

Oracle® Fusion Middleware

Concepts Guide

11g Release 1 (11.1.1)

E10103-08

January 2011

Oracle Fusion Middleware Concepts Guide, 11g Release 1 (11.1.1)

E10103-08

Copyright © 2009, 2011, Oracle and/or its affiliates. All rights reserved.

Primary Author: Lisa Jamen

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	vii
Audience.....	vii
Documentation Accessibility	vii
Related Documents	viii
Conventions	viii
1 Introduction	
1.1 What is Middleware?	1-1
1.2 Functions of Middleware.....	1-2
1.3 Middleware Architecture Design	1-3
1.4 Service-Oriented Architecture	1-4
1.5 Oracle Fusion Middleware Solution	1-4
2 Key Concepts	
2.1 Key Oracle Fusion Middleware Concepts.....	2-1
2.2 Oracle Fusion Middleware Deployment Topology	2-2
2.3 Oracle Fusion Middleware Environment.....	2-3
2.4 What Is an Oracle WebLogic Server Domain?.....	2-4
2.4.1 What Is the Administration Server?.....	2-5
2.4.2 Managed Servers and Managed Server Clusters	2-5
2.4.3 What Is Node Manager?	2-6
2.5 What Is an Oracle Instance?	2-7
2.6 What Is a Middleware Home?	2-7
2.7 What Is a WebLogic Server Home?.....	2-7
2.8 What Is an Oracle Home and the Oracle Common Home?.....	2-7
2.9 What Is the Oracle Metadata Repository?.....	2-7
2.10 What is a Farm?.....	2-8
2.11 Oracle Fusion Middleware Directory Structure and Concepts.....	2-8
2.11.1 Middleware Home and WebLogic Home Directories.....	2-8
2.11.2 Oracle Home Directory	2-9
2.11.3 WebLogic Domain.....	2-9
2.11.4 Multiple Products with Multiple Oracle WebLogic Server Domains.....	2-10
2.11.5 Domain Extension.....	2-10
2.11.6 Oracle Instance and Oracle WebLogic Server Domain	2-11

3 Oracle Fusion Middleware Components

3.1	Oracle WebLogic Server.....	3-3
3.1.1	Key Features of Oracle WebLogic Server	3-4
3.1.1.1	High Availability	3-4
3.1.1.2	System Administration.....	3-5
3.1.1.3	Diagnostic Framework.....	3-5
3.1.1.4	Security Service.....	3-6
3.1.1.5	Client Options	3-6
3.1.1.5.1	Product Integration	3-6
3.1.1.5.2	Integration with Web Servers.....	3-6
3.1.2	Oracle WebLogic Server Core Components	3-7
3.1.2.1	Oracle WebLogic Server Web Services.....	3-7
3.1.2.2	Oracle Coherence.....	3-7
3.1.2.3	JMS Messaging Standard.....	3-8
3.1.2.4	Enterprise Grid Messaging	3-8
3.1.2.5	JRockit	3-8
3.1.2.6	Oracle Application Development Framework.....	3-8
3.1.2.7	Tuxedo.....	3-9
3.1.2.8	Oracle TopLink	3-9
3.2	Oracle Metadata Repository	3-9
3.3	Oracle Identity Management.....	3-10
3.3.1	Oracle Identity Manager.....	3-10
3.3.2	Oracle Internet Directory.....	3-11
3.3.3	Oracle Directory Integration Platform.....	3-11
3.3.4	Oracle Identity Federation.....	3-12
3.3.5	Oracle Virtual Directory	3-12
3.3.6	Oracle Platform Security Services	3-13
3.3.7	Oracle Role Manager.....	3-13
3.3.8	Oracle Entitlements Server	3-14
3.3.9	Oracle Directory Services Manager.....	3-14
3.3.10	Oracle Access Manager.....	3-14
3.3.11	Oracle Adaptive Access Manager	3-15
3.3.12	Oracle Identity Navigator.....	3-15
3.3.13	Oracle Authorization Policy Manager	3-15
3.4	Oracle WebCenter.....	3-16
3.4.1	Oracle Composer	3-16
3.4.2	Oracle WebCenter Framework	3-17
3.4.3	Oracle WebCenter Services	3-17
3.4.4	Oracle WebCenter Spaces.....	3-18
3.5	Oracle Business Intelligence	3-19
3.5.1	Oracle Business Intelligence Enterprise Edition (EE).....	3-19
3.5.2	Oracle Business Intelligence Publisher.....	3-19
3.5.3	Oracle Real-Time Decisions.....	3-19
3.6	Oracle Business Intelligence Discoverer.....	3-20
3.7	Oracle Portal	3-20
3.8	Oracle Reports	3-20
3.9	Oracle Forms Services	3-21

3.10	Oracle SOA Suite.....	3-22
3.10.1	Adapters.....	3-22
3.10.2	Oracle Service Bus	3-23
3.10.3	Oracle Complex Event Processing	3-23
3.10.4	Oracle Business Rules	3-25
3.10.5	Oracle Business Activity Monitoring.....	3-25
3.10.6	Oracle B2B.....	3-25
3.10.7	Oracle BPEL Process Manager.....	3-26
3.10.8	Oracle Service Registry	3-26
3.10.9	Oracle User Messaging Service.....	3-27
3.10.10	Human Workflow.....	3-27
3.10.11	Oracle Mediator	3-28
3.11	Development Tools.....	3-28
3.11.1	Oracle JDeveloper	3-28
3.11.2	Oracle Application Development Framework.....	3-29
3.11.2.1	ADF Model Layer	3-30
3.11.2.2	ADF Business Components.....	3-30
3.11.2.3	ADF Controller	3-31
3.11.2.4	ADF Faces Rich Client	3-31
3.11.2.5	Oracle ADF Security.....	3-32
3.11.2.6	Oracle ADF Support for Other View Technologies.....	3-32
3.11.2.7	Oracle ADF Swing	3-32
3.11.2.8	Oracle ADF Mobile.....	3-32
3.11.2.9	Oracle ADF Desktop Integration.....	3-33
3.11.3	Oracle TopLink	3-33
3.11.4	Oracle Enterprise Pack for Eclipse	3-34
3.11.5	Oracle Help.....	3-35
3.12	Oracle HTTP Server.....	3-35
3.13	Oracle Web Cache.....	3-36
3.13.1	Reverse Proxying	3-36
3.13.2	Request Filtering	3-36
3.14	Oracle Web Services	3-37
3.15	Oracle Delegated Administration Services	3-37
3.16	Oracle Enterprise Content Management Suite	3-37
3.16.1	Oracle Universal Content Management (UCM).....	3-38
3.16.2	Oracle Universal Records Management (URM).....	3-38
3.16.3	Oracle Inbound Refinery (IBR)	3-38
3.16.4	Oracle Information Rights Management (IRM)	3-38
3.16.5	Oracle Imaging and Process Management (I/PM)	3-39
3.17	Oracle Data Integrator.....	3-39
3.17.1	Oracle Data Quality Products.....	3-39

4 Oracle Fusion Middleware Administration and Operations

4.1	Enterprise Deployment.....	4-1
4.2	Management	4-2
4.2.1	Fusion Middleware Control.....	4-2
4.2.2	Oracle WebLogic Server Administration Console.....	4-2

4.2.3	Oracle WebLogic Scripting Tool.....	4-3
4.2.4	Oracle Process Manager and Notification Server	4-3
4.2.5	Oracle Enterprise Manager Grid Control.....	4-4
4.3	Scalability	4-4
4.4	High Availability	4-5
4.4.1	High Availability Problems.....	4-5
4.4.2	High Availability Solutions.....	4-6
4.4.3	Disaster Recovery	4-8
4.4.3.1	Components Protected by High Availability Solutions.....	4-8
4.5	Load Balancing.....	4-9
4.6	Diagnostic Data	4-9
4.7	Cloning	4-9
4.8	Oracle RAC	4-9
4.9	Backup and Recovery.....	4-10

Index

Preface

This preface includes the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This guide is intended for anyone interested in an overview of the components and features that comprise Oracle Fusion Middleware 11g.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit

<http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

Related Documents

For more information, see these Oracle resources:

- *Oracle Fusion Middleware Administrator's Guide*
- Oracle Fusion Middleware Documentation Library 11g Release 1 (11.1.1)

Conventions

The following text conventions are used in this guide:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introduction

This chapter describes Oracle Fusion Middleware. It includes the following sections:

- [What is Middleware?](#)
- [Functions of Middleware](#)
- [Middleware Architecture Design](#)
- [Service-Oriented Architecture](#)
- [Oracle Fusion Middleware Solution](#)

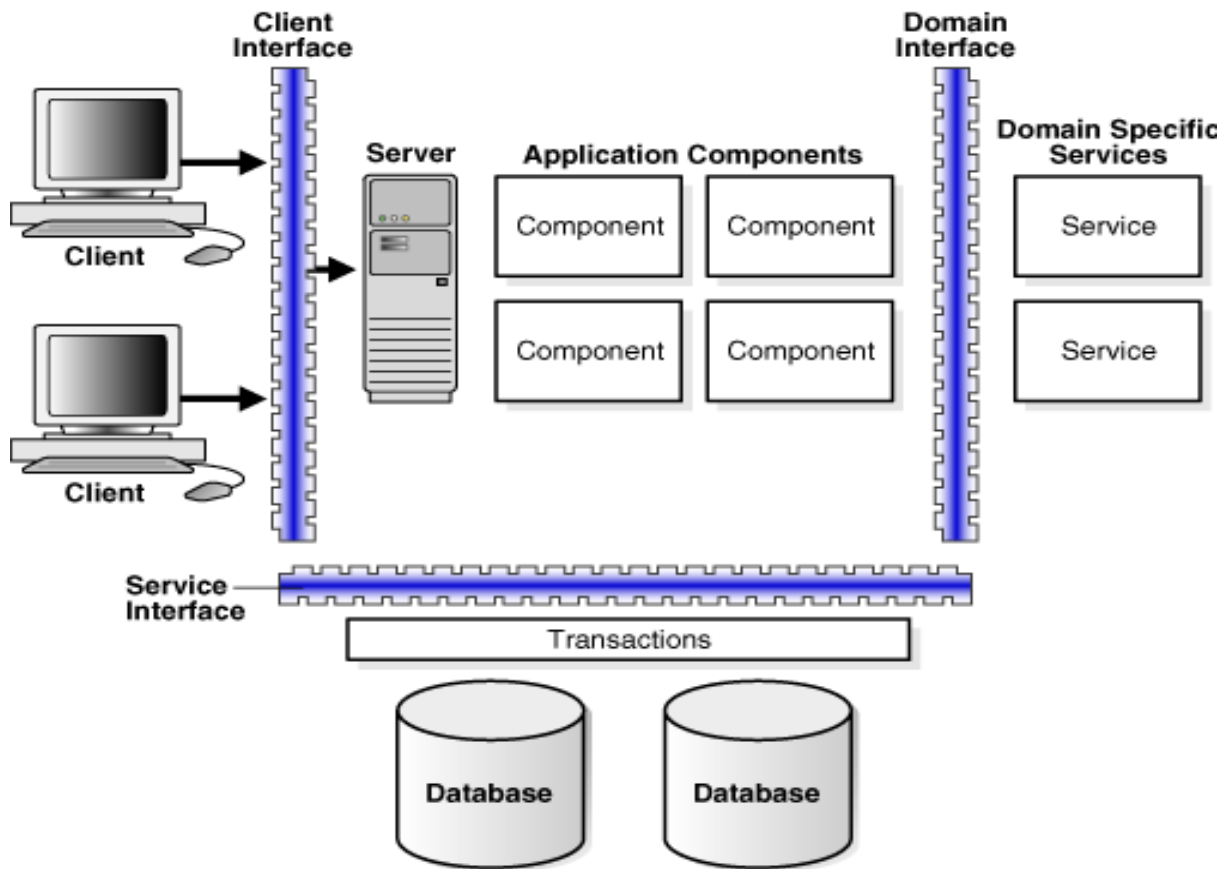
1.1 What is Middleware?

Middleware is the software that connects software components or enterprise applications. Middleware is the software layer that lies between the operating system and the applications on each side of a distributed computer network ([Figure 1-1](#)). Typically, it supports complex, distributed business software applications.

Middleware is the infrastructure which facilitates creation of business applications, and provides core services like concurrency, transactions, threading, messaging, and the SCA framework for service-oriented architecture (SOA) applications. It also provides security and enables high availability functionality to your enterprise.

Middleware includes Web servers, application servers, content management systems, and similar tools that support application development and delivery. It is especially integral to information technology based on Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web services, SOA, Web 2.0 infrastructure, and Lightweight Directory Access Protocol (LDAP)m etc.

Figure 1-1 Middleware Architecture



Due to continued growth and use of network-based applications by businesses, middleware technologies are increasingly important. Companies and organizations are now building enterprisewide information systems by integrating previously independent applications with new software developments. The integration process may involve legacy applications which may be used only with, or through a nonmodifiable interface. In some cases, rewriting the code for a legacy application may be cost-prohibitive.

Increasingly, information systems are composed of a collection of various specialized hardware devices interconnected by a network. Each device performs a function that involves receipt of real time data and remote interaction with other devices of the system. Some examples include computer networks, telecommunication systems, uninterrupted power supply units, and decentralized manufacturing units.

Interaction with the information system may span a wide range of performance. You can interact with Internet applications through a variety of devices, whose characteristics and performance figures span an increasingly wide range. Between a high performance personal computer, a smart telephone, and a personal digital assistant, the variations in bandwidth, local processing power, screen capacity, and the ability to display color pictures, are extremely large.

1.2 Functions of Middleware

Applications use intermediate software that resides on top of the operating systems and communication protocols to perform the following functions:

- Hide the distributed nature of the application. An application represents a collection of interconnected parts that are operational and running in distributed locations, out of view.
- Hide the heterogeneity of the enterprise. This includes the hardware components used, computer operating systems, and communication protocols.
- Provide uniform, standard, high-level interfaces to the application developers and integrators, so that applications can be easily composed, reused, ported, and made to interoperate.
- Supply a set of common services to perform various general purpose functions to avoid duplicating efforts, and to facilitate collaboration between applications.

Middleware makes application development easier, by providing common programming abstractions, by masking application heterogeneity and the distribution of the underlying hardware and operating systems, and by hiding low-level programming details.

1.3 Middleware Architecture Design

The function of middleware is to mediate interaction between the parts of an application, or between applications. Therefore, considerations for architectural structure play a central role in middleware design. The architectural design encompasses the organization, overall structure, and communication patterns, both for applications and for the middleware itself.

Besides architectural aspects, the main problems of middleware design pertain to various aspects of distributed systems. Any middleware system relies on a communication layer that allows its different pieces to interoperate. In addition, communication is a function provided by middleware itself to applications, in which the communicating entities may take on different roles such as client server or peer-to-peer. Middleware allows different interaction modes (synchronous invocations, asynchronous message passing, coordination through shared objects) embodied in different patterns.

Therefore, middleware system design faces several challenges:

- Middleware systems rely on interception and indirection mechanisms, which induce performance penalties. Adaptable middleware introduces additional indirections, which make the situation even worse.
- As applications become more and more interconnected and interdependent, the number of objects, users, and devices tends to increase. This poses the problem of the scalability of the communication and object management algorithms, and increases the complexity of administration. The availability, reliability, concurrency, security, and performance of applications may also be an issue.
- Widespread computing is a vision of the near future, in which an increasing number of devices embedded in various physical objects participate in a global information network. Mobility and dynamic reconfiguration will be dominant features, requiring permanent adaptation of the applications.
- Managing large applications that are heterogeneous, widely distributed, and in permanent evolution raises many questions, such as consistent observation, security, trade-offs between autonomy and interdependence for the different subsystems, and definition and implementation of resource management policies.

1.4 Service-Oriented Architecture

Service-Oriented Architecture (SOA) is an architectural style whose goal is to achieve loose coupling among diverse interacting software applications, enabling organizations to take advantage of existing investments in applications and systems. SOA facilitates the development of modular business services that can be easily integrated and reused, thus creating a flexible and adaptable infrastructure. Using a SOA approach, an organization can focus more resources and budget on innovation and on delivering new business services. Systems that can successfully use SOA can minimize the disruption of planned or unplanned outages in an enterprise.

Some of the advantages of using SOA are:

- **Reduction in development time and cost:** SOA services are easily reused and can be rapidly assembled into new, composite applications.
- **Lower maintenance cost:** Reusable services reduce the number and internal complexity of enterprise services.
- **High-quality services:** Increased service reuse creates high-quality services through multiple testing cycles from different service consumers.
- **Lower integration costs:** Standardized services know how to work together, enabling disparate applications to quickly and easily connect.
- **Reduce risk:** Fewer, reusable services provide greater control over corporate and IT governance policies, and reduce the overall compliance risk to an enterprise.

How does SOA achieve loose coupling among interacting software agents? It does so by employing two architectural constraints:

- A small set of simple and widespread interfaces to all participating software agents. Only generic semantics are encoded at the interfaces. The interfaces are universally available for all providers and consumers.
- Descriptive messages constrained by an extensible schema delivered through the interfaces. No, or only minimal, system activity is prescribed by messages. A schema limits the vocabulary and structure of messages. An extensible schema allows new versions of services to be introduced without breaking existing services.

1.5 Oracle Fusion Middleware Solution

Oracle Fusion Middleware offers solutions to and support for complex, distributed business software applications. It includes Web servers, application servers, content management systems, and similar tools that support application development and delivery.

Oracle Fusion Middleware is a collection of standards-based software products that includes a range of tools and services: from a Java Enterprise Edition 5 (Java EE) compliant environment, and developer tools, to integration services, business intelligence, collaboration, and content management. Oracle Fusion Middleware offers complete support for development, deployment, and management.

[Figure 1–2](#) provides an overview of the Oracle Fusion Middleware solution.

Figure 1–2 Oracle Fusion Middleware Solution Overview



Oracle Fusion Middleware offers the following solutions through its middleware design:

- Development Tools:** A unified SOA development tool and framework. An integrated, but modular, set of development tools to build complete applications, rather than using lots of specialized tools. The design tool includes a single design environment for user interface, business logic, service composition, business process or workflow, business rules, and business intelligence. The design tool enables simplified design and debugging, and to improve productivity. Includes Oracle JDeveloper, Oracle TopLink, Oracle Application Development Framework, and Oracle Eclipse.
- User Interaction:** A single, end-user environment that enable access to enterprise applications, business processes, business intelligence and to share information with each other. This end user environment is multi-channel allowing it to be accessed from a variety of different clients (mobile clients, desktop clients, Voice-over IP (VOIP) clients). Includes Oracle WebCenter Framework and Oracle Portal.
- Business Intelligence:** A suite of business intelligence tools from extract, transform, and load to integrate data into warehouses; query, analysis, and reporting tools for decision support; scorecards to compare how the business is doing against key performance indicators; and alerting to drive notifications to users about problems in the business software. Includes Oracle Business Intelligence Discoverer, Oracle Reports, Oracle Forms Services, and Oracle Portal.

- **Content Management:** A repository within which to manage documents, digital assets, scanned images and other forms of content; to integrate this content with a company's enterprise applications, Web sites, and business processes.
- **SOA:** A means of using existing investments in applications and systems most efficiently to be able to focus more resources and budget on innovation and on delivering new business services. Includes Oracle Service Bus, Oracle Complex Event Processing, Oracle Business Rules, Oracle Business Activity Monitoring, Oracle B2B, Oracle BPEL Process Manager, Oracle Service Registry, Oracle User Messaging Service, Human Workflow, Oracle Mediator.
- **Application Server:** A standards-based Java EE application server to run the enterprise applications and provide the Web services infrastructure for interoperability.
- **Integration and Business Process Management (BPM):** A standards-based service bus to connect applications with each other and with legacy systems using messaging; a BPM or workflow engine to connect the application into a business process or workflow; and business activity monitoring and optimize business processes in real time.
- **Security and Identity Management:** Lower cost of security administration across multiple applications and systems in an enterprise, by centralizing how users are created and provisioned, their identities, and roles and by enabling them to have single sign-on access. Includes Oracle Internet Directory, Oracle Virtual Directory, Oracle Directory Integration Platform, Oracle Identity Federation.
- **Enterprise Management:** Lower cost of operations and administration by running on a grid architecture with grouping, backup, and other high availability technologies, and integrating with Oracle Enterprise Manager for systems management. Includes Fusion Middleware Control, Oracle WebLogic Server Administration Console, Oracle WebLogic Scripting Tool, Oracle Process Manager and Notification Server, Oracle Enterprise Manager 10g Grid Control.

