Session: 1

Introduction to Business Components

Objectives

- Explain enterprise applications
- Describe the problem faced by large enterprise applications
- Explain distributed application architecture used for developing business components
- Describe Java EE platform and its API
- Describe application development architecture of enterprise application
- Describe application server containers and their services
- List various Java EE application servers
- Describe Java EE profiles and the use of EJBLite on application servers

Introduction

- **Java Enterprise Edition (Java EE)** enables enterprise application development on Java platform.

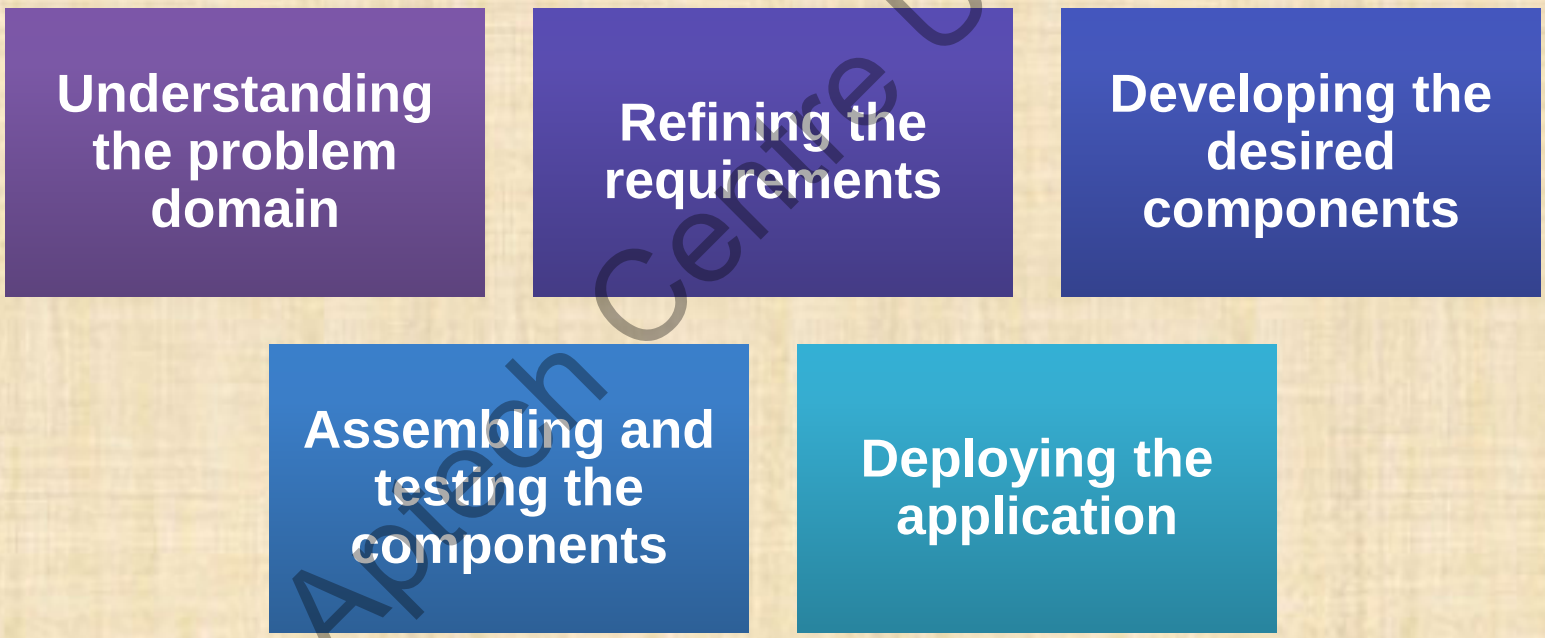
Enterprise Applications 1-2

What is an Enterprise application?

- **An enterprise application:**
  - Is a large business application.
  - Is developed to fulfil the needs of large business domains with complex functionalities.
  - Is usually hosted on servers.
  - Provides services to a large number of users over a computer network.

Enterprise Applications 2-2

- Following figure shows the process of enterprise application development:



Need for Enterprise Applications 1-3

❑ Current Scenario:

- The rapid changes have made businesses to meet the requirement of customers, communicate with other business processes, and incorporate business-to-business services.

