Oracle Identity Analytics Architecture

An Oracle White Paper
July 2010

ORACLE

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 Identity Analytics (OIA), formerly Sun Role Manager (SRM) provides comprehensive enterprise role lifecycle management and identity compliance capabilities to streamline operations, enhance compliance, and reduce costs. Enterprise role lifecycle management includes the business processes and technologies governing the creation and maintenance of roles and their assignments to users. Identity Compliance capabilities include a simple and cost-effective solution for user access certification using Cert 360, role lifecycle management and governance as well as enterprise IT Audit Policy (SoD) enforcement.

This technical whitepaper describes the robust core technology that powers Oracle Identity Analytics. The paper provides an overview of the technologies that the architecture is based on, then describes how these technologies are leveraged to deliver a scalable, high-availability solution that manages roles and provides a sustainable cost-effective identity compliance capabilities across a typically heterogeneous environment.

Core Architecture

Oracle Identity Analytics is built using the Spring Framework. Spring provides a light-weight solution for building enterprise-ready applications, while still supporting the possibility of using declarative transaction management, remote access to your logic using web services, mailing facilities and various options in persisting your data to a database. Spring also provides an MVC framework, transparent ways of integrating Aspect Oriented Programming into your software and a well-structured exception hierarchy including automatic mapping from proprietary exception hierarchies. The architecture is able to leverage the most flexible and supported cross-platform J2EE services available: a combination of Java, XML, and object technologies. Such architecture makes Oracle Identity Analytics a scalable, fault-tolerant solution for the most ambitious global deployments in the industry.

Oracle Identity Analytics runs on leading J2EE compliant application server platforms, including Glassfish, Oracle WebLogic, and IBM WebSphere, to support JSP/Java Servlet and EJB execution, as well as to exploit the performance and scalability features inherent in these servers. Oracle Identity Analytics also supports application server clustering for increased performance and virtually automatic failover in mission-critical computing environments. The standards-based approach also allows Oracle Identity Analytics to leverage multiple enterprise databases like Oracle, MySQL and MS SQL Server for its data tier.

Oracle Identity Analytics architecture is designed to meet the following goals and objectives:

- Time to Market – rapidly deploy Oracle Identity Analytics services
- Performance – speedy response times and efficient navigation
- Portability – minimizing platform and external system dependencies
- Scalability – scale from low end to thousands of users
- Maintainability – easy to support and maintain
- Reliability – consistency of application and transactions

Oracle Identity Analytics Architecture

Oracle Identity Analytics architecture utilizes the Spring framework for the J2EE components. In addition it uses components and libraries that integrate seamlessly with the Spring Framework to provide the easy to use Ajax interface with the help of DWR and advanced functionalities like Scheduling which uses Quartz and workflow capabilities in Role Management using OS Workflow.

Overview

The Oracle Identity Analytics system consists of the OIA Server, which offers the Compliance Services and the Role Management Services, and the Web Presentation layer which is offered via Web Services API and the HTTP protocols in the Web Browser.

