Oracle Identity Manager Architecture

An Oracle White Paper
July 2010

Executive Overview....3
Architecture Overview and Goals....3
The Java EE Architecture....4
Clustering....4
Load Balancing....4
Security Management....5
Messaging....5
The Oracle ADF Architecture....6
Oracle Identity Manager's Tiered Architecture....7
The Presentation Tier....7
Administration & End-User Console....7
Design Console....8
Custom Clients....8
The Business Services Tier....9
API Services....9
Integration Services....10
Pre-configured Connectors....11
Platform Services....12
The Data Tier....19
The OIM Database....19
The Metadata Store....20
The Identity Store....21
Security....22
Channel Security....22
Data Security....22
Remote Manager Security....22
Deployment Options....23
Option 1 – Simple Deployment....23
Option 2 – Clustered Deployment....24
Option 3 – Proxied Deployment....24
Option 4 – Partitioned Deployment....25
Remote Manager Deployment Options....26
Conclusion....27

Executive Overview

Oracle Identity Manager is a best-in-class identity administration and provisioning solution that automates the process of adding, updating, and deleting user accounts from applications and directories; and improves regulatory compliance by providing granular reports that attest to who has access to what. Oracle Identity Manager is available as a stand-alone product or as part of Oracle's award-winning Oracle Identity & Access Management Suite.

This technical whitepaper provides an overview of the technologies on which the OIM architecture is based, and then describes how these technologies are leveraged to deliver a scalable, high-availability provisioning solution than manages identity information across a typically heterogeneous environment.

Architecture Overview and Goals

Oracle Identity Manager is built on a state-of-the-art **Java EE architecture**. Java EE is a standard, robust, scalable, and secure platform that forms the basis for many of today's enterprise applications. Java EE provides a set of specifications for building multi-tier applications using the Java language. OIMs architecture is able to leverage the most flexible and supported cross-platform Java EE services available: a combination of Java, XML, and object technologies. This architecture makes Oracle Identity Manager a scalable, fault-tolerant solution for the most ambitious global deployments in the industry.

Oracle Identity Manager runs on leading Java EE compliant application server platforms, including Oracle WebLogic Server, to take advantage of the performance and scalability features inherent in these servers. Oracle Identity Manager also supports application server clustering for increased performance and virtually automatic failover in mission-critical computing environments.

Oracle Identity Manager's technology architecture is designed to deliver the specific functionality requirements expected of an industry-leading provisioning system. Specifically, the Oracle Identity Manager architecture is designed to meet the following goals and objectives:

- Time to Market – rapidly deploy Oracle Identity Manager services
- Performance – speedy response times and efficient navigation
- Portability – minimizing platform and external system dependencies
- Scalability – scale from low end to thousands of simultaneous users while managing millions of identities and user accounts
- Maintainability – easy to support and maintain
- Availability – always online and available when needed
- Reliability – consistency of application and transactions

Oracle Identity Manager meets all of the above goals and objectives with a well-designed architecture and application that is based on the **Oracle Application Development Framework** (Oracle ADF), an innovative, yet mature Java EE development framework available from Oracle. The architecture reflects the years of experiences and expertise possessed by Oracle in successfully producing and deploying enterprise level systems.

The Java EE Architecture

Oracle Identity Manager has been developed as a Java EE application to take advantage of the management, security, performance and scalability services provided by these industry-leading application servers to deliver a high-performance, fault tolerant enterprise application. Java EE defines a set of standardized, modular components, provides a complete set of services to those components, and handles many details of the application behavior.

The application server in which Oracle Identity Manager runs provides the life-cycle management, security, deployment and runtime services to the logical components that make up the Oracle Identity Manager Application. These services include

- Scalable Management of Resources (Clustering, Failover)
- Transaction Management
- Security Management
- Client Access
- Technology Resources (Database Connection Pooling, Messaging, etc)
- Other services required as part of a manageable server platform.

Clustering

A cluster in Java EE architecture is generally defined as a group of two or more Java EE compliant web or application servers that closely cooperate with each other through transparent object replication mechanisms to ensure each server in the group presents the same content. Each server (node) in the cluster is identical in configuration and networked to act as a single virtual server. Any Java EE server in the cluster can handle client requests directed to this virtual server independently, which gives the impression of a single entity hosting the Java EE application in the cluster.

High availability refers to the capability to ensure applications hosted in the middle tier remain consistently accessible and operational to their clients. It is achieved through the redundancy of multiple web and application servers within the cluster and is implemented by the cluster's "failover" mechanisms. If an application component fails processing its task, the cluster's failover mechanism reroutes the task and any supporting information to a copy of the object on another server to continue the task.
OIM has been architected to support a clustered environment. This includes ensuring that the EJBs and the Value Objects used to store data support serialization (for object replication to work). High availability also extends to the data tier in the OIM architecture.

Load Balancing

For a server cluster to achieve its high-availability, high-scalability, and high-performance potential, load balancing is required. *Load balancing* refers to the capability to optimally partition inbound client processing requests across all the Java EE servers that constitute a cluster based on factors such as capacity, availability, response time, current load, historical performance, and also administrative weights (priority) placed on the clustered servers.

A load balancer, which can be either software or hardware based, sits between the Internet and the physical server cluster, also acting as a virtual server. As each client request arrives, the load balancer makes near-instantaneous decisions about the Java EE server best able to satisfy that request.

The architecture of OIM takes full advantage of the built-in load-balancing capabilities of the application server it runs on.

Security Management

OIM architecture relies on the application server for certain security services as part of its overall security infrastructure. This is discussed in the Security section below that details the overall security model in Oracle Identity Manager.

Messaging

The basic concept behind messaging is that distributed applications can communicate using a self-contained package of business data and routing headers. These packages are called messages. While RMI and HTTP rely on a two-way active conversation between a client and a server, messaging relies on two or more interested parties communicating asynchronously through a messaging server (that is, without waiting for a response).

JMS (Java Messaging Service) is a wrapper API incorporated in the J2EE standard as a way to standardize messaging functionality. All industry standard application servers provide their own JMS server implementations as a part of their service offerings.
The Oracle Identity Manager architecture leverages messaging to provide better performance and load balancing. This is done within the Oracle Identity Manager application by leveraging messaging for *off-line processing*. Off-lining is a way to separate an end-user's interaction with the application from the processing that the user's interaction initiates. When the user initiates some action that will result in a lot of processing, it is desirable to return the control of the console to the user before the processing is finished. This can be accomplished by not initiating the processing right away based on the users action, but rather sending a message into the system's message queue regarding the action. Sending a message is a lightweight operation, and the user gets back a response instantly. The message can be picked up asynchronously, and processing can be initiated based on the content of the message. Since the message handlers can be distributed across the application server cluster, processing of multiple simultaneous user actions can be load-balanced across the different nodes in the cluster. Figure 1 shows an overview of message-based off-lining in the Oracle Identity Manager application.

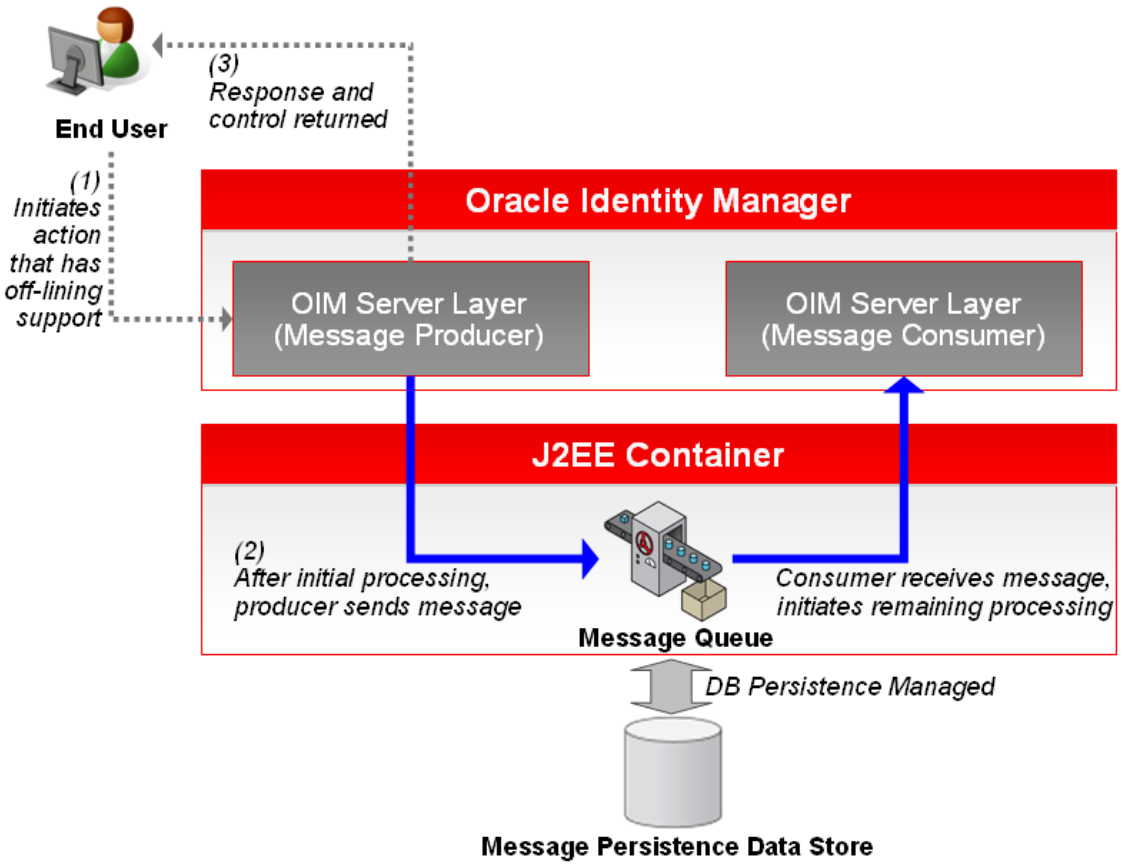

Figure 1: Message-Based Offlining

Messages are also persisted to a data store or to the file system for guaranteed delivery in the case of fail-over. The persistence of the messages is controlled by the application server, and therefore by the system administrator, in a standard way.