- Enterprise applications are developed to satisfy such business needs.

Need for Enterprise Applications 2-3

- Following figure shows some different types of domains involving complex business functionalities:

Banking

Logistics

Finance

Production Scheduling

Order Management

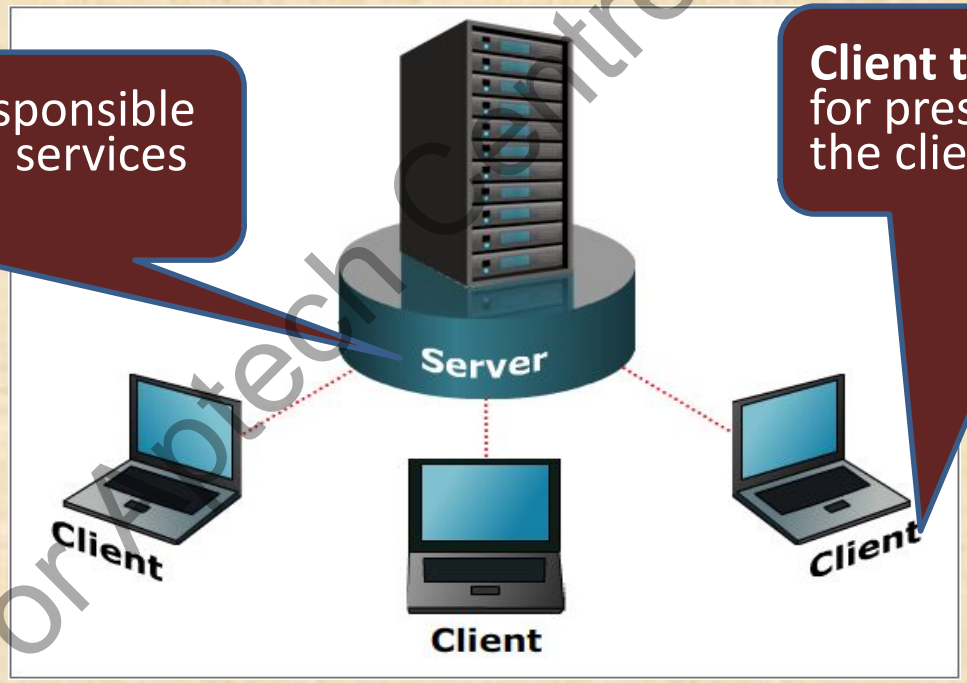
Need for Enterprise Applications 3-3

Concerns and requirements of large businesses:

- Should be long lived applications performing parallel processing.
- Should be functional across platforms.
- Should support complex business processes and domain-based constraints.
- Should follow procedures with respect to security, administration, and maintenance of the applications.
- Should follow complex business requirements that are defined through policies, constraints, rules, processes, and entities in the domain.