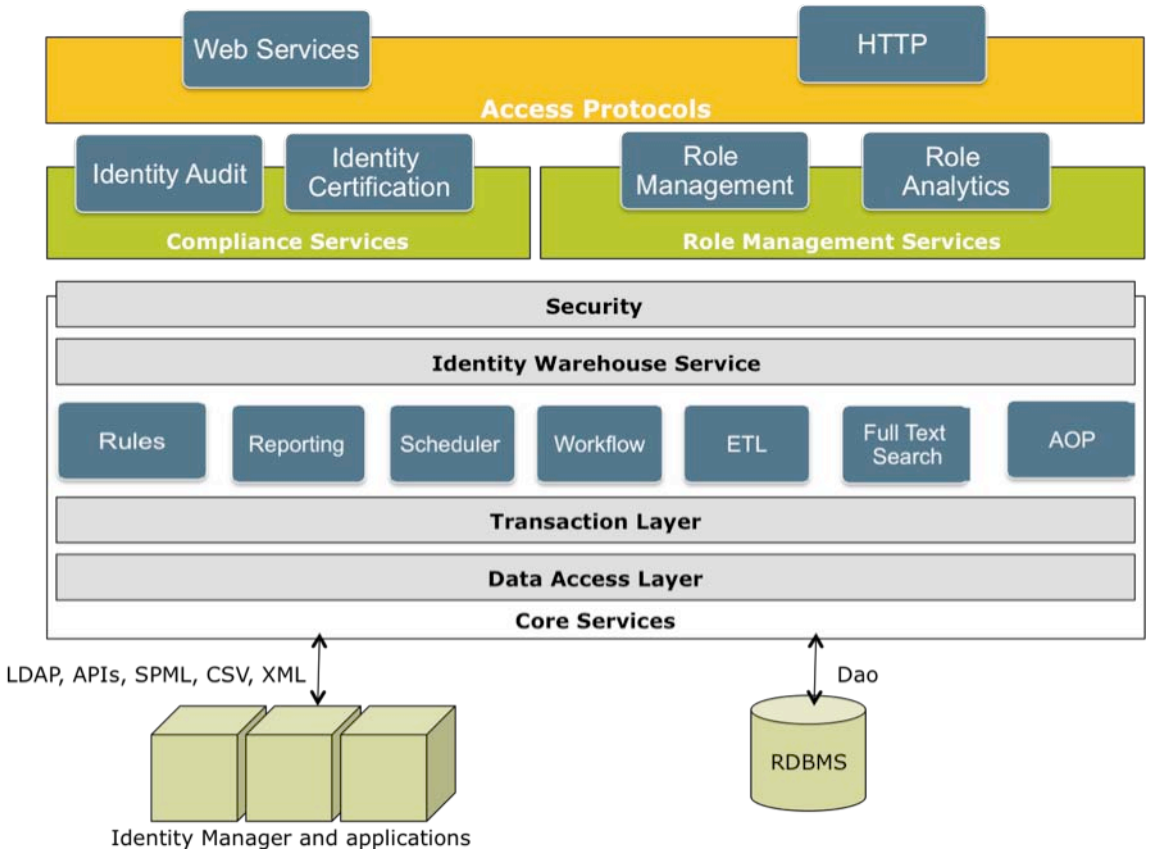

Figure 1. OIA Architecture

The tiers of the OIA server are deployed within a Java application server container. OIA uses and requires an RDBMS system for data persistency. The OIA server layer contains the OIA

Web components at the top to interact with the OIA Web interface. Below that layer there exists the Authentication and Authorization Layer. Below that is the Business Logic layer, followed by the Services layer and lastly followed by the Transaction and Persistence layer.

The OIA application is a web-based J2EE application deployed in an application server and is built using Java 1.5 and higher to make use of some of the Web 2.0 functionality.

OIA WEB UI

OIA Web UI is a full-blown thin client accessible via the web browsers. It makes use of the Spring Web MVC layer and provides AJAX support using the DWR component. It also exposes the core Identity Compliance and Role Management pieces via the web services.

AJAX Based

The OIA Web UI presentation layer utilizes Direct Web Remoting (DWR) to provide easy AJAX functionality and for performance improvements. The DWR layer interacts with the Java objects on the server layer and converts them into Java script objects, which are easier to render on the UI, and does not require any complex conversion between the objects. The Reverse Ajax allows Java Code on the server to find out what clients are viewing and send them JavaScript, generated either manually or using Java API.