The Oracle ADF Architecture

Oracle ADF is based on the Model-View-Controller (MVC) design pattern. An MVC application is separated into: 1) a model layer that handles interaction with data-sources and runs the business logic, 2) a view layer that handles the application user interface, and 3) a controller that manages the application flow and acts as the interface between the Model and the View layers.

Separating applications into these three layers simplifies maintenance and reuse of components across applications. The independence of each layer from the others results in a loosely coupled, Service Oriented Architecture (SOA).

Oracle ADF implements MVC and further separates the model layer from the business services to enable service-oriented development of applications. The Oracle ADF architecture is based on four layers:

- The Business Services layer - provides access to data from various sources and handles business logic.

- The Model layer - provides an abstraction layer on top of the Business Services layer, enabling the View and Controller layers to work with different implementations of Business Services in a consistent way.
- The Controller layer - provides a mechanism to control the flow of the Web application.
- The View layer - provides the user interface of the application.

More information about Oracle ADF can be found on the Oracle Technology Network (OTN), at http://otn.oracle.com/products/jdev.

Oracle Identity Manager's Tiered Architecture

Oracle Identity Manager is based on a multi-tiered Java EE architecture. This section discusses each of these tiers in detail. Figure 2 illustrates this tiered architecture of Oracle Identity Manager.

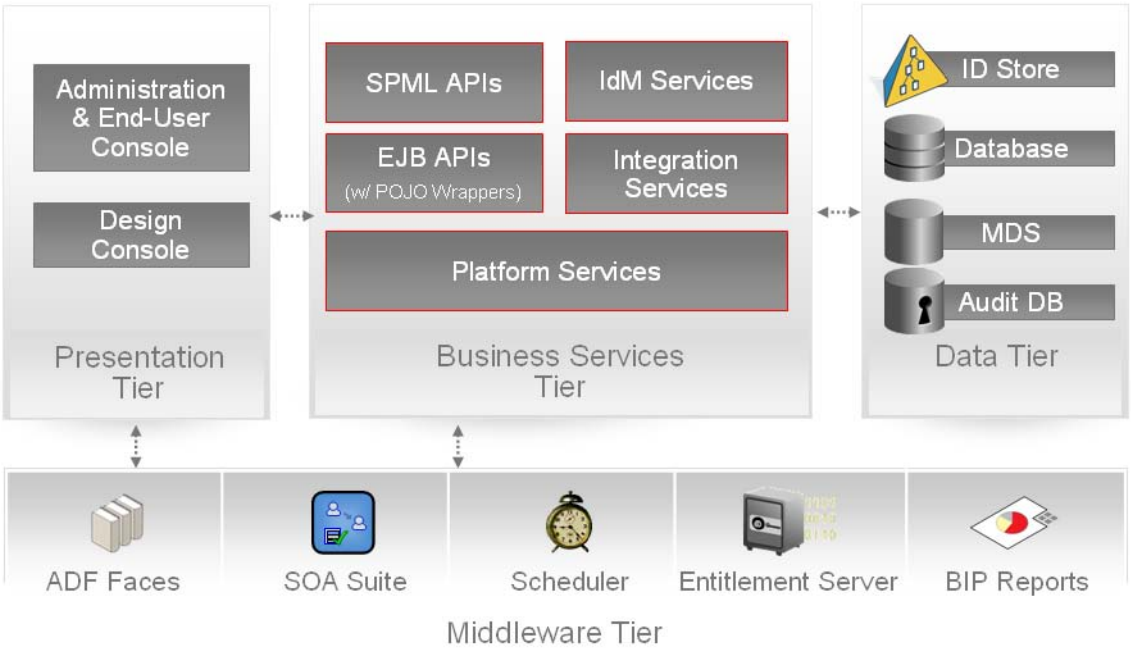

Figure 2: Oracle Identity Manager Tiered Architecture

The Presentation Tier

The Presentation layer represents the user interface of the application. In OIM, this is implemented following the MVC pattern as described in the previous section.

OIM has two UI clients – the *Administration & End-User Console* and the *Design Console*.

Administration & End-User Console

The Oracle Identity Manager *Administration & End-User Console* is a web based thin client that can be accessed from any web browser. The console provides user self-service and delegated administration features that serve the bulk of the user base of the system.

The *Administration & End-User Console* is implemented as a JSF (Java Server Faces) application that leverages *ADF Faces* - a large set of UI components built on top of the standard JSF APIs that leverage the latest technologies — including partial page rendering and Ajax — to provide a rich, interactive user interface.

Customization

Most of the screens in the Oracle Identity Manager Web Application are metadata and configuration driven, allowing customers to incorporate desired customizations easily and rapidly.

Additionally, the look and feel of the console can be customized via cascading style sheets (CSS).

Being based on the JSF framework, the web application supports a great deal of configurability and customization. Customers can easily extend the Oracle Identity Manager Web Application with pages and functional flows that are more suitable for their environment.

Modular Design

The Oracle Identity Manager Web Application is built in a very modular manner with mostly reusable components and widgets that customers can leverage when extending the application to suit their needs. This eliminates a great deal of the work necessary to build components for custom flows.

Interface Philosophy

The interface philosophy in the Oracle Identity Manager Web Application is based on ease-of-use and intuitive user interaction. A number of wizards are provided to allow users to step through commonly needed tasks in a well thought-out, intuitive manner.

Internationalization

The Oracle Identity Manager Web Application provides support for localization of the interface into 27 languages, so that users always see the interface in the language of their preference.

Design Console

The Oracle Identity Manager *Design Console* is a feature-rich, sophisticated client accessed using a desktop Java client. The Design Console provides the full range of Oracle Identity Manager's system configuration and development capabilities including Form Designer, Workflow Designer and Adapter Factory.

The *Design Console* is implemented as a Java Swing client that communicates directly with the Business Services layer in the application. It also supports a highly sophisticated delegated administration model, guaranteeing that users can only work on those parts of the application configuration that they have been given privileges to.

Custom Clients

The client environment for Oracle Identity Manager is highly customizable via well-documented Java APIs. In quite a few enterprises, there is a need for the provisioning

system to support a custom developed client. Some of the needs that may drive this include:

- Integration of the client into an existing enterprise portal
- Creation of custom flows for user interaction
- Creation of custom pages built around the customer's unique needs from the provisioning system
- Adherence to enterprise portal standards

In order to support such customization, Oracle Identity Manager exposes the bulk of the necessary functionality via its published public APIs (see below). A comprehensive Software Development Kit (SDK) is also provided to assist in the development effort

The Business Services Tier

The Business Services Layer for Oracle Identity Manager is implemented as an EJB application. The core functionality for the Oracle Identity Manager platform is implemented in Java using a highly modular, object-oriented methodology. This makes the application extremely flexible and extensible.

The Business Services Layer for Oracle Identity Manager includes the following services and capabilities:

- The *Core Services* that comprise the core of the business features offered by Oracle Identity Manager – like the User Management Service, the Policy Management Services, the Provisioning and Reconciliation Services, among others.
- The *API Services* that describe the APIs supported by the product that allow custom clients to integrate with OIM.
- The *Integration Services* based on the Adapter Factory and Connector Framework, which dynamically generates integration code based on the metadata definition of the adapters.
- The *Platform Services* that are crucial to the business features offered by Oracle Identity Manager – like the Workflow Service, the Request Service, the Entity Manager Service and the Scheduler Service.

Since this whitepaper is regarding the technical architecture of Oracle Identity Manager, it will not cover the Core Services.

API Services

Oracle Identity Manager provides a rich set of APIs that expose the business functionality of the product for use by custom clients, in product customization, and in plug-in and adapter development.

SPML APIs

SPML (Service provisioning Markup Language) is an OASIS standard for managing the provisioning and allocation of identity information and system resources within and between organizations.