Key Concepts

This chapter describes Oracle Fusion Middleware key concepts. It includes the following sections:

- [Key Oracle Fusion Middleware Concepts](#)
- [Oracle Fusion Middleware Deployment Topology](#)
- [Oracle Fusion Middleware Environment](#)
- [What Is an Oracle WebLogic Server Domain?](#)
- [What Is an Oracle Instance?](#)
- [What Is a Middleware Home?](#)
- [What Is a WebLogic Server Home?](#)
- [What Is an Oracle Home and the Oracle Common Home?](#)
- [What Is the Oracle Metadata Repository?](#)
- [What is a Farm?](#)
- [Oracle Fusion Middleware Directory Structure and Concepts](#)

2.1 Key Oracle Fusion Middleware Concepts

Oracle Fusion Middleware provides two types of components:

- A **Java component**, which is an Oracle Fusion Middleware component that is deployed as one or more Java EE applications and a set of resources. Java components are deployed to an Oracle WebLogic Server domain as part of a domain template. Examples of Java components are the Oracle SOA Suite and Oracle WebCenter components.
- A **system component**, which is a manageable process that is not deployed as a Java application. Instead, a system component is managed by Oracle Process Manager and Notification (OPMN). The system components are:
 - Oracle HTTP Server
 - Oracle Web Cache
 - Oracle Internet Directory
 - Oracle Virtual Directory
 - Oracle Forms Services
 - Oracle Reports

- Oracle Business Intelligence Discoverer
- Oracle Business Intelligence

A Java component and a system component are peers.

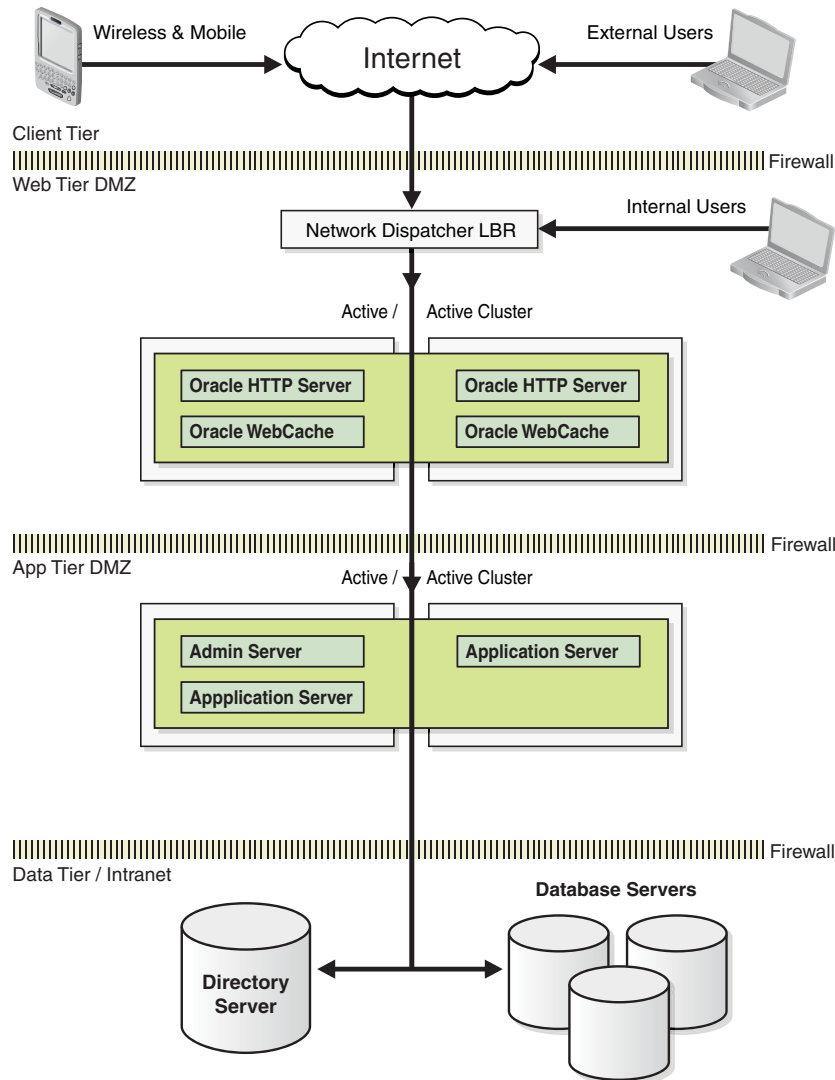
After installing and configuring Oracle Fusion Middleware, the environment contains the following:

- An Oracle WebLogic Server domain, which contains one Administration Server and one or more Managed Servers. The Administration Server contains the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control. The Managed Servers contain components, such as Oracle WebCenter and Oracle SOA Suite.
- If the environment includes system components, one or more Oracle instances.
- A metadata repository, if the installed components require one. For example, Oracle SOA Suite requires a metadata repository.

The environment also includes a Middleware home, which consists of the Oracle WebLogic Server home, and, optionally, one or more Oracle homes.

2.2 Oracle Fusion Middleware Deployment Topology

[Figure 2-1](#) shows an Oracle Fusion Middleware deployment topology.

Figure 2–1 Oracle Fusion Middleware Enterprise Deployment Architecture

As shown in [Figure 2–1](#), this topology represents a multi-tiered architecture. Users access the system from the client tier. Requests go through a hardware load balancer, which then routes them to a Web server cluster, running Oracle HTTP Server. Web servers route the requests to the WebLogic cluster. Applications running on the WebLogic cluster then interact with the database cluster to service the request.

2.3 Oracle Fusion Middleware Environment

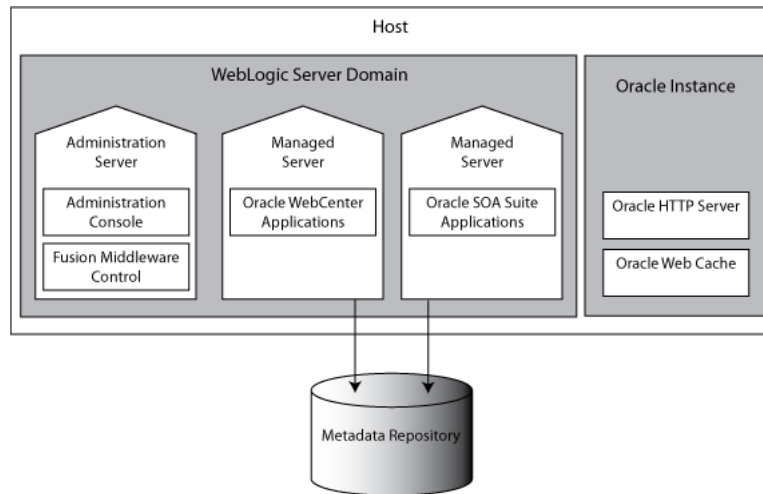
After installation, a typical Oracle Fusion Middleware environment contains the following ([Figure 2–2](#)):

- One Oracle WebLogic Server domain, which contains one Administration Server and one or more Managed Servers. See [Section 2.4, "What Is an Oracle WebLogic Server Domain?"](#).
- If system components are installed, they are configured in an Oracle Instance. See [Section 2.5, "What Is an Oracle Instance?"](#).

- A metadata repository, if the installed components require one. For example, Oracle SOA Suite requires a metadata repository. See [Section 3.2, "Oracle Metadata Repository"](#).
- A Middleware home, which contains product binary files. See [Section 2.6, "What Is a Middleware Home?"](#). Note that the product binary files will change as a result of patching. The configuration of the product binaries change as you configure and create a new data source.

Figure 2–2 shows an Oracle Fusion Middleware environment with an Oracle WebLogic Server domain that contains an Administration Server, two Managed Servers, and an Oracle instance. The environment also contains a metadata repository.

Figure 2–2 Oracle Fusion Middleware Environment



2.4 What Is an Oracle WebLogic Server Domain?

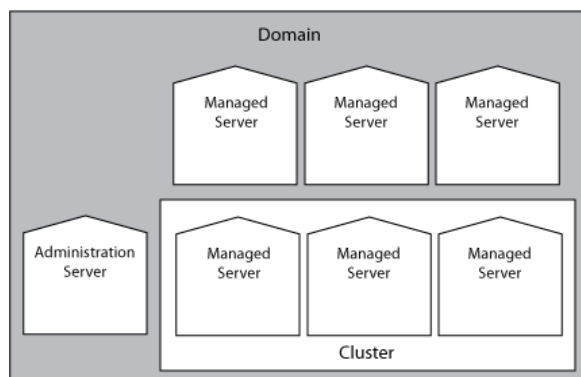
An **Oracle WebLogic Server domain** is a logically related group of Java components. A domain includes a special Oracle WebLogic Server instance called the Administration Server, which is the central point from at which configuration and management of all resources in the domain occur. Usually, you configure an additional domain to include additional Oracle WebLogic Server instances called Managed Servers. You can deploy Java components, such as Web applications, Enterprise Java Beans (EJB), and Web services, and other resources to the Managed Servers and uses the Administration Server for configuration and management purposes only.

Managed servers in a domain can be grouped together into a cluster.

The directory structure of a domain is separate from the directory structure of the Oracle WebLogic Server home. It can reside anywhere; it need not be within the Middleware home directory.

A domain is a peer of an Oracle instance. Both contain specific configurations outside of their Oracle homes.

Figure 2–3 shows a domain with an Administration Server, three standalone Managed Servers, and three Managed Servers in a cluster.

Figure 2–3 WebLogic Server Domain

For more information about domain configuration see the *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.

The following topics describe entities in the domain:

- [What Is the Administration Server?](#)
- [Managed Servers and Managed Server Clusters](#)
- [What Is Node Manager?](#)

2.4.1 What Is the Administration Server?

The **Administration Server** operates as the central control entity for the configuration of the entire domain. It maintains the domain's configuration documents and distributes changes in the configuration documents to Managed Servers. The Administration Server serves as a central location from which to monitor all resources in a domain.

Each domain must have one server instance that acts as the Administration Server.

To interact with the Administration Server, you can use the Oracle WebLogic Server Administration Console, Oracle WebLogic Scripting Tool (WLST), or create your own Java Management Extension (JMX) client. In addition, you can use Fusion Middleware Control Console for some tasks.

Oracle WebLogic Server Administration Console and Fusion Middleware Control run in the Administration Server. Oracle WebLogic Server Administration Console is the Web-based administration console used to manage the resources in an Oracle WebLogic Server domain, including the Administration Server and Managed Servers. Fusion Middleware Control is a web-based administration console used to manage Oracle Fusion Middleware, including components such as Oracle HTTP Server and Oracle Portal and product suites such as Oracle SOA Suite, Oracle WebCenter, and Oracle Identity Management.

2.4.2 Managed Servers and Managed Server Clusters

Managed Servers host business applications, application components, Web services, and their associated resources. To optimize performance, Managed Servers maintain a read-only copy of the domain's configuration document. When a Managed Server starts up, it connects to the domain's Administration Server to synchronize its configuration document with the document that the Administration Server maintains.

When you create a domain, you create it using a particular domain template. The domain template contains all of the information you would like to include in your domain configuration. The template can then be used for additional installations. That template supports a particular component or group of components, such as the Oracle SOA Suite. The Managed Servers in the domain are created specifically to host those particular Oracle Fusion Middleware components.

Oracle Fusion Middleware Java components (such as Oracle SOA Suite, Oracle WebCenter, and some Oracle Identity Management components), as well as customer-developed applications, are deployed to Managed Servers in the domain.

Managed servers are Java Virtual Machine (JVM) processes.

If you want to add other components, such as Oracle WebCenter, to a domain that was created using a template that supports another component, you can extend the domain by creating additional Managed Servers in the domain, using a domain template for the component that you want to add. For more information, see "Extending A Domain to Support Additional Components" in the *Oracle Fusion Middleware Administrator's Guide*.

For production environments that require increased application performance, throughput, or high availability, you can configure two or more Managed Servers to operate as a cluster. A cluster is a collection of multiple Oracle WebLogic Server server instances running simultaneously and working together to provide increased scalability and reliability. In a cluster, most resources and services are deployed identically to each Managed Server (as opposed to a single Managed Server), enabling failover and load balancing. A single domain can contain multiple Oracle WebLogic Server clusters, as well as multiple Managed Servers that are not configured as clusters. The key difference between clustered and non-clustered Managed Servers is support for failover and load balancing. These features are available only in a cluster of Managed Servers.

For more information see "Understanding WebLogic Server Clustering" in *Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server*

2.4.3 What Is Node Manager?

Node Manager is a Java utility that runs as a separate process from Oracle WebLogic Server and enables you to perform common operations for a Managed Server, regardless of its location with respect to its Administration Server. While use of Node Manager is optional, it provides valuable benefits if your Oracle WebLogic Server environment hosts applications with high-availability requirements. The Node Manager performs some of the following operations:

- Start
- Stop
- Process Monitoring
- Death Detection
- Restart

If you run Node Manager on a computer that hosts Managed Servers, you can start and stop the Managed Servers remotely using the Fusion Middleware Control Console or the command line. Node Manager can also automatically restart a Managed Server after an unexpected failure.

For more information about Node Manager see the *Oracle Fusion Middleware Node Manager Administrator's Guide for Oracle WebLogic Server*.

2.5 What Is an Oracle Instance?

An **Oracle instance** contains one or more system components, such as Oracle Web Cache, Oracle HTTP Server, or Oracle Internet Directory. The system components in an Oracle instance must reside on the same computer. An Oracle instance directory contains updatable files, such as configuration files, log files, and temporary files.

An Oracle instance is a peer of an Oracle WebLogic Server domain. Both contain specific configurations outside of their Oracle homes.

The directory structure of an Oracle instance is separate from the directory structure of the Oracle home. It can reside anywhere; it need not be within the Middleware home directory.

2.6 What Is a Middleware Home?

A **Middleware home** consists of the Oracle WebLogic Server home, and, optionally, one or more Oracle homes.

A Middleware home can reside on a local file system or on a remote shared disk that is accessible through a network file system (NFS).

2.7 What Is a WebLogic Server Home?

A **WebLogic Server home** contains installed files necessary to host a WebLogic Server. The WebLogic Server home directory is a peer of Oracle home directories and resides within the directory structure of the Middleware home.

2.8 What Is an Oracle Home and the Oracle Common Home?

An **Oracle home** contains installed files necessary to host a specific component or software suite. For example, the SOA Oracle home contains a directory that contains binary and library files for Oracle SOA Suite.

An Oracle home resides within the directory structure of the Middleware home. Each Oracle home can be associated with multiple Oracle instances or Oracle WebLogic Server domains. There can be multiple Oracle homes within each Middleware home.

The **Oracle Common home** contains the binary and library files required for Fusion Middleware Control and Java Required Files (JRF). There can be only one Oracle Common home within each Middleware home. The Oracle Common home is installed as part of a typical Fusion Middleware component installation. It is important to note, however, that a domain created with the WebLogic standard Configuration Wizard does not include the Fusion Middleware (EM) Control or Java Required Files (JRF). To manually add the JRF files to a managed server or cluster-only installation, see "Applying Oracle JRF to a Managed Server or Cluster" in *Oracle Fusion Middleware Administrator's Guide*.

2.9 What Is the Oracle Metadata Repository?

The **Oracle Metadata Repository** contains metadata for Oracle Fusion Middleware components, such as Oracle BPEL Process Manager, Oracle B2B, and Oracle Portal. It can also contain metadata about the configuration of Oracle Fusion Middleware and metadata for your applications.

A metadata repository can be database-based or file-based. If it is database-based, you can create it in an existing database using the Repository Creation Utility (RCU).

Oracle Fusion Middleware supports multiple repository types. A repository type represents a specific schema or set of schemas that belong to a specific Oracle Fusion Middleware component (for example, Oracle SOA Suite or Oracle Internet Directory.)

A particular type of repository, the Oracle Metadata Services (MDS) repository, contains metadata for most Oracle Fusion Middleware components, such as Oracle B2B, and for certain types of applications.

2.10 What is a Farm?

A **farm** is a collection of components managed by Fusion Middleware Control. It can contain Oracle WebLogic Server domains, one Administration Server, one or more Managed Servers, and the Oracle Fusion Middleware components that are installed, configured, and running in the domain.

2.11 Oracle Fusion Middleware Directory Structure and Concepts

During the installation process, you are asked for a variety of directory locations. This section defines each of these directories and explains the contents of each directory.

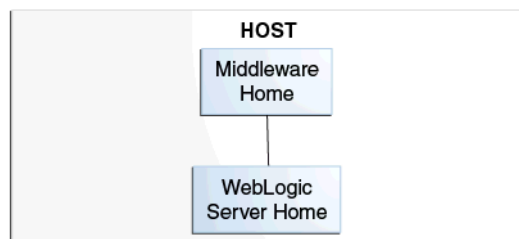
This section includes the following topics:

- [Middleware Home and WebLogic Home Directories](#)
- [Oracle Home Directory](#)
- [WebLogic Domain](#)
- [Multiple Products with Multiple Oracle WebLogic Server Domains](#)
- [Domain Extension](#)
- [Oracle Instance and Oracle WebLogic Server Domain](#)

2.11.1 Middleware Home and WebLogic Home Directories

The top-level directory for all Oracle Fusion Middleware products is called the Middleware home; this directory is created when Oracle WebLogic Server is installed. The WebLogic home directory is inside the Middleware home, and is also created when Oracle WebLogic Server is installed. [Figure 2-4](#) shows the directory structure after a Oracle WebLogic Server installation.

Figure 2-4 Directory Structure After Oracle WebLogic Server Installation

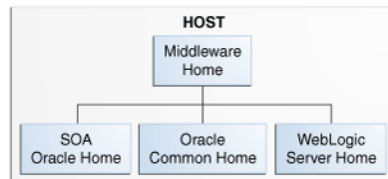


2.11.2 Oracle Home Directory

Each Oracle Fusion Middleware product must be installed in its own Oracle home location. The software binaries are installed into the Oracle home; no run time process can write to this directory.

The Oracle home directories (in this case, the SOA Oracle home) for each product must reside inside an existing Middleware home directory (see [Figure 2-5](#)).

Figure 2-5 Directory Structure with An Oracle Home Directory



The Oracle home directory can be created in the following ways:

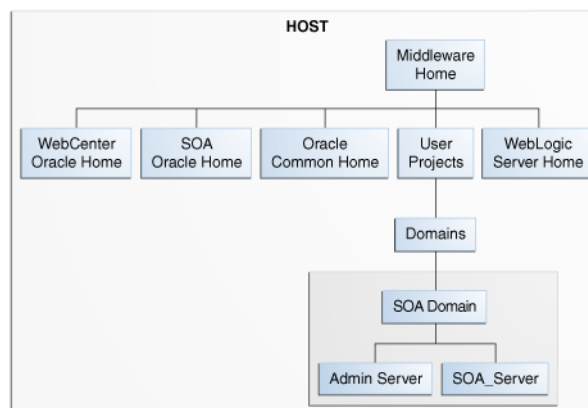
- Before running the installation, you can create your Oracle home directory on your system. Remember that the Oracle home must reside inside the Middleware home directory, and it must also be an empty directory. Then, when you run the installer and are asked to provide the location of your Oracle home, you can specify the directory you have created.
- While running the installer, you can specify the name of a new directory. This directory are automatically be created for you by the installer and will be located inside the Middleware home directory.

2.11.3 WebLogic Domain

After a product is installed, it can be configured into an Oracle WebLogic Server Domain (see [Section 2.4, "What Is an Oracle WebLogic Server Domain?"](#)).

The User Projects directory is created when you create or configure a domain (in this case, a SOA domain). By default, new domains are created inside the Domains folder, but you can choose to create these elsewhere.