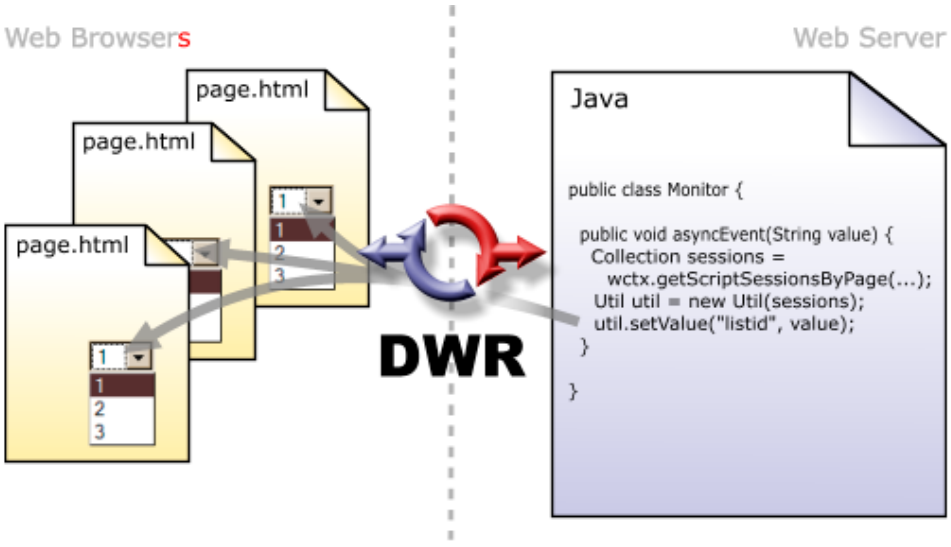

Figure 2. DWR

Spring Web MVC

Spring Web MVC provides the Model-View-Controller implementation, which provides a clean separation between domain model code and web forms and also provides other features like the validation. It also helps integrate with other components like the widgets in order to provide OIA with the Web 2.0 technologies available on the application UI with the help of accordions.

Configurability

OIA UI is built so that pieces can be customized without having to modify the server code or the application level code. Changes can be made on the Web UI layer and the information could be customized for deployments.

- Branding: Customizable icons, labels and images. These are customizable via CSS modifications.
- Localization: As the labels are customizable, they can even be localized based on the browser settings of the application.
- Business Logic: There is some business Logic built in certain modules on the UI that could be customized for the full deployment.

Web UI Security

The OIA UI is divided into menus, submenus and screens. The access to the user is secured using Roles within the system. The Roles define the access/view the users will have view to. The menu and screens are all security based and can be only be viewed by getting the appropriate roles. Apart from the menus and screens, some objects within the screens that get displayed to the user is also controlled by the Objects the user can have access to. All these are controlled by the OIA Roles defined within the application.

SSO

OIA can be configured to utilize any header based single sign-on application for authentication into the application. OIA can also be configured to have SSO from OIA into other SSO enabled application by making corresponding changes to configurations. OIA officially supports Oracle Access Manager enablement for Single Sign On services.

Web services

The OIA UI also provides a set of functions that can be utilized by external code to invoke certain functionalities within the application. These are achieved using the Web services that are being exposed via XFire. This is another component that integrates tightly with spring from the context layer and also to provide the services with the correct required set of authorization. OIA is officially certified with Oracle Web Services Manager (OWSM).

OIA Server

The OIA Server Architecture is divided into modules. Each of the functionality with OIA is treated individually and broken down into modules as shown in the diagram below.

Technologies

The OIA server uses and requires Java 1.5 and specifically uses Apache Lucene for Full text searching, Quartz for scheduling, Clover ETL for data manipulation, Jasper for reporting and OS Workflow for the workflow components of Role managements.

Modules

The OIA server is divided into various modules based on the functionality offered by the application. The main modules are described below

Identity Warehouse

Identity Warehouse is the core and forms the basis of the OIA application. This module corresponds to the data being populated and present in OIA for the functionalities to be provided. The Identity Warehouse could be populated multiple ways, including pulling in information from existing IDM solutions like Oracle Identity Manager (OIM) and other authoritative sources via the Web UI or various methods like API, CSV, XML, etc as listed in the above Figure 1.

Identity Certification

Identity Certification is one of the compliance modules present in OIA. This module is responsible for compliance certification of user information and roles within the product. This module provides different types of certification like User, Role, Resource Owner & Data Owner. This module is also responsible for tracking changes made after certification and remediating the changes with appropriate comments.

