CHAPTER 1

Introduction to Technical Architecture
This chapter will provide a bird's-eye view of the key technical components of Oracle Fusion Applications with emphasis on those that are typically relevant to designers and developers of extensions and customizations. Detailed deep-dive and technical insights will be provided in the subsequent chapters in this book, and the highlights provided in this chapter give a cursory glimpse of the technology components that are used to build Fusion Applications.

While reading this chapter, and of course the rest of this book, keep in mind that some of the key drivers that shaped the technical architecture of Fusion Applications were to have an efficient, productive, and predictable user experience, and to build an adaptable applications platform based on established standards such as Java, Service Oriented Architecture (SOA), Extensible Markup Language (XML), Business Process Execution Language (BPEL), and technology acceptance that is suitable for current and emerging trends such as cloud and mobile computing, social media, embedded business intelligence, predictive analytics, and so forth.

Fusion Applications (11.1.X) are built using Oracle Fusion Middleware 11g components, Oracle Database 11g, and Oracle Essbase, and the focus of this chapter is to outline the way those technologies are leveraged and exploited to build Oracle Fusion Applications.

Technical Architecture Overview
On the level of technology, Fusion Applications primarily use Oracle Fusion Middleware and Oracle Database as depicted in Figure 1-1. In addition to the standard Fusion Middleware components, you can see an extra technology layer called Fusion Middleware Infrastructure Components, which are specific to Fusion Applications to provide a common and reusable framework that can be applied to any Fusion Applications product.

In comparison to other Oracle Applications products such as Oracle E-Business Suite, there is a visible shift in the pronounced use of standard application middleware features such as Identity Management and SOA. However, Fusion Middleware Extensions for Applications (Applications Core or applcore) still feature prominently in the technology stack, and we will cover them extensively throughout the book as they are essential tools in the application designer's arsenal. The components, such as flexfields, UI templates, document attachments, and hierarchical trees, are some of the tools that promote common development standards across applications.

Fusion Middleware Components
It shouldn't come as a surprise that Oracle Fusion Middleware and its components provide the principal infrastructure foundation to Fusion Applications and its product families. Oracle has heavily invested in making Fusion Middleware a complete,
open, and integrated set of components that fall into different categories and functional areas:

- Development Tools and Frameworks (JDeveloper, Applications Development Framework, Enterprise Pack for Eclipse)
- User Experience (WebCenter)
- Service Oriented Architecture (SOA Suite)
- Content Management
- Business Intelligence
- Data Integration
Oracle Fusion Applications Development and Extensibility Handbook

- Application Grid (WebLogic Server, JRockit, Oracle Coherence, Oracle HTTP Server, Web Cache)
- Identity Management (Directory Services, Access Management, Identity Federation, Identity Administration, Entitlement Services, Identity Analytics)

While our book touches on most of the previously listed topics in later chapters, we will now take this opportunity to have a detailed look at Oracle WebLogic Server, the foundation component on which all Java-based Oracle Fusion Middleware components are built.

**Oracle WebLogic Server**

WebLogic Server is where applications and products are deployed, managed, and monitored. It is also a place where security, database connection management, messaging, and other key features are configured and administered.

The most important concepts that are fundamental to WebLogic Server are server instance, domain, and cluster. A WebLogic Server instance is a Java Virtual Machine (JVM) process that is executing Java code written to Java EE and/or WebLogic API specifications. The server consists of a number of deployment containers such as Web Container that run deployed Java EE Web applications, JDBC and JMS Services that manage communication with database and Java Messaging Service resources, Security Services, Request Management Services, and Java Runtime Environment (JRE).