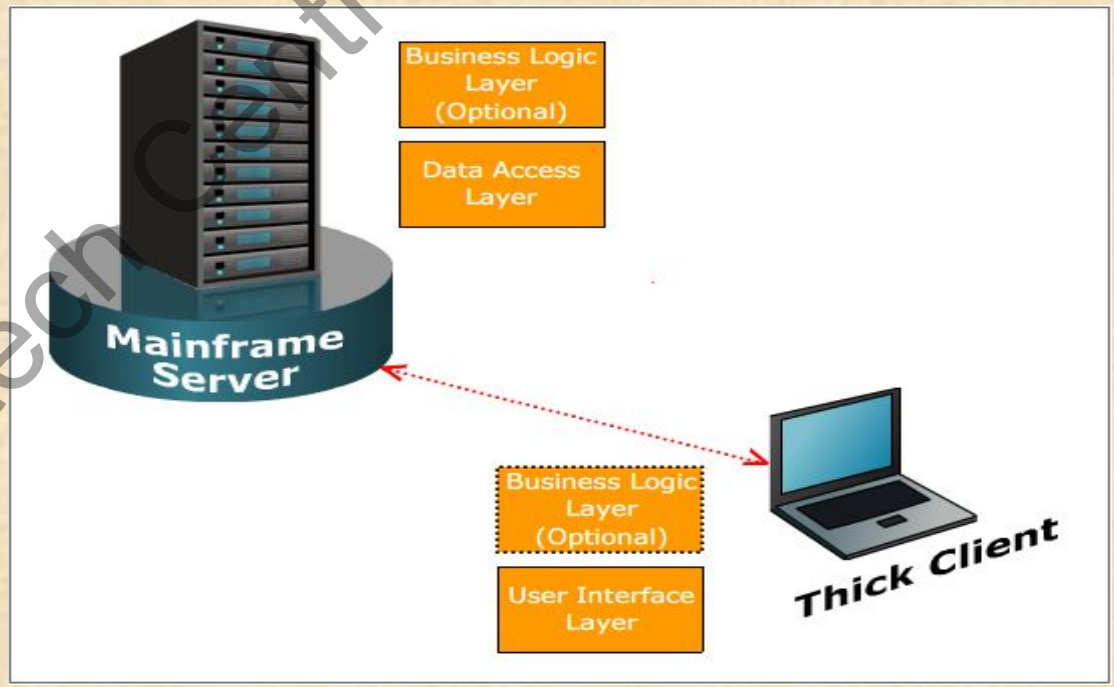
Enterprise Application Architecture 1-5

- Application architecture divides application components into tiers or layers based on their functionality in the environment.
- Following figure shows a typical client-server architecture:



Enterprise Application Architecture 2-5

- There is another variant of two tier architecture where the client is a thick client.
  - A thick client contains some business logic residing on it.
  - A significant part of the executable code resides on the client-tier.



Enterprise Application Architecture 3-5

- Some of the drawbacks with the client-server architecture are as follows:
  - If the server has multiple clients trying to access the server, then there might be resource contention resulting in response delays.
  - Any changes in the user interface have to be updated at all client nodes.

To overcome these problems, **distributed application architecture** has been introduced.

