Fundamentals of Java Enterprise Components

Session: 1

Introduction to Java EE 7

Objectives

- Explain Java Application model
- Explain multitier applications
- Describe various components of multitier applications
- Describe containers and services provided
- Describe Web services
- Describe various APIs used in Java EE 7 for enterprise and Web applications

Java Enterprise Application Model 1-3

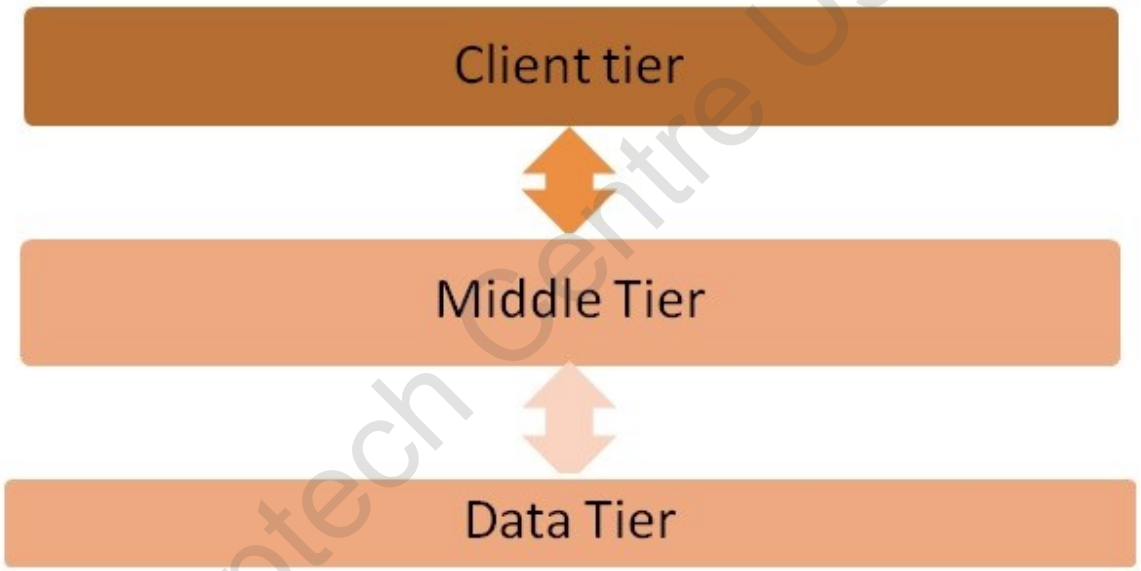
Defines the organization of various application components to create robust and portable applications

Three tier architecture

- Evolved from client server architecture, by adding a middle tier
- Supports different types of clients
- Middle tier is responsible for controlling the data flow between the application/data servers and clients
- Middle tier ensures that the client receives correct data from the database according to the business logic

Java Enterprise Application Model 2-3

Following figure shows an abstract representation of the Java enterprise application model:



Java Enterprise Application Model 3-3

Client tier - includes interfaces through which the application is accessed. Interfaces are developed to be used by customers, employees, suppliers of the enterprise.

Middle tier - implements the business logic of the application through beans and other class files of the application.

Data tier - includes all the enterprise data which is accessed by the application.

Java EE 7 also provides various services such as security, JDBC for database access and so on, which simplify the process of application development.

Multitier Applications 1-2

A multitier application has the application logic divided into multiple components based on their functionality.