Identity Audit

Identity Audit is the second compliance module available in OIA to provide SOX compliance capabilities. This module is responsible for identifying the users with access' that are in conflict with each other, or with the users job responsibilities.

Role Management and Role Analytics

The Role Management and Role Analytics are the other modules of OIA. These modules form the core of the Role Management services that OIA offers. The Role Management module is responsible for the Role mining and life cycle management that includes assignment of roles based on certain rules. Role Analytics is the module that performs the Role consolidation to remove redundant roles.

Layers

The OIA server is divided into different abstraction layers with each layer performing different functions. Some of the layers are common for the modules previously mentioned while other layers have their corresponding modularity maintained in the layers as well.

Security

Security layer is the first layer on the OIA server. This layer is responsible for the authentication and authorization into the OIA application. OIA makes use of the Acegi Security. Acegi Security provides comprehensive authentication and authorization services for enterprise applications based on Spring.

Acegi uses a chain of (at least) three filters to enable web application security.

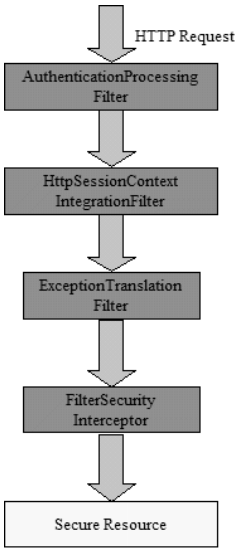

Figure 3. Acegi Filters

- The Authentication Processing Filter handles the Authentication Request Check ("logging into the application"). It uses the Authentication Manager to do its work.
- The Http Session Context Integration Filter maintains the Authentication object between various requests and passes it around to the Authentication Manager and the Access Decision Manager when needed
- The Exception Translation Filter performs the Existing Authentication Check, handles security exceptions and takes the appropriate action. This action can be either spawning the authentication dialog or returning the appropriate HTTP security error code. Exception Translation Filter depends on the next filter, Filter Security Interceptor, to do its work.
- Filter Security Interceptor manages the Restricted Access check, and the Authorization check. It knows which resources are secure and which roles have access to them. Filter Security Interceptor uses the Authentication Manager and Access Decision Manager to do its work.

Authentication

The Authentication part of security could be handled natively by Acegi by using the passwords that are stored encrypted in the users identity in the OIA repository.

Acegi also provides the ability to delegate the authentication of users externally to either LDAP or a SSO solution like Oracle Access Manager (OAM). Once authenticated and identifier is provided to OIA to allow the users to log in.

Authorization

Whether the authentication is done natively or externally, the authorization of the users access is managed internally within OIA with the help of Acegi Role Based Access and the Access Control lists. Some of the accesses for the users are also identified by the Spring Interceptors that is tightly integrated with Acegi using the "AfterMethodInvocation Interceptor" which removes objects from collections when user can't read them as mentioned earlier.

Business Logic Layer

This layer is the layer just below Security. This is the layer that manages the Business Logic for the various module. Depending on the modules this layer could also have common code. All the requests sent from the Web UI and the web services of OIA is received and processed by this layer.

Since the business logic layer is implemented in a modularized fashion it is easy to modify and extend the business functionality. This layer also depends on some of configuration that can be set using the OIA Web UI. Some of the business functionality is configurable for the modules and all that is managed by this layer.

Services Layer

The services layer is the layer that abstracts the Business Logic layer from the transaction and the data access layers. The services layer is managed using the "container level services" to offer the services to the Business layer. The services layer has the following "containers level" or modules for the various services.

Identity Warehouse Service

This is the service layer that provides access to the data stored in the Identity Warehouse of OIA. This Service layer makes use of a various other components to provide the functionality supported by OIA.

Rules Service

This service layer provides the Rules service across the OIA product. OIA has a custom rule engine in order to provide functionality for the Rule based Role Assignments. OIA uses the same rule engine for the Identity Audit and other Rules that are present within the application.