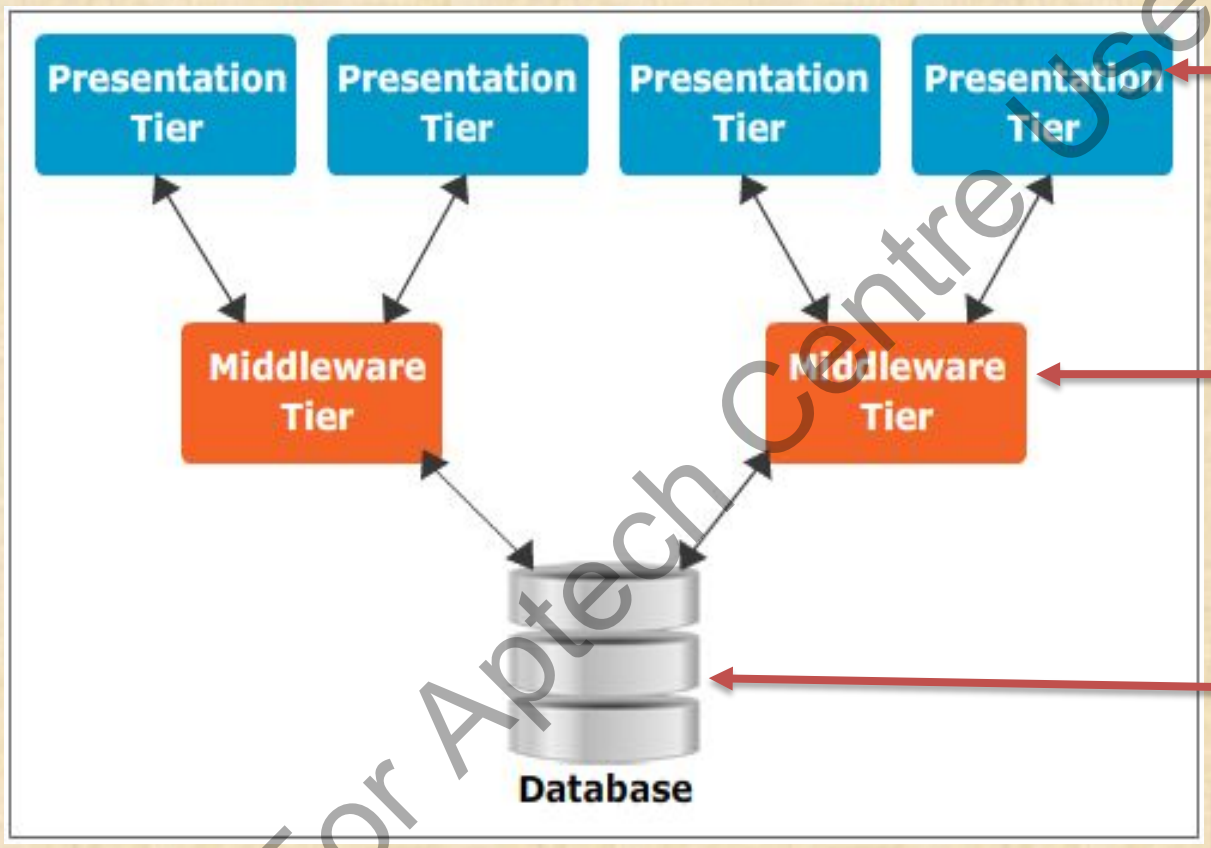
Enterprise Application Architecture 4-5

What is a Distributed application architecture?

- **Distributed application architecture:**
  - Divides the large monolithic application into layers.
  - Introduces middleware layer whose main purpose is to bridge the gap between different hardware systems.

Enterprise Application Architecture 5-5

- Following figure shows distributed architecture of applications:



Client Tier

Initiates service requests

Business Tier

Implements business logics and communicates with clients

Enterprise Information System (EIS)

Holds all the data pertaining to the application

Requirements of Distributed Applications

- Reusability
- Remote methods
- Resource pooling
- Multi-user
- Access control
- Failover support
- Transactional
- Shared data
- Logging and auditing
- Security

Java in Enterprise Development

- Following figure shows different editions of Java used in application development:

Java EE Platform Stack

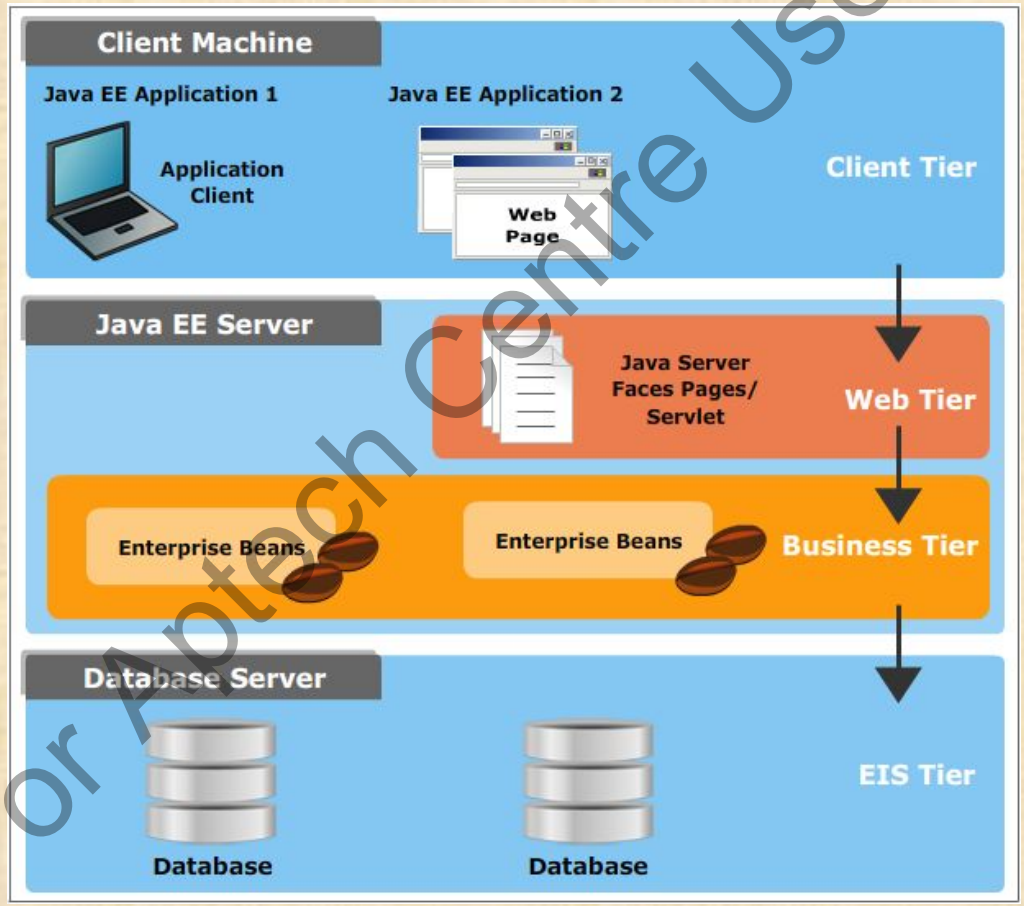
- Java EE platform is the application environment where developers can write Java enterprise applications.
- It comprises the following components:
  - A set of specifications for the platform
  - Implementation of specifications
  - Java EE software development kit
  - Java EE components and applications