Oracle Identity Manager supports a set of SPML-based web services that expose identity administration functionality to consuming clients The APIs are based on version

2.0 of the OASIS SPML spec, and implement the XSD profile. The APIs provide support for coarse-grained functionality like:

- Add, modify, and delete of **Identities**
- Add, modify, delete of **Roles**
- Add and delete of **Role Memberships**

Please note that these APIs support requests coming into OIM for administration purposes, which is distinct and separate from SPML as the protocol used to integrate with provisioning targets (covered in the Integration Services below).

EJB APIs

Highly granular access to the functionality of the platform is via a set of EJB (Enterprise Java Beans). These session beans are the basis for functionality implemented in the Oracle Identity Manager Web Application. It is also the interface that custom clients can use to access Oracle Identity Manager capabilities.

The APIs, which are implemented as Stateless Session EJBs, use the Java EE infrastructure to provide the lookup and communication mechanisms.

Integration Services

A scalable and flexible integration architecture is critical for the successful deployment of provisioning solutions. Oracle Identity Manager offers a proven integration architecture for fast and low-cost deployments.

Connector Framework provides rapid integration to commercial or custom systems. The connector framework separates connector code (integration libraries specific and optimized for the target system) from connector meta-data (data models, forms, connectivity information and process). This separation makes extending, maintaining and upgrading connectors a manageable and straightforward process. Version changes on the target system simply require replacement of the connector code without requiring a revisit of the overall integration design. Custom logic is more easily plugged in through custom extensions that do not prevent customers from leveraging improved versions of the connector code.

This connector architecture is a best-of-breed combination of the capabilities from the Oracle Identity Manager and Oracle Waveset (formerly Sun Identity Manager) products. It relies on the Identity Connectors that were part of the Oracle Waveset product, and provided robust integration libraries to a variety of managed targets based on a well-defined and complete integration SDK.

Oracle has defined the common connector framework that allows these identity connectors to be used in the same manner, without need for modification, in both products moving forward. This allows Oracle to offer an unprecedented level of support for both connector development and pre-defined connectors to its customers.

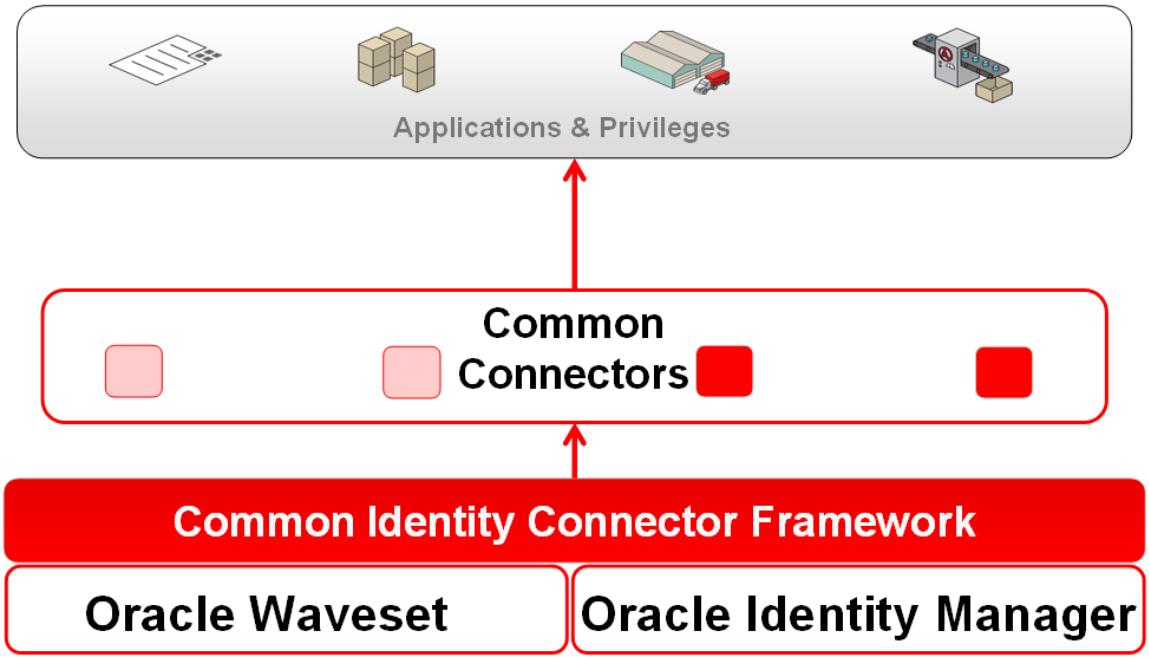

Figure 3: Oracle Identity Manager Connector Framework

The connector framework also provides comprehensive connector lifecycle management through wizard-driven install and upgrade of connectors with version control, one-click cloning of installed connectors and a complete uninstall process.

Oracle Identity Managers integration services provide all the pieces necessary to support the development, deployment and maintenance of connectors in the product. In addition to the connector framework, this includes entity management services, metadata management, code generation tools and deployment tools.

Pre-configured Connectors

For the most popular commercial applications and interface technologies, Oracle Identity Manager offers an extensive and rapidly expanding library of pre-configured connectors. With these connectors, an enterprise can get a head start on application integration. Each connector supports a wide range of identity management functions and uses the most appropriate integration technology recommended for the target resource, whether it's proprietary or based on open standards. These connectors enable out-of-the-box integration, but can be enhanced to work with each enterprise's unique integration requirements.

Remote Manager

The Remote Manager is an Oracle Identity Manager server component that runs on a target system machine, providing the network and security layer necessary to integrate with applications that do not have network-aware APIs or do not provide security. It is built as a lightweight RMI (Remote Method Invocation) server. The communication protocol is RMI tunneled over HTTP/S.

The J2EE RMI framework enables the creation of virtually transparent, distributed services and applications. RMI-based applications consist of Java objects making method calls to one another without regard for their location. This allows one Java object to invoke methods on another Java object residing in another virtual machine in the same manner in which methods are invoked on a Java object residing in the same virtual machine. An overview of the Remote Manager architecture is shown in Figure 4.

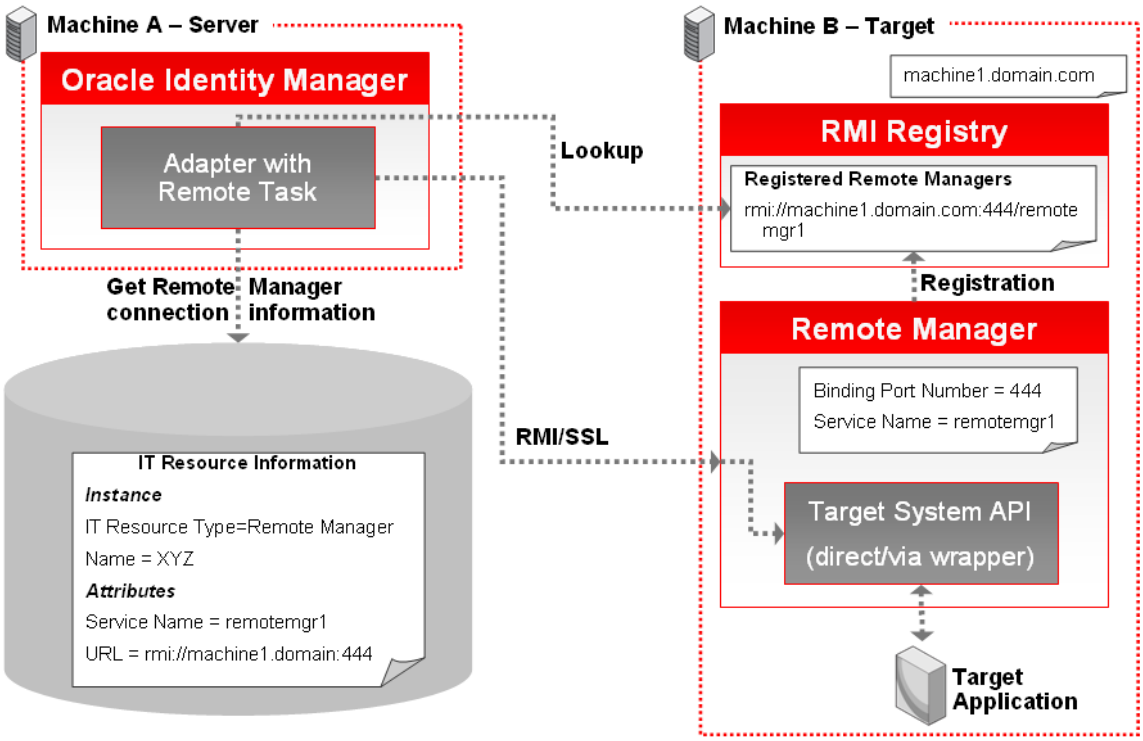

Figure 4: Remote Manager Architecture

Platform Services

Workflow Service

The ability to impose workflow based controls on the identity management operations being executed in Oracle Identity Manager is a critical component of its ability to satisfy the regulatory and compliance needs of an organization. To that end, the Oracle Identity Manager architecture includes a robust workflow service that allows customers to configure approval and provisioning workflows in the product. To deliver this functionality, Oracle Identity Manager relies on the Oracle SOA Suite.

