J2EE Architecture

Basic

About the Author

Created By:	**Mohamed, Hajura (Cognizant), 117960**
Credential Information:	**Executive – Cognizant Academy**
Version and Date:	**J2EE/PPT/0306/1.0**

Cognizant Certified Official Curriculum

Icons Used

Questions

Hands-on Exercise

A Welcome Break

Test Your Understanding

Coding Standards

Reference

Demo

Key Contacts

J2EE Architecture: Overview

- **Introduction:**

The chapter provides a brief overview of the J2EE architecture and lists the important terms and concepts. In addition, it discusses how to approach J2EE application programming, assembly, and deployment.

J2EE Architecture: Objectives

- **Objective:**

  After completing this chapter, you will be able to:

  - Explain a distributed multi-tiered applications
  - Explain J2EE components
  - Explain J2EE containers
  - Describe container services
  - Describe container types
  - Explain packaging and deployment
  - Describe J2EE APIs

J2EE Architecture: Distributed Multi-tiered Applications

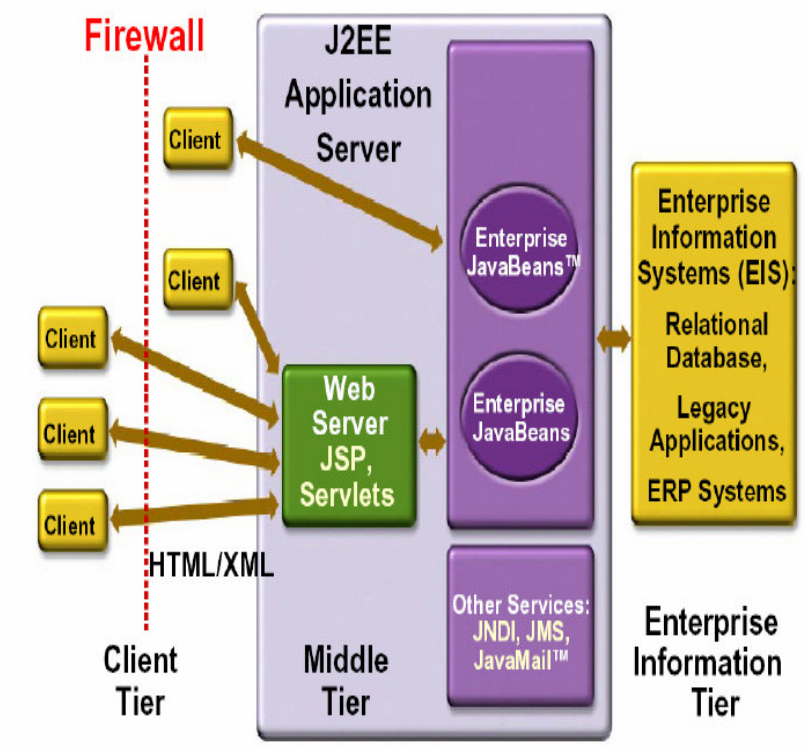
- J2EE stands for Java 2 Enterprise Edition.
- In the past, two-tier applications were used. Two-tier applications are also known as client/server applications. In most of the cases, these architectures were providing only the database functionality and all the business logic and presentation of the data logic were installed on the client machine. This architecture was very difficult to maintain.

J2EE Architecture: Distributed Multi-tiered Applications (Contd.)