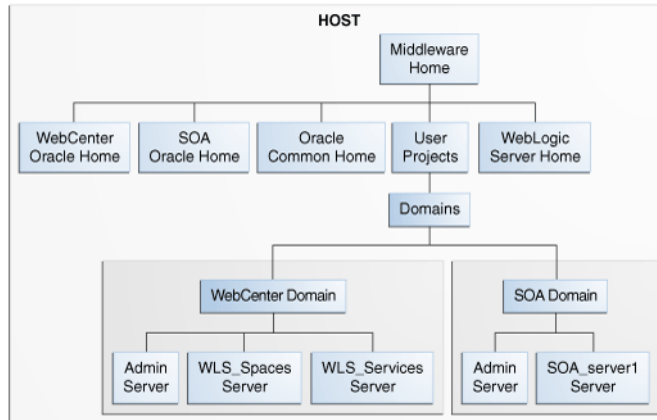
Figure 2-6 Directory Structure after an WebLogic Domain is Created



2.11.4 Multiple Products with Multiple Oracle WebLogic Server Domains

If you install multiple products and choose to create a separate domain for each product, then your directory structure would look something like the one shown in [Figure 2-7](#).

Figure 2-7 Directory Structure with Multiple Oracle Fusion Middleware Products



Each product has its own Oracle home directory. To differentiate among all the product Oracle home directories, the installation guides typically refer to each Oracle home by product name; for example, the Oracle home for Oracle SOA Suite is referred to as the SOA Oracle home, while the Oracle home for Oracle WebCenter Suite is referred to as the WebCenter Oracle home.

If you choose to create multiple domains on a single computer, ensure that each domain has a unique name.

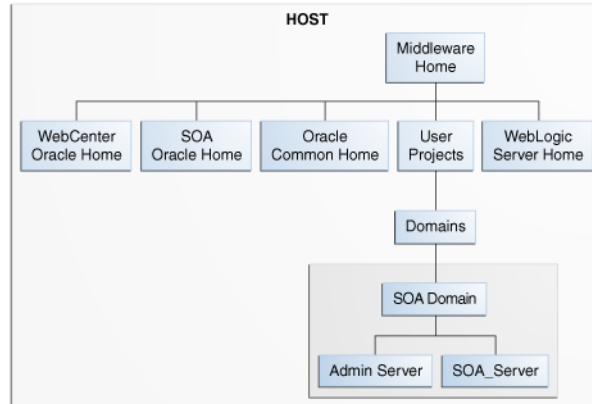
For more information about multiple products and multiple domains, see the following documentation:

- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle WebCenter*
- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management*

2.11.5 Domain Extension

During installation and configuration, you may choose to create an existing domain rather than to create a new domain. Extending a domain means that you add products and functionality to an existing domain. If, for example, you first install Oracle SOA Suite to create a new domain, then install Oracle WebCenter Suite while choosing to extend the existing Oracle SOA Suite domain, then your topology would look like the one shown in [Figure 2-8](#).

Figure 2–8 Directory Structure with Multiple Oracle Fusion Middleware Products in a Single Domain

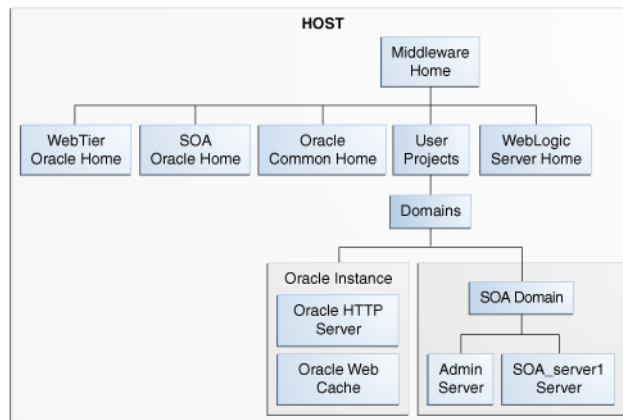


In essence, you are adding the products and functionality of Oracle WebCenter Suite to the existing Oracle SOA Suite domain.

2.11.6 Oracle Instance and Oracle WebLogic Server Domain

Figure 2–9 shows the directory structure when two products are installed, but one product (Oracle SOA Suite) is configured in a Oracle WebLogic Server domain (because it consists of Java components) while the other products (Oracle HTTP Server and Oracle Web Cache) is configured in an Oracle Instance (because it consists of system components):

Figure 2–9 Directory Structure with an Oracle Instance and a Oracle WebLogic Server Domain



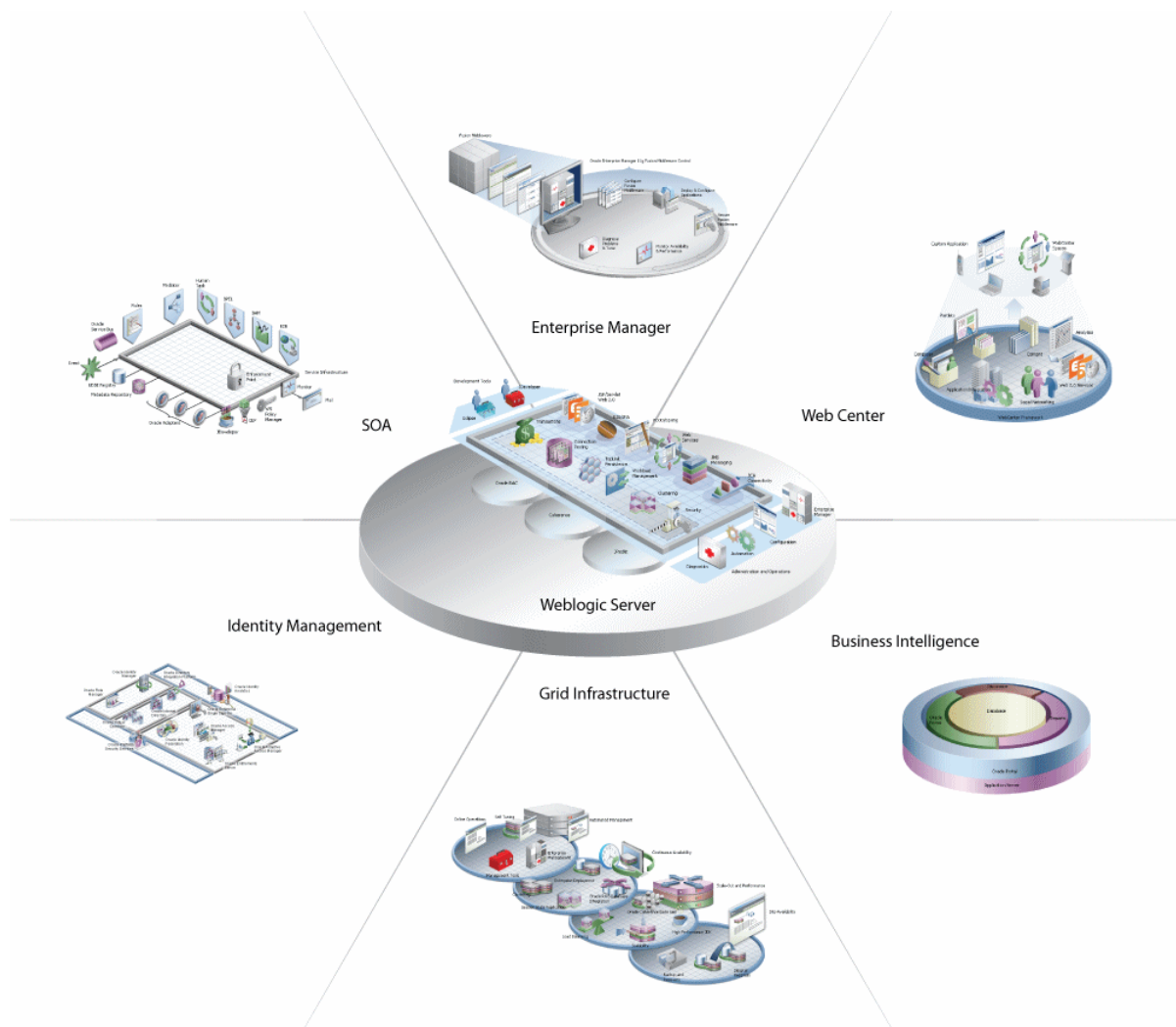
Oracle Fusion Middleware Components

This chapter describes Oracle Fusion Middleware components. It includes the following sections:

- Section 3.1, "Oracle WebLogic Server"
- Section 3.2, "Oracle Metadata Repository"
- Section 3.3, "Oracle Identity Management"
- Section 3.4, "Oracle WebCenter"
- Section 3.5, "Oracle Business Intelligence"
- Section 3.6, "Oracle Business Intelligence Discoverer"
- Section 3.7, "Oracle Portal"
- Section 3.8, "Oracle Reports"
- Section 3.9, "Oracle Forms Services"
- Section 3.10, "Oracle SOA Suite"
- Section 3.11, "Development Tools"
- Section 3.12, "Oracle HTTP Server"
- Section 3.13, "Oracle Web Cache"
- Section 3.14, "Oracle Web Services"
- Section 3.15, "Oracle Delegated Administration Services"
- Section 3.16, "Oracle Enterprise Content Management Suite"
- Section 3.17, "Oracle Data Integrator"

Figure 3–1 provides an overview of Oracle Fusion Middleware.

Figure 3–1 Oracle Fusion Middleware Overview



The following [Figure 3–2](#) provides an overview of the Oracle Fusion Middleware architecture.

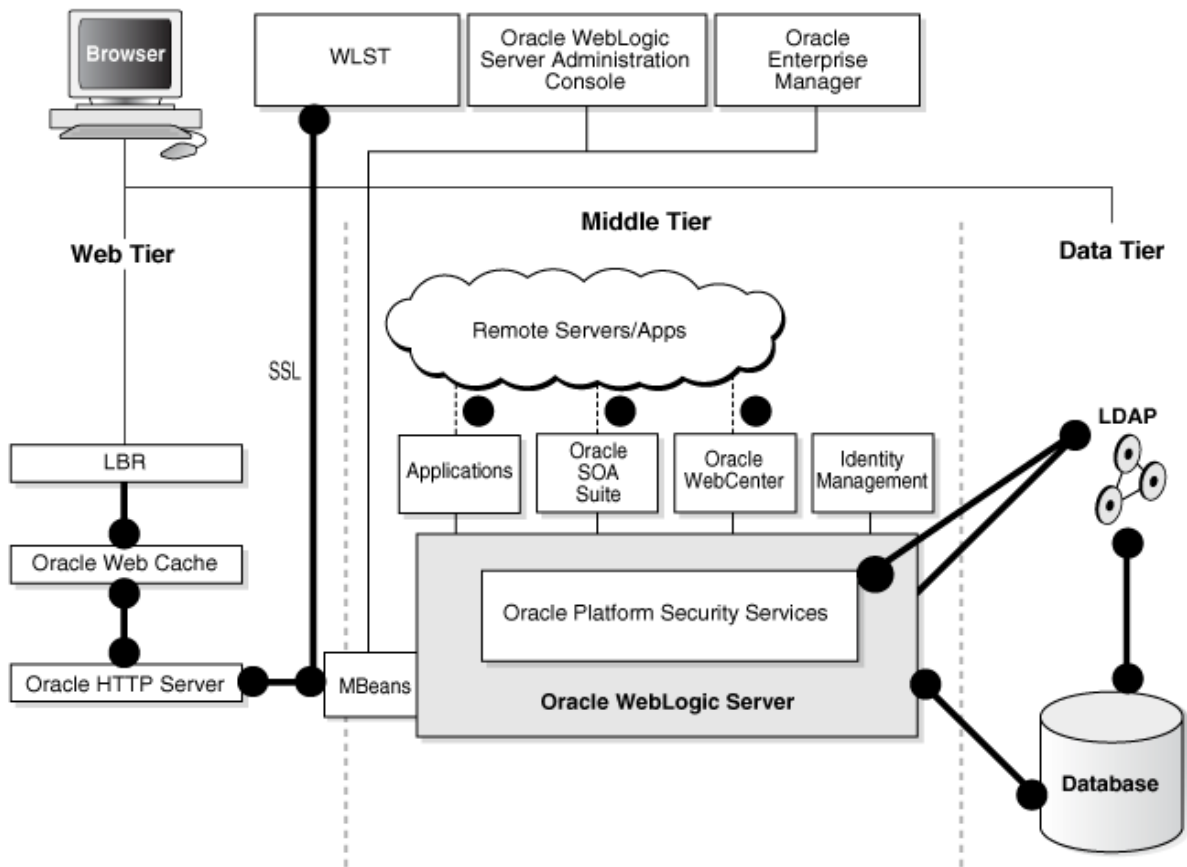
For more information see the following guides:

- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle WebCenter*
- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware High Availability Guide*

For more information on high-availability practices, visit:

<http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm>

Figure 3–2 Oracle Fusion Middleware Architecture Overview



3.1 Oracle WebLogic Server

Oracle WebLogic Server is a scalable, enterprise-ready Java Platform, Enterprise Edition (Java EE) application server. The Oracle WebLogic Server infrastructure supports the deployment of many types of distributed applications and is an ideal foundation for building applications based on SOA (Figure 3–2).

Java EE is a widely used platform for server programming in the Java programming language. The Java EE Platform differs from the standard edition of Java in that it adds libraries which provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on an application server.

Oracle WebLogic Server is a complete implementation of the Sun Microsystems Java EE 5.0 specification, which provides a standard set of APIs for creating distributed Java applications that can access a wide variety of services, such as databases, messaging services, and connections to external enterprise systems. End-user clients access these applications using Web browser clients or Java clients. Oracle WebLogic Server also supports the Spring Framework, a programming model for Java applications which provides an alternative to aspects of the Java EE model.

In addition to the Java EE implementation, Oracle WebLogic Server enables enterprises to deploy business-critical applications in a robust, secure, highly available, manageable, and scalable environment. These features allow enterprises to configure clusters of Oracle WebLogic Server instances to distribute load and provide

extra capacity in case of hardware or other failures. New diagnostic tools allow system administrators to monitor and tune the performance of deployed applications and the Oracle WebLogic Server environment itself. You can also configure Oracle WebLogic Server to monitor and tune application throughput automatically, without human intervention. Extensive security features protect access to services, keep enterprise data secure, and prevent malicious attacks.

3.1.1 Key Features of Oracle WebLogic Server

Oracle WebLogic Server includes the following key feature categories:

- [High Availability](#)
- [System Administration](#)
- [Diagnostic Framework](#)
- [Security Service](#)
- [Client Options](#)

3.1.1.1 High Availability

Oracle WebLogic Server includes the following features to support high availability:

- Oracle WebLogic Server clusters provide scalability and reliability for your applications by distributing the workload among multiple instances of Oracle WebLogic Server. Incoming requests can be routed to a Oracle WebLogic Server instance in the cluster based on the volume of work being processed. In case of hardware or other failures, the session state is available to other cluster nodes that can resume the work of the failed node.

For information about Oracle WebLogic Server clusters, see *Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server*.

- Work Managers prioritize work based on rules you define and by monitoring actual run time performance statistics. This information is then used to optimize the performance of your application. Work Managers may be applied globally to a Oracle WebLogic Server domain or to a specific application or component.

For information about Work Managers, see "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*.

- Overload protection gives Oracle WebLogic Server the ability to detect, avoid, and recover from overload conditions.

For information about overload protection, see "Avoiding and Managing Overload" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*.

- Network channels facilitate the effective use of network resources by segregating network traffic into channels based on the type of traffic.

For information about network channels, see "Configuring Network Resources" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*.

- The persistent store is a built-in, high-performance storage solution for Oracle WebLogic Server subsystems and services that require persistence. For example, it can store persistent Java Message Server (JMS) messages or temporarily store messages sent using the store-and-forward (SAF) feature.

For information about the Oracle WebLogic Server persistent store, see "Using the WebLogic Persistent Store" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*.

- SAF services enable Oracle WebLogic Server to deliver messages reliably between applications that are distributed across Oracle WebLogic Server instances. If the message destination is not available at the moment the messages are sent, either because of network problems or system failures, then the messages are saved on a local server instance and are forwarded to the remote destination once it becomes available.

For information about Oracle WebLogic Server SAF, see *Oracle Fusion Middleware Configuring and Managing Store-and-Forward for Oracle WebLogic Server*.

3.1.1.2 System Administration

System administration of a Oracle WebLogic Server environment includes tasks such as creating Oracle WebLogic Server domains; deploying applications; migrating domains from development environments to production environments; monitoring and configuring the performance of the Oracle WebLogic Server domain; and diagnosing and troubleshooting problems. Oracle WebLogic Server provides many tools for system administrators to help with these tasks, including a browser-based Administration Console, the WebLogic Scripting Tool (WLST), a scripting language for automation of WebLogic system administration tasks based on Jython, Simple Network Management Protocol (SNMP), the Configuration Wizard, and command-line utilities.

Because Oracle WebLogic Server's management system is based on Java EE and other standards, it integrates well with tools used to manage other enterprise software and hardware components. In addition, Oracle WebLogic Server implements the Java EE Java Management Extension (JMX) specification, which allows programmatic access to the Oracle WebLogic Server management system. Using this API, you can create your own administration utilities or automate frequent tasks using Java classes.

For more information about the tools used for these system administration tasks, see the following:

- "Overview of the Configuration Wizard" in *Oracle Fusion Middleware Creating Domains Using the Configuration Wizard*
- "Overview of the Administration Console" in *Oracle Fusion Middleware Introduction to Oracle WebLogic Server*
- *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*
- "SNMP Introduction and Roadmap" in *Oracle Fusion Middleware SNMP Management Guide for Oracle WebLogic Server*
- "Using the Oracle WebLogic Server Java Utilities" and "Weblogic.Server Command-Line Reference" in *Oracle Fusion Middleware Command Reference for Oracle WebLogic Server*

3.1.1.3 Diagnostic Framework

The Oracle WebLogic Server Diagnostic Framework is a monitoring and diagnostic service that lets you create, collect, analyze, archive, and access diagnostic data generated by a running server and its deployed applications. This data provides insight into the run time performance of Oracle WebLogic Server instances and deployed applications, and lets you isolate and diagnose faults and performance bottlenecks.

For more information, see *Oracle Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

3.1.1.4 Security Service

The Oracle WebLogic Server security architecture provides a comprehensive, flexible security infrastructure designed to address the security challenges of making applications available on the Web. Oracle WebLogic Server security can be used standalone to secure Oracle WebLogic Server applications or as part of an enterprisewide, security management system that represents an outstanding security management solution.

For more information, see "Overview of the WebLogic Security Service" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.

3.1.1.5 Client Options

In addition to support for browser-based Web application clients, Oracle WebLogic Server also supports a variety of client types for creating graphical user interface applications or simple command-line utilities. These client types include: RMI-IIOP, T3, J2SE clients, Java EE thin clients, CORBA/IDL clients, and C++ clients that communicate with Oracle Tuxedo.

For more information, see "Overview of Stand-alone Clients" in *Oracle Fusion Middleware Programming Stand-alone Clients for Oracle WebLogic Server*.

This section also includes the following topics:

- [Product Integration](#)
- [Integration with Web Servers](#)

3.1.1.5.1 Product Integration Oracle WebLogic Server provides the core application server run time within the integrated Oracle WebLogic Suite Java infrastructure. WebLogic Suite contains the following server-side components:

- Oracle WebLogic Server
- Oracle Coherence
- Oracle WebLogic Operations Control
- Oracle WebLogic Real Time

This integrated infrastructure enhances application performance, improves application availability, and enables predictable and reliable application scalability with high quality of service.

Oracle WebLogic Server also provides a variety of tools to integrate end-user applications with disparate systems. These tools include Web Services, Resource Adapters, the JMS .NET client, Messaging Bridge, and Remote Method Invocation (RMI).

3.1.1.5.2 Integration with Web Servers Plug-ins are provided with your Oracle WebLogic Server installation that allow Oracle WebLogic Server to operate with Web servers from Apache, Microsoft, and Sun Microsystems. Typically, these Web servers serve static HTML content while requests for dynamic Web content such as Java Server Pages (JSP) are directed to the Oracle WebLogic Server environment.