Oracle SOA Suite is a standards-based best of breed suite that enables you to build service-oriented applications and deploy them to your choice of middleware platform. It consists of a number of components, but for the purposes of delivering comprehensive workflow capabilities, Oracle Identity Manager relies on a few key components:

- BPEL Process Manager
- Human Workflow Service
- BPEL Designer

Oracle BPEL Process Manager provides a comprehensive, standards-based and easy-to-use solution for creating, deploying and managing cross-application business processes with both automated and human workflow steps. It also provides audit trails for both completed and in-flight processes, and process history that enables process improvement.

While the BPEL standard does not itself cover manual tasks, it has rich support for asynchronous services. Therefore, the Oracle SOA Suite supports a manual task service called the Human Workflow Service, so that people and manual steps can be included in standard BPEL processes. The Oracle Identity Manager Administrative Console includes a task list that allows users to view and interact with assigned tasks being managed within the Human Workflow Service.

The Oracle BPEL Designer is available as a plug-in for Eclipse and JDeveloper and offers a rich visual design paradigm for creating and deploying BPEL based processes.

Oracle Identity Manager provides an abstraction service on top of the SOA suite that optimizes and simplifies the interaction of users with the SOA suite. This service includes capabilities to register BPEL composites for use in OIM, define parameterized variables for use in the BPEL and HW modules, and APIs that are used by the OIM task list and custom development. This service follows a provider model that can be extended in future releases to accommodate other workflow engines.

Request Service

The Request Service provides the services used to raise and track requests in the system. A request allows a user (or the system on behalf of a user) to ask that an action be taken after obtaining the necessary approvals, and that a tracking record of the entire process and its status be maintained. The request could be for any kind of action (defined as Request Types):

- Creating, modifying or deleting an identity
- Enabling or disabling an identity
- Provisioning a resource to a user or a set of users
- Adding or removing an identity as a member of a role

The request service will support different types of requests through the ability to accommodate multiple **request models**. A request model defines the behavior of the request service for a particular type of request. It provides the details that the request service needs in order to take that request type through its entire lifecycle from start to finish. Oracle Identity Manager ships with a number of predefined request models that cover the most common use cases. The pluggable request model feature will allow new request types to be added into the product in future versions without requiring an update to the core. Also, future versions will allow a customer to define a new model for, say, a vacation request.

The request service defines the flow models by which data provided in a request flows through the various services in OIM. This includes invoking approval workflows at the correct time, monitoring the status of the workflows, and executing the request if approval is received.

The request service also provides support for **request templates**. A request template defines a particular scenario for creating requests of a certain type. The template is a simplified overlay on top of a request model that allows the person defining that template to control how a request gets created, and add additional layers of approval, authorization and data restrictions over those already defined in the model. For example, a request template created for the “Provision Resource” model could restrict which resources can be requested through the template.

Both transaction data and history data for requests is maintained, supporting audit and compliance requirements.

Authorization Service

Oracle Identity Manager being a security product requires a strong level of access control over what users can do within the application. Customers need to be able to define authorization policies that determine at runtime whether a particular action is allowed or not. These policies should be able to satisfy their complex authorization needs in a simple, easy to understand and maintainable manner.

This is controlled by the authorization service embedded within Oracle Identity Manager, which is built on a sophisticated authorization product called Oracle Entitlements Server OES). OES enables centralized management of entitlements and authorization policies to more granularly determine access to both application components and application business objects.

The Oracle Entitlements Server architecture is made up of two major components. The administration application acts as the policy administration point (PAP) and is used to manage policy, configuration, roles, and entitlements. The second major component is the use of one or more Security Modules (SMs). The Security Modules evaluate fine-grain access control polices at the policy decision point (PDP) and enforce it at the policy enforcement point (PEP). The Security Modules are also the integration point for access to external attributes that may be incorporated into the policy. The Security Module resides inside an application container.

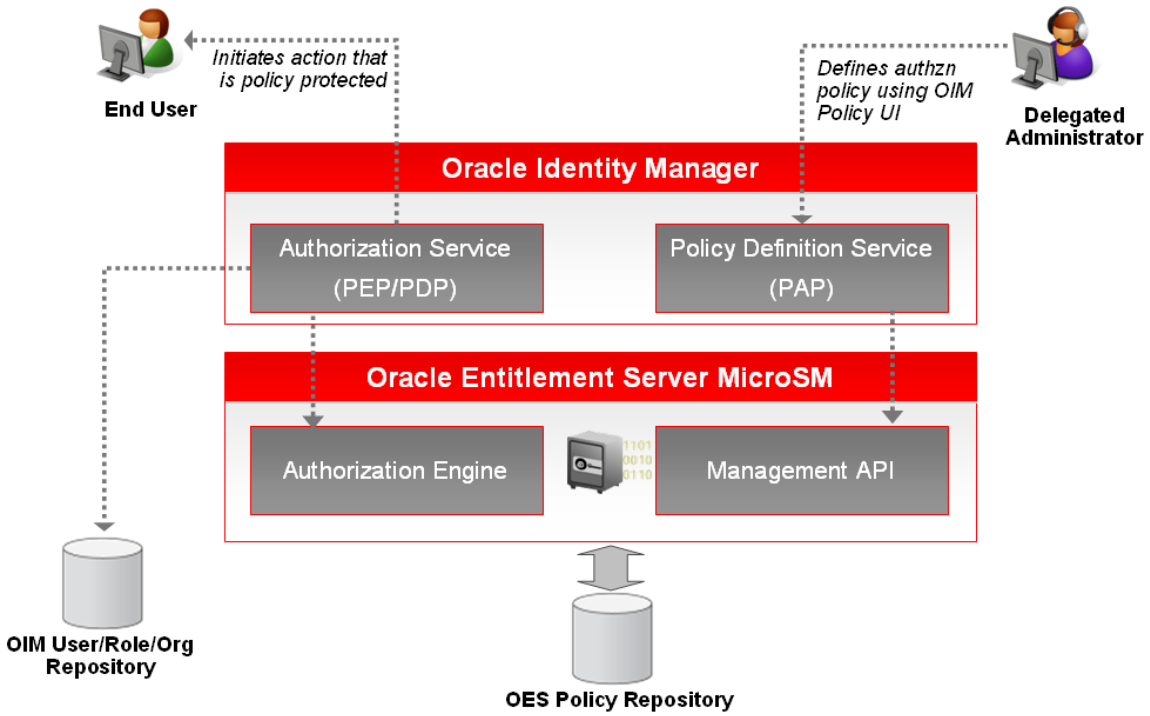

Figure 5: Oracle Entitlement Server based Authorization Service

Each time a privilege check is requested, several things happen. The application container asks the authorization service for an access decision each time a protected resource is accessed. The service then finds the policy or policies that apply to the resource and evaluates them. All information required to evaluate a policy is collected by the Security Modules at run time. If the policy references subject by role, all roles are evaluated and the access decision is made.

Oracle Identity Manager provides an abstraction service on top of OES that optimizes and simplifies the definition of policies within the OIM context. This service includes a policy definition UI that allows the definition of authorization policies that are feature specific and support fine-grained controls (including attribute and function level controls on entities like users and roles).

The general form of an authorization policy in Oracle Identity Manager can be represented as follows:

- Feature: The Oracle Identity Manager feature for which the policy is being defined (like “User Management”, “Scheduler Management”).
- Assignee: The role or roles to who privileges are being granted. Additionally, relationships like manager can also be supported as assignees for contextual authorization.
- Entitlements: The list of actions over which privileges are being granted (like “Modify User Profile”, “View User Detail”).
- Fine-Grained Attribute Controls: If the feature and entitlement supports attribute level controls (for instance, the list of attributes that can be modified under the “Modify User Profile” entitlement), then the lost of attributes can be specified.
- Data Security: The entities managed by the feature over which privileges are being granted (like “All users in Marketing Organization”, “All Users”).

Entity Manager Service

The Entity Manager Service defines the structure of the identity data within Oracle Identity Manager and provides basic capabilities to manage this data. More importantly, it then organizes this data into constructs that are easily consumed by the other identity services within Oracle Identity Manager to make policy, permission, workflow and provisioning decisions.

Identity data tends to be distributed among multiple repositories, each containing information about different types of entities –users, roles, etc. The nature and structure of this identity data tends to differ from one enterprise to the next. Each repository can be contacted using different mechanisms – such as native APIs, JDBC, LDAP, JCA or SPML. The Entity Manager abstracts all of this away from the other services within Oracle Identity Manager that rely on this data.

Because of the complexity of the identity data mentioned above, the Entity Manager Service relies on a provider model. A *Data Provider* is essentially a plug-in that lets OIM talk to the configured repository to utilize and modify the identity data in that repository. Each provider is metadata driven, with the metadata defining which repository to connect to and how, what data to manage, and what capabilities (read-only/read-write/write-only) it has. The Entity Manager Service provides a number of other key services on top of this, including the ability to support bulk operations for optimal performance. The diagram below summarizes the approach taken by the Entity Manager service.