Reporting Service

This service layer provides the Reporting functionality of OIA. This layer makes use of the Jasper Reports in order to provide the ability to add customized reports run time into the application within a deployment. The Reporting service also offers the ability to re-design the reports using iReports and plug it into the Web UI.

Scheduler Service

As Business systems frequently need to be run, OIA offers advanced scheduling capabilities, which can be configured to run programs at specified times and repeatedly. This gives OIA the ability to run the jobs automatically without any manual interaction. OIA makes use of the J2EE scheduling product call Quartz. The Quartz Service is managed by OIA using the Web UI that reflects the configurations within Quartz in a user-friendly manner.

Workflow Service

The workflow service is the core component of the Role Management module of OIA. The workflow service provides the capability to define business process for approvals of Role definitions and Role grants. The OS Workflow component is used to provide the functionality and service for OIA. OIA makes use of the Web UI to configure the workflows and edit them to match the business process.

ETL Service

OIA makes use of Clover ETL in order to transform data and import it into the Identity Warehouse. The ETL service is embedded in OIA and can be invoked during the import and export functionalities of OIA. During the Import and Export Process, the ETL code is invoked to transform the data that is OIA readable before updating the system.

Full Text Search Service

Full Text Search is one of the most advanced search functionalities that OIA offers. This is offered by leveraging the Apache Lucene component in the service layer. This gives the ability to search for complex conditions within the Web UI search. It also helps in the Rule Engine to improve performance while searching data within a very large dataset.

AOP Service

OIA uses AOP interceptors for Security, audit and logging. AOP Interceptors are easily configurable using Spring and all Security including automatic creation of user access within OIA is handled by AOPs. OIA uses AOPs to also perform audit and logging into the warehouse for operations that the users perform.

Transaction and Data Access Layer

The lowest layer in the OIA server is the transaction and the data access layer. OIA uses the built in data access layer provided by Spring. OIA uses the iBatis integration of Spring to persist the data. It also has the ability to for connection pooling and Batch Updates. OIA uses a robust

infrastructure for declarative transaction management that supports local transactions as well as global transactions via JTA.

Oracle Identity Analytics is also able to use data sources to communicate with the database tier.

Backend System Integration Layer

Database

Oracle Identity Analytics data tier consists of the Oracle Identity Analytics repository, which manages and stores Oracle Identity Analytics metadata in an SQL relational database. Oracle Identity Analytics is heavily metadata driven, with all the data residing in the Oracle Identity Analytics repository. As such, the database is a critical component in the Oracle Identity Analytics architecture.

The Database system must provide a truly scalable and redundant data layer. The architecture relies heavily on the corresponding capabilities provided by the Enterprise Database Management System that is used with the product. These include, but are not limited to:

- Clustering
- Standby Databases
- Replication

Deployment Options

The Oracle Identity Analytics platform provides a number of different deployment options to the customer, depending on their performance and highly available requirements. This section reviews some common deployment options.

Option 1 – Simple Deployment

The Simple Deployment shown in Figure below uses a single application server and a single database server. This scenario allows for limited scalability by adding additional CPUs and memory to each server.