3.1.2 Oracle WebLogic Server Core Components

The following sections briefly describe some of the Oracle WebLogic Server core components:

- [Oracle WebLogic Server Web Services](#)
- [Oracle Coherence](#)
- [JMS Messaging Standard](#)
- [Enterprise Grid Messaging](#)
- [JRockit](#)
- [Tuxedo](#)
- [Oracle TopLink](#)

3.1.2.1 Oracle WebLogic Server Web Services

A Oracle WebLogic Server Web Service is a set of functions packaged into a single application that is available to other systems on a network. The network can be a corporate intranet or the Internet. Because Web Services rely on basic, standard technologies that most systems provide, they are an excellent means for connecting distributed systems together. They can be shared and used as components of distributed Web-based applications. Other systems, such as customer relationship management systems, order-processing systems, and other existing back-end applications, can call a Web Service function to request data or perform an operation.

Oracle WebLogic Server Web Services feature the following capabilities:

- Interoperability among distributed applications that span diverse hardware and software platforms
- Easy, widespread access to applications through firewalls using Web protocols
- A cross-platform, cross-language data model Extensible Markup Language (XML) that facilitates developing heterogeneous distributed applications

Oracle WebLogic Server Web Services are implemented according to the *Web Services for Java EE 1.2* specification (<http://www.jcp.org/en/jsr/detail?id=109>), which defines the standard Java EE run time architecture for implementing Web Services in Java. The specification also describes a standard Java EE Web Service packaging format, deployment model, and run time services, all of which are implemented by Oracle WebLogic Server Web Services.

For more information about Oracle WebLogic Server Web Services see the *Oracle Fusion Middleware Introducing Web Services for Oracle WebLogic Server*.

3.1.2.2 Oracle Coherence

Oracle Coherence is a component of Oracle Fusion Middleware that enables organizations to predictably scale mission-critical applications by providing fast and reliable access to frequently used data. By automatically and dynamically partitioning data in memory across multiple servers, Oracle Coherence enables continuous data availability and transactional integrity, even in the event of a server failure. As a shared infrastructure, Oracle Coherence combines data locality with local processing power to perform real-time data analysis, in-memory grid computations, and parallel transaction and event processing. Oracle Coherence comes in three editions.

For more information about Oracle Coherence see:

<http://www.oracle.com/technology/products/coherence/index.html>

3.1.2.3 JMS Messaging Standard

Java Message Service (JMS) is a messaging standard defined by Sun Microsystems. JMS is a set of interfaces and associated semantics that defines how a JMS client accesses the facilities of an enterprise messaging product.

Oracle WebLogic Server JMS provides a Java API for Advanced Queuing (AQ) based on the JMS standard. Oracle WebLogic Server JMS supports the standard JMS interfaces and has extensions to support the AQ administrative operations and other AQ features that are not a part of the standard.

For more information about JMS see the *Oracle Fusion Middleware Configuring and Managing JMS for Oracle WebLogic Server*.

3.1.2.4 Enterprise Grid Messaging

Oracle WebLogic Server includes a highly performant and reliable Java Messaging engine built into the application server. As part of the application server, it shares the management infrastructure, services, and processes with the server, which means reduced overhead both from an operations and run time perspective.

Oracle WebLogic Server Enterprise Grid Messaging is a key component in Oracle Fusion Middleware and is commonly used as a full-fledged enterprise messaging backbone. Besides high performance, WebLogic Enterprise Messaging Grid includes the reliability, availability, and integration capabilities you would expect from an enterprise messaging solution. It also includes unique innovative features such as out-of-the-box integration with Oracle AQ and strict message processing order in enterprise-scale clustered applications.

For more information see the *Oracle Fusion Middleware Information Roadmap for Oracle WebLogic Server* guide.

3.1.2.5 JRockit

JRockit is a portfolio of Java run time solutions for Java platform, Standard Edition (Java SE).

JRockit provides a solution for latency-sensitive applications, such as financial front office response, military command and control, and telecommunications networks, resulting in significant performance increases and reduced costs.

The JRockit JVM is a high-performance JVM developed to ensure reliability, scalability, manageability, and flexibility for Java applications. The JRockit JVM delivers a new level of performance for Java applications deployed on Intel 32-bit (Xeon) and 64-bit (Xeon, Itanium, and SPARC) architectures at significantly lower costs to the enterprise. Furthermore, it is the only enterprise-class JVM optimized for Intel architectures, providing seamless interoperability across multiple hardware and operating configurations. The JRockit JVM makes it possible to gain optimal performance for Java applications when running it on either the Windows or Linux operating platforms on either 32-bit or 64-bit architectures. The JRockit JVM is especially well-suited for running Oracle WebLogic Server.

For more information about JRockit see the *Oracle Fusion Middleware Introduction to Oracle WebLogic Server*.

3.1.2.6 Oracle Application Development Framework

The Oracle Application Development Framework (Oracle ADF) is an end-to-end application framework that builds on Java EE standards and open-source technologies to simplify and accelerate implementing service-oriented applications. If you develop enterprise solutions that search, display, create, modify, and validate data using Web,

wireless, desktop, or Web services interfaces, Oracle ADF can simplify your job. Used in tandem, Oracle JDeveloper and Oracle ADF give you an environment that covers the full development lifecycle from design to deployment, with drag-and-drop data binding, visual UI design, and team development features built-in.

For more information see the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

3.1.2.7 Tuxedo

Tuxedo is a scalable multilanguage, high-performance messaging and highly distributed transaction processing platform for business-critical applications.

For more information see the *Oracle Fusion Middleware Tuxedo Connector Administration Guide for Oracle WebLogic Server*.

3.1.2.8 Oracle TopLink

Oracle TopLink builds high-performance applications that store persistent object-oriented data in a relational database. It successfully transforms object-oriented data into either relational data or Extensible Markup Language (XML) elements.

Oracle TopLink is an advanced, object-persistence and object-transformation framework that provides development tools and run time capabilities that reduce development and maintenance efforts, and increase enterprise application functionality.

Oracle TopLink includes support for EJB 3.0 in Java EE and Java SE environments, as well as support for EJB 2.*n* container-managed persistence (CMP). You can integrate Oracle TopLink with a variety of application servers, including Oracle WebLogic Server, OC4J, SunAS, JBoss, and IBM WebSphere.

For more information about developing, deploying, and configuring Oracle TopLink applications, see the *Oracle Fusion Middleware Developer's Guide for Oracle TopLink*.

3.2 Oracle Metadata Repository

A metadata repository contains metadata for Oracle Fusion Middleware system components, such as Oracle BPEL Process Manager, Oracle B2B, and Oracle Portal. It can also contain metadata about the configuration of Oracle Fusion Middleware and metadata for enterprise applications.

A metadata repository can be database-based or file-based. If it is database-based, Oracle Metadata Repository can be installed into an existing database using the Repository Creation Utility (RCU).

Oracle Fusion Middleware supports multiple repository types. A repository type represents a specific schema or a set of schemas that belongs to a specific Oracle Fusion Middleware component (for example, Oracle SOA Suite or Oracle Internet Directory.)

For more information, see "Managing the Oracle Metadata Repository" in the *Oracle Fusion Middleware Administrator's Guide*.

A particular type of repository, the MDS Repository, contains metadata for most Oracle Fusion Middleware components, such as Oracle B2B, and for certain types of applications. For more information about the MDS Repository see "Managing the MDS Repository" in the *Oracle Fusion Middleware Administrator's Guide*.

3.3 Oracle Identity Management

Oracle Identity Management is an enterprise identity management system that automatically manages users' access privileges within the resources of an enterprise. The architecture of Oracle Identity Management works with the most demanding business requirements without requiring changes to existing infrastructure, policies, or procedures.

Oracle Identity Management is designed to manage user access privileges across all the resources in an enterprise. Oracle Identity Management manages throughout the entire identity management lifecycle; from initial creation of access privileges to dynamically adapting to changes in enterprise business requirements.

Oracle Identity Management products provide for a shared infrastructure for all Oracle applications. It also provides services and interfaces that facilitate third-party enterprise application development. These interfaces are useful for application developers who must incorporate identity management into their applications.

Note: Oracle Application Server Single Sign-On is now part of Oracle Access Manager. For more information see the *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.

Oracle Identity Management includes the following components:

- [Oracle Identity Manager](#)
- [Oracle Internet Directory](#)
- [Oracle Directory Integration Platform](#)
- [Oracle Identity Federation](#)
- [Oracle Virtual Directory](#)
- [Oracle Platform Security Services](#)
- [Oracle Role Manager](#)
- [Oracle Entitlements Server](#)
- [Oracle Directory Services Manager](#)
- [Oracle Access Manager](#)
- [Oracle Adaptive Access Manager](#)
- [Oracle Identity Navigator](#)
- [Oracle Authorization Policy Manager](#)

3.3.1 Oracle Identity Manager

Oracle Identity Manager is a user provisioning and administration 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 identify which users have access to which applications. Oracle Identity Manager is available as a stand-alone product or as part of Oracle's Identity and Access Management Suite.

Automating user identity provisioning can reduce Information Technology (IT) administration costs and improve security. Provisioning also plays an important role in regulatory compliance. Key features of Oracle Identity Manager include password

management, workflow and policy management, identity reconciliation, reporting and auditing, and extensibility through adapters.

See the following for more information about Oracle Identity Manager:

- "Application Management" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Manager*
- "Concepts" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*
- "Feature Overview" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager*

3.3.2 Oracle Internet Directory

Oracle Internet Directory is a Lightweight Directory Access Protocol (LDAP) v3 directory that takes advantage of the scalability, high availability, and security features of the Oracle Database. Oracle Internet Directory serves as the central user repository for Oracle Identity Management, a component of Oracle Fusion Middleware, simplifying user administration in the Oracle environment and providing a standards-based general purpose directory for the heterogeneous enterprise.

Oracle Internet Directory has a multi threaded, multi-instance process model with Oracle Database as the directory data store. This unique architecture translates to tremendous flexibility, enabling several high availability configurations, and opens up scalability possibilities on all kinds of hardware configuration.

For security, Oracle Internet Directory supports Oracle Database Vault and Oracle Transparent Data Encryption, as well as multiple levels of access control and sophisticated password policy management capabilities.

See the following for more information about Oracle Internet Directory:

- "Introduction to Directory Services" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
- "Understanding Oracle Internet Directory Concepts and Architecture" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
- "Understanding Process Control of Oracle Internet Directory Components" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
- "Understanding Oracle Internet Directory Organization" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
- "Understanding Oracle Internet Directory Replication " in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*

3.3.3 Oracle Directory Integration Platform

Oracle Directory Integration Platform is a Java EE application that runs on an Oracle WebLogic Managed server and enables you to synchronize data between different repositories and Oracle Internet Directory. Synchronization can be one-way or two-way.

Oracle Directory Integration Platform includes services and interfaces that allow you to develop synchronization solutions with other enterprise repositories. It can also be used to provide Oracle Internet Directory interoperability with third party metadirectory solutions.

Oracle Directory Integration Platform uses the Oracle Fusion Middleware infrastructure for monitoring, management, auditing, logging and credential storing.

For more information about Oracle Directory Integration Platform concepts, see Chapter 1, "Introduction to Oracle Identity Management Integration," in the *Oracle Fusion Middleware Integration Guide for Oracle Identity Management*.

3.3.4 Oracle Identity Federation

Oracle Identity Federation enables identities and their relevant entitlements to be propagated across security environments, whether these secure environments reside within an organization or in different organizations.

A self-contained and flexible multi-protocol federation server, Oracle Identity Federation integrates with leading identity and access management systems, enabling organizations to continue using their existing infrastructures and share identities across vendors, customers, and business partners without the increased costs of managing, maintaining, and administering additional identities and credentials.

Using an event-based model, Oracle Identity Federation can receive, process, and respond to HTTP and SOAP-based messages. Once an assertion is received, the federation server processes the assertion using its core protocol and business logic. Oracle Identity Federation implements the major federation protocols including the OASIS SAML 2.0, OASIS SAML 1.0 and 1.1, and WS-Federation specifications.

To make authentication and authorization decisions, Oracle Identity Federation integrates with third-party identity and access management systems, including:

- AAA engines
- LDAP and RDBMS user data repositories such as Oracle Internet Directory and Oracle Database

For more information about Oracle Identity Federation see:

- "About Oracle Identity Federation" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation*
- "Architecture Options" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation*
- "Profiles and Bindings" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation*
- "Authentication Engines" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation*
- "Data Repositories" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation*
- "Deployment Scenarios" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation*

3.3.5 Oracle Virtual Directory

Oracle Virtual Directory is a component of Oracle Fusion Middleware as a standalone JavaSE process. It is an LDAP service that provides a single, abstracted view of enterprise directory servers and databases from a variety of vendors. Oracle Virtual Directory can serve as a single source of facts in an environment with multiple data sources.

Oracle Virtual Directory minimizes or eliminates the need to change existing infrastructure when you add new applications, saving time and reducing expenses. Oracle Virtual Directory's data translation and joining capabilities allow you to create an integrated view of multiple data sources without changing their structures. This enables organizations to share data in their own repositories, yet retain full control of them and monitor usage. The sources can be separately owned and need not be synchronized. Users see only a single, logical LDAP tree, although there may be multiple data sources.

Oracle Virtual Directory uses the Oracle Fusion Middleware infrastructure for monitoring, management, auditing, logging, and credential storing. You can also manage Oracle Virtual Directory using Oracle Directory Services Manager, the unified browser-based graphical user interface (GUI) for both Oracle Virtual Directory and Oracle Internet Directory. Oracle Directory Services Manager simplifies the administration and configuration of Oracle Virtual Directory and Oracle Internet Directory by allowing you to use Web-based forms and templates.

For more information about Oracle Virtual Directory concepts, see "Understanding Oracle Virtual Directory," in the *Oracle Fusion Middleware Administrator's Guide for Oracle Virtual Directory*.

3.3.6 Oracle Platform Security Services

Oracle Platform Security Services (OPSS) provides enterprise product development teams, systems integrators, and independent software vendors with a standards-based, portable, integrated, enterprise-grade security framework for Java SE and Java EE applications.

OPSS provides an abstraction layer in the form of standards-based application programming interfaces (APIs) that insulate developers from security and identity management implementation details. With OPSS, developers do not need to know the details of cryptographic key management or interfaces with user repositories and other identity management infrastructures.

OPSS is the underlying security platform that provides security to Oracle Fusion Middleware including products like Oracle WebLogic Server, Oracle SOA, Oracle WebCenter, Oracle ADF, and Oracle Entitlements Server to name a few. OPSS is designed to be portable to third-party application servers. As a result, developers can use OPSS as the single security framework for both Oracle and third-party environments, thus decreasing application development, administration, and maintenance costs.

For more information about OPSS, see the *Oracle Fusion Middleware Security Guide*.

3.3.7 Oracle Role Manager

Oracle Role Manager, previously known as Bridgestream SmartRoles, is an enterprise-class application for managing business and organizational relationships, roles, and resources. As the most comprehensive role management product on the market and the system of record for role lifecycle management, it also provides tools for role mining, organizational modeling, and administration. Oracle Role Manager uses your business policy and real-time business views of data to derive privileges, and integrates with Oracle Identity Management applications to automate role-based provisioning and access control across the IT infrastructure.

3.3.8 Oracle Entitlements Server

Oracle Entitlements Server, formerly AquaLogic Enterprise Security, is a fine-grained authorization and entitlement management solution that can be used to precisely control the protection of application resources. It simplifies and centralizes security for enterprise applications and SOA by providing comprehensive, reusable, and fully auditable authorization policies and a simple, easy-to-use administration model.

3.3.9 Oracle Directory Services Manager

Oracle Directory Services Manager is the unified browser-based GUI for managing instances of Oracle Internet Directory and Oracle Virtual Directory. It is a replacement for Oracle Directory Manager.

Oracle Directory Services Manager simplifies the administration and configuration of Oracle Virtual Directory and Oracle Internet Directory by allowing you to use Web-based forms and templates. With Oracle Directory Services Manager, you can configure the structure of the directory, define objects in the directory and add and configure users, groups, and other entries. Oracle Directory Services Manager is also the interface you use to manage entries, schema, security, and other directory features.

Additionally, you can use Oracle Directory Services Manager to manage system configuration attributes, which can be useful if Fusion Middleware Control is not available or if you must modify an attribute that has no Fusion Middleware Control interface.

Note: You can invoke Oracle Directory Services Manager directly or from Oracle Enterprise Manager Fusion Middleware Control.

For more information about Oracle Directory Services Manager see:

- "Using Oracle Directory Services Manager to Manage Oracle Internet Directory" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
- "Getting Started with Oracle Directory Services Manager" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Virtual Directory*

3.3.10 Oracle Access Manager

Oracle Access Manager allows users of your applications or information technology (IT) systems to log in once and gain access to a broad range of IT resources. Oracle Access Manager provides an identity management and access control system that is shared by all your applications. The result is a new centralized and automated single sign-on (SSO) solution for managing who has access to what information across your entire IT infrastructure. Oracle Access Manager is available as a stand-alone product or as part of Oracle's Identity and Access Management Suite.

For more information about Oracle Access Manager see:

- "Introduction to Oracle Access Manager Administration" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Access Manager*
- "Introduction to Oracle Access Manager Integration" in the *Oracle Access Manager Integration Guide*

3.3.11 Oracle Adaptive Access Manager

Oracle Adaptive Access Manager is the Oracle Identity Management solution for Web-access real-time fraud detection and multifactor online authentication security for the enterprise. Oracle Adaptive Access Manager is designed to support complex, heterogeneous enterprise environments. It consists of two tightly integrated components that provide powerful protection against online identity theft.

Oracle Adaptive Access Manager (OAAM) protects companies exposing Web applications, services and their end-users from online threats and insider fraud. OAAM provides risk-aware authentication, real-time behavior profiling, transaction and event risk analysis.

Oracle Adaptive Access Manager contains functionality in two major areas:

- Real-time or offline risk analysis
- Functionality to prevent fraud

Adaptive access systems can provide the highest levels of security with context-sensitive online authentication and authorization. Thus, situations are evaluated and proactively acted upon based on various types of data.

For more information about Oracle Adaptive Access Manager see:

- "Introduction to Oracle Adaptive Access Manager" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Adaptive Access Manager*

3.3.12 Oracle Identity Navigator

Oracle Identity Navigator is an administrative portal designed to act as an application console for all of the Oracle Identity Management products. It does not replace the individual product consoles. You access Oracle Identity Navigator through a browser and use it to access consoles for Oracle Access Manager, Oracle Adaptive Access Manager, Oracle Identity Manager, Directory Services (ODSM), and other Oracle Identity Management services. You configure Oracle Identity Navigator to connect to these consoles either by specifying the URLs directly, or by employing the product discovery feature. The Oracle Identity Management Navigator simplifies the management of all of the Oracle Identity consoles from one site.

For more information about Oracle Identity Navigator see:

- "Introduction to Oracle Identity Navigator" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Navigator*.

3.3.13 Oracle Authorization Policy Manager

Oracle Authorization Policy Manager (APM) is a GUI tool to manage authorization policy for Oracle Platform Security Services based applications. APM is a tool for security administrators and greatly simplifies creation, modification, configuration and administration of application authorization policy by offering the following:

- User-friendly names and descriptions of security artifacts
- A way to organize application roles by business, product, or any other parameter specific to an application
- A uniform graphic interface to search, create, browse, and edit security artifacts
- A way to specify a subset of applications that a role can manage

For more information about Oracle Authorization Policy Manager see:

- "Introduction to Oracle Authorization Policy Manager" in the *Oracle Fusion Middleware Authorization Policy Manager Administrator's Guide*.

3.4 Oracle WebCenter

Oracle WebCenter includes the following components:

- [Oracle Composer](#)
- [Oracle WebCenter Framework](#)
- [Oracle WebCenter Services](#)
- [Oracle WebCenter Spaces](#)

3.4.1 Oracle Composer

Oracle Composer is an innovative component that enables any application or portal to be customized or personalized after it has been deployed and is in use. Oracle Composer is designed to be extremely easy to use so that business users and end users can edit any page with a few clicks of the mouse. Oracle Composer runs in all modern browsers and provides a dynamic way to edit Java Server Faces (JSF) application and portal pages by selecting information and components from the Business Dictionary or Resource Catalog.