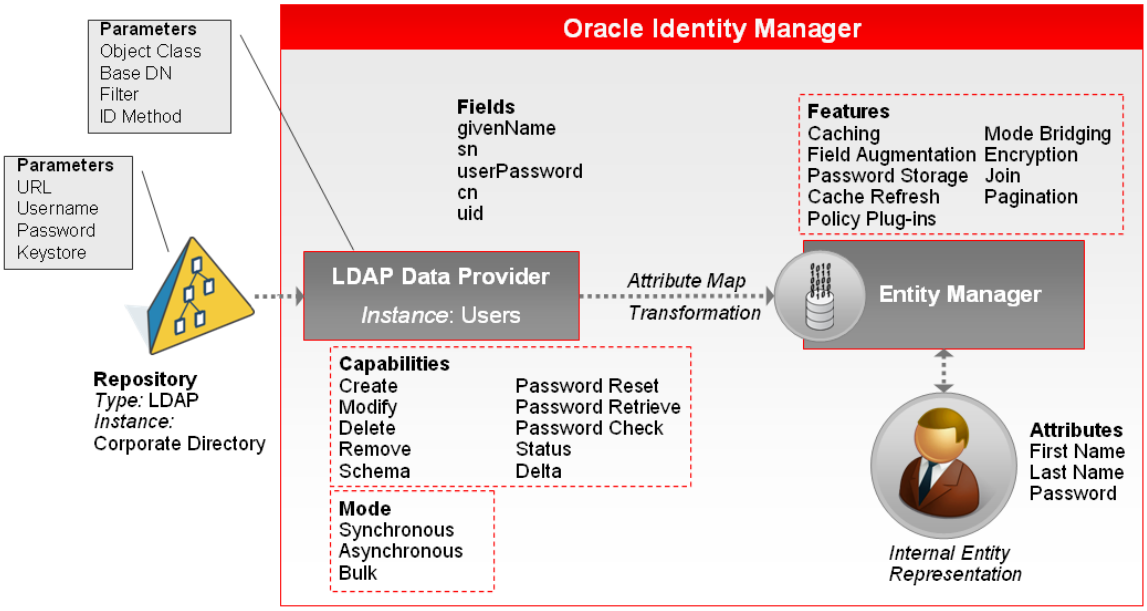

Figure 6: Entity Manager Architecture

Plug-in Framework

The Plug-in Framework allows customers to easily extend and customize the capabilities of the out-of-the-box Oracle Identity Manager features. OIM Features expose *plug-in points* - specific points in the business logic where extensibility can be provided. An interface definition accompanies each such point and is called the *plug-in interface*. Customers can create code that extends these plug-in interfaces and defines

customizations based on their business needs. These plug-ins are deployed and registered with OIM using the *Plug-in Manager*. OIM will then incorporate them into normal feature processing from that point onwards.

Feature developers or customers don't need to worry about where the custom implementations are stored and how they are loaded. The plug-in framework supports loading plug-ins from the classpath, from the file system and from the database.

SoD Engine Framework

An attempt to enforce good compliance practices is through the definition of Separation of Duties (SoD) policies. SoD is broadly defined as a way of preventing a user from acquiring a conflicting set of entitlements. This conflicting set is also referred to as a "toxic" combination. The classic example of a "toxic" combination is - a person should not have the ability to create and approve the same purchase order. Enterprises often have business application specific SoD engines that define and enforce SoD policies on the entitlements users have within those business applications. Examples of such SoD engines are OAACG and SAP GRC.

The SoD Engine Framework allows customers to integrate Oracle Identity Manager with their choice of SoD Engine to enable SoD checks at appropriate points in the request and provisioning process. As shown in the figure below, OIM can send a request for an SoD check to the SoD Engine through the SoD Invocation Library (SIL). SIL provides a common service interface to all supported SoD Engines, abstracting the business components within OIM that need SoD checks from having to worry about the correct data formats needed by the SoD Engine and also the interpretation of the results returned.

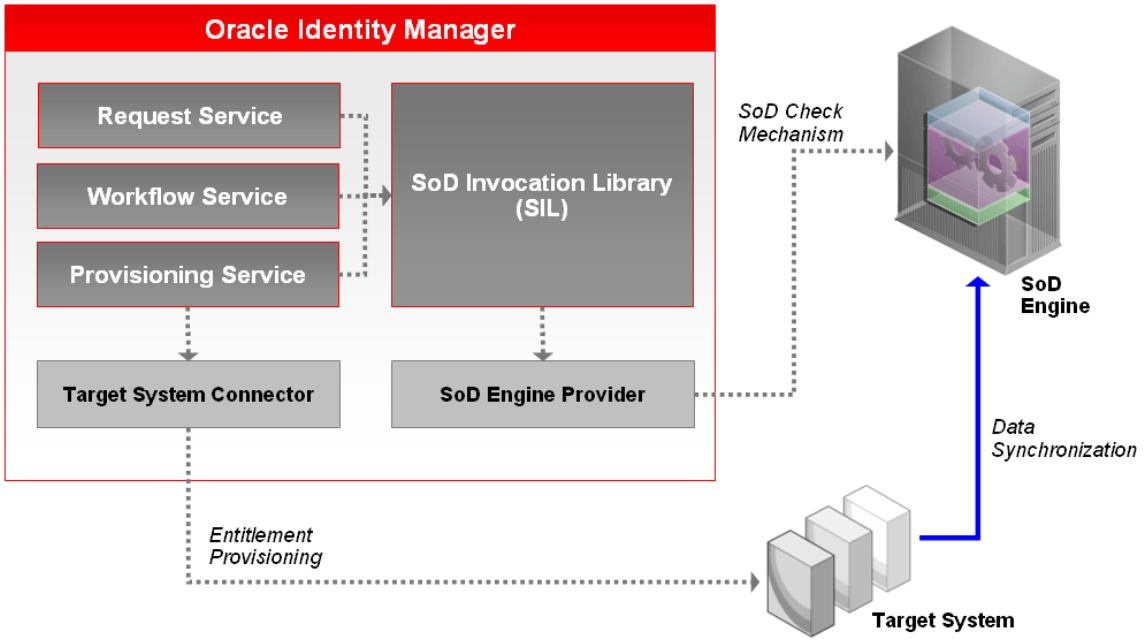

Figure 7: Oracle Identity Manager SoD Enablement

SoD checks can be executed at various times in the provisioning lifecycle – during an access request, during the approval workflow execution, or during the actual

provisioning execution. If a violation is detected, the request or resource is marked as "in violation", and the approver/administrator is given the responsibility of deciding whether to proceed or not. In the case of violations detected during request processing, different approval workflows can be invoked that allow for higher levels of approval.

Scheduler Service

Business systems frequently make use of scheduling systems, which are configured to run other programs at specified times. In many cases, scheduling systems run applications that generate reports, reformat data, or do audit work at night. Scheduling systems often run "batch jobs" (a.k.a. "scheduled jobs"), which perform routine work automatically at a prescribed time.

Scheduling systems are an integral part of any enterprise provisioning solution. Provisioning often involves tasks that need to be done in a time-based manner. Some examples are:

- Running a nightly job to reconcile all changes made directly on a managed application
- Do escalations of assigned tasks that have not been handled within a specified time period
- Execute requests that have been raised to be executed at a specific time

The Oracle Identity Manager platform includes a sophisticated scheduling product to provide the scheduling capabilities necessary for enterprise provisioning needs. This is built on a high performance scheduling product called Quartz. The Quartz Service is managed as part of the Oracle Identity Manager platform and not as an independent product. An overview of the Oracle Identity Manager Scheduler architecture is shown in Figure 8.

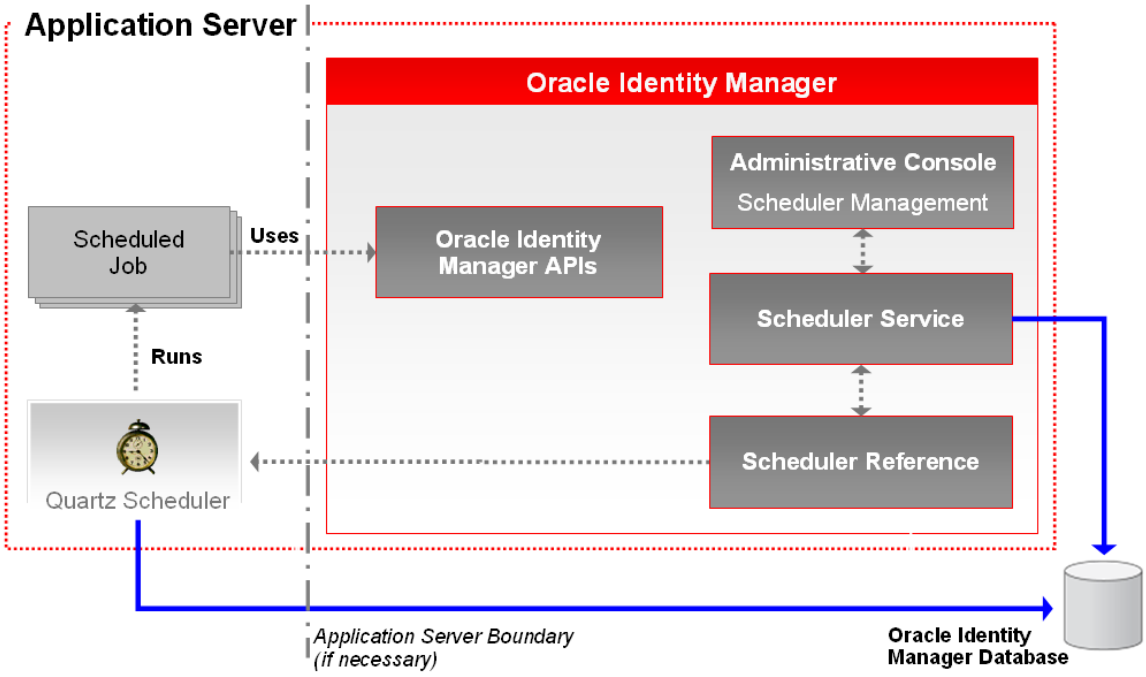

Figure 8: Oracle Identity Manager Scheduler Architecture

Key capabilities that Oracle Identity Manager harnesses from Quartz are:

- The ability to create simple/complex schedules for executing anywhere from ten to thousands of jobs
- The ability to run the scheduling service as a clustered service to provide the necessary high availability (fail-over and load balancing) capabilities
- The ability to persist the job definitions for management and fail-over support
- The ability to manage (create/modify/enable/disable/delete) jobs and individual job runs via an administrative UI
- The ability to execute a job in an ad-hoc fashion outside of regularly scheduled runs
- The ability to manage errors and failures in a graceful manner
- The ability to maintain history of job runs, including statistics and results of these runs
- The ability to manage the scheduler service itself

The Quartz Scheduler Service can run in the same application server as the Oracle Identity Manager application, or it can be run in a different application server. The jobs that are run in the Oracle Identity Manager Scheduler can interact with Oracle Identity Manager using the published Oracle Identity Manager APIs. They can also run any code to talk to any other systems that they may need to, especially in the case of reconciliation jobs.

The Data Tier

The Oracle Identity Manager application is heavily data and metadata-driven, with all of OIMs own data residing in the Oracle Identity Manager repository. It is this ability to be data and metadata driven that allows Oracle Identity Manager to be so flexible and adaptable from a functional perspective.

Oracle Identity Manager's data tier consists of the Oracle Identity Manager repository, which manages and stores Oracle Identity Manager data and metadata in an ANSI SQL 92-compliant relational database, and an optional LDAP Identity Store.

The OIM Database

The Oracle Identity Manager repository is the authoritative store for the "Who Has What, When, How and Why" data that is the core value of the identity administration and provisioning system. As such, the database is a critical component in the Oracle Identity Manager architecture.

The data stored in the Oracle Identity Manager database falls into the following broad categories:

- **Entity Data**: Users, organizations, roles, role memberships, resources, provisioned resources
- **Transactional Data**: Requests, Approval and Provisioning Workflow Instances, Human Tasks
- **Audit Data**: Request History, User Profile History

High Availability

The database system must provide a truly scalable and redundant data layer to avoid downtime and performance issues. Reliability, recoverability, timely error detection, and continuous operations are primary characteristics of a highly available solution.

The Oracle Identity Manager architecture relies heavily on the corresponding capabilities provided by the Database Management System that is used with the product. These capabilities must:

- Encompass redundancy across all components
- Provide protection and tolerance from computer failures, storage failures, human errors, data corruption, lost writes, system hangs or slowdown, and site disasters
- Recover from outages as quickly and transparently as possible
- Provide solutions to eliminate or reduce planned downtime
- Provide consistent high performance
- Be easy to deploy, manage, and scale
- Achieve SLA's at the lowest possible total cost of ownership

A broad range of high availability and business continuity solutions exists today. You can find out more about maximizing database availability using technologies like Oracle RAC (Real Application Clusters) and Oracle Data Guard at http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm.

Reporting

The rich set of data that gets collected and stored in the OIM repository can be surfaced through detailed reports that support management and compliance requirements. Oracle Identity Manager provides support for data reporting through the use of Oracle BI Publisher. Oracle BI Publisher is an enterprise reporting solution and provides a single reporting environment to author, manage, and deliver all of your reports and business documents. Utilizing a set of familiar desktop tools, such as Microsoft Word, Microsoft Excel, or Adobe Acrobat, you can create and maintain report layouts based on data from diverse sources, including Oracle Identity Management products.

Oracle Identity Manager ships with some standard Oracle BI Publisher report templates. However, you can customize each template to change its look and feel. Additionally, customers are able to create their own custom reports by leveraging the OIM database schema.

The Metadata Store

The logic underlying the Oracle Identity Manager application is metadata driven. The structural and behavioral aspects are declaratively described using metadata. To that end, the Oracle Identity Manager architecture relies on Oracle Metadata Services (MDS) to provide a unified store for metadata. This ensures consistent, reliable access to the metadata for OIM itself and for the other Fusion Middleware components that it is built on. The same metadata that is used during the design phase of an application is used at application runtime through the metadata services layer. This ensures consistency through the lifecycle of the application, illustrated in Figure 9 below. MDS also provides common administrative tooling for the metadata that can be used across various types of metadata stored in the common repository.

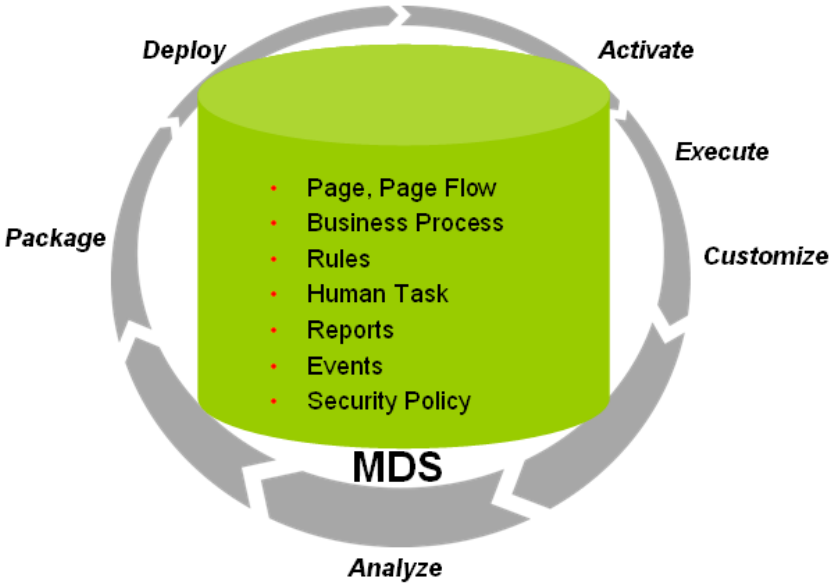

Figure 9: Meta Data Service as heart of Application Lifecycle

MDS addresses a number of design-time and runtime use cases within the Oracle Identity Manager infrastructure. Key features and architectural principles include:

- Simplified resource management through a single, unified repository for all artifacts used by various FMW components.
- Management of the metadata lifecycle for each artifact as it moves through the various stages of development, testing, staging and production.
- Sharing and reuse of metadata across components
- Categorization and reuse of artifacts, encouraging reuse and promoting consistency.
- Versioning capabilities, which form the basis for various features.
- An upgrade-safe, layered customization mechanism through which metadata and application logic can be tailored per usage of the metadata.
- Advanced caching and assembling techniques coupled with configurable tuning options to optimize performance.

Metadata accessed and managed via MDS can be in either a file-based repository or the database-based repository. In Oracle Identity Manager architecture, we have chosen to put it in the OIM database to take advantage of some of the advanced performance and availability features that this mode provides.

The Identity Store

New to Oracle Identity Manager 11g is the ability to integrate an LDAP-based identity store into the OIM architecture. Until 11g, the OIM identity store exists within the OIM database itself. OIM was able to integrate with LDAP as a provisioning target, or customers could build custom integration between OIM users and LDAP users. However, in 11g, customers have the option to connect and manage an LDAP-based identity store directly from Oracle Identity Manager. Using this feature, customers can use Oracle Identity Manager's advanced user management capabilities, including request based creation and management of identities, to manage the identities within their corporate identity store.

In this deployment architecture, user identity information will exist both in the OIM database (to support the relational functionality necessary for the product to function) and in the LDAP store, and all data will be kept in sync transparently (without the need for provisioning actions and setting up policies and rules). Identity operations initiated within OIM (such as user creation or modification) will be executed on both stores in a manner that maintains transactional integrity. Additionally, any changes in the LDAP store made outside of OIM will be pulled into OIM and made available as part of the identity context.

The LDAP Store integration is built on the Entity Manager service using an LDAP Provider that optionally leverages Oracle Virtual Directory. This brings a great deal of flexibility and scalability to the LDAP Store integration.

Security

Oracle Identity Manager is a highly secure enterprise application providing complete security of all sensitive data as it flows through the enterprise. Oracle Identity Manager leverages the Java EE security framework to provide a secure application environment.
It also has a highly flexible permission model to provide control over the various functions within the application, discussion of which is outside the scope of this document.