Java EE Application Model 1-3

- Following are the application tiers implemented in most applications:
  - Client tier
  - Middle tier which in turn may have a Web tier, business tier
  - Enterprise Information Systems (EIS) tier

Java EE Application Model 2-3

- Following figure shows a graphical representation of distributed multitier application model for enterprise applications in Java:



Java EE Application Model 3-3

Client tier comprises the application clients or Web pages which access the application.

Web tier is responsible for handling communication of the application components over the Internet.

Business tier implements the business logic of the application.

EIS tier comprises the data repositories of the application.

EJB as Business Components 1-4

- EJBs are server-side components.
- EJBs can communicate with both the end user and the database of the application.
- Processing of data is defined by the business requirements of the application and implemented through EJBs in case of Java.

EJB as Business Components 2-4

- Following are the functions that can be carried out by EJBs:

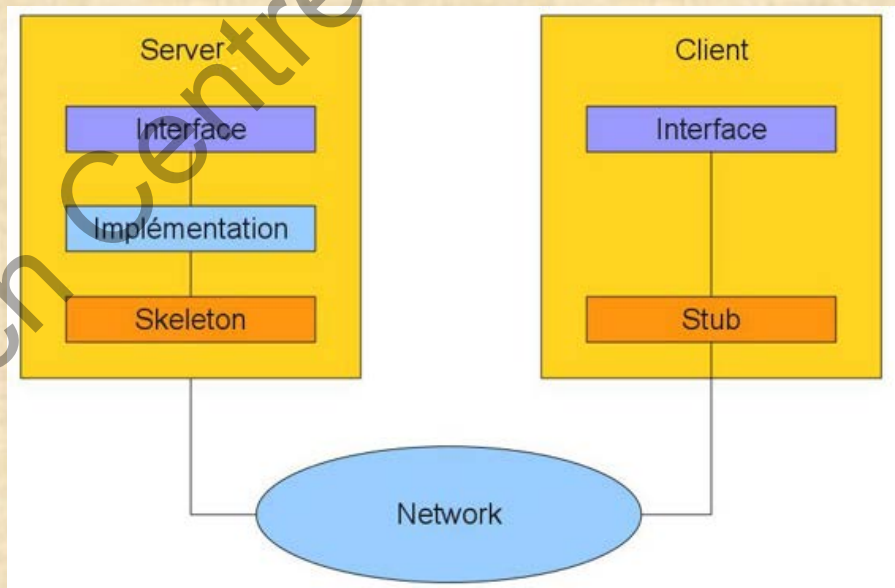
Implements business logic	Access Database
Integrates with other systems	Enables deployment and execution of business components in distributed, multi-user environment

EJB as Business Components 3-4

- Remote components were earlier accessed through Remote Method Invocation (RMI).
- Following are the steps based on which RMI is implemented:
  - When a component has to access another component located over the network, the client invokes a stub.
  - Stub invokes the component through a skeleton.
  - Skeleton extracts the parameters from the request and invokes right methods to generate the response for the request received.
  - Once the response is generated, the response is sent to the client.
- Some drawbacks of RMI are loss of object identity which creates performance bottlenecks and so on.

EJB as Business Components 4-4

- EJB overcomes the drawbacks of RMI and other technologies like Common Object Request Broken Architecture (CORBA) through component and container framework model.