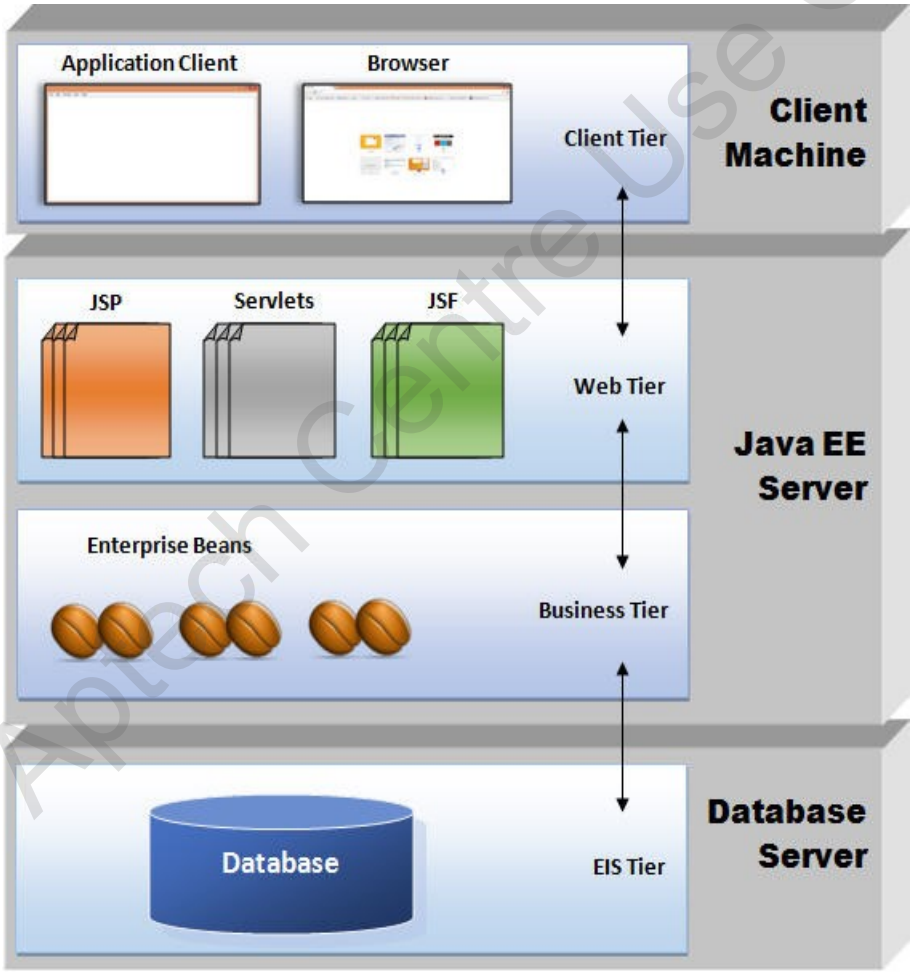
These multitier applications are distributed where each of the application components may be installed on the same or on different machines depending on the tier to which they belong.

Various components of Java EE multitier application are:

- Java EE clients
- Web components
- Java EE components
- Database components
- Security components

Multitier Applications 2-2

Following figure illustrates the multitier architecture and its respective components:



Java EE Clients 1-2

Java EE clients refer to the end users or entities for which the application is designed.

- Web clients
- Application clients
- Applets
- Java Bean components

Java EE Clients 2-2

Web clients refers to user or entities that access the application through Web pages and Web browsers primarily through Internet.

An application client runs on the client machine and has a user interface such as forms developed through Abstract Window Toolkit (AWT) or Swing. These application clients communicate with the business logic which in turn is implemented through EJB.

Applets are small independent programs written in Java for a specific task. They can be embedded in Web pages or run independently.

Java bean components manage the data flow between the application client and components running on the Java EE server and also, the data flow between the server components and a database.

Web Components and Java EE Components

Both Web components and Java EE components implement the business logic for Web applications and business applications respectively.

