Unit 4 – JEE Technology Concepts 10 hrs

Multi-tier architecture for application development – Meaning, need, advantages. Meaning of enterprise application and web application, various tiers in enterprise application – client tier, web tier, business tier, and Enterprise information system tier. Introduction to JEE concepts – Need, advantages, characteristics of JEE technology, the concepts of containers, components and services – meaning of web container, application client container, EJB container.

Multi-tier architecture for application development: [definition/meaning]

Tier- A tier is an abstract concept that decides a group of technologies that provides one or more services to its clients.

Multi-tier [n-tier] –multi tier architecture is a combination of certain tiers, in which the functionality of the application is divided into logical components that are associated with each tier. Here each component is a service that is built and maintained independently. These services are bound by protocols which enables the services.

Clients, Resources and components:

Multi tier architecture is composed of clients, resources and components, and containers.

Client-client refers to a program that request service from a component.
Resource-a resource is anything a component needs to provide a service.
Component- a component is part of a tier that consists of a collection of classes or a program that performs a function to provide the service.
Container-a container is software that manages a component and provides a component with system services.

The relationship between a container and a component is sometimes referred to as a contract, whose terms are governed by an application programming interface (API).

An API defines the rules a component must follow and the services a component will receive from the container.

A container handles persistence, resource management, security, threading and other system level services for components that are associated with the container.

Components are responsible for implementation of business logic .Which enables the programmers to focus on encoding business rules into components, without concern of low-level system services.

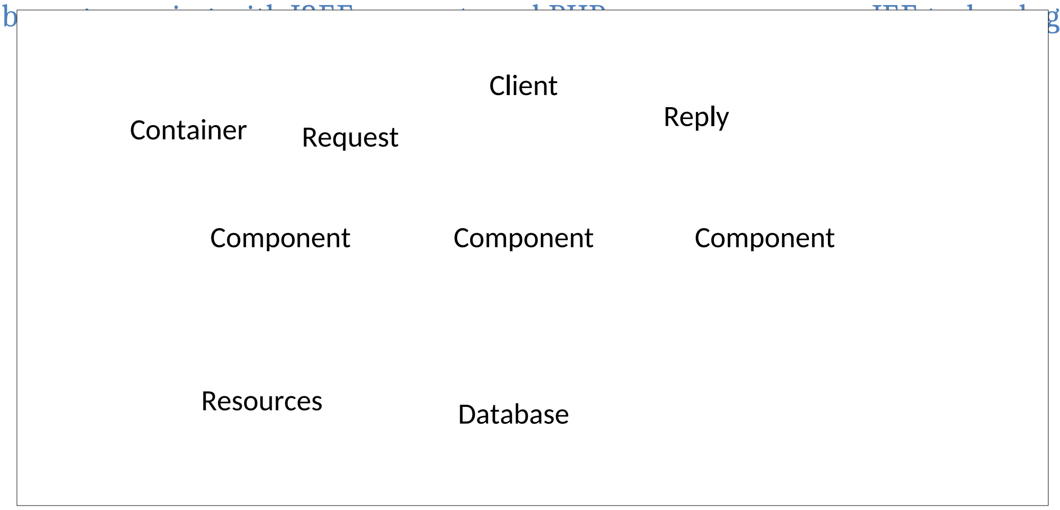


Fig [1]: client, component and database relationship in multi-tier architecture.

Multi-tier Architecture: [need]

J2ee has four tier architecture .These consists of the client tier [also referred as presentation or application tier], web tier, enterprise java beans tier [also referred as business tier], and the information system tier.