Channel Security

The Java EE security framework supports the encryption of all channels of communication within the framework using standard SSL. Oracle Identity Manager relies on this to ensure that all communication of provisioning data is secured against inspection and hacking.

Data Security

Oracle Identity Manager secures all data in the database using state-of-the-art encryption techniques. All sensitive data (like user passwords, system passwords) is automatically encrypted by the system before it is stored in the database.

Customers also have the ability to define which attributes are sensitive and need to be secured before storage.

Remote Manager Security

One of the key benefits of the Remote Manager component of Oracle Identity Manager is that it provides a mechanism for Oracle Identity Manager to communicate securely with targets that do not provide a security layer of their own in their APIs.

All Remote Managers in the Oracle Identity Manager deployment have certificates that they use to identify themselves to the Oracle Identity Manager server. These certificates are also used as the basis for setting up SSL encryption of the RMI channel between the Remote Manager and the Oracle Identity Manager server. To be trusted by the Oracle Identity Manager server, all Remote Manager certificates must be registered. This prevents spoofing to gain unauthorized access to data. In addition, only the Oracle Identity Manager server can initiate communication with a Remote Manager.

A Remote Manager can also be deployed to function in a mutual authentication mode. In this mode, in addition to the normal registration of the Remote Manager certificates with the Oracle Identity Manager server, the Oracle Identity Manager server certificate needs to be registered with the Remote Manager. Upon initiating a SSL connection, the Oracle Identity Manager server sends this certificate as a part of the handshake it performs with the Remote Manager. The Remote Manager rejects any communication if verification of the certificate fails. Customers can use this mode to secure their Remote Managers from unauthorized or erroneous access. The Remote Manager security architecture is shown in Figure 10.

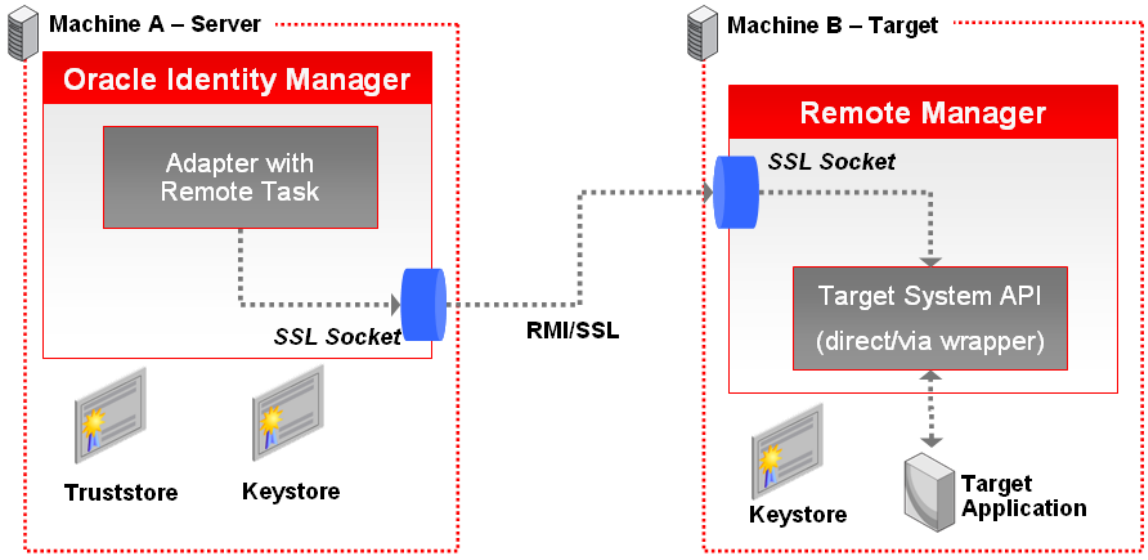

Figure 10: Remote Manager Security

Deployment Options

The Oracle Identity Manager platform leverages the flexibility of the J2EE framework along with its scalability features to provide a number of different deployment options to the customer, depending on their requirements. This section reviews some common deployment options.

Option 1 – Simple Deployment

The simplest deployment involves a single application server serving up the entire Oracle Identity Manager deployment, including the server, web and scheduler components. This is illustrated in Figure 11.

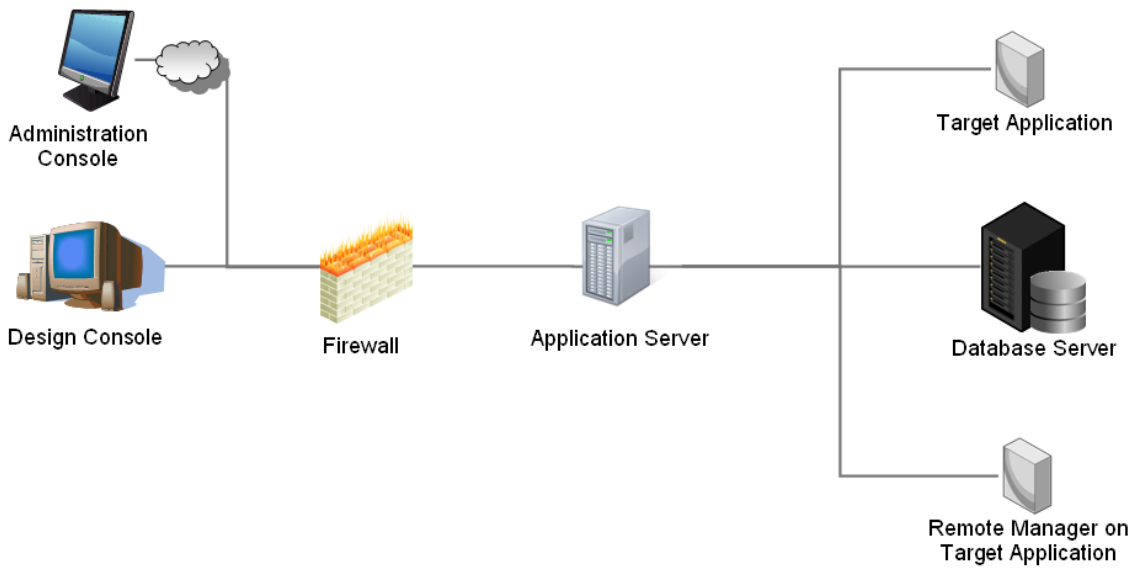

Figure 11: Simple Oracle Identity Manager Deployment

Option 2 – Clustered Deployment

This option clusters the application server in Option 1 to provide load balancing and fail over capabilities. This deployment is therefore able to support high availability requirements. The system can also be configured to work with firewalls. The database server can be configured in a number of different ways to support high availability at the data tier as well. A clustered Oracle Identity Manager deployment is shown in Figure 12.

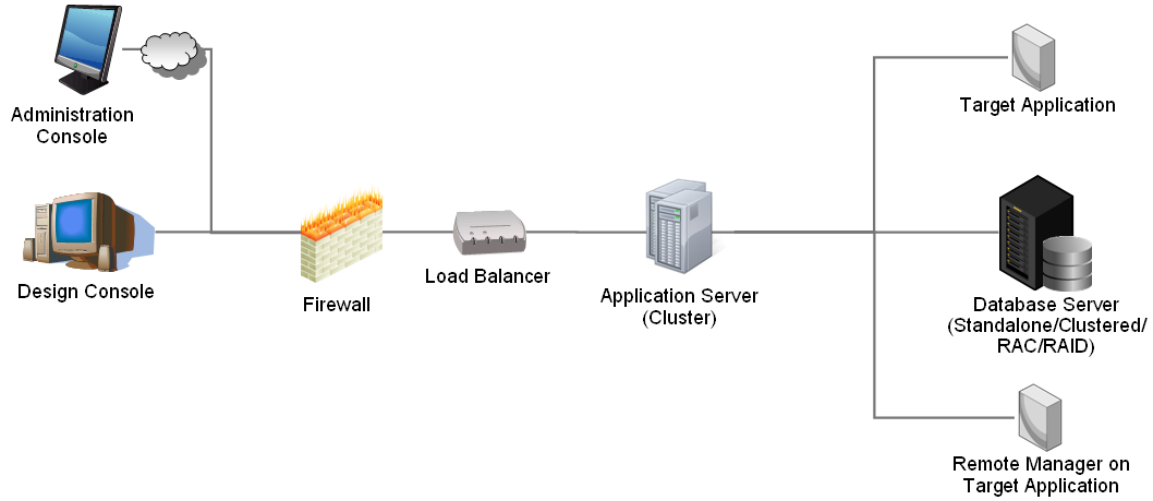

Figure 12: Clustered Oracle Identity Manager Deployment

Option 3 – Proxied Deployment

The proxied deployment option adds an extra enterprise element to the deployment in Option 2, allowing for the web interface to be served up to end-users via a Web Server (like IIS or Apache) that proxies the web page requests to the Oracle Identity Manager application component in the application server. This provides the following additional capabilities:

- Transparent support for web client fail-over using the application server plugins for the web servers
- Support for SSO-based authentication
- Static content and images could be off-loaded to the web server for better performance

As user load increases, the number of Web Servers can be increased independent of the application servers in order to scale horizontally. A proxied Oracle Identity Manager deployment is shown in Figure 13.