The domains are a logical grouping of WebLogic Server instances for administration and configuration purposes. There are two types of WebLogic Servers:

- Administration (Admin) Servers
- Managed Servers

Admin Server is a special type of WebLogic Server and is used for domain administration, configuration, and monitoring. There can only be one Admin Server within a domain. What distinguishes the Admin Server from other (managed) servers is that it performs some additional functions:

- Configuration management for all servers that belong to its domain. Stores the master copy of the domain configuration including the configuration of other servers (managed servers).
- Provides administration consoles to manage the domain and its servers; the consoles are HTML-based and can be accessed through a browser to perform administration, configuration, and monitoring tasks.
- Enables server and service migration as well as deployment of applications within the domain.
Managed servers, on the other hand, are running instances of WebLogic Server where Java EE applications are usually deployed, and they also host other resources required by applications. There is nothing preventing you from deploying an application to the Admin Server; however, that is not recommended in a production environment as all applications should be deployed to the managed servers within the domain. Managed servers are independent of each other unless they form a part of a cluster (discussed shortly); the number of managed servers is unlimited, and they can be added as necessary to either increase capacity or group similar applications together.

When managed servers are started, they connect to the Admin Server to synchronize its local copy of configuration files with the master copy on Admin Server. Every time the configuration changes, Admin Server notifies the managed servers and sends the changed configuration to them to keep the configuration in synch.

Managed servers within the domain can be grouped to form a cluster. A WebLogic domain can have many clusters, but each managed server can only belong to one cluster. One such example is Oracle Fusion Applications HCM Domain, illustrated in Figure 1-2.
One of the main benefits of clusters is application scalability since one can keep adding as many managed servers as necessary to increase the capacity. WebLogic clusters also provide high availability and reliability through deployment across more than one server. To most client applications, a WebLogic cluster will appear as a single server instance, and applications deployed to the cluster will be deployed to all servers within that cluster. Clusters enable some other advanced features such as whole server migration and server migration, but the details and inner workings of those are covered in the Oracle Fusion Middleware System Administrator’s Guide rather than in this text.

NOTE

In Figure 1-2 you see a screen shot of Enterprise Manager Fusion Applications Control that shows HCMDomain configuration installed on an instance of Fusion Applications on a single node. Therefore, you only see one managed server per cluster, but additional managed servers as represented by elements with a dashed boundary in the diagram can be added on Host 2, for example.
Each host can have a node manager, which is a WebLogic Server utility that allows system administrators to start, stop, and restart administration and managed servers from a remote location. The node manager is a process that is not tied to a particular WebLogic domain but to the physical machine, for example, a Linux or Windows host.

The WebLogic domain directory structure reflects the fact that the server configuration is segmented by domain where each domain has one set of configuration files. Among many configuration files, config.xml is the main configuration file located under the config directory as shown in Figure 1-3. This central configuration file is where the details about server instances, clusters, and other resources for the domain are provided.

In Fusion Applications, the applications from product families, such as Oracle Fusion Human Capital Management (HCM), are all deployed within a single domain, which is called HCMDomain in this case. In addition to the product family–related applications, Fusion Applications domains usually have two additional clustered servers: Oracle Enterprise Scheduler Service (ESS) cluster and SOA cluster. The ESS server hosts Enterprise Scheduler Application (ESS App), which manages and schedules jobs for that product family similarly to what concurrent request managers do in Oracle E-Business Suite, for those who are familiar with one of Oracle’s earlier enterprise applications products. Equally, SOA servers host composite applications for the applications that belong to the particular product family.

**Fusion Applications Product Families**

In the previous section, we mentioned the Oracle Fusion HCM product family without introducing the concept of product families. Product families are simply collections of products associated with a particular business function, and they consist of one or many applications.
FIGURE 1-3.  WebLogic domain directory structure
In Figure 1-4, you can see that the Oracle HCM product family consists of four Fusion Applications: HcmCoreSetupApp, HcmCoreApp, HcmEssApp, and HcmAnalyticsApp. Also, you notice that individual products can be part of many applications and that applications may extend across many products. The products,
like Global Human Resources, which is shown next, are normally deployed as an Enterprise Archive File (ear) file on the middle tier.

In addition to Human Capital Management, the current release of Fusion Applications features the following product families: Customer Relationship Management (CRM); Procurement; Governance, Risk, and Compliance (GRC); Project Portfolio Management (PPM); and Supply Chain Management (SCM).