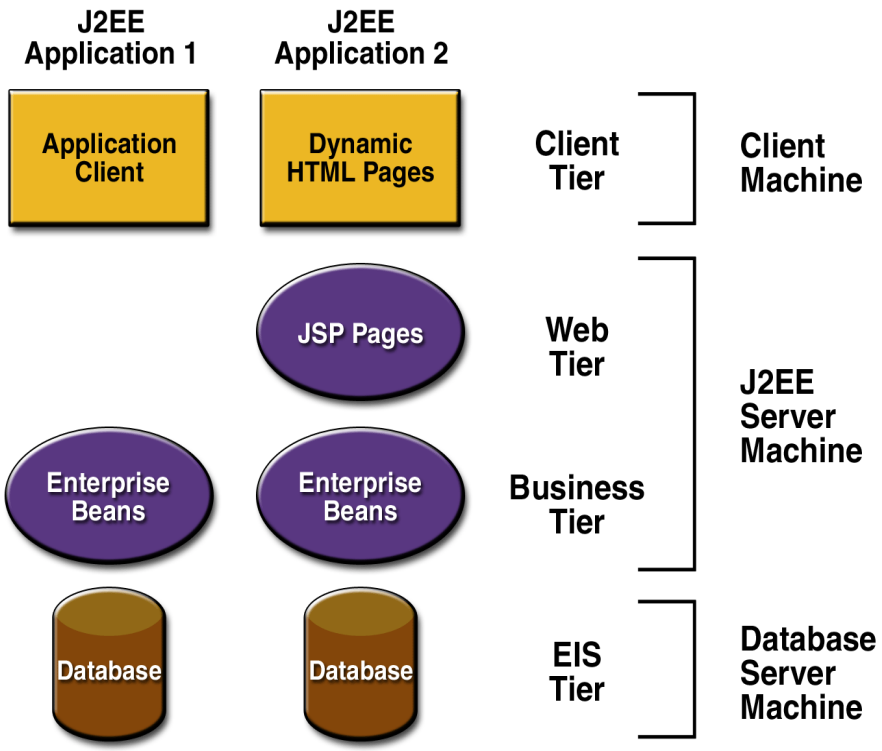
- The J2EE platform uses a distributed multi-tiered application model for enterprise applications.
- Application logic is divided into “components” according to functions. These components, which make up a J2EE application, are installed on different computers that are placed distributively on different tiers. For example:
  - Client-tier components run on the client computer
  - Web-tier components run on the J2EE server
  - Business-tier components run on the J2EE server
  - Enterprise Information System or EIS-tier software runs on the EIS server

J2EE Architecture: Distributed Multi-tiered Applications (Contd.)

- Three-tiered applications extend the standard two-tiered client and server mode.
- A multithreaded application server is placed between the client application and back-end storage.



J2EE Architecture: Distributed Multi-tiered Applications (Contd.)



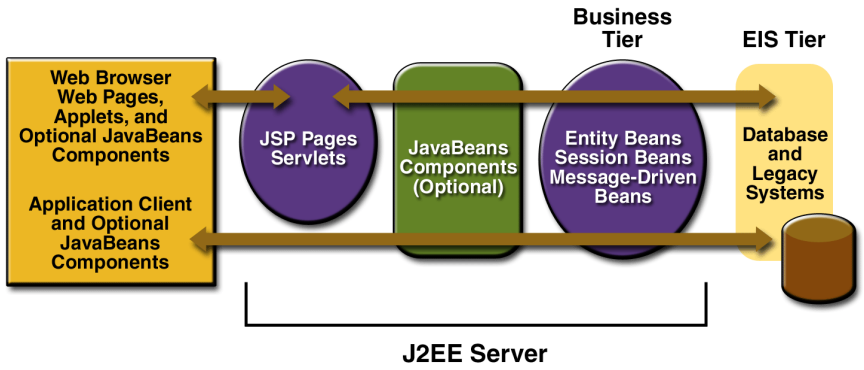
- J2EE multi-tiered applications are generally considered to be three-tiered applications because they are distributed over three different locations:
  - Client computers
  - The J2EE server machine
  - The database or legacy machines at the backend

J2EE Components

- A J2EE component
  - Is a self-contained functional software unit
  - Is assembled into a J2EE application with its related classes and files
  - Communicates with other components