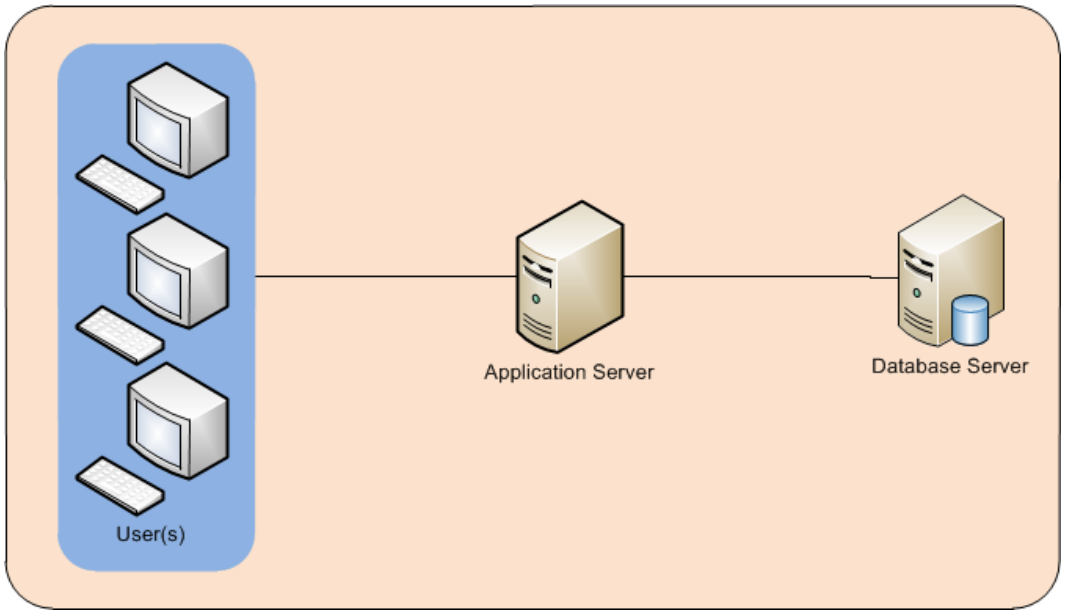

Figure 4. Simple Deployment

The above deployment scenario is appropriate for development and staging but not for production.

Option 2 – Cluster 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 database server can be configured in a number of different ways to support high availability at the data tier as well. A hardware or software based HTTP/HTTPS load balancer is used to distribute the user sessions to the application server.

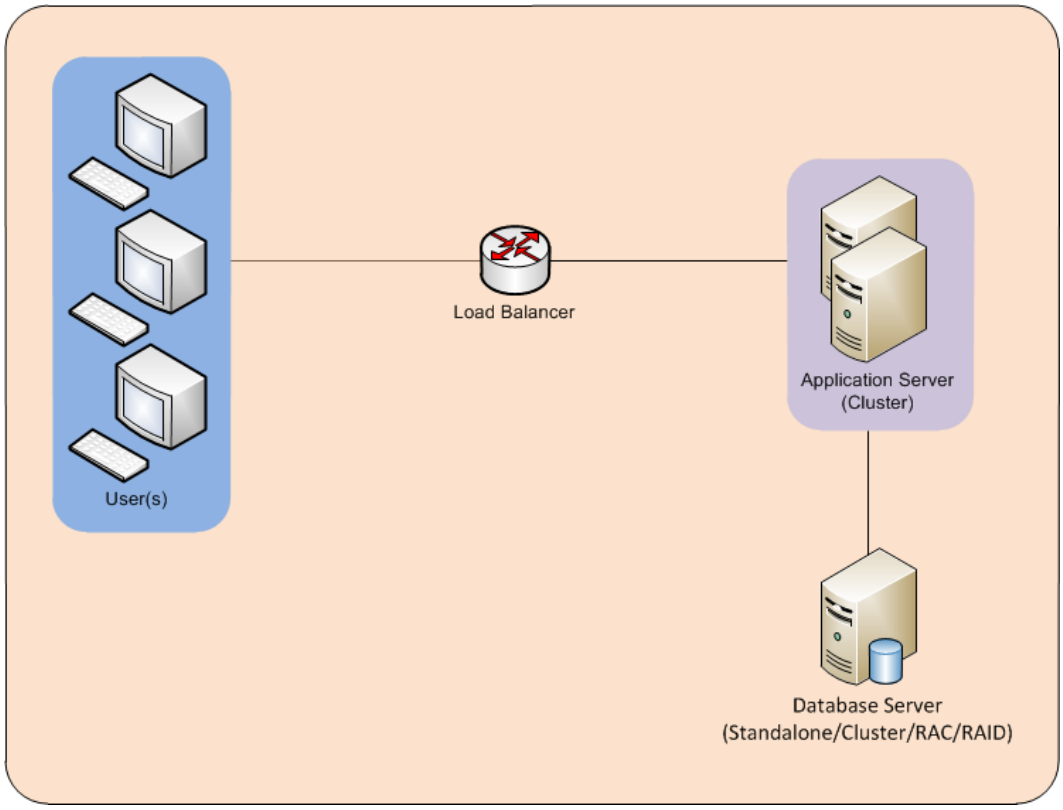

Figure 5. Cluster Deployment

The above deployment is ideal for Production. It is easy to scale in a "horizontal" manner when necessary by adding more application servers to the cluster as the load increases.