Fusion Middleware Infrastructure Components for Fusion Applications

Most enterprises and organizations accomplish common business functions like accounting in similar ways; however, an enterprise application such as Oracle Fusion Financials has to provide a great deal of flexibility to accommodate requirements that are specific to a particular organization and business environment. Take the example of a Chart of Accounts, which is an accounting structure that is different from one company to another.
This drives the need for a specialized type of infrastructure that underpins the extensibility and other features specific to Fusion Applications that go beyond what Oracle Fusion Middleware offers out of the box. Oracle Fusion Applications borrows some of the best architectural concepts from its previous application product offerings such as flexfields from E-Business Suite, PeopleSoft tree hierarchies, effective dates, shared reference data (SetIds), and so on.

As shown earlier in Figure 1-1, the main components of Fusion Middleware Infrastructure for Fusion Applications are

- Oracle Fusion Middleware Extensions for Applications
- Oracle Enterprise Scheduler
- Oracle Enterprise Crawl and Search Framework (ECSF)

Since this book is almost entirely dedicated to extensions, customizations, and development techniques, in the next sections we’ll provide only a brief description and summary of the Fusion Middleware Infrastructure for Fusion Applications components as they will be covered extensively in the subsequent chapters.

**Oracle Fusion Middleware Extensions for Applications**

Oracle Fusion Middleware Extensions for Applications are reusable common components that can be used in both standard (shipped by Oracle) or bespoke applications written by Fusion Applications developers. From a programming point of view, the extensions are packaged as a Java library that needs to be added to the project so that common components can be used in that project.

Following is a partial list of components that are commonly used in Fusion Applications along with a brief summary of their features:

- **Flexfields**  Allow customers to configure Fusion Applications products to meet the business requirements without writing custom code. Similarly to Oracle E-Business Suite, a flexfield is a data field that is expandable and consists of segments. Generally, they either capture and store additional information about some business entity such as customers through the user interface, or they can provide a multipart key identifier for a business entity, such as account numbers so it can be uniquely identified. In Fusion Applications there are three types of flexfields: descriptive, extensible, and key.

- **UI Shell**  Provides a default page template for all Fusion Application pages apart from login and the password preferences page. UI Shell provides predictable behavior of all pages, and it also enforces the use of standards such as Oracle Fusion Guidelines, Patterns, and Standards (GPS). UI Shell supports global search, navigation menus, and cross-application navigation.
Trees  Allow data to be organized in hierarchies and allow creating of tree hierarchies based on data. Fusion Applications developers design and deploy tree structures, while application users use trees, which are individual instances of those structures that can be used to group and roll up data.

Attachments  Provide a mechanism of attaching additional content to a business object. An example is a purchase order with, for example, a supplementary PDF attached to it. Users of applications can attach URLs, plain text, desktop, and repository files to the business objects. From a development point of view, attachments are declaratively added to the Fusion Applications UI pages. When added, they appear on the page as Attachment item (field), Attachment column in the table, or Attachments table.

Oracle Fusion Application developers spend most of their time working with different components of Fusion Middleware Extensions for Applications, which is also known as Applications Core or applcore. For example, if a developer needs to throw a Java exception in their code, he or she will do it with help of a Java class from the Applications Core (applcore) library:

```java
import oracle.apps.fnd.applcore.messages.ApplcoreException;
...
Map<String, Object> tokens = new HashMap<String, Object>();
tokens.put("TEXT", "Exception text");
tokens.put("NUMBER", new BigDecimal(2001));
...
throw new ApplcoreException("XXAPP:::XXAPP_MESSAGE_NAME", tokens);
```

Other examples of Applications Core extensions are PL/SQL entities, FND services, Unique IDs, document sequencing, Set IDs, TL (translatable) tables, and WHO columns, and we will discuss them as they appear in detail in subsequent chapters.

Oracle Enterprise Scheduler (ESS)

In addition to being able to provide an efficient user experience through browser-based user interaction, enterprise systems like Fusion Applications are required to have the capability of running scheduled background processes for long-running, back-office, and reporting tasks. Users of Oracle E-Business Suite will undoubtedly be familiar with the concept of concurrent requests, and similar functionality is available in Fusion Applications through the Oracle Enterprise Scheduler component.

