal information is used and propagated within Oracle products.
• As more products use declarative CARML, Java middleware will become more powerful and substantially easier to configure and manage in addition to continued support for varied identity services environments.

Oracle Fusion Applications’ Platform Security

Oracle Fusion Applications designers develop instances of resources such as Oracle ADF task flows (e.g., submit_cash_in_transit_report or add_capitalized_asset task flows) and they leverage the OPSS framework to secure these resources (permissions are granted to application roles). In other words, Oracle Fusion Application designers use OPSS to link entitlements to privileges and security administrators grant privileges to application roles. These operations result into corresponding entitlement grants in the OPSS Policy Store.

At runtime, Oracle Fusion Applications resolve the task flow instance and action into a checkPermission call. The OPSS framework’s authorization service uses assigned policies to check whether the user is authorized to perform an operation.

Oracle Fusion Applications are designed to run off-line if necessary, for example the CRM applications can run on a laptop disconnected from the network. Security in off-line mode consists in storing and applying a local version of Oracle Fusion Applications security as well as user policies, and synchronizing the possible server-level security changes with the local copy of the user policies to make sure the latest version of these policies is enforced.

Typically, Oracle Fusion Applications invoke OPSS to get a copy of the user’s authorization information (OPSS queries the LDAP directory to get the user’s group membership which it uses to extract the user’s policies from the OPSS Policy Store and provide the authorization privileges across all the Oracle Fusion Applications for which the user is provisioned). Oracle Fusion Applications upload the transactional and authorization data onto the end user’s machine via an on-demand synchronization process.

Oracle Fusion Applications and Oracle Identity Management

In the first part of this document, we saw how Oracle Fusion Applications designers leverage the OPSS framework (platform security) to make Oracle Fusion Applications secure.

In this second part, we look at how the various Oracle identity services and Oracle Identity Management components support Oracle Fusion Applications (product security).

Platform Security Versus Product Security

Platform security (the OPSS framework) is an integral part of Oracle Fusion Middleware ensuring functional security. It runs on Oracle WebLogic Server, and will soon support other Java EE containers such as IBM WebSphere and Red Hat JBoss.

Product security, on the other hand, is delivered by the identity and access control services provided by Oracle Identity Management. Because Oracle Identity Management, like any other Oracle Fusion
Middleware component, consumes OPSS framework services, product security includes platform security plus enterprise identity services.

Product security is designed to meet the requirements of centralized management and high scalability. Whereas platform security requires that an instance of the solution be installed per deployment, product security allows for administration of policies and identities across multiple deployments through a single instance of the Oracle Identity Management suite’s components (i.e., a single point of administration for the whole Oracle Fusion Applications environment deployed across the enterprise).

Identity As A Service

Oracle Identity Management leverages the OPSS framework to provide “identity as a service.” Identity services take the functionality of an identity management solution that would otherwise be bolted onto Oracle Fusion Applications and make the set of identity services available in a SOA environment.

Because Oracle Fusion Applications follow SOA guidelines, they are able to leverage these services without any concern about how these services are provided. Shared identity services enable enterprises to make identity a reusable, standard, transparent, and ubiquitous part of their applications.

Oracle Identity Management's Key Services

Oracle Identity Management 11g provides a comprehensive set of services as shown in Figure 8: Identity administration; access management; directory services; identity and access governance; platform security; operational manageability.

Instead of cobbling together a heterogeneous environment from diverse, separate products, each service (for example user on-boarding) works with other identity services through standard interfaces to provide a complete, homogeneous environment.

---

<table>
<thead>
<tr>
<th>Identity Administration</th>
<th>Access Management</th>
<th>Directory Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity Life Cycle Management (Provisioning, Reconciliation)</td>
<td>Single Sign-On; Federation Identity Assertion</td>
<td>Identity Persistent Storage</td>
</tr>
<tr>
<td>Approvals Workflow</td>
<td></td>
<td>Identity Virtualization</td>
</tr>
<tr>
<td>Self-Service Account Request</td>
<td>Multifactor, Strong Authentication</td>
<td>Identity Aggregation</td>
</tr>
<tr>
<td>Password Management</td>
<td>Risk-Based Authorization</td>
<td>Directory Integration</td>
</tr>
<tr>
<td>Enterprise Role Management</td>
<td>Fine-Grained Entitlements</td>
<td>Database User Security</td>
</tr>
</tbody>
</table>

**Figure 8: Oracle Identity Management 11g Services**
A SOA environment allows each service to leverage the environment within and outside identity management. For example, the workflow engine used in user provisioning approvals is the same, standards-based workflow engine used by Oracle SOA Suite. Likewise, the same standard cryptographic libraries are used throughout the identity management environment and other Oracle Fusion Middleware components.