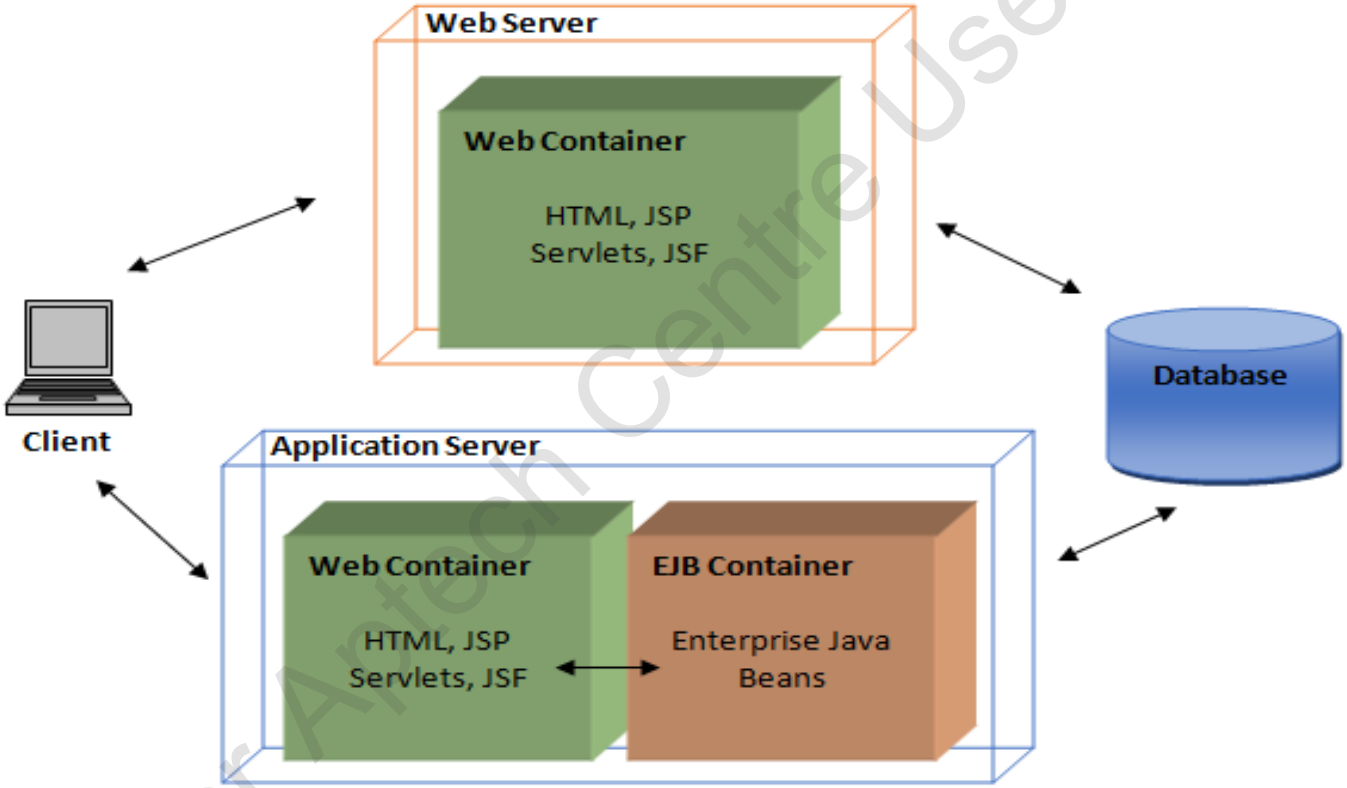
- Servlets, JSPs, and JSFs are Web components in a Web application.
- Static HTML pages are also Web components.
- Enterprise beans implement the business logic in case of Enterprise applications.

The Enterprise Information tier comprises the database of the application.

Security component refers to the implementation of the security methodologies pertaining to the application data.

Web and Application Servers 1-2

Following figure shows the Web and Application servers:



Web and Application Servers 2-2

The Web server consists of a Web container that holds all the Web components.

The application server consists of Web container as well as an EJB container that holds all the enterprise Java bean components.

- Web servers accept requests from clients through Web pages and respond to these requests with the appropriate Web page.
- The communication between the client and Web server takes place through Hyper Text Transfer Protocol (HTTP).
- An application server is a software component which is responsible for all the operations of an enterprise and for implementing the business logic of the enterprise.