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 Analytics component in the application server. This provides the following additional capabilities:

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

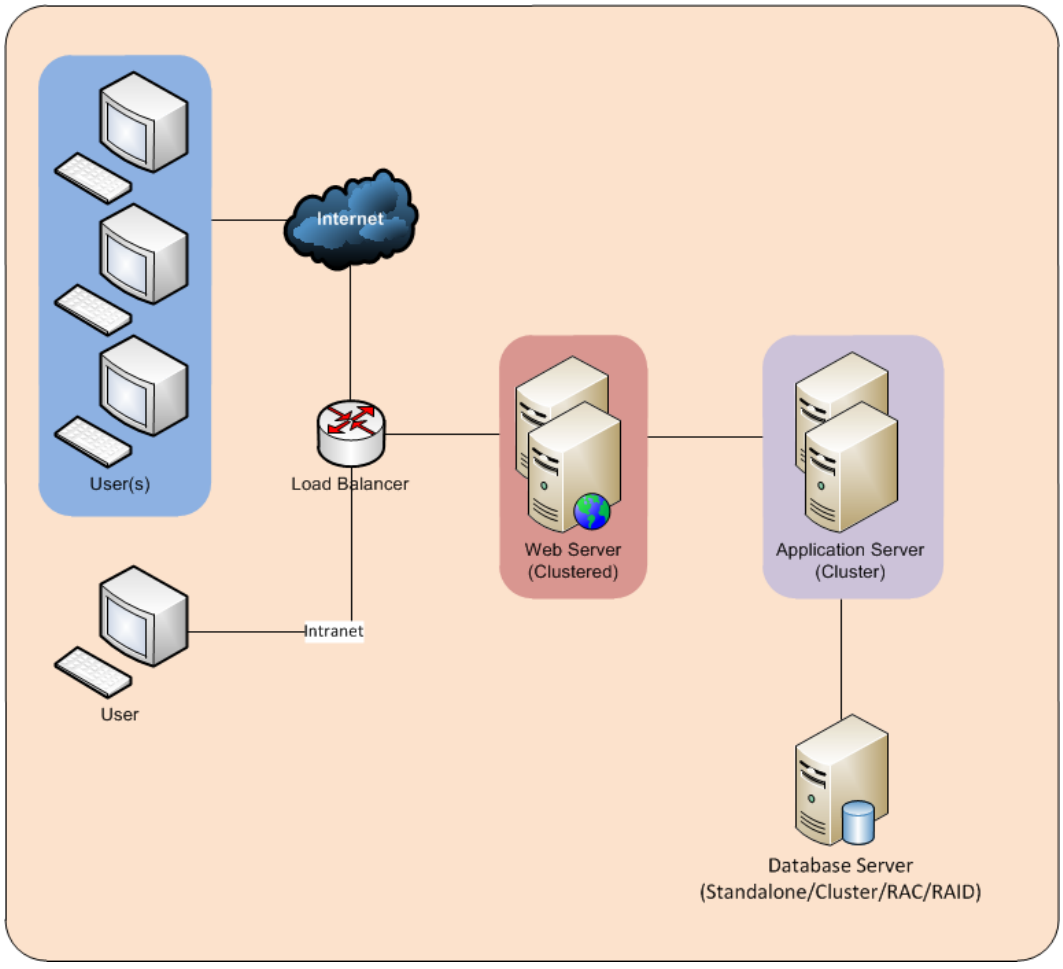

Figure 6. Proxied Deployment

As user load increases, the number of Web Servers can be increased independent of the application servers in order to scale horizontally. In addition the number of Application Servers could also be increased to scale horizontally if more processing is required by the application.