Oracle Composer can be easily added to applications and portals during development at any time to enable this Web 2.0 style of operation, without writing a single line of code. Oracle Composer can be simply dragged and dropped onto a JSF page to enable this powerful capability.

Customizations change everyone's view of an application or portal and are typically required to *brand* a delivered application or portal for a specific customer or purpose. Oracle WebCenter and Oracle Composer provide an extremely flexible model for storing customizations in the file system or directly into any database using MDS.

Personalizations change views of a portal or application page. Other users are not affected by the changes you make to a page. Users decide what is important to them and organize information so that it is quick and easy to find. Oracle Composer enables personalizations by displaying information from Oracle's Business Dictionary or Resource Catalog in a role-based view, so that users see only components relevant to them. The important point is that the page can be tailored by individuals to add any combination of components to their page whenever they want or need them without affecting everyone else's view of the page.

Oracle Composer is integrated with Oracle WebCenter Framework and Oracle WebCenter Services. You can add Oracle Composer components to your JSF application pages to enable users to edit those pages at run time. You can add Oracle Composer components at any time during the development lifecycle, when the requirements of the application demand it. Additionally, you can use the Page service to enable users to create pages at run time.

Oracle Composer has been used extensively inside Oracle WebCenter Spaces to allow you to customize and personalize group and personal spaces. The Oracle WebCenter Spaces application provides a working example of how end users can take an active role in managing and altering their work environment to match their specific requirements.

For more information about Oracle Composer, see the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

3.4.2 Oracle WebCenter Framework

Oracle WebCenter Framework is an Oracle JDeveloper design-time extension that removes the boundaries between Web-based portals and enterprise applications. It also provides the run time portal and Web 2.0 framework on which all Oracle WebCenter technology runs.

Oracle WebCenter Framework enables you to embed asynchronous Java Script and XML (AJAX) based components, portlets, services, and content into context-rich customizable applications and enterprise portals. Traditionally, developers had to decide whether they were building a portal, a Web application, a composite application, or a social networking or community site. Oracle WebCenter Framework provides a modern development framework that bridges the differences and speeds delivery of all of these different types of applications. Developers do not have to decide ahead of time whether they are building a portal or an application; with Oracle WebCenter Framework, any application can be a portal and any portal can be an application. Most importantly, Oracle WebCenter Framework is designed in such a way that each type of component can be added to the portal or application when required, or left out if not required. This significantly improves overall application delivery and maintenance.

Oracle WebCenter Framework augments the JSF declarative development environment with additional components, integration, and run time options by integrating capabilities that are historically included only in portal products. These capabilities enable you to:

- Build and consume standards-based portlets.
- Integrate content from many different data stores through standard interfaces.
- Make the application customizable at run time, which lets you edit application pages according to their requirements using Oracle Composer.
- Enable administrators to customize the behavior of components in ways that are helpful to users.

Oracle WebCenter Framework includes ready-to-use components to easily add social networking and personal productivity services to your application or portal. These components are provided by Oracle WebCenter Services.

Oracle WebCenter Framework includes adapters for Oracle Content Server and Oracle Portal.

For more information about Oracle WebCenter Framework, see the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

3.4.3 Oracle WebCenter Services

Oracle WebCenter Services exposes social networking and personal productivity features through a comprehensive set of services. The services are integrated so that you can use them together (for example, you can add an Instant Messaging and Presence link to a discussion forum to talk with a product expert directly from a forum topic).

All Oracle WebCenter Services have a complete set of ready-to-use task flows or portlets to speed their adoption, making developers and end users quickly productive. For example, the Documents service is exposed through the Documents task flow, the Document List Viewer task flow, and the Recent Document task flow. You can add any of these task flows to an application or portal when you are building it, and users can add these service task flows to a deployed application or portal with Oracle Composer.

If the prebuilt task flows do not meet the requirements, developers have two additional choices:

1. Customize the prebuilt task flows to match the requirements directly. These customizations are created inside JDeveloper and provide developers with an extremely productive and fast way of delivering their required functionality.
2. Use the prebuilt data controls and bind their own user interface to meet the requirements, without having to build the back-end integration.

Oracle WebCenter Services are built to industry standards, such as IMAP for e-mail, and is integrated into Oracle Composer and Oracle WebCenter Framework.

For more information about Oracle WebCenter Services, see the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

3.4.4 Oracle WebCenter Spaces

Oracle WebCenter Spaces is a ready-to-use application that pulls together all the capabilities of Oracle WebCenter Services to empower teams to quickly and efficiently manage their information, applications, projects, and people without requiring IT assistance.

Built using Oracle WebCenter Framework in conjunction with Oracle JDeveloper and Oracle ADF, Oracle WebCenter Spaces brings together the latest technology around social computing, communities, personal productivity, and ad-hoc team interactions with no development effort. It eliminates or reduces duplication of effort and content inconsistencies and enables the sharing of valuable team resources for solving business problems, tapping into new ideas, and reducing time to market.

Oracle WebCenter Spaces delivers the following powerful features that enable you to provide a productive environment of team-based sites, along with enterprise and departmental portals:

- Personal Spaces provide individual users with their own sets of personal pages that they can add to, change, and share with others.
- Business Role Pages provide an efficient way to associate a page or set of pages with a specific enterprise role so that all users are kept up-to-date with information specific to them and their job function.
- Group Spaces make it easy to deliver a departmental or enterprise portal without having to spend a lot of development time building the site. Administrators and end users get a role-based view of what they can add to the project or portal sites. Group spaces can be completely customized and insulates the IT team from application upgrades.
- Group Space Templates provide an efficient means of start up a group space. While Oracle WebCenter Spaces is optimally configured to enable you to be productive immediately, you can also correct different components together or tailor the space in line with your company look and feel. You can make such adjustments to a group space and then save it as a template for others to use directly.

For more information, see the *Oracle Fusion Middleware User's Guide for Oracle WebCenter*.

3.5 Oracle Business Intelligence

Oracle Business Intelligence (BI) is a portfolio of technology and applications that provides the industry's first integrated, end-to-end Enterprise Performance Management System, including BI foundation and tools as well as financial performance management applications, operational BI applications, and data warehousing.

Oracle Business Intelligence includes the following:

- [Oracle Business Intelligence Enterprise Edition \(EE\)](#)
- [Oracle Business Intelligence Publisher](#)
- [Oracle Real-Time Decisions](#)

3.5.1 Oracle Business Intelligence Enterprise Edition (EE)

Oracle Business Intelligence Enterprise Edition (Oracle BI EE) is a comprehensive set of enterprise business intelligence tools and infrastructure, including a scalable and efficient query and analysis server, an ad-hoc query and analysis tool, interactive dashboards, proactive intelligence and alerts, real-time predictive intelligence, and an enterprise reporting engine. Oracle Business Intelligence Enterprise Edition is designed to bring greater business visibility and insight to a wide variety of users.

The components of Oracle Business Intelligence Enterprise Edition share a common service-oriented architecture, data access services, analytic and calculation infrastructure, metadata management services, semantic business model, security model and user preferences, and administration tools. Oracle Business Intelligence Enterprise Edition provides scalability and performance with data-source specific optimized analysis generation, optimized data access, advanced calculation, intelligent caching services, and clustering.

For more information, see "Introducing Oracle Business Intelligence Enterprise Edition" in the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* and "Introduction to Oracle Business Intelligence System Administration" in the *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*

3.5.2 Oracle Business Intelligence Publisher

Oracle Business Intelligence Publisher (BI Publisher, formerly XML Publisher) allows you to create highly formatted reports that are suitable for printing. BI Publisher reports are built on top of BI Publisher data models. A BI Publisher data model can consist of data sets from a wide range of sources, such as subject areas from the BI Server or analyses, SQL queries against relational data bases, MDX queries against Essbase or other OLAP sources, LDAP, Web Services, Microsoft Excel, HTTP feeds, or XML files. BI Publisher supports a wide range of layout types, so you can create the full range of documents that your organization might need. Within Oracle BI EE, you can view, create, edit, and schedule BI Publisher reports and then include them in dashboard pages.

For more information on running reports with Oracle BI Publisher, see *Oracle Business Intelligence Publisher Report Designer's Guide*.

3.5.3 Oracle Real-Time Decisions

Oracle Real-Time Decisions (Oracle RTD) enables you to develop adaptive enterprise software solutions. These adaptive solutions continuously learn from business

transactions while they execute and automatically optimize each transaction, in real time, by way of close loop business rules and predictive models.

For more information, see *Oracle Fusion Middleware Platform Developer's Guide for Oracle Real-Time Decisions*.

3.6 Oracle Business Intelligence Discoverer

Oracle Business Intelligence Discoverer (Oracle BI Discoverer) is a business intelligence tool for analyzing data, and is a key component of Oracle Fusion Middleware. The latest release of Oracle Business Intelligence Discoverer 11g offers new functionality, including published Discoverer Webservice APIs, integration with Oracle WebCenter, integration with Oracle WebLogic Server, integration with Enterprise Manager (Fusion Middleware Control) and improved performance and scalability.

For more information, see "What is Oracle BI Discoverer?" in the *Oracle Fusion Middleware Configuration Guide for Oracle Business Intelligence Discoverer*.

3.7 Oracle Portal

Oracle Portal offers a complete and integrated framework for building, deploying, and managing enterprise portals. Only Oracle Portal delivers a unified and secure point of access to all enterprise information and services to improve business visibility and collaboration, reduce integration costs, and ensure investment protection.

Oracle Portal offers the following advantages:

- Improved productivity
Employees, customers, partners, and suppliers are empowered with role-based, self-service access to enterprise content and services to improve business visibility and responsiveness to key business events.
- Reduced costs and complexity
A pre-integrated, portal solution that enables you to quickly build and deploy an enterprise portal that delivers powerful business intelligence, search, collaboration, and security capabilities.
- Investment protection through open standards
Support for open standards to take advantage of existing enterprise investments by exposing heterogeneous applications and services through a unified, intuitive user interface.

For more information about Oracle Portal, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Portal*.

3.8 Oracle Reports

Oracle Reports is an enterprise reporting tool that enables businesses to give immediate access to information to all levels within and outside of the organization in a scalable and secure environment.

With Oracle Reports, you can rapidly develop and deploy sophisticated Web and paper reports against any data source, including an Oracle database, JDBC, XML, and text files. By leveraging Java EE technologies, such as JSP and XML, you can publish your reports in a variety of formats, including HTML, XML, PDF, Spreadsheet,

delimited text, PostScript, and RTF to any destination (including e-mail, Web browser, WebDav, FTP, printer, Oracle Portal, and file system) in a scalable, efficient manner.

Oracle Reports includes the following components:

- Oracle Reports Builder: Report developers use the Oracle Reports Builder design-time user interface to create and maintain report definitions.
- Oracle Reports Services: This component executes, distributes, and publishes your reports for enterprisewide reporting. Using Oracle Reports Services to deploy your reports results in gains of flexibility, time savings, and processing capacity. It includes the following components:
 - Oracle Reports Bridge that provides functionality for discovering a Reports Server across farms.
 - Oracle Reports run time that runs a report by starting its own in-process server (not to be confused with the default in-process Reports Server), which runs in the same JVM as the `rwr` process.
 - Oracle Reports Client that provides a command-line interface to send a report to a remote Reports Server.
 - Oracle Reports Servlet that translates and delivers information between either a Web Server or a Java EE Container (for example, Oracle WebLogic Server) and the Reports Server, enabling you to run a report dynamically from your Web browser.
 - Oracle Reports Server that provides reporting services to execute, distribute, and publish your reports for enterprisewide reporting. This component processes client requests, including user authentication, scheduling, caching, and report distribution.

For more information about Oracle Reports, see the *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services*.

3.9 Oracle Forms Services

Oracle Forms Services is a middle-tier application framework for deploying complex, transactional forms applications to a network such as an intranet or the Internet.

Developers build Forms applications with Oracle Forms Builder and deploy them with Oracle Forms Services.

With Oracle Forms Services, business application developers can quickly build comprehensive Java client applications that are optimized for the Internet without writing any Java code, and that meet (and exceed) the requirements of professional user communities. These Java client applications are Web-deployed applications available on demand for rapid processing of large amounts of data and rapid completion of complex calculations, analysis, and transactions.

Oracle Forms Services consists of a three-tier architecture:

- The client tier, contains the Web browser, where the application is displayed. In addition to the browser, Java Runtime Environment (JRE) and Java Plug-In (JPI) are required.
- The middle tier is the application server, where application logic and server software are stored.
- The database tier is the database server, where database server software is stored.

For more information, see the following:

- *Oracle Fusion Middleware Forms Services Deployment Guide*
- *Oracle Forms Upgrading Oracle Forms 6i to Oracle Forms 11g*

3.10 Oracle SOA Suite

Oracle SOA Suite is a middleware component of Oracle Fusion Middleware. Oracle SOA Suite enables services to be created, managed, and orchestrated into SOA composite applications. Composite applications enable you to easily assemble multiple technology components into one SOA composite application. Oracle SOA Suite plugs into heterogeneous infrastructures and enables enterprises to incrementally adopt SOA.

You can administer the components of Oracle SOA Suite from Oracle Enterprise Manager Fusion Middleware Control Console.

Software architects and developers can address the complexity of their application and enterprise environments with Oracle SOA Suite. Oracle SOA Suite facilitates the development of enterprise applications as modular business services that can be easily integrated and reused, thus creating a flexible, adaptable enterprise infrastructure.

This section describes the following Oracle SOA Suite components:

- [Adapters](#)
- [Oracle Service Bus](#)
- [Oracle Complex Event Processing](#)
- [Oracle Business Rules](#)
- [Oracle Business Activity Monitoring](#)
- [Oracle B2B](#)
- [Oracle BPEL Process Manager](#)
- [Oracle Service Registry](#)
- [Oracle User Messaging Service](#)
- [Human Workflow](#)
- [Oracle Mediator](#)

3.10.1 Adapters

With the growing need for business process optimization, efficient integration with existing back-end applications has become the key to success. To optimize business processes, you can integrate applications by using Java Connector Architecture (JCA) 1.5 compliant resource adapters. Adapters support a robust, lightweight, highly scalable, and standards-based integration framework, which enables disparate applications to communicate with each other. For example, adapters enables integration of packaged applications, legacy applications, databases, and Web services. Using Oracle JCA adapters insures interoperability by integrating applications that are heterogeneous, provided by different vendors, based on different technologies, and run on different platforms.

Oracle JCA Adapters integrate Oracle Fusion Middleware with transport protocols, data stores, and messaging middleware. These adapters include Oracle JCA Adapter for Files, Oracle JCA Adapter for FTP, Oracle JCA Adapter for JMS, Oracle JCA Adapter for Database, Oracle JCA Adapter for Advanced Queuing, Oracle JCA Adapter for MQ Series, Oracle JCA Adapter for Sockets, and Oracle Adapter for

Oracle Applications. Oracle technology adapters are installed as part of Oracle Fusion Middleware.

3.10.2 Oracle Service Bus

Oracle Service Bus, the backbone of SOA, is a service integration platform that lets you turn complex, brittle, point-to-point integration implementations into loosely coupled services that service consumers can use across multiple domains and between heterogeneous endpoints such as Java and Microsoft.Net. Oracle Service Bus lets you quickly adapt to change, scale, manage risk, and achieve higher value with simple, code-free, configuration-based service integration.

Key features of Oracle Service Bus, which focus on enhanced productivity, performance and scalability, a unified run time, and adaptive connectivity, include:

- message transformation
- service-level agreement (SLA) enforcement
- service monitoring
- quality of service (QoS) control
- support for multiple messaging protocols
- native connectivity with enterprise information systems (EIS)
- built-in multi-level security
- integration with other Oracle products such as Oracle Enterprise Repository, Oracle Service Registry, Oracle Web Services Manager, and Oracle Enterprise Manager

3.10.3 Oracle Complex Event Processing

Databases are best equipped to run queries over finite stored data sets. However, many modern applications require long-running queries over continuous unbounded sets of data. By design, a stored data set is appropriate when significant portions of the data are queried repeatedly and updates are relatively infrequent. In contrast, data streams represent data that is changing constantly, often exclusively through insertions of new elements. It is either unnecessary or impractical to operate on large portions of the data multiple times.

Many types of applications generate data streams as opposed to data sets, including sensor data applications, financial tickers, network performance measuring tools, network monitoring and traffic management applications, and clickstream analysis tools. Managing and processing data for these types of applications involves building data management and querying capabilities with a strong temporal focus.

To address this requirement, Oracle introduces Oracle Complex Event Processing (Oracle CEP), a data management infrastructure that supports streams of structured data records together with stored relations.

Oracle CEP (formerly known as the Oracle WebLogic Server Event Server) is a Java server for the development and deployment of high-performance event driven applications. Oracle CEP supports ultra-high throughput and microsecond latency using JRockit Real Time and provides Oracle CEP Visualizer and Oracle CEP IDE for Eclipse developer tooling for a complete real time end-to-end Java Event-Driven Architecture (EDA) development platform.

You can use Oracle CEP to perform advanced context creation, filtering, correlation and aggregation, and pattern matching on the events of an Oracle Fusion Middleware application.

Oracle CEP provides the following principle features:

- **Data and event source support:** An Oracle CEP event source identifies a producer of data on which Oracle CQL queries operate. Event sources include data feeds such as wire services and stock tickers, sensors such as temperature, motion, or radio frequency identification (RFID) detectors, and other devices. Oracle CEP provides a variety of adapters that connect such real-world event sources to your Oracle CQL queries. Oracle CEP adapters support the following event sources: JMS, HTTP publisher/subscriber, and file.
- **Context creation:** Oracle CEP offers a variety of sliding window operators and views (subqueries) that allow you to define the temporal or semantic context in which filtering, correlation and aggregation, and pattern matching takes place. Oracle CQL provides a variety of sliding windows, including: range-based (time or constant value), tuple-based, and partitioned.
- **Filtering:** Using Oracle CQL, you can specify queries that select any of the attributes of the events offered by event sources. You use such queries to filter the event sources to obtain events of interest. Oracle CQL provides a set of operators, expressions, conditions, and statements for this purpose.
- **Correlation and aggregation:** Using Oracle CQL, you can perform advanced statistical and arithmetic operations on the attributes of the events offered by event sources. Oracle CQL provides: single-row functions that return a single result row for every row of a queried stream or view; aggregate functions that return a single aggregate result based on a group of tuples, rather than on a single tuple; statistical and advanced arithmetic operations for high-performance scientific and technical computing; and statistical and advanced arithmetic operations based on the `java.lang.Math` class. In addition, you can easily define custom single-row and aggregate functions.
- **Pattern matching:** Using the Oracle CQL `MATCH_RECOGNIZE` condition, you can succinctly express complex pattern matching operations for a wide variety of tasks such as algorithmic trading, double-bottom detection, non-event detection, and so on.
- **Complex event sink support:** An Oracle CQL event sink identifies a consumer of Oracle CQL query results. That is, a consumer of notable events that Oracle CQL queries have extracted from event sources by executing filtering, correlation and aggregation, and pattern matching within various contexts. Typically, noteworthy events are fewer in number (and much higher in value) than the events offered by event sources. Oracle CEP adapters support the following event sinks: JMS, HTTP publisher/subscriber, file, and event beans.
- **Development and run time tools:** Oracle CEP provides Oracle CEP IDE for Eclipse, a set of plug-ins for the Eclipse IDE designed to help develop, deploy, and debug Oracle CEP applications. Oracle CEP also provides an advanced run time administration console called Oracle CEP Visualizer. Using Oracle CEP Visualizer, you can manage, tune, and monitor Oracle CEP server domains and the Oracle CEP applications you deploy to them all from a browser. Oracle CEP Visualizer provides a variety of sophisticated run time administration tools, including support for Oracle CQL and EPL rule maintenance and creation.

For more information, see "Introduction to Oracle Complex Event Processing" in the *Oracle CEP Getting Started*.