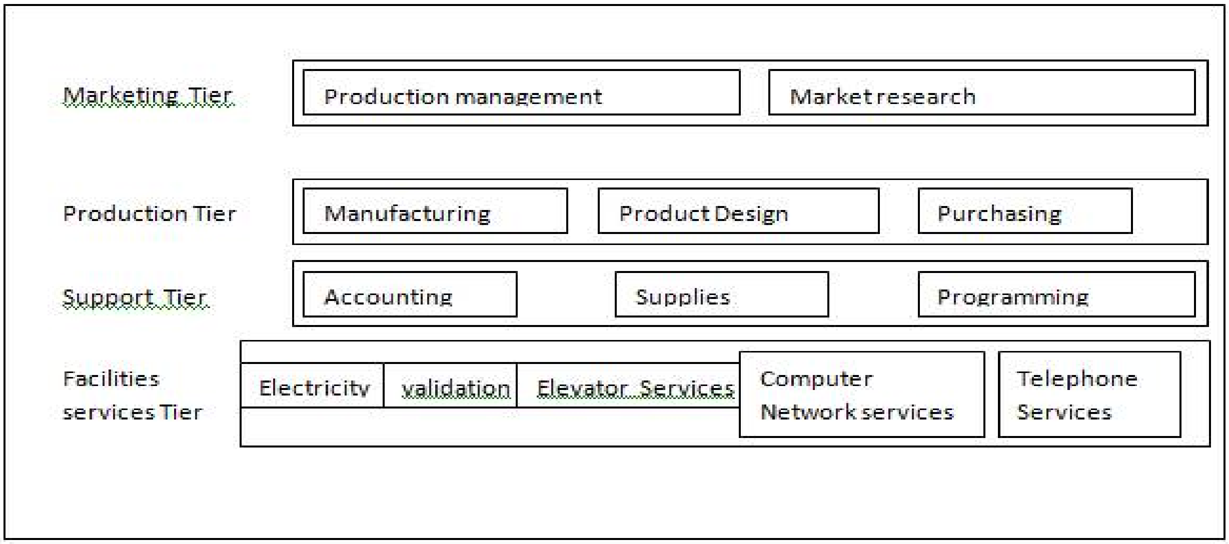
In multi tier architecture, each tier contains services that include software objects, DBMS, or connectivity to legacy systems.

The information systems in multi-tier architecture are cost –efficient. Here functionality is divided into logical components that area associated with each tier.

Here the client request can access service without knowing the functionality of the service. If the functionality of the service is modified, it would not affect the client request response.

Hence there is need for multi-tier Architecture.

Fig 2: Tier architecture in distributed systems.



Advantages of the Multi-Tier Architecture:

- **Provides Security:** we can secure each of the tiers separately using different methods. The main security resides on the server side, which makes the multi threading process more secure whenever the java beans accessed. The access control list [ACL] doesn't allow direct access to the resources, providing security.
- **Easy to manage:** we can manage each tier separately, adding or modifying each tier without affecting the other tiers. For example programmer can change the functionality of the service without affecting the client request.
- **Scalability:** If we need to add more resources, you can do it per tier, without affecting the other tiers. For example resources added to the databases effectively.
- **Flexibility:** Apart from isolated scalability, we can also expand each tier in any manner that the requirements dictate. As multi-tier architecture interfaces with variety of technologies, connectors.
- **Object distribution:** Enterprise JavaBeans are contained on the enterprise JavaBeans server, which is a distributed object sever that works on enterprise Java Bean tier. Only server object might update not the objects on client.
- **Availability:** The EIS [enterprise information system] link a j2ee application to resources and legacy systems are made available, which are corporate backbone network. This availability makes mainframes, which are part of the mission-critical systems are kept operational.

Enterprise Application [Meaning]: Enterprise java application contains server, which stores and manages enterprise Java beans .Majorly implemented on government or business fields. This provides concurrency, scalability, lifecycle management, and fault tolerance. This also manages instances of components.

Web application [Meaning]: web applications are the crucial applications for any client server architecture. Basically they run on browser and conforms HTTP, implements web based languages like HTML, Java script, CSS etc.