Option 4 – Oracle Integrated Deployment

This deployment is a typical Oracle Integrated Deployment in Production environment. As mentioned earlier, OIA can leverage OAM to provide SSO authentication to OIA. In addition it can also connect to OIM as shown below.

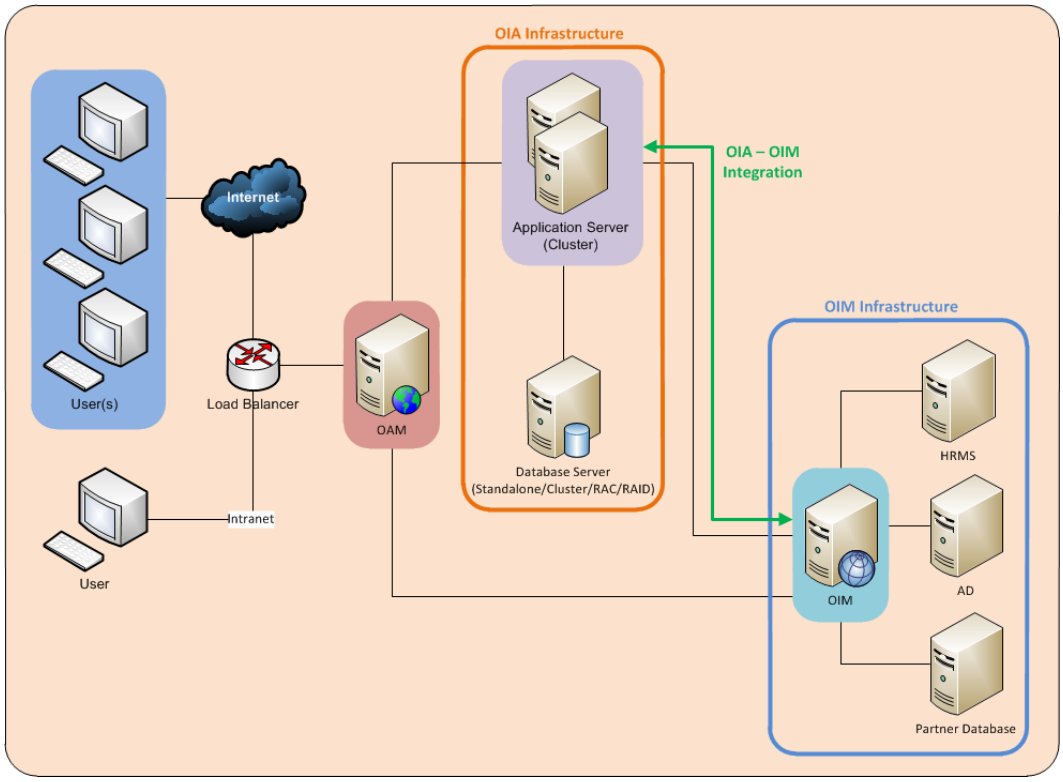

Figure 7. Oracle Integrated Deployment

In the above diagram, OIA acts as both a consumer and an enforcer. OIA consumes information from OIM to populate the Identity Warehouse data by connection to OIM using API based mechanism. OIA also has the ability to send in updates and role definitions to OIM where OIM becomes the consumer of roles defined in OIA.

Conclusion

Oracle Identity Analytics provides a scalable, cost-effective, secure and flexible role management solution that could be tailored as per requirements. Spring Framework forms the basis of OIA and using the J2EE model OIA can be deployed on any J2EE application server in single or clustered deployments with the ability to scale horizontally and vertically in the case of later deployment setup and scale vertically in the case of the former deployment setup. In addition it also provides the capability of load balancing and failover settings to have a highly available solution.

Oracle Identity Analytics Architecture
July 2010

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

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

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

0109