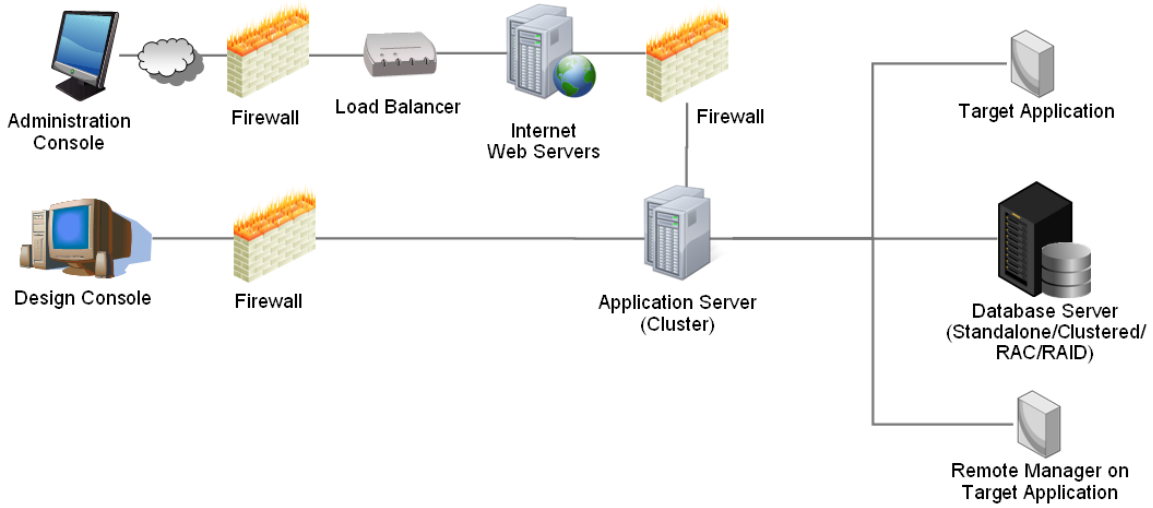

Figure 13: Proxied Oracle Identity Manager Deployment

Option 4 – Partitioned Deployment

Option 4 partitions the deployment configuration in Option 3, separating the Oracle Identity Manager server component into two logical components that handle different types of processing:

- The *Front-Office Application Server* provides all the Oracle Identity Manager application services necessary to support the Oracle Identity Manager Web Application processing requirements.
- The *Back-Office Application Server* runs the scheduler, thus doing all the heavy duty processing associated with scheduling jobs for reconciliation. It can also be configured to process the message queues for off-lining activities. Thus the Back-Office Application Server provides all the Oracle Identity Manager Application services necessary to support the processing-intensive tasks.

A partitioned Oracle Identity Manager deployment is shown in Figure 14. In this figure, each of these partitions is an independent Oracle Identity Manager deployment (individual clusters) that must be managed as a unit (from an adapter and other configuration perspective) that share the same configuration items like security keys, etc. Partitioning is managed by simple configuration of the deployment on each with regards to scheduling.

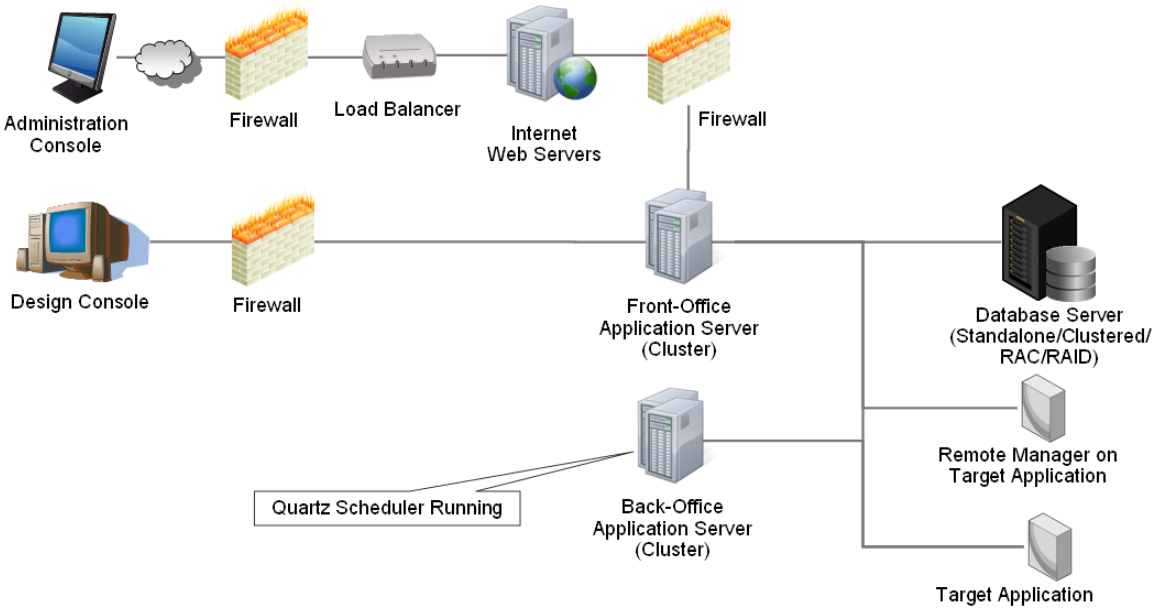

Figure 14: Partitioned Oracle Identity Manager Deployment

Remote Manager Deployment Options

The deployment of the Remote Managers in the Oracle Identity Manager solution depends on the requirements that the Remote Manager aims to achieve.

If the Remote Manager is needed to execute APIs that can only be executed on the physical machine hosting the target application, or to provide a secure channel to the target application, then the Remote Manager must be deployed on the same physical machine, like Remote Manager R1 in Figure 15.

If the Remote Manager is needed to execute APIs not directly on the same physical machine, but on a machine within the same domain/network, or on the same OS as the target machine (which is different from the OS of the Oracle Identity Manager application server), then it can be deployed on a proxy target machine, like Remote Manager R2 in Figure 15.

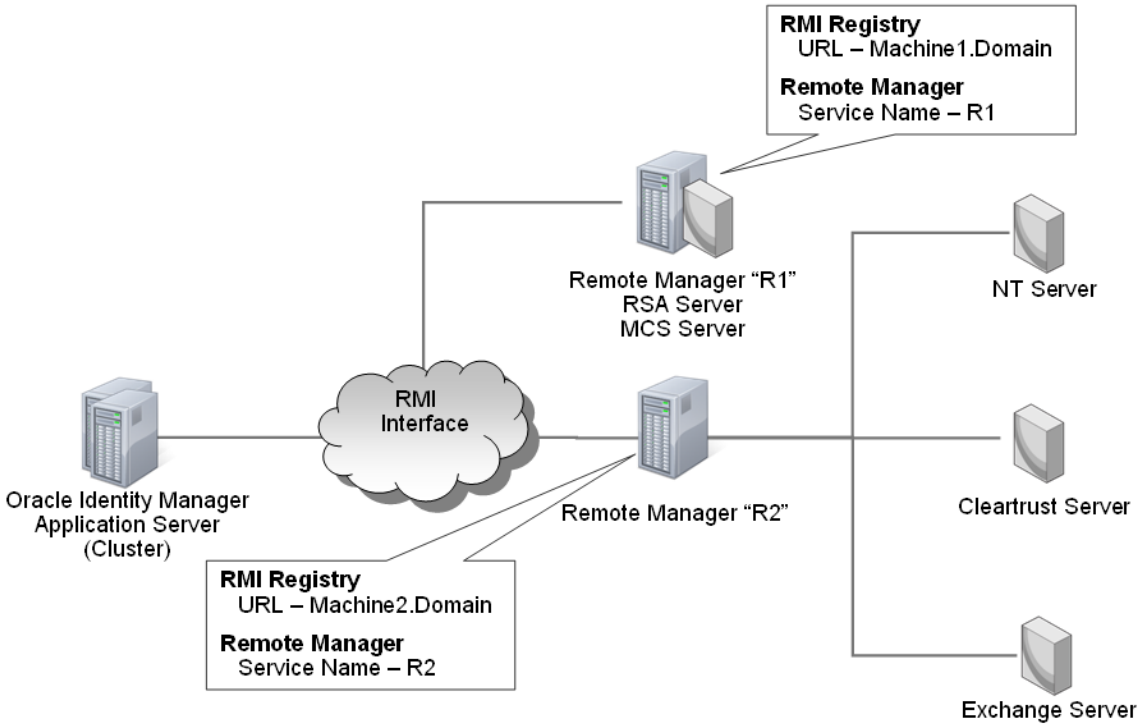

Figure 15: Remote Manager Deployment Options

Conclusion

Oracle Identity Manager provides a secure, scalable and flexible enterprise provisioning solution that can be tailored to a variety of needs. At the heart of these capabilities is the Oracle Identity Manager architecture, which reflects the latest best practices for Java EE based N-tier architectures. As a result, Oracle Identity Manager can be deployed in a number of different ways, from simple deployments on a single application server, to clustered, proxied deployments that provide transparent failover, vertical scalability and high performance.