Containers 1-2

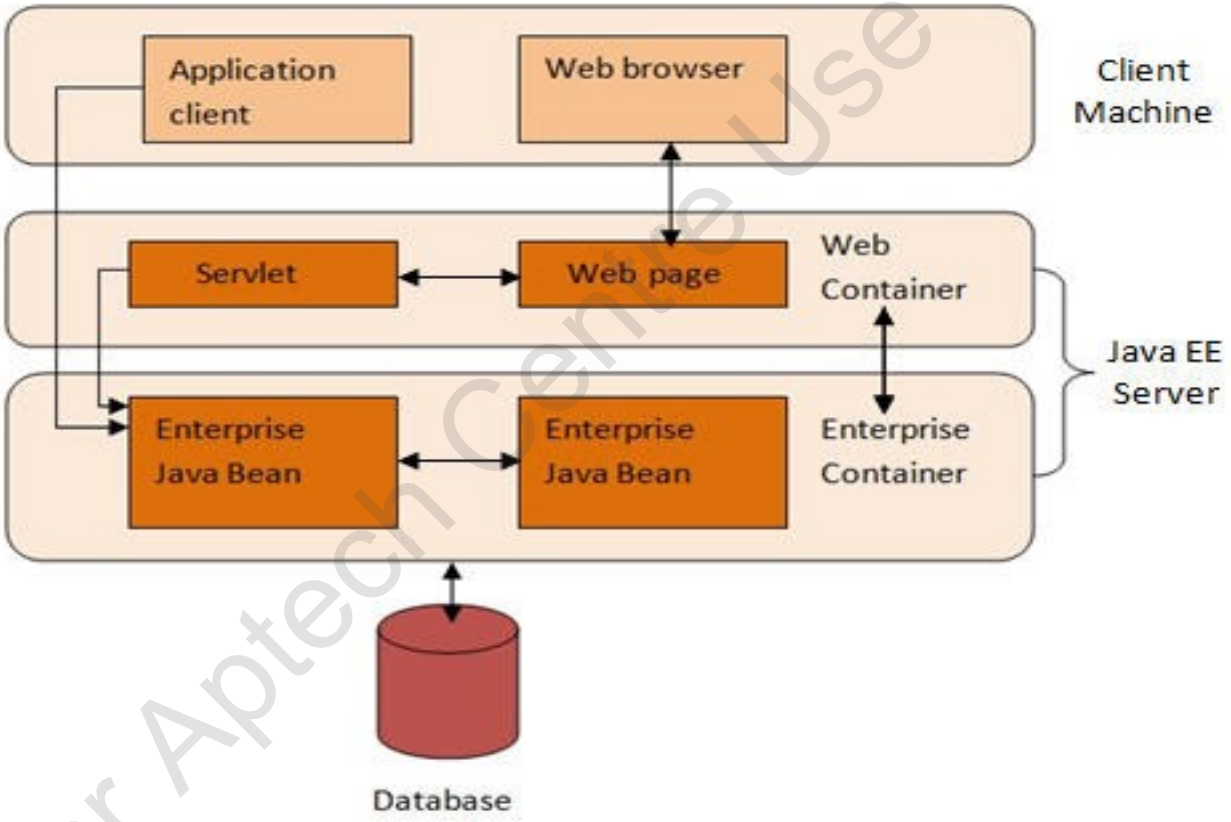
A container contains various components which together provide some services to the application, such as transaction management, state management, resource pooling, thread pooling, security, and so on.

These services help in reducing the effort of developing these functionalities from scratch.

The support and services provided by containers and components enable the developer to concentrate on implementing the actual business logic.

Containers 2-2

Following figure displays containers in Java EE:



Container Services

In order to use the container as per the requirement of the application, it is essential to configure the container settings. Following is a list of services provided by containers in a Java enterprise application:

- Configures components to implement security mechanisms
- Enables linking of methods to a transaction
- Manages connectivity between the client and the server even if the client and server are remotely located
- Java Naming and Directory Interface (JNDI) provides naming service to access different objects and components

The container allows different configurations of the same application component based on the context of the application.

The container manages non-configurable services such as enterprise beans, servlet lifecycles, database access, and so on.

Container Types

The **Web container** is installed on the Java EE server, which comprises all the components of the Web application.

The **EJB container** is also present on the Java EE server, which consists of the EJB components implementing the business logic of the application.

The **Applet container** manages the execution of applets. This container comprises the Web browser and plug-ins.

The **Application Client container** manages various application clients.

Development and Deployment 1-2

Following are the stages in developing a Java application:

- Procuring Java platform and tools for development
- Development of components by application component providers
- The components generated by the application providers are assembled by the assemblers and further deployed by the deployers

Product providers are the vendors who develop Java EE APIs according to standard specification.

Tool providers are those vendors who provide application development tools such as Eclipse, NetBeans, and other tools for assembly, deployment, and packaging.