3.10.4 Oracle Business Rules

Oracle Business Rules enables dynamic decisions at run time allowing you to automate policies, computations, and reasoning while separating rule logic from underlying application code. This allows more agile rule maintenance and empowers business analysts with the ability to modify rule logic without programmer assistance and without interrupting business processes.

Business rules are statements that describe business policies or describe key business decisions.

Oracle Business Rules supports Decision Tables. A Decision Table is an alternative business rule format that is more compact and intuitive when many rules are needed to analyze many combinations of property values. You can use a Decision Table to create a set of rules that covers all combinations or where no two combinations conflict.

Oracle Fusion Middleware provides support for business rules as follows:

- In a Decision component. A Decision component is a mechanism for publishing business rules as a reusable service that can be invoked from multiple business processes. You can use business rules to integrate a SOA composite application with Oracle Business Rules. A business rule service component is also called a Decision component. You can add business rules as part of an SCA application or as part of a Business Process Execution Language (BPEL) process.
- In a rule-enabled non-SOA Java EE application with Oracle Business Rules. Oracle Business Rules Rules Engine (Rules Engine) is available as a library for use in a Java EE application (non-SOA). The Rules Engine efficiently applies rules to facts and defines and processes rules.

For more information, see the following:

- *Oracle Fusion Middleware User's Guide for Oracle Business Rules*
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware Language Reference Guide for Oracle Business Rules*

3.10.5 Oracle Business Activity Monitoring

Oracle Business Activity Monitoring helps businesses improve decision-making processes by aggregating service metrics and delivering actionable information on critical business services. Oracle Business Activity Monitoring provides information to business users by using visual dashboards and alerts.

Oracle Business Activity Monitoring also gives users the ability to change the business processes and take corrective action as the business environment changes.

Oracle Business Activity Monitoring is a complete solution for building real-time operational dashboards and monitoring and alerting applications over the Web.

For more information see the *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring*.

3.10.6 Oracle B2B

Oracle B2B (Business to Business) is an e-commerce gateway that enables the secure and reliable exchange of business documents between an enterprise and its trading partners. Oracle B2B is used for typical business-to-business e-commerce, such as buying and selling products and services over the Internet, and also for Health Level 7 support, to enable healthcare systems to communicate with each other. Oracle B2B

supports business-to-business document standards, packaging, transports, messaging services, and trading partner and agreement management. The Oracle SOA Suite platform, in which Oracle B2B is a binding component, enables the implementation of end-to-end e-commerce business processes.

For more information see "Introduction to Oracle B2B" in the *Oracle Fusion Middleware User's Guide for Oracle B2B*.

3.10.7 Oracle BPEL Process Manager

Oracle BPEL Process Manager provides the standard for assembling a set of discrete services into an end-to-end process flow, radically reducing the cost and complexity of process integration initiatives. Oracle BPEL Process Manager enables you to orchestrate synchronous and asynchronous services into end-to-end BPEL process flows.

You integrate BPEL processes with external services (known as partner links). You also integrate technology adapters and services, such as human tasks, transformations, notifications, and business rules within the process.

Oracle BPEL Process Manager is included as a service component in Oracle SOA Suite. Oracle SOA Suite is an essential layer of Oracle Fusion Middleware. It provides a complete set of components for designing, deploying, and managing composite applications. Oracle SOA Suite enables services to be created, managed, and orchestrated into composite applications and business processes.

For more information about Oracle BPEL Process Manager, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

3.10.8 Oracle Service Registry

Oracle Service Registry is an easy-to-use, standards-based mechanism for publishing and discovering Web services and related resources like XML Schemas or Extensible Stylesheet Language Transformations (XSLT).

Oracle Service Registry is a fully compliant implementation of Universal Description, Discovery and Integration (UDDI), and is a key component of a SOA. Oracle Service Registry can be deployed in almost any Java environment and works with all popular database systems. In addition, the registry has been designed specifically for enterprise deployment and includes many advanced features that make it easy to configure, deploy, manage and secure. Oracle Service Registry is also easy to customize to support different enterprise user communities.

Oracle Service Registry extends the core UDDI standard with unique functionality designed for enterprise applications, including:

- **Advanced Security**
Allows for defining granular access control for registered components. Component publisher can specify find, get, modify and delete access permissions for every published object.
- **Data Accuracy and Quality**
Enforcement mechanisms ensures that component registrations are accurate and current. Oracle Service Registry clearly defines responsibility for every registered component. It offers component promotion & approval mechanisms for promoting components between development, quality assurance, and production environments
- **Subscription and Notification**

Provides automatic notification to registry users about changes to components on which they depend.

- Selective Replication

Allows multiple registries to perform automated propagation (for example, between internal and external registries).

- Advanced Taxonomy Management

Allows the enforcement of well-defined taxonomies.

- Powerful Management

Allows granular control, logging, and auditing of the publishing and discovery processes.

- Performance and Scalability

Provides maximum performance and scalability by efficient implementation of Web services stack and database algorithms and by supporting of a load balancing and clustering mechanism.

Oracle Service Registry is a platform-independent solution that can easily be deployed in a wide variety of settings. Oracle Service Registry also integrates with LDAP directories, including Oracle Internet Directory and Microsoft ActiveDirectory.

For more information, see the *Oracle Fusion Middleware Service Registry Guide*.

3.10.9 Oracle User Messaging Service

Oracle User Messaging Service supports a simple and reliable way of integrating multi channel messaging into. Oracle User Messaging Service supports both Java APIs as well as Web-Services for integration. The channels supported include SMS, E-Mail, and Instant Messaging as well as Voice messages. Oracle User Messaging Service also supports intelligent messaging whereby the final destination of a message is determined by a user's preferences.

For more information, see the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

3.10.10 Human Workflow

Human workflow enables you to model a flow that describes the tasks for users or groups to perform as part of an end-to-end business process flow. Many end-to-end business processes require human interactions with the process. For example, humans may be needed for approvals, exception management, or performing activities required to advance the business process. Human workflow provides the following features:

- Human interactions with processes, including assignment and routing of tasks to the correct users or groups
- Deadlines, escalations, notifications, and other features required for ensuring the timely performance of a task (human task activity)
- Presentation of tasks to end users through a variety of mechanisms, including a worklist application (Oracle BPM Worklist)
- Organization, filtering, prioritization, and other features required for end users to productively perform their tasks

- Reports, reassignments, load balancing, and other features required by supervisors and business owners to manage the performance of tasks

For more information about human workflow, see the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

3.10.11 Oracle Mediator

Oracle Mediator provides a lightweight framework to mediate between various components within a composite application. Mediator converts data to facilitate communication between different interfaces exposed by different components, which are wired together to build a SOA composite application. For example, a Mediator can accept data contained in a text file from an application or service, transform it to a format appropriate for updating a database that serves as a customer repository, and then route and deliver the data to that database.

Oracle Mediator works as a service component in the SOA Suite to facilitate integration between events and services, where service invocations and events can be mixed and matched. You can use a Mediator component to consume a business event or to receive a service invocation. A Mediator component can evaluate routing rules, perform transformations, validate, and either invoke another service or raise another business event. You can use a Mediator component to handle returned responses, callbacks, faults, and timeouts.

For information about Oracle Mediator, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

3.11 Development Tools

This section describes Oracle Fusion Middleware support for building Fusion Web applications with Oracle ADF and the integrated development environment, Oracle JDeveloper. A Fusion Web application is any database-centric application that is targeted to run in the Oracle Fusion Middleware environment.

This section includes the following topics:

- [Oracle JDeveloper](#)
- [Oracle Application Development Framework](#)
- [Oracle TopLink](#)
- [Oracle Enterprise Pack for Eclipse](#)
- [Oracle Help](#)

3.11.1 Oracle JDeveloper

Oracle JDeveloper is an integrated development environment (IDE) for building SOA applications using the latest standards for Java, XML, Web services, and SQL. It supports the complete development lifecycle with integrated features for modeling, coding, debugging, testing, profiling, tuning, and deploying applications. Oracle JDeveloper is the main development platform for the Oracle Fusion Middleware suite of products. It is a cross-platform IDE that runs on Windows, Linux, Mac OS X, and other UNIX-based systems.

Oracle JDeveloper provides a visual and declarative development approach and works together with the Oracle ADF to simplify development.

Key features of Oracle JDeveloper include:

- A consistent development environment that can be used for various technology stacks including Java, SOA, Oracle WebCenter, SQL and PL/SQL, HTML, and JavaScript.
- XML-based application development.
- A full development and modeling environment for building database objects and stored procedures.
- A wide range of application deployment options, including Integrated Oracle WebLogic Server, an integrated run time service for running and testing applications before deploying to a production environment.
- Extension capabilities that enable customization of the IDE based on development needs and add additional functionality.

For more information about Oracle JDeveloper, see the Oracle JDeveloper online documentation.

3.11.2 Oracle Application Development Framework

The Oracle ADF is an end-to-end application framework that builds on Java EE standards and open-source technologies to simplify and accelerate implementing service-oriented applications. Oracle ADF is suitable for enterprise developers who want to create applications that search, display, create, modify, and validate data using Web, wireless, desktop, or Web services interfaces. Used in tandem, Oracle JDeveloper 11g and Oracle ADF provide an environment that covers the full development lifecycle from design to deployment, with drag-and-drop data binding, visual UI design, and team development features built in.

The core module in the framework is Oracle ADF Model, a declarative data binding facility that implements the latest specification. This specification provides an API for accessing declarative data binding metadata. The Oracle ADF Model layer enables a unified approach to bind any user interface to any business service, without the need to write code. The other modules that make up a Fusion Web application technology stack are:

- Oracle ADF Business Components, the business layer that simplifies building business services.
- Oracle ADF Faces rich client, the view layer that offers a library of AJAX-enabled UI components for Web applications built with JavaServer Faces (JSF).
- Oracle ADF Controller, the controller layer that integrates JSF with Oracle ADF Model. The ADF Controller extends the standard JSF controller by providing additional functionality, such as reusable task flows that pass control not only between JSF pages, but also between other activities, for instance method calls or other task flows.

In addition to ADF Faces, Oracle ADF also supports using the Swing, JSP, and standard JSF view technologies, as well as Microsoft Excel as a view layer for your application. For more information about these technologies, see [Section 3.11.2.6, "Oracle ADF Support for Other View Technologies."](#)

Oracle also offers the following resources for learning about Oracle ADF:

- Cue Cards in JDeveloper: JDeveloper cue cards provide step-by-step support for the application development process using Oracle ADF. They are designed to be used either with the included examples and a sample schema, or with your own data.
- Tutorials on the Oracle Technology Network.

For more information, see the Oracle JDeveloper documentation page on the Oracle Technology Network:

<http://www.oracle.com/technology/documentation/jdev.html>

3.11.2.1 ADF Model Layer

In the model layer, Oracle ADF Model implements the JSR-227 service abstraction called the data control. Data controls abstract the implementation technology of a business service by using standard metadata interfaces to describe the service's operations and data collections, including information about the properties, methods, and types involved. In Oracle JDeveloper, developers can view that information as icons that they can easily drag and drop onto a page. When the developer drags the representation of the service onto the page, Oracle JDeveloper automatically creates the bindings from the page to the services. At run time, the ADF Model layer reads the information describing the application's data controls and data bindings from appropriate XML files and implements the two-way connection between the user interface and the application's business service.

Oracle ADF provides ready-to-use data control implementations for the most common business service technologies. Using Oracle JDeveloper and Oracle ADF together provides a declarative, drag-and-drop data binding experience for building user interfaces. Along with support for ADF Business Components application modules, the ADF Model layer also provides support for the following service technologies:

- Enterprise JavaBeans (EJB) session beans and JPA Persistence API entities
- JavaBeans
- Web services
- XML
- CSV files

For more information about Oracle ADF Model, see "Using Oracle ADF Model in a Fusion Web Application" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

3.11.2.2 ADF Business Components

ADF Business Components are prebuilt application objects that accelerate the delivery and maintenance high-performance, highly functional, database-centric services. When building service-oriented Java EE applications, developers implement the core business logic as one or more business services. These back-end services provide clients with a way to query, insert, update, and delete business data as required while enforcing appropriate business rules. ADF Business Components provides a ready-to-use implementation of Java EE design patterns and best practices.

Oracle ADF Business Components provides the following key components to simplify building database-centric business services:

- Entity object

An entity object represents a row in a database table and simplifies modifying its data by handling all data manipulation language (DML) operations. It can encapsulate business logic to ensure that business rules are consistently enforced. Developers can associate an entity object with others to reflect relationships in the underlying database schema to create a layer of business domain objects to reuse in multiple applications.
- View object

A view object represents a SQL query and simplifies working with its results. Developers use the SQL language to join, project, filter, sort, and aggregate data into the shape required by the end-user task represented in the user interface. This includes the ability to link a view object with other entity objects to create master-detail hierarchies of any complexity. When end users modify data in the user interface, view objects collaborate with entity objects to consistently validate and save the changes.

- Application module

An application module is the transactional component that UI clients use to work with application data. It defines an updatable data model and top-level procedures and functions (called service methods) related to a logical unit of work related to an end-user task.

For more information about Oracle ADF Business Components, see "Getting Started with ADF Business Components" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

3.11.2.3 ADF Controller

In the controller layer, where handling page flow of the Web applications is a key concern, ADF Controller provides an enhanced navigation and state management model on top of JSF. JDeveloper supports declarative creation of task flows that can manage application control between different types of activities, such as pages, methods on managed beans, declarative case statements, or calls to other task flows.

For more information about Oracle ADF Controller, see "Getting Started with ADF Task Flows" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

3.11.2.4 ADF Faces Rich Client

ADF Faces rich client (ADF Faces for short), is a set of standard JSF components that includes built-in AJAX functionality. AJAX is a combination of asynchronous JavaScript, dynamic HTML (DHTML), XML, and an XMLHttpRequest communication channel. This combination allows requests to be made to the server without fully re-rendering the page. While AJAX allows client-like applications to use standard Internet technologies, JSF provides server-side control, which reduces the dependency on an abundance of JavaScript often found in typical AJAX applications.

ADF Faces provides over 100 components, including hierarchical data tables, tree menus, in-page dialogs, accordions, dividers, and sortable tables. ADF Faces also provides ADF Data Visualization components, which are Flash- and SVG-enabled components capable of rendering dynamic charts, graphs, gauges, and other graphics that can provide a real-time view of underlying data. Each component also supports customization and skinning, along with internationalization and accessibility.

To achieve these front-end capabilities, ADF Faces components use a rendering kit that handles display of the component and also provides the JavaScript objects needed for the functionality. This built-in support enables developers to build applications without needing extensive knowledge of the individual technologies on the front or back end.

For more information about ADF Faces, including the architecture and detailed information about each of the components, see "Introduction to ADF Faces Rich Client" in the *Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework*.

3.11.2.5 Oracle ADF Security

Oracle ADF Security (ADF Security) is a framework that provides a security implementation that is based on JAAS. The Oracle ADF implementation of JAAS is permission-based. Oracle JDeveloper supports making permission grants to enable fine-grained security for Oracle ADF resources of the Fusion Web application.

For more information about Oracle ADF Security, see "Enabling ADF Security in a Fusion Web Application" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

3.11.2.6 Oracle ADF Support for Other View Technologies

Along with ADF Faces, Oracle ADF also supports the following view technologies:

- Apache MyFaces Trinidad: This is the open source code donation from Oracle to the Apache Software Foundation. ADF Faces components are based on these Trinidad components.
- Java Swing and ADF Swing: ADF Swing is the development environment for building Java Swing applications that use the ADF Model layer.
- ADF Mobile: This is a standards-based framework for building mobile applications built on the component model of JSF.
- ADF Desktop Integration: Integration is supported through Microsoft Excel to create spreadsheets that are bound to data using the same binding principles as do other view technologies.

3.11.2.7 Oracle ADF Swing

Oracle Application Development Framework Swing (Oracle ADF Swing) consists of Java classes and an API to handle the binding of Swing components to business services and to simplify the task of creating databound forms in a desktop application. JDeveloper and Oracle ADF Swing provide comprehensive support for developing desktop applications, which includes features such as form creation wizards, form layout editor, and drag-and-drop data binding with business components.

For more information about Oracle ADF Swing, see the Oracle JDeveloper online documentation available from within the Oracle JDeveloper application.

3.11.2.8 Oracle ADF Mobile

Oracle Application Development Framework Mobile (Oracle ADF Mobile) is a standards-based framework that enables the rapid development of enterprise mobile applications. The current version of Oracle ADF Mobile extends Oracle ADF to browsers running on mobile devices. Oracle ADF Mobile is built upon the component model of JSF, which enables development of applications for mobile browsers.

Oracle ADF Mobile's mobile-specific extensions to JSF enables development of mobile applications using the same methodologies for developing JSF applications for the desktop. Oracle ADF Mobile application development uses only mobile JSF pages that consist of Apache MyFaces Trinidad components. Currently, Oracle ADF Mobile supports more than 60 Trinidad components for a component set that renders appropriately to the screens of mobile devices. For more information, see the Apache MyFaces Trinidad home page at <http://myfaces.apache.org/trinidad/index.html>.

For more information about Oracle ADF Mobile, see the *Oracle Fusion Middleware Mobile Browser Client Developer's Guide for Oracle Application Development Framework*.

3.11.2.9 Oracle ADF Desktop Integration

Oracle Application Development Framework Desktop Integration (Oracle ADF Desktop Integration) provides tools and components to integrate Microsoft Excel workbooks with Web applications built using Oracle ADF. At run time, Excel workbooks configured by the Oracle ADF Desktop Integration module can invoke Oracle ADF components to manage data retrieved from Web applications.

For more information, see the *Oracle Fusion Middleware Desktop Integration Developer's Guide for Oracle Application Development Framework*.

3.11.3 Oracle TopLink

Oracle TopLink is an advanced, object-persistence and object-transformation framework that provides development tools and run time capabilities that reduce development and maintenance efforts, and increase enterprise application functionality.

Oracle TopLink builds high-performance applications that store persistent object-oriented data in a relational database. It successfully transforms object-oriented data into either relational data, or Extensible Markup Language (XML) elements.

Oracle TopLink is used to design, implement, deploy, and optimize an advanced, object-persistence and object-transformation layer that supports a variety of data sources and formats, including the following:

- Relational: For transactional persistence of Java objects to a relational database accessed using Java Database Connectivity (JDBC) drivers.
- Object-Relational Data Type: For transactional persistence of Java objects to special purpose structured data source representations optimized for storage in object-relational data type databases such as Oracle Database.
- Enterprise information system (EIS): For transactional persistence of Java objects to a non-relational data source accessed using a JCA adapter, and any supported EIS record type, including indexed, mapped, or XML.
- XML: For non-transactional, nonpersistent (in-memory) conversion between Java objects and XML Schema Document (XSD)-based XML documents using Java Architecture for XML Binding (JAXB).

Oracle TopLink supports EJB 3.0 in Java EE and Java SE environments including integration with a variety of application servers, such as Oracle WebLogic Server and Oracle Containers for J2EE (OC4J) 10g (10.1.3.4.x). Oracle TopLink also includes support for EJB 2.*n* container-managed persistence (CMP) in OC4J.

Oracle TopLink provides a suite of development tools, such as Oracle JDeveloper TopLink Editor, Eclipse Dali, and Oracle TopLink Workbench, to capture and define object-to-data source and object-to-data representation mappings in a metadata format.

Some of the key features of Oracle TopLink are the following:

- Metadata-based architecture: Plain Old Java Objects (POJO), Container-Managed Persistence (CMP), as well as Java Persistence API (JPA), Java Architecture for XML Binding (JAXB), Service Data Objects (SDO), and Web services provided by EclipseLink.
- Advanced mapping support.
- Optimization and performance tuning.

- Object caching support, including cluster integration for some application servers (such as Oracle Application Server).
- Query capability including TopLink Expressions framework, Java Persistence Query Language (JP QL), Enterprise JavaBeans Query Language (EJB QL), and native SQL.
- Just-in-time reading.
- Object-level transaction support and integration with popular application servers and databases.
- Optimistic and pessimistic locking options and locking policies.
- Visual design tools.