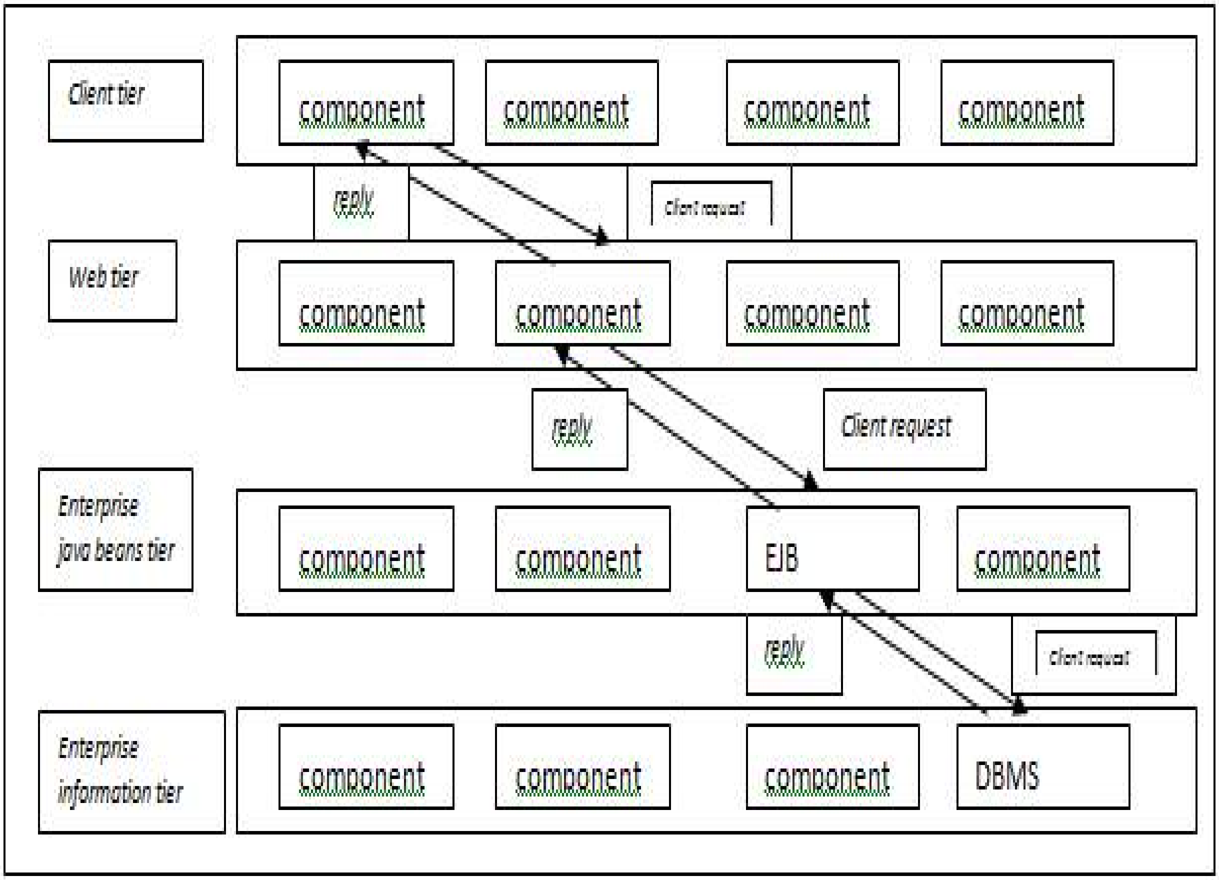
Various Tiers in enterprise application [or Multi –tier or JEE architecture]:

There exist four tiers in enterprise application. This is also referred as multi-tier architecture or n-tier architecture.

- Client Tier
- Web Tier.
- Enterprise Java beans Tier or Business tier.

- Enterprise Information system Tier.

Fig 3: architecture of enterprise application /or multi tier architecture.



Client Tier:

This tier is also referred as presentation tier or application tier. There exist two types of client .Applet clients and application clients. Applet client is a component used by web client in the applet container.

An application client is a Java application that operates within the application client container of JRE standard.

Web client consists of software, usually a browser, which access resources located on the web tier. These resources are typically XML or HTML pages.

EJB [enterprise Java Bean] client interfaces the J2EE application with the user.

EIS [enterprise Information system] clients are the interfaces between users resources located on EIS tier.

Multi-Tier clients can access components located on tiers other than the tier where the multi-tier client resides.

Web Tier:

Web tier acts as an intermediary between components working on the web tier and other tiers. The intermediary activities include

- Accepting requests from other software that was sent using POST, GET, and PUT operations, which are part of HTTP transformations.
- Transmit data such as images and dynamic content.

Two types of components work on web tier. These are Servlets and Java Server Pages[JSP].a Servlet is java class that resides on the web tier and is called by request from browser client that operates on the client tier.