The following tables summarize Oracle’s identity services and components by category.

### Platform Security Services

<table>
<thead>
<tr>
<th>COMPONENTS</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Authorization Policy Manager (APM)</td>
<td>APM is a graphical user interface console for administering OPSS-based authorization policies.</td>
<td>APM is intended for customers relying on Oracle Fusion Middleware products based on OPSS, custom or in-house applications built with Oracle ADF, and Oracle Fusion Applications.</td>
</tr>
<tr>
<td>Identity Governance Framework (IGF)</td>
<td>The IGF standard specification is designed to help enterprises control how identity-related information (e.g., attributes and entitlements) is used, stored, and propagated between</td>
<td>Originally started by Oracle, IGF is an open-source project hosted by The Liberty Alliance. ArisID is an open source</td>
</tr>
</tbody>
</table>
Oracle Fusion Applications Security

Authorization API (OpenAz)

Oracle’s Authorization API provides a standard interface between an application and a general authorization service. It also provides an effective way to enable authorization providers to plug in client-side authorization functionality.

Oracle Web Services Manager (OWSM)

OWSM secures standards-compliant web services (Java EE, Microsoft .NET, PL/SQL, etc.), SOA composites, and Oracle WebCenter’s remote portlets.

Directory Services

<table>
<thead>
<tr>
<th>COMPONENTS</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Internet Directory (OID)</td>
<td>LDAP directory server and directory integration platform implemented on top of Oracle Database technology providing unsurpassed level of scalability, high-availability, and information security. OID includes Oracle Directory Services Manager (ODSM), a web-based administration user interface for server configuration.</td>
<td>Highly scalable LDAP directory integrated with Oracle Fusion Middleware and Oracle Fusion Applications.</td>
</tr>
<tr>
<td>Oracle Virtual Directory (OVD)</td>
<td>Java-based environment designed to provide real-time identity aggregation and transformation without data copying or data synchronzation. OVD includes two primary components: the OVD Server to which applications connect, and ODSM (described above).</td>
<td>OVD provides a single standard interface to access identity data no matter where it resides while hiding the complexity of the underlying data infrastructure (OVD does not store information, this role is left to the persistence systems used for that purpose, such as OID and ODSEE).</td>
</tr>
</tbody>
</table>

Access Management

<table>
<thead>
<tr>
<th>COMPONENTS</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Access Manager (OAM)</td>
<td>OAM provides centralized, policy driven services for web applications authentication, web single sign-on (SSO), and identity</td>
<td>OAM integrates with a broad array of authentication mechanisms, third-party web servers and application servers.</td>
</tr>
</tbody>
</table>
assertion.

and standards-based federated SSO solutions to ensure maximum flexibility and a well-integrated, comprehensive web access control solution.

Oracle Identity Federation (OIF)

OIF is a self-contained solution enabling browser-based, cross-domain single sign-on using industry standards (SAML, Liberty ID-FF, WS-Federation, Microsoft Windows CardSpace).

OIF seamlessly integrates with third-party identity and access management solutions. OIF is specifically designed for identity providers.

Oracle OpenSSO Fedlet

A lightweight federation extension allowing a service provider to immediately federate with an identity provider without requiring a full-blown federation solution in place.

Oracle’s Fedlet is specifically designed for service providers and fully integrated with OIF.

Oracle OpenSSO Security Token Service (STS)

Oracle’s STS establishes a trust relationship between online partners through web services. STS provides both standard and proprietary security token issuance, validation, and exchange.

STS is currently available with the Oracle Access Management Suite Plus. Going forward, Oracle’s STS will be integrated with OAM.