For more information about Oracle TopLink, see "Introduction to TopLink" in the *Oracle Fusion Middleware Developer's Guide for Oracle TopLink*.

3.11.4 Oracle Enterprise Pack for Eclipse

Oracle Enterprise Pack for Eclipse (OEPE) is a set of plug-ins designed to support Java EE development, especially if the following technologies are present:

- Eclipse IDE
- Oracle WebLogic Server
- Oracle Database

Key features of OEPE include:

- Oracle WebLogic Server support: OEPE provides features to develop, deploy and debug Oracle WebLogic Server applications:
 - Faster application development with virtual enterprise archive (EAR) technology.
 - Remote deployment of applications.
 - Deployment descriptor editing using the graphical design view.
 - Support for the Oracle WebLogic Server shared library.
 - Support for XMLBeans.
 - Support for EJBGen.
 - Support for Web services: OEPE enables building of enterprise-class Web services that employ standard Web service technologies, such as XML, SOAP, and WSDL. OEPE simplifies Web service development by allowing developers to focus on application logic, rather than the complex implementation details traditionally required by these technologies.
- Oracle Database support: OEPE provides the means to easily connect to, create, explore, and query Oracle databases. Support includes database visualization through the Data Source Explorer view and data descriptive language (DDL) generation.
- JSF facelets tools: Facelets enable development of JSF pages using XHTML instead of JSP.
- Support for object-relational mappings: OEPE enables creation of a persistence layer that uses EJB 3.0 JPA, and provides the means to do the following:
 - Configure the persistence provider for JPA projects

- Generate JPA entities
- Annotate classes
- Use entity diagram editors
- Use a SQL schema viewer
- Use the Spring framework
- Web page editors.

For more information about OEPE, see:

http://download.oracle.com/docs/cd/E14545_01/help/oracle.eclipse.tools.common.doc/html/index.html

3.11.5 Oracle Help

Oracle Help provides the means for developing and displaying HTML-based help systems for Java applications and Web applications. Authors can create a single help system that can be displayed - without modification - both in a Java environment, using Oracle Help for Java, and in a Web environment, using Oracle Help for the Web. Oracle Help for the Web is available in two formats: Oracle Help for the Web - UIX and Oracle Help for the Web - Rich Client.

Oracle Help for Java (OHJ) is a set of Java components, a Java API, and a file formats specification for developing and displaying HTML-based help content in a Java environment. OHJ is designed primarily for displaying help for Java applications, although it can also be implemented as a standalone document viewer for use in a Java environment.

Oracle Help for the Web - UIX (OHW-UIX) is a Java servlet and a file formats specification for developing and delivering HTML-based help content in a Web environment. OHW-UIX can be used to provide context-sensitive help for Web applications or as a means for processing and displaying structured views of independent HTML content on the Web.

Oracle Help for the Web - Rich Client (OHW-RC), similar to OHW-UIX, delivers HTML-based Help content in a Web environment. It uses Oracle Application Development Framework (Oracle ADF), which is based on the JavaServer Faces (JSF) technology, to build a user interface that follows Oracle's Browser Look And Feel Plus (BLAF+) guidelines.

For more information about Oracle Help, see "Introduction to Oracle Help" in the *Oracle Fusion Middleware Developer's Guide for Oracle Help*.

3.12 Oracle HTTP Server

Oracle HTTP Server (OHS) is the underlying deployment platform for all programming languages and technologies that Oracle Fusion Middleware supports. It provides a Web listener and the framework for hosting static and dynamic pages and applications over the Web. Based on the proven technology of the Apache 2.2.10 infrastructure, OHS includes significant enhancements that facilitate load balancing, administration, and configuration. It also includes a number of enhanced modules, or mods, which are extensions to the HTTP server that extend its functionality for other enterprise applications and services.

OHS is, and includes modules developed specifically by Oracle. The features of single sign-on, clustered deployment, and high availability enhance the operation of the Oracle HTTP Server.

OHS allows developers to program their sites in a variety of languages and technologies, such as Java, Perl, C, C++, PHP, and PL/SQL. Additionally, it can serve as either a forward or reverse proxy server.

For more information about OHS see the *Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server*.

3.13 Oracle Web Cache

This section describes Oracle Web Cache. It includes the following two topics:

- [Reverse Proxying](#)
- [Request Filtering](#)

3.13.1 Reverse Proxying

Oracle Web Cache is a content-aware server accelerator, or reverse proxy, for the Web tier that improves the performance, scalability, and availability of Web sites that run on any Web server or application server, such as Oracle HTTP Server and Oracle WebLogic Server.

Oracle Web Cache is the primary caching mechanism provided with Oracle Fusion Middleware. Caching improves the performance, scalability, and availability of Web sites that run on Oracle Fusion Middleware by storing frequently accessed URLs in memory.

By storing frequently accessed URLs in memory, Oracle Web Cache eliminates the need to repeatedly process requests for those URLs on the application Web server and database tiers. Unlike legacy proxies that handle only static objects, Oracle Web Cache caches both static and dynamically generated content from one or more application Web servers. Because Oracle Web Cache can cache more content than legacy proxies, it provides optimal performance by greatly reducing the load on application Web server and database tiers. As an external cache, Oracle Web Cache is also an order of magnitude faster than object caches that run within the application tier.

Because Web Cache is fully compliant with HTTP 1.0 and 1.1 specifications, it can accelerate Web sites that are hosted by any standard Web servers, such as Apache Tomcat and Microsoft IIS. In Oracle Fusion Middleware, Oracle Web Cache resides in front of one or more instances of Oracle HTTP Server. Responses to browser based HTTP requests are directed to the Oracle HTTP Server instance and transmitted through Oracle Web Cache. The Oracle Web Cache instance can handle any Web content transmitted with the standard HTTP protocol.

For more information about reverse proxying see the *Oracle Fusion Middleware Administrator's Guide for Oracle Web Cache*.

3.13.2 Request Filtering

Oracle Web Cache provides request filters to filter incoming HTTP or HTTPS requests to configured sites on the origin server.

Request filtering aids administrators in controlling access to Web sites and providing the following:

- The planting of malicious code within the Web site that, when executed, steals a user's identity or personal information
- Attacks that try to exploit software vulnerabilities on the site that enable the attacker to execute arbitrary code on the application server

- Attacks that try to render a Web site unusable by bombarding it with extremely high volumes of bogus requests that effectively consume the application server resources or bandwidth, thereby preventing access for other users

In addition, request filtering controls which clients and requests are allowed to access to a Web site or certain parts of a Web site.

For more information about request filtering, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Web Cache*.

3.14 Oracle Web Services

A Web Service is a set of functions packaged into a single application that is available to other systems on a network. The network can be a corporate intranet or the Internet. Because Web Services rely on basic, standard technologies which most systems provide, they are an excellent means for connecting distributed systems together. They can be shared by and used as a component of distributed Web-based applications. Other systems, such as customer relationship management systems, order-processing systems, and other existing back-end applications, can call a Web Service function to request data or perform an operation.

Web Services feature the following capabilities:

- Interoperability among distributed applications that span diverse hardware and software platforms
- Easy, widespread access to applications through firewalls using Web protocols
- A cross-platform, an XML cross-language data model that facilitates developing heterogeneous distributed applications

In Oracle Fusion Middleware 11g, there are two categories of Web services to support the development, security, and administration of the following types of Web services:

- SOA, ADF, and WebCenter services
- WebLogic Web services (Java EE Web services)

For an overview of Web services in Oracle Fusion Middleware 11g, a description of the Web services and clients supported in each category, and a roadmap for implementing Oracle Fusion Middleware Web services, see *Oracle Fusion Middleware Introducing Web Services*.

3.15 Oracle Delegated Administration Services

Oracle Delegated Administration Services is an important feature of the Oracle Identity Management infrastructure. It enables you to store all data for users, groups, and services in a central directory, while distributing the administration of that data to various administrators and end users. It does this in a way that respects the various security requirements in your environment.

3.16 Oracle Enterprise Content Management Suite

Oracle Enterprise Content Management Suite is an integrated suite of products designed for managing content. It features the following components:

- [Oracle Universal Content Management \(UCM\)](#)
- [Oracle Universal Records Management \(URM\)](#)
- [Oracle Inbound Refinery \(IBR\)](#)

- [Oracle Information Rights Management \(IRM\)](#)
- [Oracle Imaging and Process Management \(I/PM\)](#)

These components can be deployed as applications to an Oracle WebLogic Server domain.

3.16.1 Oracle Universal Content Management (UCM)

Oracle Universal Content Management (Oracle UCM) provides a unified application for several different kinds of content management.

Oracle Universal Content Management (UCM) is an enterprise content management platform that enables you to leverage document management, Web content management, digital asset management, and records retention functionality to build and complement your business applications. Building a strategic enterprise content management infrastructure for content and applications helps you to reduce costs, easily share content across the enterprise, minimize risk, automate expensive, time-intensive and manual processes, and consolidate multiple Web sites onto a single platform for centralized management. Through user-friendly interfaces, roles-based authentication and security models, Oracle Universal Content Management empowers users throughout the enterprise to view, collaborate on or retire content, ensuring that all accessible distributed or published information is secure, accurate and up-to-date.

Oracle UCM is now deployed in WebLogic Server (WLS), resulting in many significant changes from previous releases of UCM.

3.16.2 Oracle Universal Records Management (URM)

Oracle Universal Records Management (Oracle URM) effectively manages content items on a retention schedule, which determines the life cycle of that content item.

The focus of records management tends to be the preservation of content for historical, legal, or archival purposes while also performing retention management functions. The focus of retention management tends to be the scheduled elimination of content in which the costs of retaining content outweighs the value of keeping it.

Oracle URM combines both record and retention management into one software system. You can use Oracle URM to track and to preserve content as needed, or to dispose of content when it is no longer required.

3.16.3 Oracle Inbound Refinery (IBR)

Oracle Inbound Refinery (Oracle IBR) is a conversion server that manages file conversions for electronic assets such as documents, digital images, and motion video. In addition to conversion, Oracle IBR provides thumbnail functionality for documents and images, story boarding for video, and the ability to extract and use EXIF data from digital images and XMP data from electronic files generated from programs such as Adobe Photoshop and Adobe Illustrator. You can use Oracle IBR to convert content items stored in Oracle Content Server.

3.16.4 Oracle Information Rights Management (IRM)

Oracle Information Rights Management (Oracle IRM) distributes rights management between centralized servers and desktop agents.

Oracle IRM enables documents or emails to be automatically or manually sealed at any stage in their lifecycle, using sealing tools integrated into the Microsoft Windows desktop, authoring applications, email clients, and content management and

collaborative repositories. Sealing wraps documents and emails within a layer of strong encryption and digital signatures, together with indelible links back to network-hosted servers (operated by the organization to which the information belongs) that store the decryption keys and associated access rights.

Sealed documents and emails can be distributed by any existing means, including email, the Web, and file sharing.

For more information about Oracle IRM refer to the *Oracle Information Rights Management Server Administrator's Guide for Microsoft Windows (32-Bit)*.

3.16.5 Oracle Imaging and Process Management (I/PM)

Oracle Imaging and Process Management (Oracle I/PM) provides organizations with a scalable solution upon which to develop process-oriented imaging applications and image-enablement solutions for enterprise applications. Oracle I/PM enables image capture via Oracle Document Capture and Oracle Distributed Document Capture, annotation and markup of images, BPEL support for routing and approval automation, and support for high-volume applications for billions of items. With Oracle I/PM, organizations can quickly integrate their content and processes directly with Oracle enterprise applications, such as Oracle E-Business Suite, PeopleSoft Enterprise, and JD Edwards EnterpriseOne. Users benefit by having a single source for all transaction-based content, eliminating the need for double entry.

For more information about Oracle I/PM refer to the *Oracle Fusion Middleware Administrator's Guide for Oracle Imaging and Process Management*.

3.17 Oracle Data Integrator

Oracle Data Integrator provides a fully unified solution for building, deploying, and managing complex data warehouses or as part of data-centric architectures in a SOA or business intelligence environment. In addition, it combines all the elements of data integration—data movement, data synchronization, data quality, data management, and data services—to ensure that information is timely, accurate, and consistent across complex systems.

Oracle Data Integrator (ODI) features an active integration platform that includes all styles of data integration: data-based, event-based and service-based. ODI unifies silos of integration by transforming large volumes of data efficiently, processing events in real time through its advanced Changed Data Capture (CDC) capability, and providing data services to the Oracle SOA Suite. It also provides robust data integrity control features, assuring the consistency and correctness of data. With powerful core differentiators - heterogeneous E-LT, Declarative Design and Knowledge Modules - Oracle Data Integrator meets the performance, flexibility, productivity, modularity and hot-pluggability requirements of an integration platform.

For more information about Oracle Data Integrator refer to the following:

- *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*

3.17.1 Oracle Data Quality Products

Oracle Data Profiling and Oracle Data Quality for Data Integrator (also referred to as Oracle Data Quality Products) extend the inline Data Quality features of Oracle Data Integrator to provide more advanced data governance capabilities.

A complete Data Quality system includes data profiling, integrity and quality:

- Profiling

Profiling makes possible data investigation and quality assessment. It allows business users to get a clear picture of their data quality challenges, to monitor and track the quality of their data over time. Profiling is handled by Oracle Data Profiling. It allows business users to assess the quality of their data through metrics, to discover or infer rules based on this data, and finally to monitor over time the evolution of the data quality.

- Integrity Control

Integrity control is essential in ensuring the overall consistency of the data in your information system's applications. Application data is not always valid for the constraints and declarative rules imposed by the information system. You may, for instance, find orders with no customer, or order lines with no product, and so forth. Oracle Data Integrator provides built-in working environment to detect these constraint violation and store them for recycling or reporting purposes. Static and Flow checks in Oracle Data Integrator are integrity checks.

- Quality

Quality includes integrity and extends to more complex quality processing. A rule-based engine apply data quality standards as part of an integration process to cleanse, standardize, enrich, match and de-duplicate any type of data, including names and addresses. Oracle Data Quality for Data Integrator places data quality as well as name and address cleansing at the heart of the enterprise integration strategy.

For more information about Oracle Data Quality products refer to following:

- "Working with Oracle Data Quality Products" in the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*

Oracle Fusion Middleware Administration and Operations

This chapter describes Oracle Fusion Middleware administration. It includes the following sections:

- [Section 4.1, "Enterprise Deployment"](#)
- [Section 4.2, "Management"](#)
- [Section 4.3, "Scalability"](#)
- [Section 4.4, "High Availability"](#)
- [Section 4.5, "Load Balancing"](#)
- [Section 4.6, "Diagnostic Data"](#)
- [Section 4.7, "Cloning"](#)
- [Section 4.8, "Oracle RAC"](#)
- [Section 4.9, "Backup and Recovery"](#)

4.1 Enterprise Deployment

An enterprise deployment is an Oracle best practices blueprint based on proven Oracle high-availability technologies and recommendations for Oracle Fusion Middleware. The high-availability best practices include all Oracle products across the entire technology stack—Oracle Database, Oracle Fusion Middleware, Oracle Applications, Oracle Collaboration Suite, and Oracle Grid Control.

An Oracle Fusion Middleware enterprise deployment:

- Considers various business service level agreements (SLA) to make high-availability best practices as widely applicable as possible
- Takes advantage of database grid servers and storage grids with low-cost storage to provide highly resilient, lower cost infrastructure
- Uses results from extensive performance impact studies for different configurations to ensure that the high-availability architecture is optimally configured to perform and scale to business needs
- Enables control over the length of time to recover from an outage and the amount of acceptable data loss from a natural disaster
- Evolves with each Oracle release and is completely independent of hardware and operating system

For more information on enterprise deployment see the following guides:

- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle WebCenter*
- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*
- Oracle Fusion Middleware High Availability Guide

For more information on high-availability practices, visit:

<http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm>

4.2 Management

You manage your Oracle Fusion Middleware environment using one of the following management tools:

- [Fusion Middleware Control](#)
- [Oracle WebLogic Server Administration Console](#)
- [Oracle WebLogic Scripting Tool](#)
- [Oracle Process Manager and Notification Server](#)
- [Oracle Enterprise Manager Grid Control](#)

4.2.1 Fusion Middleware Control

Oracle Enterprise Manager Fusion Middleware Control (Fusion Middleware Control) is a Web browser-based, graphical user interface that you can use to monitor and administer a farm or cluster.

A farm is a collection of components managed by Fusion Middleware Control. It can contain Oracle WebLogic Server domains, one Administration Server, one or more Managed Servers, clusters, and the Oracle Fusion Middleware components that are installed, configured, and running in the domain.

Fusion Middleware Control organizes a wide variety of performance data and administrative functions into distinct, Web-based home pages for the farm, cluster, domain, servers, components, and applications. The Fusion Middleware Control home pages make it easy to locate the most important monitoring data and the most commonly used administrative functions all from your Web browser.

Fusion Middleware Control provides direct access to Oracle WebLogic Server Administration Console. The Web pages in the Fusion Middleware Control interface contain links that take enable you to access the Administration Console. For example, on the Domain home page, if you go to the Summary area there is a link that takes you to the Administration Console in order to configure and manage the domain.

For more information, see "Getting Started Using Oracle Enterprise Manager Fusion Middleware Control" in the *Oracle Fusion Middleware Administrator's Guide*.

4.2.2 Oracle WebLogic Server Administration Console

Oracle WebLogic Server Administration Console is a Web browser-based, graphical user interface that you use to manage an Oracle WebLogic Server domain. It is accessible from any supported Web browser with network access to the Administration Server.

Use the Administration Console to:

- Configure, start, and stop Oracle WebLogic Server instances
- Configure Oracle WebLogic Server clusters
- Configure Oracle WebLogic Server services, such as database connectivity (JDBC) and JMS messaging
- Configure security parameters, including creating and managing users, groups, and roles
- Configure and deploy Java EE applications
- Monitor server and application performance
- View server and domain log files
- View application deployment descriptors
- Edit selected run time application deployment descriptor elements

For more information, see "Getting Started Using Oracle WebLogic Server Administration Console" in the *Oracle Fusion Middleware Administrator's Guide*.

4.2.3 Oracle WebLogic Scripting Tool

The Oracle WebLogic Scripting Tool (WLST) is a command-line scripting environment that you can use to create, manage, and monitor Oracle WebLogic Server domains. It is based on the Java scripting interpreter, Jython. In addition to supporting standard Jython features such as local variables, conditional variables, and flow control statements, WLST provides a set of scripting functions (commands) that are specific to Oracle WebLogic Server. You can extend the WebLogic scripting language to suit your needs by following the Jython language syntax.

You can use any of the following techniques to invoke WLST commands:

- Interactively, on the command line
- In script mode, supplied in a file
- Embedded in Java code

For more information, see "Getting Started Using Oracle WebLogic Server Scripting Tool (WLST)" in the *Oracle Fusion Middleware Administrator's Guide*.

For more information about the WebLogic scripting tool see the Oracle Fusion Middleware WebLogic Scripting Tool Command Reference .

4.2.4 Oracle Process Manager and Notification Server

Oracle Process Manager and Notification Server (OPMN) manages and monitors the following Oracle Fusion Middleware components, referred to as system components:

- Oracle HTTP Server
- Oracle Web Cache
- Oracle Internet Directory
- Oracle Virtual Directory
- Oracle Forms Services
- Oracle Reports
- Oracle Business Intelligence Discoverer
- Oracle Business Intelligence

For more information, see "Getting Started Using Oracle Process Manager and Notification Server" in the *Oracle Fusion Middleware Administrator's Guide*.

4.2.5 Oracle Enterprise Manager Grid Control

Oracle Enterprise Manager Grid Control is a Web browser-based, graphical user interface that you can use to monitor multiple Oracle Fusion Middleware Farms and Oracle WebLogic Server Domains. In fact, Grid Control provides deep management solutions for Oracle technologies including Oracle packaged applications, Oracle Database and Oracle VM. Grid Control also offers extensive support for non-Oracle technologies through more than two dozen heterogeneous management plug-ins and connectors including Microsoft MOM, IBM WebSphere, JBoss, EMC storage, F5 BIG IP, Check Point Firewall, and Remedy.