The servlet is associated with a URL that is mapped by the servlet container.

Enterprise Java bean Tier or business tier:

The enterprise java bean contains server, which is object server that stores and manages enterprise JavaBeans .This provides concurrency, Scalability, lifecycle management , and fault tolerance.

This tier automatically handles concurrency issues that assure multiple clients have simultaneous access to the same object.

Enterprise JavaBeans tier manages instances of components. It also has container within the collection of enterprises java beans. This also implements certain business logic which provides services such as,

- Resource pooling.
- Distributed object protocols.
- Thread management.
- Security.

Enterprise information system tier:

Enterprise information system tier provides connectivity to the resources such as DBMS, legacy systems provided by third parties etc.

The EIS provides the connectivity between a J2EE application and non-J2EE software. This includes rules for connecting each other and conducting secured transactions.

Introduction to J2EE concepts:

J2EE stands for Java 2Enterprise Edition .this version was developed in order provide platform independent, web-centric applications. This is mainly used to develop web based applications, API's which are portable and scalable.

Need of J2EE:

- The web applications in earlier days needed workers to be engaged on server side 24*7.which were dependent on certain middleware.J2ee simplifies the creation of enterprise-wide application, because the functionality is encapsulated within the components of j2ee.

- The collaboration of the industry leaders resulted in j2ee standard enterprise environment. This provides uniformity for the clients or the vendors of J2ee applications.
- The uniformity of the standard edition also helped in cost effectiveness of j2ee applications by customizing them as for the requirements of the clients.
- J2ee is a versatile technology because application components built using j2ee are able to communicate with each other behind the scenes using standard communications methods such as HTTP, SSL, HTML, XML, RMI, and IIOP.
- J2ee applications are written in java enabling the same java programmers to work in the multi tier environment. This is helpful in multi threading and synchronization.
- Java beans, java servlets and java server pages are components of j2ee. This also consists of seven standard services, these enable efficient web development environment.

Advantages of J2EE:

- Simplified architecture and development
    - Component-based, labour division, dynamic assembly / deployment.
- Scalability to meet demand variations
    - Transaction support, DB connection pooling, load-balancing.
- Integration with existing information systems
    - Integration APIs for: DBs, mail, CORBA, messaging, directories.
- Choices of servers, tools, components
    - Server choices, tool (IDE) support, component market place.
- Flexible security model
    - Support a wide range of security requirements.
- J2ee provide solid backend for wireless applications .hence many wireless application companies use j2ee for this purpose.
- J2ee provides standardized development. So that the code will be uniform for the third party as well as for the vendors.
- J2ee has well built infrastructure, which is easily deployed in application which needs messaging and security features.
- J2ee is fully-fledged server side development platform with all the necessary provisions like JSP, servlets, applets etc.

Characteristics of j2EE:

- **J2ee is component based model:**

Here components are the basic units for the module .Which is collection of classes and functions.

- **Container provided services:**

A container is software that manages a component and provides component with system services.

- **Highly Scalable:**

The most advantageous feature of J2ee is it is scalable from smaller to larger applications.

- **Simplified Architecture:**

J2EE has got simplified architecture, which is delineate with respect to different tiers and legit.

- **Flexible security model :**

J2ee provides two types of security declarative security and programmatic security. Declarative security is with respect to the containers and programmatic security is applied when declarative security is not sufficient.

Concept of component and services:

A component is part of a tier that consists of a collection of classes or a program or other containers that performs a function to provide the service. Several different components are grouped together to form the tier in a multi tier architecture.

Services are provided by the containers of the components. There might be several services provided depending the type of the containers. Such as web container, servlets, applets , EJB container etc.

Concept of containers:

A container is software that manages a component and provides a component with system services. This could also be referred as the sub class of the component. The application server which control and provide services through an interface is known as a *container*.

J2EE container helps in deployment and execution of J2EE component. In J2EE specification we can classify container types as five.

Three of these are server-side containers:

- The server, which provides the J2EE runtime environment
- An EJB container that manage EJB components.
- A Web container that manage servlets and Java Server Pages.