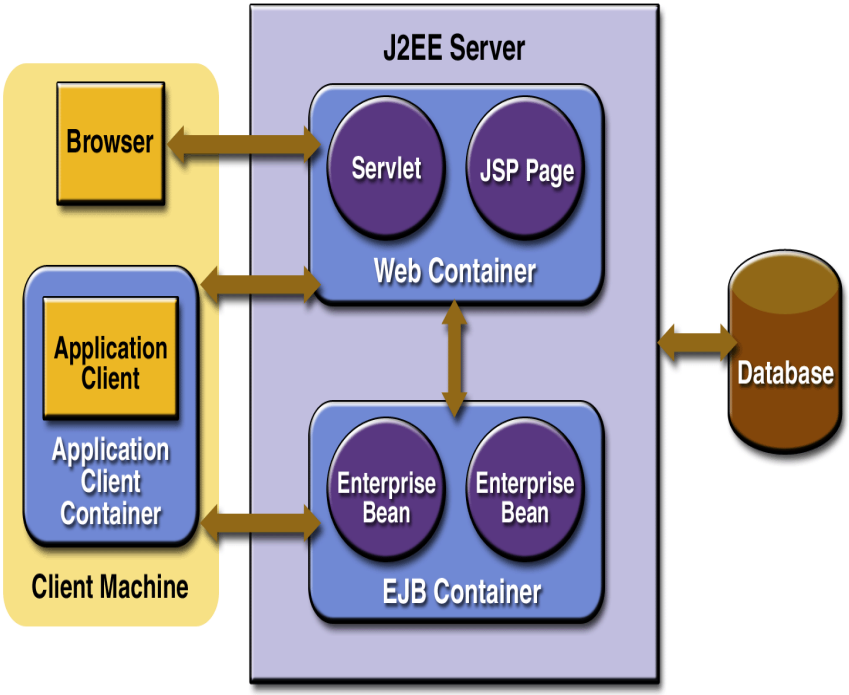
The following are the types of J2EE components:

- Client components run on the client machine, which correlate to the client containers
- Web components -servlets and JSP pages
- EJB Components- run on the Business Tier



J2EE Containers

- Containers are the interface between a component and the low-level, platform-specific functionality that supports the component.
- The following are the server-side containers:
  - The server itself, which provides the J2EE runtime environment and the other two containers:
    - An EJB container to manage EJB components
    - A Web container to manage servlets and JSP pages
- The other two container types are client-side:
  - An application container for standalone GUIs and console
  - An applet container, which is a browser, which comes usually with the Java plug-in



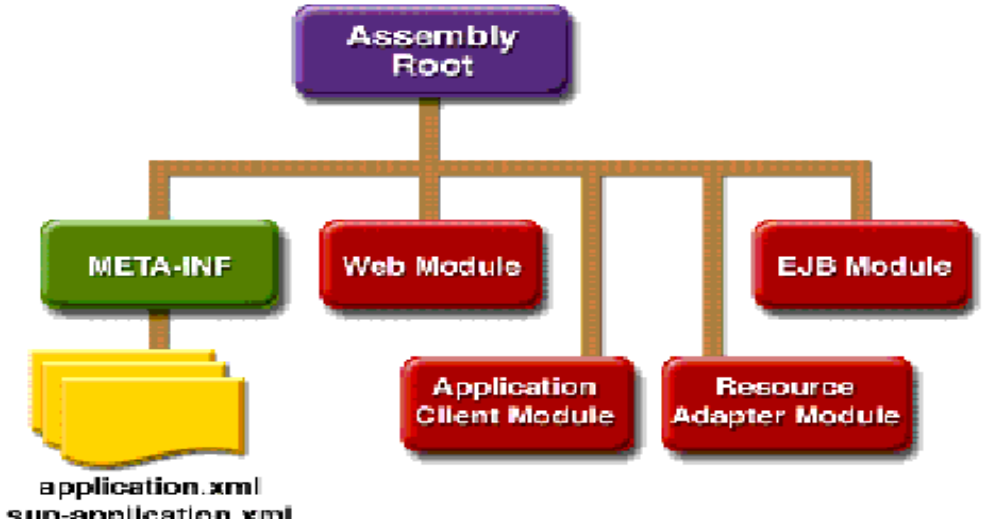
Container Services

Container settings customize the underlying support provided by the J2EE server, including services such as security, transaction management, Java Naming and Directory Interface (JNDI) lookups, and remote connectivity. The following are some of the highlights of container services:

- **Security:** The J2EE security model lets you to configure a Web component or enterprise bean so that the system resources are accessed only by authorized users.
- **Transaction Management:** The J2EE transaction model lets you to specify the relationships among methods that make up a single transaction, so that all methods in one transaction are treated as a single unit.
- **Java Naming and Directory Interface (JNDI) lookups:** JNDI lookup services provide a unified interface to multiple naming and directory services in the enterprise so that application components can access naming and directory services.
- **Remote connectivity:** The J2EE remote connectivity model manages low-level communications between clients and enterprise beans. After an enterprise bean is created, a client invokes methods on it as if it were in the same virtual machine.

Packaging Applications and Components

- Under J2EE, applications and components are packaged into Java Archive (JAR) files.
- EJB modules contain class files for enterprise beans and an EJB deployment descriptor. EJB modules are packaged as JAR files with a `.jar` extension.



Packaging Applications and Components (Contd.)

- Web modules contain servlet class files, JSP files, supporting class files, GIF and HTML files, and a Web application deployment descriptor called `web.xml`. Web modules are packaged as JAR files with a .war (Web archive) extension.
- Application client modules contain class files and an application client deployment descriptor. Application client modules are packaged as JAR files with a `.jar` extension.
- Resource adapter modules contain all Java interfaces, classes, native libraries, and other documentation, along with the resource adapter deployment descriptor. Together, these implement the connector architecture for a particular EIS. Resource adapter modules are packaged as JAR files with an resource adapter archive (`.rar`) extension.
- Together all the archives will form a `.ear` (Enterprise Archive) file for an application.

Deployment Descriptors

- Deployment descriptors are included in the JARs, along with component-related resources.
- Deployment descriptors are XML documents that describe configuration and other deployment settings to the J2EE Server.
- The statements in the deployment descriptor are declarative instructions to the J2EE container.