Oracle Enterprise Single Sign-On (eSSO)

Oracle eSSO is a Microsoft Windows desktop-based set of components providing unified authentication and single sign-on to both thick- and thin-client applications with no modification required to existing applications.

Using Oracle eSSO, enterprise users benefit from single sign-on to all of their applications, whether users are connected to the corporate network, traveling away from the office, roaming between computers, or working at a shared workstation.

Oracle Entitlement Server (OES)

OES is a fine-grained authorization engine that externalizes, unifies, and simplifies the management of complex entitlement policies.

OES provides a centralized administration point for complex entitlement policies across a diverse range of business and IT systems.

Oracle Adaptive Access Manager (OAAM)

OAAM provides resource protection through real-time fraud prevention, software-based multifactor authentication, and unique authentication strengthening.

OAAM consists of components that create one of the most powerful and flexible weapons in the war against fraud.

Identity Management, Identity and Access Governance

<table>
<thead>
<tr>
<th>COMPONENTS</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Identity Manager (OIM)</td>
<td>OIM is designed to administer both intranet and extranet user access privileges across a company’s resources throughout the entire identity management life cycle, from initial onboarding to final de-provisioning of an identity.</td>
<td>In extranet environments, OIM’s superior scalability allows enterprises to support millions of customers accessing the company’s resources using traditional clients (e.g., browsers) or smart phones.</td>
</tr>
</tbody>
</table>
Oracle Identity Analytics (OIA)

OIA helps enterprises address regulatory mandates, automate processes, and quickly make compliance a repeatable and sustainable part of business. OIA provides a comprehensive solution for attestation (access certification), role governance, and enterprise-level segregation-of-duties enforcement.

Integrates with OIM for role administration and role-based provisioning automation as part of Oracle remediation.

Operational Manageability

<table>
<thead>
<tr>
<th>COMPONENTS</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Identity Navigator (OIN)</td>
<td>OIN is an SSO-enabled launch pad for all of Oracle Identity Management services' administrative consoles.</td>
<td>OIN acts as a user experience consolidation point for Oracle Identity Management.</td>
</tr>
</tbody>
</table>

Oracle Identity Management Components In Oracle Fusion Applications

Oracle Fusion Applications ship with Oracle Identity Management and use a subset of the Oracle Identity Management services described above to meet the specific use cases of multiple Oracle Fusion Applications deployed across the enterprise:

- Oracle Directory Services (ODS), including Oracle Internet Directory (OID) and Oracle Virtual Directory (OVD)
- Oracle Identity Manager (OIM)
- Oracle Access Manager (OAM)
- Oracle Entitlement Server (OES)
- Oracle Authorization Policy Manager (APM)
- Oracle Web Services Manager (OWSM)

The following sections describe the deployment topologies for Oracle Fusion Application use cases together with a description of the Oracle Identity Management components used to meet the requirements of each use case.

Oracle Directory Services

Identity data in an Oracle Fusion Applications environment can be stored in Oracle Internet Directory (OID) or in the customer’s existing LDAP directory servers (e.g., Microsoft Active Directory or Oracle Directory Server, Enterprise Edition (DSEE), via Oracle Virtual Directory (OVD).
Customers benefit most using both OVD and OID: OVD to abstract access to identity data and OID to store the policy, role, and entitlement information used by other identity management components. In addition, OVD can be used to abstract access to a relational database containing information necessary for authentication or authorization.

Oracle Identity Manager

Oracle Identity Manager (OIM) is designed to administer both intranet and extranet user access privileges across a company's resources throughout the entire identity management life cycle, from initial on-boarding to final de-provisioning of an identity.

OIM exposes Service Provisioning Markup Language (SPML) interfaces that allow Oracle Fusion Applications to make direct calls to manage identity data, which avoids sending users to the OIM user interface to perform this type of operation.

OIM's functional layers include:

New metadata model: All configurations in various components of OIM are stored centrally in an XML store (Metadata Store –MDS) common to the various services provided by Oracle Fusion Middleware (Oracle SOA, WebCenter, etc.). This new metadata model allows you to run multiple jobs performing different types of reconciliation against the same target.

User provisioning: Provisioning provides outward flow of user information from OIM to a target system (e.g., Oracle Fusion Applications). Provisioning is the process by which an action to create, modify, or delete user information in a resource is started from OIM and passed into the resource (or target). The provisioning system communicates with the resource and specifies changes to be made to the account.