The other two containers are client-side:

- An application container for standalone GUIs, console, and batch-type programs the familiar Java applications started with the java command.
- An applet container, meaning a browser, usually with the Java Plug-in.

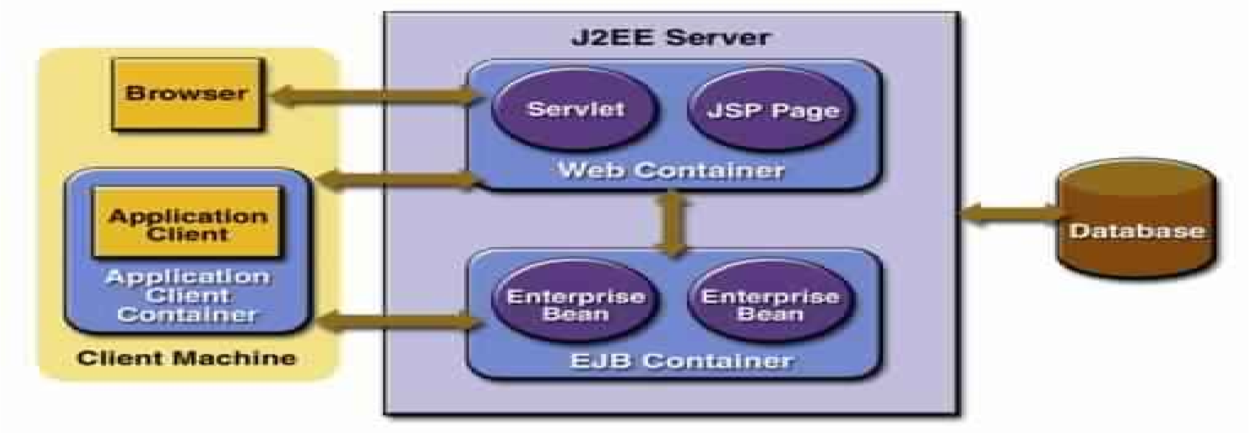


Fig 3: server and containers

- **J2EE server**: The runtime portion of a J2EE product. A J2EE server provides EJB and Web containers
- **Applet Container:** Applet container includes support for the applet programming model. The applet container is a mixture of web browser and java plug in on client machine.
- **Web Container**: A Web application runs within a Web container of a Web server which is also known as Servlet container. The Web container provides the runtime environment through components. Web container manages the execution of JSP page and Servlet components for J2EE applications. A web container provides the same services as a JSP container .Web components and their container run on the J2EE server. Web server is a server which is capable of handling HTTP request send by a client and respond back with a HTTP response.
- **Enterprise Java Bean (EJB) Container**: EJB container is server side component architecture. It is mainly used for modular construction of enterprise application. An EJB container provides a run-time environment for enterprise beans within the application server.

  The container handles all aspects of an enterprise bean's operation within the application server. EJB container acts as an intermediary between the user-written business logic within the bean and the rest of the application server environment. It manages the execution of enterprise beans for J2EE applications.

  Enterprise beans and their container run on the J2EE server. The EJB container provides local and remote access to enterprise beans. Enterprise bean can be divided into session beans, entity beans, and message-driven beans depend on the data. Session beans represent transient objects and processes and typically are used by a single client.

Entity beans represent persistent data, typically maintained in a database. Message-driven beans asynchronously pass messages to application modules and services.

- **Application Client Container:** To host application components, application clients, the application client container is used. It runs on the client computer and it can be interacted with each other.

  It helps developers to create robust Java applications that also have access to J2EE resources such as data sources or EJBs. This container manages the execution of application client components.

  Application clients and their container run on the client. Application server can handle all application operations between users and databases.

J2ee advantages

1. Portability

As we know that the servlets are written in java and follow well known standardized APIs so they are highly portable across operating systems and server implementations. We can develop a servlet on Windows machine running the tomcat server or any other server and later we can deploy that servlet effortlessly on any other operating system like Unix server running on the iPlanet/Netscape Application server. So servlets are write once, run anywhere (WORA) program.