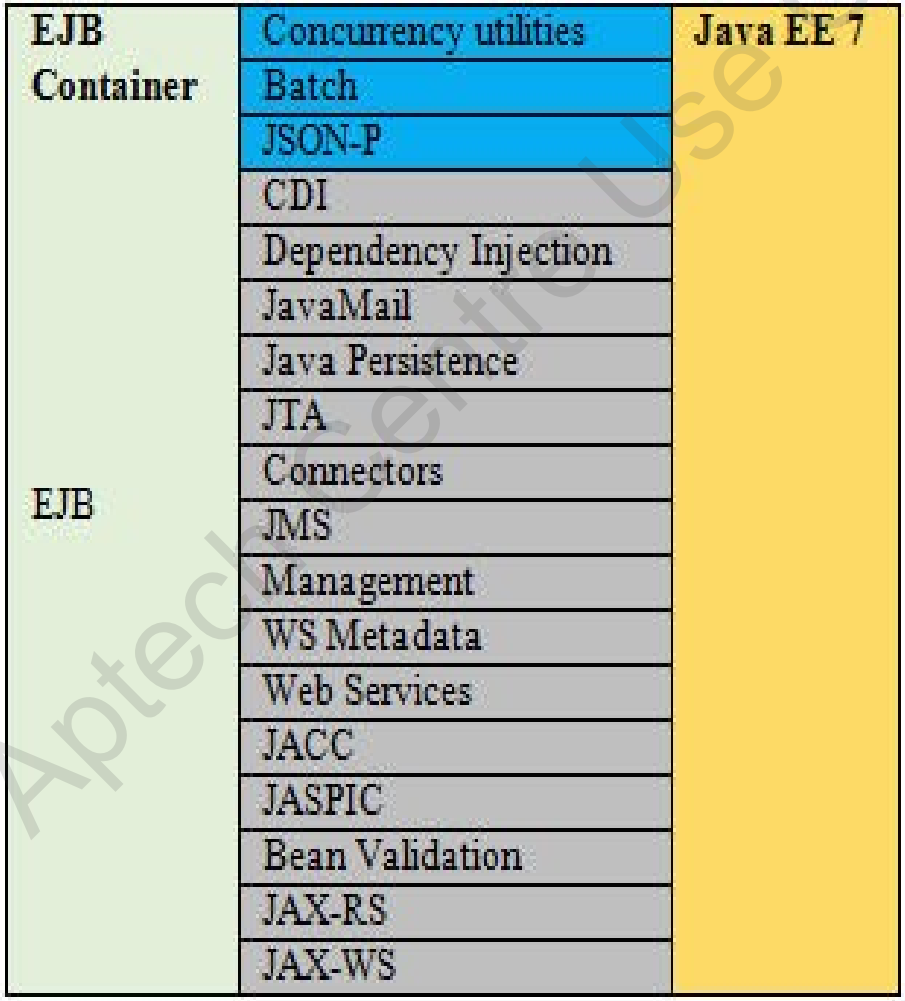
Development and Deployment 2-2

Following are the various developer roles that provide these components:

- **Bean developer** - writes and compiles the source code, creates the deployment descriptor, and packages the .class file as an EJB JAR file.
- **Web component developer** - writes source codes for servlets, JSPs, JSFs, and HTML files. The .class files, .jsp files, and .html files created are packaged as WAR files.
- **Application assembler** - receives all the components from the application component providers and assembles the respective JAR and WAR files to EAR (Enterprise Archive) file.
- **Application deployer** - deploys the application in the operational environment.