User administration: User administration includes self-service profile management (users can view and edit their own profile), administrative profile management (one can view and manage the profiles of other users subject to access permissions), request management (enables users to create provisioning requests for resources with fine-grained entitlements, profile management requests, and role
membership requests – approvers use the same user interface to process requests), delegated administration (by moving administration points as close to the user as possible, an organization can achieve tighter control and better security).

Policy Management: OIM enables policy-based automated provisioning of resources with fine-grained entitlements. For any set of users, administrators can specify access levels for each resource to be provisioned, granting each user only the exact level of access required to complete the job. These policies can be driven by user roles or attributes, enabling implementation of role based access control (RBAC) as well as attribute based access control (ABAC).

Workflow Management: OIM supports the separation of approval and provisioning workflows. An approval workflow enables an organization to model its preferred approval processes for managing resource access requests. A provisioning workflow enables an organization to automate IT tasks for provisioning resources with the most complex of provisioning procedures. OIM provides a Workflow Visualizer that allows business users, administrators, and auditors to visualize task sequences and dependencies to understand process flow, and a Workflow Designer to edit and manage the process flow. OIM's workflow leverages Oracle SOA's BPEL engine and Oracle JDeveloper at design time.

Password management: Password management includes self-service (users can reset their own passwords), advanced password policies (password length, alphanumerics and special characters usage, etc.), password synchronization (OIM can synchronize or map passwords across managed resources and enforce differences in password policies among these resources). OIM is tightly integrated with Oracle Access Manager to support password management.

Audit and compliance management: Audit and compliance management includes identity reconciliation (OIM tracks the creation, update, and deletion of an account across all managed resources – reconciliation is performed by the reconciliation engine described in the following paragraph), rogue and orphan account management (A rogue account is an account created "out of process" or beyond the control of
the provisioning system; an orphan account is an operational account without a valid owner), attestation (also referred to as recertification, attestation is mandated by the Sarbanes-Oxley Act -- OIM offers an attestation feature that can be deployed quickly to enable an organization-wide attestation process that provides automated report generation, delivery, and notification).

Reconciliation: The reconciliation process involves generation of events to be applied to OIM. These events reflect atomic changes in the target system, and contain the data that has changed, the type of change, along with other information. The reconciliation events that are generated as a result of changes occurring in the target system must be managed in such a way that they meet various business requirements. OIM's event management APIs, the reconciliation APIs, and the UI to manage reconciliation events are protected by using authorization policies controlled by Oracle Entitlement Server.

Segregation of Duties: The concept of Segregation of Duties (SoD) is aimed at applying checks and balances on business processes. Each stage of a business process may require the involvement of more than one individual. An organization can convert this possibility into a requirement for all IT-enabled business processes by implementing SoD as part of its user provisioning solution. The overall benefit of SoD is the mitigation of risk arising from intentional or accidental misuse of an organization's resources. In the OIM implementation of SoD, IT privilege (entitlement) requests submitted by a user are checked and approved by an SoD engine and other users. Multiple levels of system and human checks can be introduced to ensure that even changes to the original request are vetted before the request is cleared. This preventive simulation approach helps identify and correct potentially conflicting assignment of entitlements to a user, before the requested entitlements are granted to the user.

Approval and request management: With OIM, account request and approval processes can be automated to meet your organization’s needs. In intranet and extranet deployments, administrators, peers, or users themselves can initiate requests for access to resources and track the status of their requests through web applications and email notifications. OIM 11g features a new request model based on Oracle SOA’s (BPEL) approval workflow (design and orchestration). Approval workflows are highly configurable to accommodate multiple approval processes and stakeholders. OIM 11g provides Request Templates for persona-specific request catalogs.

Policy-based entitlement management: OIM’s policy engine controls fine-grained, attribute-level entitlements across managed applications through Oracle Entitlement Server-based authorization policies, automating IT processes and enforcing security and compliance requirements such as segregation of duties. Policy-based management of entitlements allows multiple-request and approval processes to be implemented and refined over time in parallel, reducing the total cost of implementation. Universal Delegated Administration is provided through the embedded Oracle Entitlement Server.