2. Powerful

We can do several things with the servlets which were difficult or even impossible to do with CGI, for example the servlets can talk directly to the web server while the CGI programs can't do. Servlets can share data among each other, they even make the database connection pools easy to implement. They can maintain the session by using the session tracking mechanism which helps them to maintain information from request to request. It can do many other things which are difficult to implement in the CGI programs.

3.Efficiency

As compared to CGI the servlets invocation is highly efficient. When the servlet get loaded in the server, it remains in the server's memory as a single object instance. However with servlets there are N threads but only a single copy of the servlet class. Multiple concurrent requests are handled by separate threads so we can say that the servlets are highly scalable.

4.Safety

As servlets are written in java, servlets inherit the strong type safety of java language. Java's automatic garbage collection and a lack of pointers means that servlets are generally safe from memory management problems. In servlets we can easily handle the errors due to Java's exception handling mechanism. If any exception occurs then it will throw an exception.

5.Integration

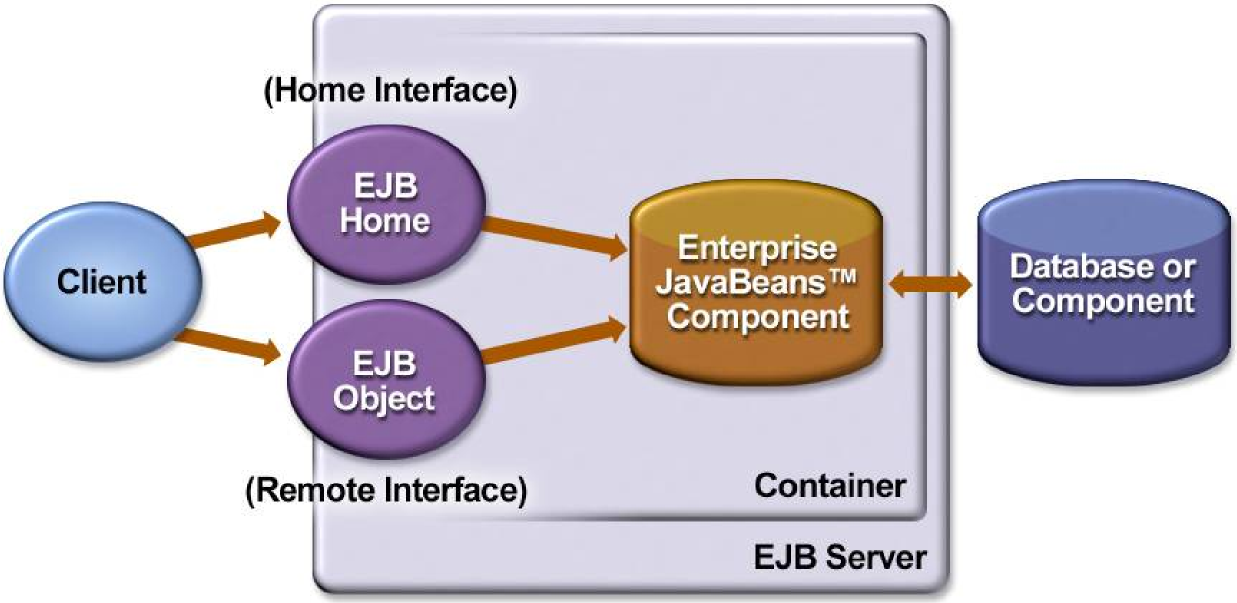
Servlets are tightly integrated with the server. Servlet can use the server to translate the file paths,
perform logging, check authorization, and MIME type mapping etc.

6.Extensibility
The servlet API is designed in such a way that it can be easily extensible. As it stands today, the servlet API support Http Servlets, but in later date it can be extended for another type of servlets.

7.Inexpensive
There are number of free web servers available for personal use or for commercial purpose. Web servers are relatively expensive. So by using the free available web servers you can add servlet support to it.

Why EJB Technology?

Leverages the benefits of component-model on the server side
Separates business logic from system code
−Container provides system services
Provides framework for portable components
−Over different J2EE-compliant servers,Over different operational environments
Enables deployment-time configuration
−Deployment descriptor


(Home Interface)
Client
EJB Home
EJB Object
Enterprise JavaBeans™ Component
Database or Component
(Remote Interface)
Container
EJB Server