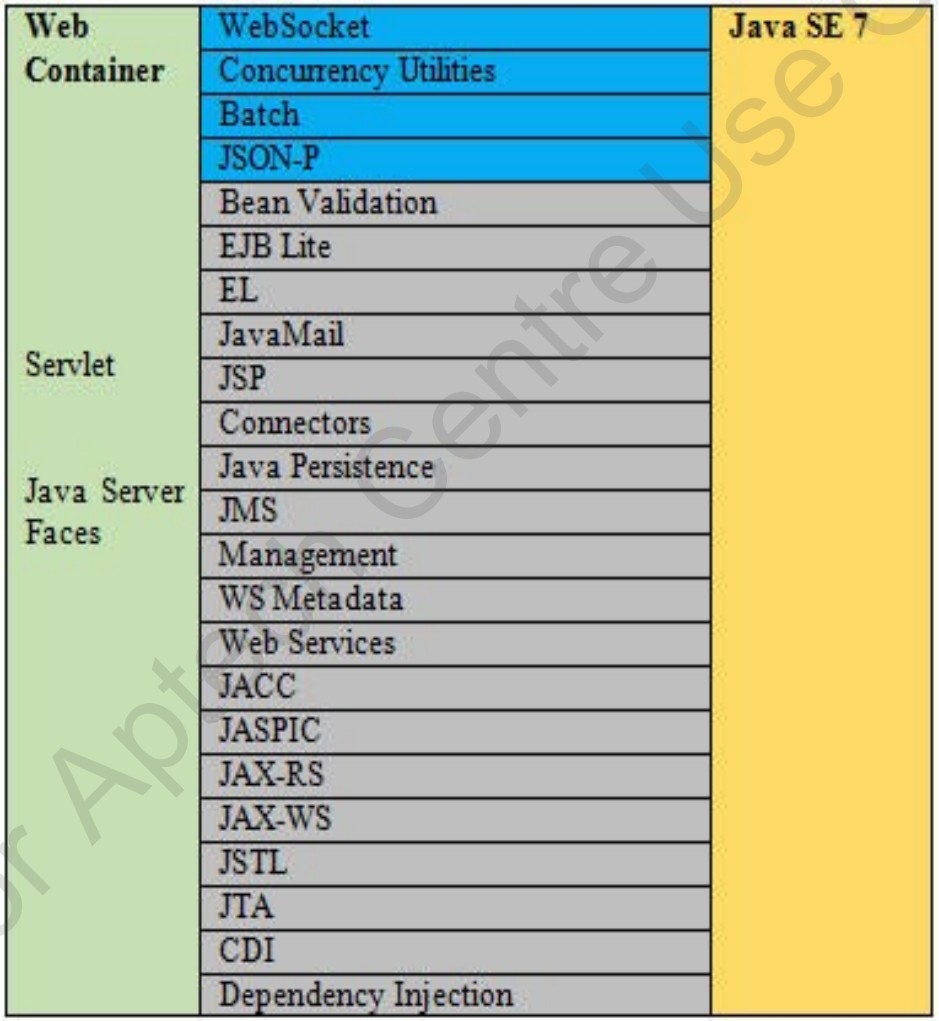
APIs in Java EE 7 1-7

Following figure displays the APIs used in EJB containers:



APIs in Java EE 7 2-7

Following figure displays the APIs used in Web containers:



APIs in Java EE 7 3-7

Java Server Pages Standard Tag Library

- Comprises tags used for formatting the Web page, tags for iterating and managing the flow control, and so on.

Java Persistence API

- Defined to bridge the gap between object-oriented interpretation of data and relational data in databases.

Java Transaction API

- Performs transaction management in the application database.

APIs in Java EE 7 4-7

Java API for Restful Web Services

- Provides means to develop Web services according to REST architecture.

Managed Beans

- Responsible for reporting events such as faults, state changes, performance of the application, and so on.

Contexts and Dependency Injection in Java EE

- Enables addition of components into the context of the application lifecycle based on certain events and ensure that dependencies hold good.

APIs in Java EE 7 5-7

Bean Validation

- Responsible for validating the data in beans in various tiers of application.

Java Messaging Service API

- Used by the components of application to create, send, receive, and read messages.

Java EE Connecter Architecture

- Used for creating resource adapters that will access Enterprise Information Systems.

APIs in Java EE 7 6-7

Java Mail API

- Enables sending mail notifications.

Java Authorization Contract for Containers

- Defines a contract between the Java EE components of the application and authorization policy providers.

Java API for Web Socket

- Helps in creating the communication end point and allows specifying the configuration parameters.

APIs in Java EE 7 7-7

Java API for JSON Processing

- Enables Java EE applications to parse, transform, and query data from JSON using the object model or streaming model.

Concurrency Utilities

- Provide asynchronous capabilities to enterprise application components.

Batch Application Framework

- Enables creating and executing batch jobs.

Summary

- The Java enterprise application model defines how various components of an enterprise application are organized.
- A component can be defined as a fully functional software unit which can communicate with other components.
- A Web client consists of two parts – dynamic Web pages and Web browsers.
- A container contains various components which together provide some services to the application. The services include transaction management, state management, resource pooling, thread pooling, security, and so on.
- The EJB container is also present on the Java EE server, which consists of the EJB components implementing the business logic of the application.
- Application assembler receives all the components from the application component providers and assembles the respective JAR and WAR files to EAR (Enterprise Archive) file.
- Java Persistence API is used for bridging the gap between object-oriented interpretation of data and relational data in databases.