Unlike Concurrent Request Managers in Oracle E-Business Suite, Oracle Enterprise Scheduler has a mere role of time and resource controller for the job and its schedules; the responsibility of running the actual jobs is delegated to the client applications, which receive callbacks from the Enterprise Scheduler engine.
Chapter 1: Introduction to Technical Architecture

Each Fusion Applications Domain has a dedicated ESS cluster with an Ess-app .ear file deployed. The client server cluster runs a client application along with its UI, which allows the application user to submit and monitor jobs.

Developers use Oracle JDeveloper and Enterprise Manager (EM) Control to develop ESS client applications, job definitions, sets, and incompatibilities between the jobs. Business users can use ESS Submission Request Screen (SRS) UI and ESS Monitor UI to submit and monitor the jobs they own. The primary interface for System Administrators is EM Control, where they can define security, monitor the jobs for all users, define job schedules, perform ESS troubleshooting, and control other aspects of job execution such as stopping and resuming.

**Oracle Enterprise Crawl and Search Framework (ECSF)**

The ECSF redundant integrates with Oracle Secure Enterprise Search (Oracle SES) through Security and Crawler plugins to provide the ability to perform secure searches on ADF view objects, Fusion file attachments, and WebCenter tags. It exposes application programming APIs, which developers can use to design and incorporate search capability in their Fusion Applications. Developers use the View Object editor’s Search page in JDeveloper to set search properties for business objects.
Searches are category- or keyword-based, and ECSF takes care of security and user access rights.

**Oracle Database and Oracle Essbase**

Oracle Fusion Applications use Oracle Database 11g to store and retrieve transactional data from applications. However, before Fusion Applications can be installed, Oracle Identity Management components need to be installed first, which also use Oracle databases for data storage and retrieval. In addition to Oracle Database 11g, the Fusion Applications environment uses Oracle Essbase to manage multidimensional data in analytic and performance management applications.

The transactional database, in the current release of Fusion Applications 11.1.x, keeps data for all product families and middleware in a single database. Oracle Fusion Applications Repository Creation Utility (Applications RCU) is used during the environment installation to create a repository of application-specific schemas and tablespaces for Oracle Database. The Applications RCU creates Oracle Database schema users that own appropriate middleware and application components as shown in Figure 1-5.

![FIGURE 1-5. Fusion Applications transactional database schemas in Oracle Database](image-url)
Enterprise Manager Controls
(Administration Tools)

Oracle Enterprise Manager is certainly one of the most important tools for monitoring and management of applications, middleware, and Oracle databases. Fusion Applications are shipped with a special version of Enterprise Manager (EM) known as Fusion Applications Control (see the earlier Figure 1-4).

Fusion Applications Control enables end users and system administrators to monitor and administer a product family within the Oracle Fusion Applications environment. Information is presented in the form of dashboards to show real-time performance details about running ESS jobs, WebLogic server statuses, Service Oriented Architecture (SOA) processes, execution statuses, and so on. In addition to Fusion Applications Control, system administrators can separately install Oracle EM Grid Control, which provides even more centralized monitoring and management capability to monitor Oracle Database and Oracle Fusion Middleware components across the entire Oracle Fusion Applications environment.

Summary

In this chapter, we have glanced over most of the major technology components in Fusion Applications. Undoubtedly, Oracle Fusion Middleware with Fusion Applications-specific extensions functions as the principal technology toolset used to build, deploy, and run Fusion Applications.

Its technologies provide a platform to build a user interface with the Application Development Framework (ADF), orchestrate services with SOA Suite components, offer security and extensibility frameworks, integrate data with the Oracle Data Integrator tool, as well as provide tools such as Enterprise Crawl and Search, Enterprise Scheduler Service, and so on.

In addition to the middleware components, back-end technologies such as Oracle Database and Oracle Essbase are the primary places where Fusion Applications store and retrieve transactional and multidimensional data.