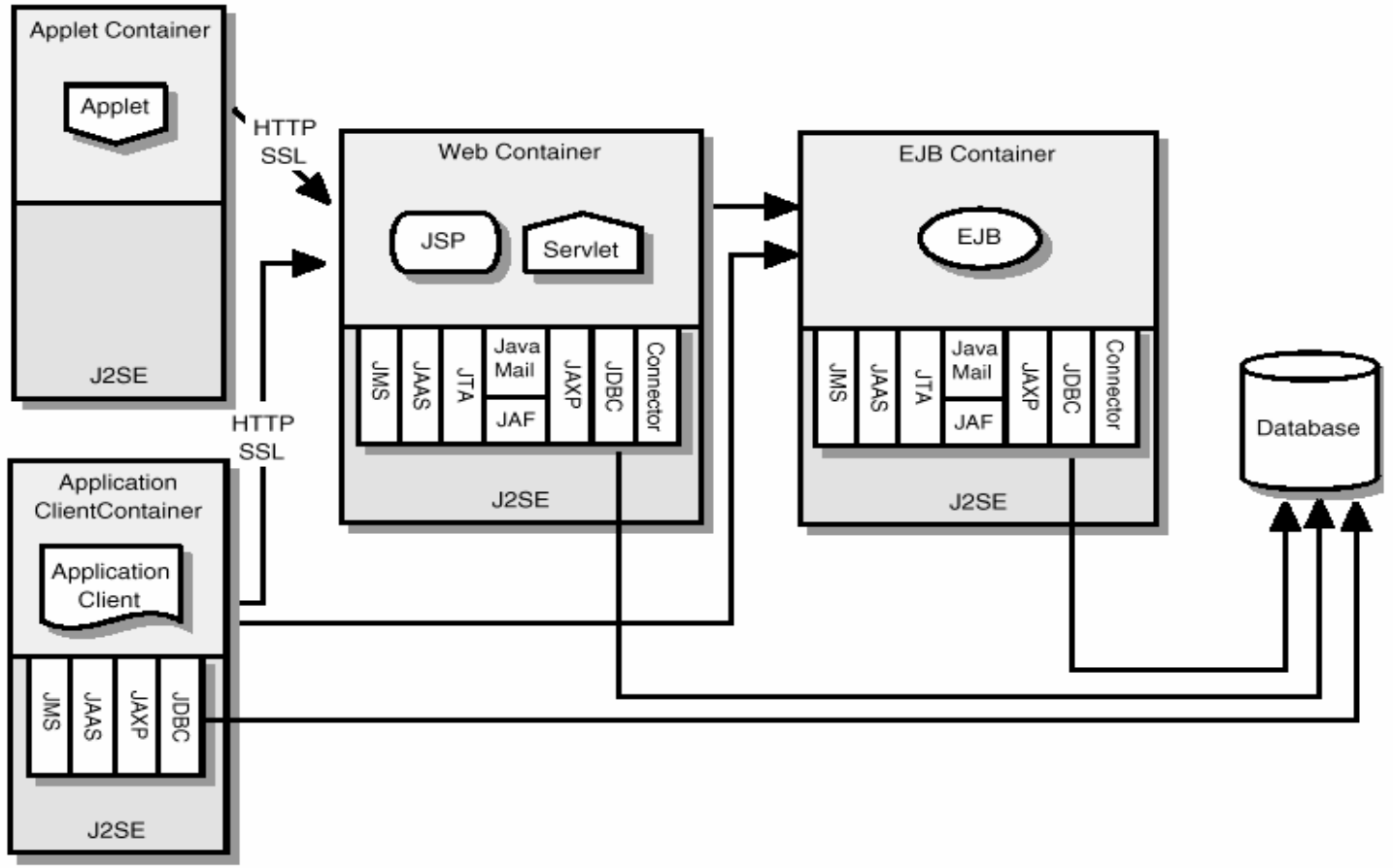
  Most J2EE vendors provide a GUI tool for generating deployment descriptors and performing deployment, because creating manual entries is tedious and error prone.
- The deployment descriptor for an EJB component must be named `ejb-jar.xml`, and it resides in the `META-INF` directory inside the EJB JAR file.
- The deployment descriptor for an Web component should be name as `web.xml` and it resides in the `WEB-INF` directory.

Example of web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?> <web-app>
<servlet>
   <servlet-name> snoop </servlet-name>
   <servlet-class> SnoopServlet </servlet-class>
</servlet>
<servlet>
   <servlet-name> file </servlet-name>
   <servlet-class> ViewFile </servlet-class>
   <init-param> <param-name> initial </param-name> <param-value> 1000
   </param-value> <description> The initial value for the counter <!--
   optional --> </description>
   </init-param>
</servlet>
<distributed/>
<security-role>
<role-name> manager </role-name>
<role-name> director </role-name> <role-name> president </role-name>
</security-role>
</web-app>
```

J2EE 1.4 APIs

The figure illustrates the availability of the J2EE 1.4 platform APIs in each J2EE container type. The sections of the following diagram specify the technologies required by the J2EE platform and the J2SE Enterprise APIs that would be used in J2EE applications.



- **Allow time for questions from participants**

J2EE Architecture: Summary

- J2EE stands for Java 2 Enterprise Edition.
- Two-tier applications are also known as client/server applications.
- The J2EE platform uses a multi-tiered distributed application model.
- Application logic is divided into “components” according to functions.
- A multithreaded application server is placed between the client application and back-end storage.
- The following are the types of J2EE components:
  - Client components run on the client machine, which correlate to the client containers
  - Web components: servlets and JSP pages
  - EJB Components: run on the Business Tier
- Containers are the interface between a component and the low-level, platform-specific functionality that supports the component.
- Deployment descriptors are XML documents that describe configuration and other deployment settings to the J2EE Server.

J2EE Architecture: Source

- http://java.sun.com/j2ee/tutorial/1_3-fcs/index.html
- Professional Java Server Programming J2EE Edition By Wrox Author Team

You have successfully completed J2EE Architecture.