Beyond managing your entire data center from a single interface, Grid Control offers critical features that help you manage Oracle Fusion Middleware and Oracle WebLogic Server more effectively and efficiently. Such additional management capabilities include:

- Analyze and report on trends based upon collected availability and performance data.
- Receive alert notifications (via email, page, SNMP) for metrics which have crossed thresholds previously defined by you.
- Automate common administrative operations (e.g. start/stop, WLST scripts).
- Resolve problems faster through visibility into all Java activity - including in-flight transactions - and tracing transactions from Java to Database and vice-versa.
- Detect, validate, and report authorized and unauthorized configuration changes in real time.
- Ensure configuration consistency across development and production environments.

For more information about Oracle Enterprise Manager Grid Control refer to the *Oracle Enterprise Manager Grid Control Concepts Guide* available on OTN.

4.3 Scalability

Scalability is the ability of a system to provide throughput in proportion to, and limited only by, available hardware resources. A scalable system is one that can handle increasing numbers of requests without adversely affecting response time and throughput.

The growth of computational power within one operating environment is called vertical scaling. Horizontal scaling is leveraging multiple systems to work together on a common problem, in parallel.

Oracle Fusion Middleware scales both vertically and horizontally. Horizontally, Oracle Fusion Middleware can increase its throughput with several Managed Servers grouped together to share a workload. Also, Oracle Fusion Middleware provides great vertical scalability, allowing you to add more Managed Servers or components in the same node.

High availability refers to the ability of users to access a system. Deploying a high-availability system minimizes the time when the system is down, or unavailable and maximizes the time when it is running, or available. Oracle Fusion Middleware is designed to provide a wide variety of high-availability solutions, ranging from load

balancing and basic clustering to providing maximum system availability during catastrophic hardware and software failures.

High-availability solutions can be divided into two basic categories: local high availability and disaster recover.

For more information about high availability see the *Oracle Fusion Middleware High Availability Guide*.

4.4 High Availability

A high availability architecture is one of the key requirements for any Enterprise Deployment. Oracle Fusion Middleware has an extensive set of high availability features, which protect its components and applications from unplanned down time and minimize planned downtime to achieve your business goals.

High availability refers to the ability of users to access a system without loss of service. Deploying a high availability system minimizes the time when the system is down, or unavailable and maximizes the time when it is running, or available. This section provides an overview of high availability from a problem-solution perspective. This section includes the following topics:

- [High Availability Problems](#)
- [High Availability Solutions](#)
- [Disaster Recovery](#)

4.4.1 High Availability Problems

Mission critical computer systems need to be available 24 hours a day, 7 days a week, and 365 days a year. However, part or all of the system may be down during planned or unplanned downtime. A system's availability is measured by the percentage of time that it is providing service in the total time since it is deployed. [Table 4-1](#) provides an example.

Table 4-1 Availability Percentages and Corresponding Downtime Values

Availability Percentage	Approximate Downtime Per Year
95%	18 days
99%	4 days
99.9%	9 hours
99.99%	1 hour
99.999%	5 minutes

System downtime may be categorized as planned or unplanned. Unplanned downtime is any sort of unexpected failure. Planned downtime refers to scheduled operations that are known in advance and that render the system unavailable. The effect of planned downtime on end users is typically minimized by scheduling operational windows when system traffic is slow. Unplanned downtime may have a larger effect because it can happen at peak hours, causing a greater impact on system users.

These two types of downtimes (planned and unplanned) are usually considered separately when designing a system's availability requirements. A system's needs may be very restrictive regarding its unplanned downtimes, but very flexible for planned downtimes. This is the typical case for applications with high peak loads during

working hours, but that remain practically inactive at night and during weekends. You may choose different high availability features depending on the type of failure is being addressed.

4.4.2 High Availability Solutions

High availability solutions can be categorized into local high availability solutions that provide high availability in a single data center deployment, and disaster recovery solutions, which are usually geographically distributed deployments that protect your applications from disasters such as floods or regional network outages.

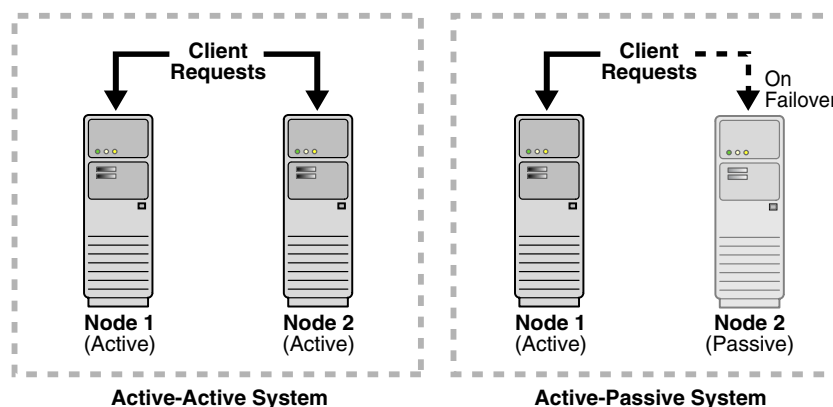
Amongst possible types of failures, process, node, and media failures as well as human errors can be protected by local high availability solutions. Local physical disasters that affect an entire data center can be protected by geographically distributed disaster recovery solutions.

To solve the high availability problem, a number of technologies and best practices are needed. The most important mechanism is redundancy. High availability comes from redundant systems and components. You can categorize local high availability solutions by their level of redundancy, into active-active solutions and active-passive solutions (see [Figure 4-1](#)):

- **Active-active solutions** deploy two or more active system instances and can be used to improve scalability as well as provide high availability. In active-active deployments, all instances handle requests concurrently.
- **Active-passive solutions** deploy an active instance that handles requests and a passive instance that is on standby. In addition, a heartbeat mechanism is set up between these two instances. This mechanism is provided and managed through operating system vendor-specific clusterware. Generally, vendor-specific cluster agents are also available to automatically monitor and failover between cluster nodes, so that when the active instance fails, an agent shuts down the active instance completely, brings up the passive instance, and application services can successfully resume processing. As a result, the active-passive roles are now switched. The same procedure can be done manually for planned or unplanned downtime. Active-passive solutions are also generally referred to as cold failover clusters.

You can use Oracle Cluster Ready Services (CRS) to manage the Fusion Middleware Active-Passive (CFC) solutions.

Figure 4-1 Active-Active and Active-Passive High Availability Solutions



In addition to architectural redundancies, the following local high availability technologies are also necessary in a comprehensive high availability system:

- **Process death detection and automatic restart**

Processes may die unexpectedly due to configuration or software problems. A proper process monitoring and restart system should monitor all system processes constantly and restart them should problems appear.

A system process should also maintain the number of restarts within a specified time interval. This is also important since continually restarting within short time periods may lead to additional faults or failures. Therefore a maximum number of restarts or retries within a specified time interval should also be designed as well.

- **Clustering**

Clustering components of a system together allows the components to be viewed functionally as a single entity from the perspective of a client for run time processing and manageability. A cluster is a set of processes running on single or multiple computers that share the same workload. There is a close correlation between clustering and redundancy. A cluster provides redundancy for a system.

If failover occurs during a transaction in a clustered environment, the session data is retained as long as there is at least one surviving instance available in the cluster.

- **State replication and routing**

For stateful applications, client state can be replicated to enable stateful failover of requests in the event that processes servicing these requests fail.

- **Failover**

With a load-balancing mechanism in place, the instances are redundant. If any of the instances fail, requests to the failed instance can be sent to the surviving instances.

- **Server load balancing**

When multiple instances of identical server components are available, client requests to these components can be load balanced to ensure that the instances have roughly the same workload.

- **Server Migration**

Some services can only have one instance running at any given point of time. If the active instance becomes unavailable, the service is automatically started on a different cluster member. Alternatively, the whole server process can be automatically started on a different machine in the cluster.

- **Integrated High Availability**

Components depend on other components to provide services. The component should be able to recover from dependent component failures without any service interruption.

- **Rolling Patching**

Patching product binaries often requires down time. Patching a running cluster in a rolling fashion can avoid downtime. Patches can be uninstalled in a rolling fashion as well.

- **Configuration management**

A clustered group of similar components often need to share common configuration. Proper configuration management ensures that components provide the same reply to the same incoming request, allows these components to synchronize their configurations, and provides high availability configuration management for less administration downtime.

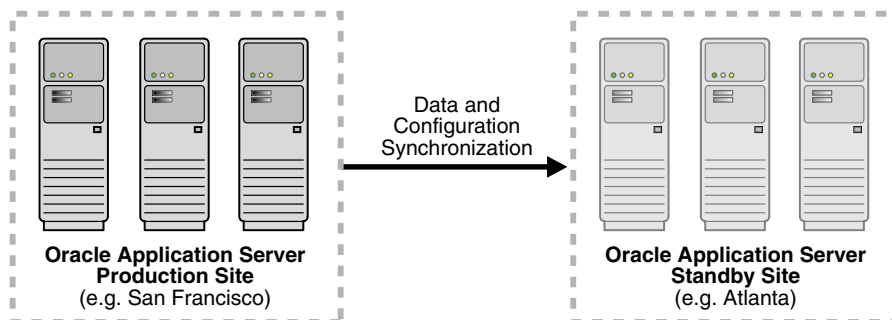
- **Backup and Recovery**

User errors may cause a system to malfunction. In certain circumstances, a component or system failure may not be repairable. A backup and recovery facility should be available to back up the system at certain intervals and restore a backup when a failure occurs.

4.4.3 Disaster Recovery

Disaster recovery solutions typically set up two homogeneous sites, one active and one passive. Each site is a self-contained system. The active site is generally called the production site, and the passive site is called the standby site. During normal operation, the production site services requests; in the event of a site failover or switchover, the standby site takes over the production role and all requests are routed to that site. To maintain the standby site for failover, not only must the standby site contain homogeneous installations and applications, data and configurations must also be synchronized constantly from the production site to the standby site.

Figure 4–2 Geographically Distributed Disaster Recovery



4.4.3.1 Components Protected by High Availability Solutions

The Oracle Fusion Middleware High Availability Guide discusses high availability solutions for the following components:

- Oracle WebLogic Server
- Oracle SOA Suite
- Oracle ADF
- Oracle WebCenter
- Oracle Identity Management Components
- Oracle HTTP Server
- Oracle Web Cache
- Oracle Portal, Forms, Reports, and Discoverer

4.5 Load Balancing

Origin server load balancing is a feature in which HTTP requests are distributed among origin servers so that no single origin server is overloaded.

Oracle Web Cache supports load balancing and failover detection for application Web servers.

Oracle Web Cache ensures that cache misses are directed to the most available, highest-performing Web server in the server farm. A capacity heuristic guarantees performance and provides surge protection when the application Web server load increases.

For more information about load balancing and failover, see Oracle Fusion Middleware Administrator's Guide for Oracle Web Cache.

4.6 Diagnostic Data

Oracle Fusion Middleware components generate log files containing messages that record all types of events, including startup and shutdown information, errors, warning messages, and access information on HTTP requests. This chapter describes how to find information about the cause of an error and its corrective action, to view and manage log files to assist in monitoring system activity and to diagnose problems.

For more information see the *Oracle Fusion Middleware Administrator's Guide*.

4.7 Cloning

Cloning is the process of copying an existing entity to a different location while preserving its state. Some situations in which cloning Oracle Fusion Middleware is useful are:

- Creating a Middleware home or Oracle home that is a copy of a production, test, or development environment. Cloning enables you to create a new Middleware home or an Oracle home with all patches applied to it in a single step. This is in contrast to separately installing, configuring and applying any patches to separate Oracle homes.
- Preparing a "gold" image of a patched home and deploying it to many hosts.

The cloned entity behaves the same as the source entity. For example, a cloned Oracle home can be deinstalled or patched using the installer. It can also be used as the source for another cloning operation.

For more information about cloning see *Oracle Fusion Middleware Administrator's Guide*.

4.8 Oracle RAC

Oracle Real Application Clusters (Oracle RAC) is a computing environment that harnesses the processing power of multiple, interconnected computers. Along with a collection of hardware, called a cluster, it unites the processing power of each component to become a single, robust computing environment. A cluster comprises two or more computers, also called nodes. Oracle Real Application Clusters simultaneously provides a highly scalable and highly available database for Oracle Fusion Middleware.

Every Oracle RAC instance in the cluster has equal access and authority, therefore, node and instance failure may affect performance, but does not result in downtime,

since the database service is available or can be made available on surviving server instances.

For more information on Oracle Real Application Clusters, see the Oracle Fusion Middleware High Availability Guide.

4.9 Backup and Recovery

An Oracle Fusion Middleware environment can consist of different components and configurations. A typical Oracle Fusion Middleware environment contains an Oracle WebLogic Server domain with Java component such as Oracle SOA Suite and an Oracle WebLogic Server domain with Oracle Identity Management components. It can also include one or more Oracle instances.

The installations of an Oracle Fusion Middleware environment are interdependent in that they contain configuration information, applications, and data that are kept in synchronization. For example, when you perform a configuration change, you might update configuration files in the installation. When you deploy an application, you might deploy it to all Managed Servers in a domain or cluster.

It is important to consider your entire Oracle Fusion Middleware environment when performing backup and recovery. You should back up your entire Oracle Fusion Middleware environment once, then periodically thereafter perform incremental backup and recovery operations. If a loss occurs, you can restore your environment to a consistent state.

For information about backup and recovery, see "Advanced Administration: Backup and Recovery" in the *Oracle Fusion Middleware Administrator's Guide*.

Index

A

active-active topologies, 4-6
active-passive topologies, 4-6
ADF Business Components, 3-30
ADF Controller, 3-31
ADF faces rich client, 3-31
Administration Server, 2-2, 2-4, 2-5
application module, 3-31
automatic restart, 4-7

B

Backup, 4-8
backup and recovery, 4-8, 4-10

C

CEP Service Engine. *See* Oracle CEP Service Engine
cloning, 4-9
cluster, 2-6
clustering, 4-7
creating a domain, 2-9
creating the Oracle Home directory, 2-9

D

death detection, 4-7
deployment architecture, 2-3
development tools, 3-28
directory structure, 2-4
domain template, 2-6

E

enterprise deployment, defined, 4-1
entity object, 3-30
Event-Driven Architecture. *See* EDA
extending a domain, 2-10

F

Failover, 4-7
failover, 4-7
farm, 2-8
functions
 middleware, 1-2

Fusion Middleware Control
 using, 4-2

G

grid servers, 4-1

H

high availability, 4-4
high availability practices, Oracle site, 3-2, 4-2
human workflow, 3-27

I

Integrated, 4-7
integrated high availability, 4-7

J

Java component, 2-1
Java EE, 3-3
Java Message Service, 3-8
Java Platform, Enterprise Edition, 3-3
JRockit, 3-8

M

Managed Servers, 2-2, 2-4
Managed servers, 2-5
metadata repository, 2-2, 2-7, 3-9
Middleware, 1-1
middleware
 functions, 1-2
middleware architecture design, 1-3
Middleware home, 2-2, 2-7
Middleware Home directory, 2-8

N

Node Manager, 2-6

O

Oracle Access Manager, 3-14
Oracle Adaptive Access Manager, 3-15
Oracle ADF, 3-29

- Oracle ADF Business Components, 3-30
- Oracle ADF Desktop Integration, 3-33
- Oracle ADF Mobile, 3-32
- Oracle ADF Model, 3-30
- Oracle ADF Security, 3-32
- Oracle Application Development Framework, 3-8
- Oracle Authorization Policy Manager, 3-15
- Oracle B2B, 3-25
- Oracle BPEL Process Manager, 3-26
- Oracle Business Activity Monitoring, 3-25
- Oracle Business Intelligence Discoverer, 3-20
- Oracle Business Intelligence Enterprise Edition, 3-19
- Oracle Business Intelligence Publisher, 3-19
- Oracle Business Rules, 3-25
- Oracle CEP Service Engine
 - about, 3-23
- Oracle Coherence, 3-7
- Oracle Common home, 2-7
- Oracle Complex Event Processing, 3-23
- Oracle Composer, 3-16
- Oracle Data Integrator, 3-39
- Oracle Data Profiling, 3-39
- Oracle Data Quality, 3-39
- Oracle Delegated Administration Services, 3-37
- Oracle Directory Integration Platform, 3-11
- Oracle Directory Services Manager, 3-14
- Oracle Enterprise Content Management Suite, 3-37
- Oracle Enterprise Manager 10g Grid Control, 4-4
- Oracle Enterprise Pack for Eclipse, 3-34
- Oracle Entitlements Server, 3-14
- Oracle Forms Services, 3-21
- Oracle Fusion Middleware
 - creating the Oracle Home directory, 2-9
 - directory structure concepts, 2-8
 - environment, 2-3
 - Middleware Home directory, 2-8
 - Oracle Home directory, 2-9
 - WebLogic Home directory, 2-8
 - with Oracle Web Cache, 3-36
- Oracle Fusion Middleware enterprise deployment
 - functions, 4-1
- Oracle Help, 3-35
- Oracle home, 2-7
- Oracle Home directory, 2-9
- Oracle HTTP Server, 3-35
- Oracle Identity Federation, 3-12
- Oracle Identity Management, 3-10
- Oracle Identity Manager, 3-10
- Oracle Identity Navigator, 3-15
- Oracle Imaging and Process Management, 3-39
- Oracle Information Rights Management, 3-38
- Oracle instance, 2-7
- Oracle Instance in a WebLogic Domain, 2-11
- Oracle Internet Directory, 3-11
- Oracle JDeveloper, 3-28
- Oracle Mediator, 3-28
- Oracle Metadata Repository, 3-9
- Oracle Platform Security Services, 3-13
- Oracle Portal, 3-20
- Oracle Process Manager and Notification Server
 - (OPMN), 4-3
- Oracle Real Application Clusters, 4-9
- Oracle Reports, 3-20
- Oracle Reports Bridge, 3-21
- Oracle Reports Builder, 3-21
- Oracle Reports Client, 3-21
- Oracle Reports Runtime, 3-21
- Oracle Reports Server, 3-21
- Oracle Reports Services, 3-21
- Oracle Reports Servlet, 3-21
- Oracle Role Manager, 3-13
- Oracle Service Bus, 3-23
- Oracle Service Registry, 3-26
- Oracle SOA Suite, 3-22
- Oracle TopLink, 3-9, 3-33
- Oracle User Messaging Service, 3-27
- Oracle Virtual Directory, 3-12
- Oracle Web Cache, 3-36
 - features
 - origin server load balancing and failover, 4-9
 - with Oracle HTTP Server, 3-36
 - with Oracle WebLogic Server, 3-36
- Oracle Web Tier
 - Oracle HTTP Server, 3-36
- Oracle WebCenter, 3-16
- Oracle WebCenter Framework, 3-17
- Oracle WebCenter Services, 3-17
- Oracle WebCenter Spaces, 3-18
- Oracle WebLogic Scripting Tool (WLST)
 - See Also WLST commands
- Oracle WebLogic Server, 3-3
- Oracle WebLogic Server Administration
 - Console, 4-2
- Oracle WebLogic Server Diagnostic Framework, 3-5
- Oracle WebLogic Server domain, 2-2
- Oracle WebLogic Server Enterprise Grid
 - Messaging, 3-8
- Oracle WebLogic Server Web Service, 3-7

P

- performance, enterprise deployment and, 4-1
- process death detection, 4-7

R

- Rolling, 4-7
- rolling patching, 4-7

S

- scalability, 4-4
- security architecture
 - Oracle WebLogic Server, 3-6
- Server, 4-7
- server load balancing, 4-7
- server migration, 4-7
- service level agreements, 4-1
- Service-Oriented Architecture, 1-4
- Simple Network Management Protocol, 3-5
- state replication and routing, 4-7

system component, 2-1

T

Tuxedo, 3-9

V

view object, 3-30

W

Web tier

Oracle HTTP Server, 3-36

WebLogic Domain

creating, 2-9

directory structure for multiple products, 2-10

Domains directory, 2-9

extending, 2-10

User Projects directory, 2-9

with Oracle Instance, 2-11

WebLogic Home directory, 2-8

WebLogic Scripting Tool, 3-5

WebLogic Server home, 2-7