Adapter Factory: OIM integrates with any application or resource through highly configurable, agentless interface technology. Oracle provides a growing library of pre-configured connectors to popular applications and user repositories. Each connector supports a wide range of identity management functions and uses the most appropriate method of integration recommended for the target resource, whether it’s proprietary or based on open standards (e.g., SPML). Connecting to proprietary systems might be difficult. OIM’s Adapter Factory eliminates the complexity associated with creating and
maintaining these connections. The *Adapter Factory* provided by OIM is a code-generation tool that enables you to create Java classes.

*Integration with Governance, Risk, and Compliance:* Oracle Identity Manager is part of the multiple products making up Oracle’s Governance, Risk, and Compliance (GRC) infrastructure controls. Oracle Application Access Controls Governor, a key product in the Oracle GRC platform, allows customers to manage, remediate, and enforce enterprise resource planning SoD policies (the reference implementation that comes with Oracle Fusion Applications was built with these SoD policies). Enterprise resource planning roles and responsibilities are effectively segregated, thus minimizing the risk of fraud and ensuring regulatory compliance. Oracle Application Access Controls Governor also provides a comprehensive library of real-world, best-practice SoD controls for Oracle Fusion Applications. Oracle Identity Manager integrates with Oracle Application Access Controls Governor by performing real-time SoD validation prior to provisioning roles to Oracle Fusion Applications.

**Oracle Access Manager**

Oracle Fusion Applications support single sign-on (SSO) through the use of Oracle Access Manager (OAM).

![Figure 12: Oracle Fusion Applications and OAM](image)

OAM provides centralized, policy-driven services for authentication, single sign-on (SSO), and identity assertion. OAM integrates with a broad array of authentication mechanisms, third-party web servers and application servers, and standards-based federated SSO solutions to ensure maximum flexibility and a well-integrated, comprehensive web access control solution.
OAM provides authentication and SSO services in the web tier and integrates with applications and data providers by asserting authenticated identities to application access control systems.

OAM’s functional layers include:

**Authentication:** OAM’s Access Server, Policy Manager, and out-of-the-box web server plug-ins called WebGates (or AccessGates for integration with application servers, packaged applications, and other enterprise resources) work together to intercept access requests to resources, check for a pre-existing authentication, validate credentials, and authenticate users.

**Single Sign-On:** Typically, when a browser user attempts to access Oracle Fusion Applications, OAM first checks whether the applications are protected. If they are, OAM (through a WebGate) challenges the user for credentials (e.g., simple username / password, X.509 certificates, smart cards, etc.). Based on these credentials, OAM enforces its security policies to authenticate the user against a user store and creates a session ticket (in the form of an HTTP (browser) cookie) enabling single sign-on or repeated access to the same Oracle Fusion Applications without re-logging.

**Access control:** OAM allows coarse-grained authorization to resources based on user roles and access policies. Typically, following successful authentication, OAM provides access to a specific resource (e.g., an Oracle Fusion Applications main page) based on the authenticated user’s role. For example, a basic user and an administrator authorized to the same web application may have access to different levels of functionality through a personalized web page based on their role’s attributes.

**Support for Windows Native Authentication:** OAM enables Microsoft Internet Explorer users to automatically authenticate to their web applications using their desktop credentials. This is known as Windows Native Authentication (WNA). Cross-platform authentication is achieved by emulating the negotiate behavior of native Windows-to-Windows authentication services that use the Kerberos protocol. In order for cross-platform authentication to work, non-Windows servers (in this case, OAM) must parse SPNEGO tokens in order to extract Kerberos tokens subsequently used for authentication. With OAM single sign-on combined with WNA, a Kerberos session ticket is generated that contains the user’s log-in credentials (this Kerberos session ticket is not visible to the user). With WNA implemented, the user can click on a web application without another challenge for credentials; the Kerberos session ticket including the user’s credentials is passed through the browser to OAM. OAM validates the credentials by checking them against the Key Distribution Center (KDC) server on the Windows domain server.

**Compliance Reporting:** OAM includes unified and centralized audit reporting for all OAM components, with all operations stored and correlated in a secure database for analysis. OAM comes with pre-built reports and the ability to create custom reports through Oracle Business Intelligence Publisher in order to provide greater visibility and reporting on common events such as user access attempts, successful or failed authentications, and single sign-on events. These features improve an organization’s ability to meet common governmental and industry regulations.

Thanks to its tight integration with OPSS, OAM is able to make calls to container-managed applications (such as Oracle Fusion Applications) in order to invoke authentication events that are
enforced by OAM. In this case, the application makes the decision to authenticate by calling OPSS for log-in.

**Oracle Entitlement Server**

Oracle Fusion Applications that require support for fine-grained authorization leverage Oracle Entitlement Server (OES).

OES is an authorization engine that externalizes, unifies, and simplifies the management of complex entitlement policies. OES secures access to application resources and software components (such as URLs, Enterprise JavaBeans, and JavaServer Pages) as well as arbitrary business objects (such as customer accounts or patient records in a database).

OES's unique architecture allows Security Modules to be combined as a single policy decision point and policy enforcement point that runs in process with Oracle Fusion Applications to vastly increase the performance and reduce latency of runtime authorization decisions.

OES presents advantages over OPSS. As mentioned before, Oracle Fusion Applications can make standard authorization calls to OPSS and use APM to define authorization policies. However, OES has advanced features that OPSS doesn’t have such as a comprehensive policy language, obligations (responses), or constraints (e.g., “if x is greater than 3, then perform some action”). As a result, OES supports a richer set of authorization models. OES also directly supports data security, and has the Security Modules for various target systems with policy distribution and local decision caching, policy simulation, resource discovery, and a richer resource model.

**Oracle Web Services Manager**

Oracle Web Services Manager (OWSM) is designed to protect access to multiple types of resources including standards-compliant web services (Java EE, Microsoft .NET, PL/SQL, etc.); SOA composites including BPEL and enterprise service bus (ESB) processes; Oracle WebCenter’s remote portlets, and web services exposed by Oracle Fusion Applications.

OWSM 11g is installed as part of Oracle SOA 11g and Oracle WebCenter 11g. In addition, OWSM 11g is the runtime policy governance component for the Oracle SOA Governance solution. In this case, OWSM provides production assurance for deployed SOA artifacts through policy-based security and participates at various stages of the closed-loop life cycle control.

OWSM 11g includes a policy manager and interceptors or enforcement points (also known as agents). Both policy manager and agents run on Oracle WebLogic Server. Agents can be on the service requester side (client) and/or the service provider side (endpoint server). Typically, a request made to a web service is intercepted by an OWSM agent that enforces security policies defined in the OWSM policy manager. OWSM’s policy model, based on the WS-Policy and WS-SecurityPolicy standards, is the security lynchpin for Oracle Fusion Middleware’s web-services-based components.
Oracle Fusion Applications Security Process Flow

Figure 13 summarizes Oracle Fusion Applications’ security process flow.

Putting It All Together

Figure 14 represents a logical view of Oracle Fusion Applications security including all of the Oracle Identity Management components involved.

Figure 13: Oracle Fusion Applications Security Process Flow

Figure 14: Oracle Fusion Applications Security Logical View
Other Oracle Identity Management components can optionally be used, such as Oracle Adaptive Access Manager (OAAM) for resource protection through real-time fraud prevention, software-based multi-factor authentication, unique authentication strengthening, offline risk analysis and proactive actions to prevent fraud at critical log-in and transaction checkpoints, and Oracle Identity Federation in cases where Oracle Fusion Applications require single sign-on across multiple Internet domains. In addition, Oracle Identity Analytics (OIA) is used to complement Oracle Identity Manager in the area of identity and access governance. OIA provides a comprehensive solution for certification, role governance, enterprise-level SoD enforcement, a 360-degree view of user access (Cert360), and an Identity Warehouse designed to consolidate identities, resources, and entitlement information.

Conclusion

Oracle Fusion Applications are Oracle’s Java EE-based, next-generation enterprise resource planning applications. Oracle Fusion Applications leverage Oracle Fusion Middleware’s service-oriented security to protect access to resources.

For large-scale enterprise environments, Oracle Fusion Applications take advantage of Oracle Identity Management’s services, thus abstracting security from the applications, and administering the enterprise environment from a single point of control.