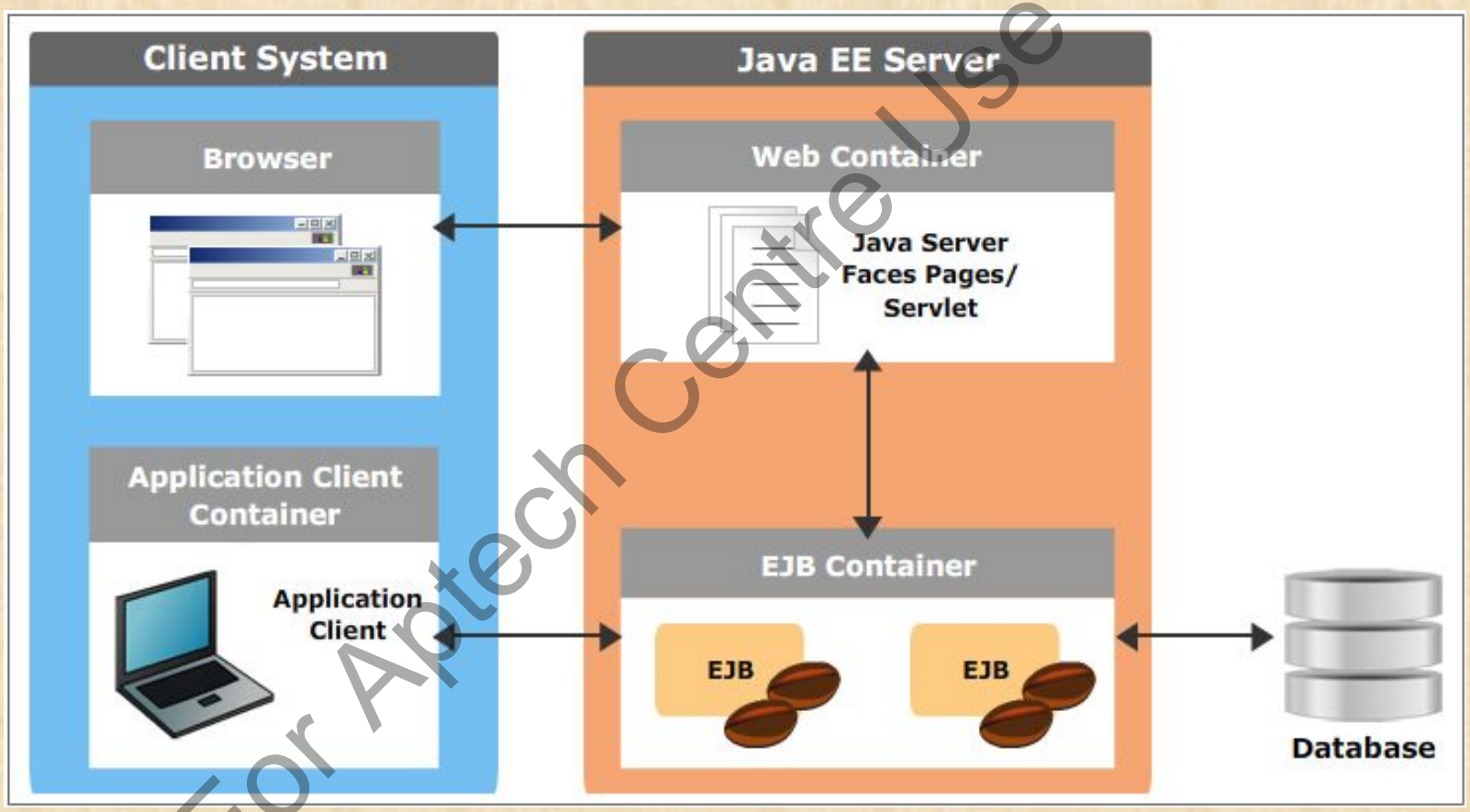


Java EE Containers 1-6

- Java EE provides various underlying services for enterprise application components through containers.
- Containers can be defined as an interface between the application component and the underlying hardware/software on which the component is residing.
- Containers provide various services to the enterprise application components, such as transaction management, security, and so on.
- Containers allows application developers to concentrate on the business logic specific for the application, rather than implementation of services for the application.
- All the application components are deployed into a container before execution.

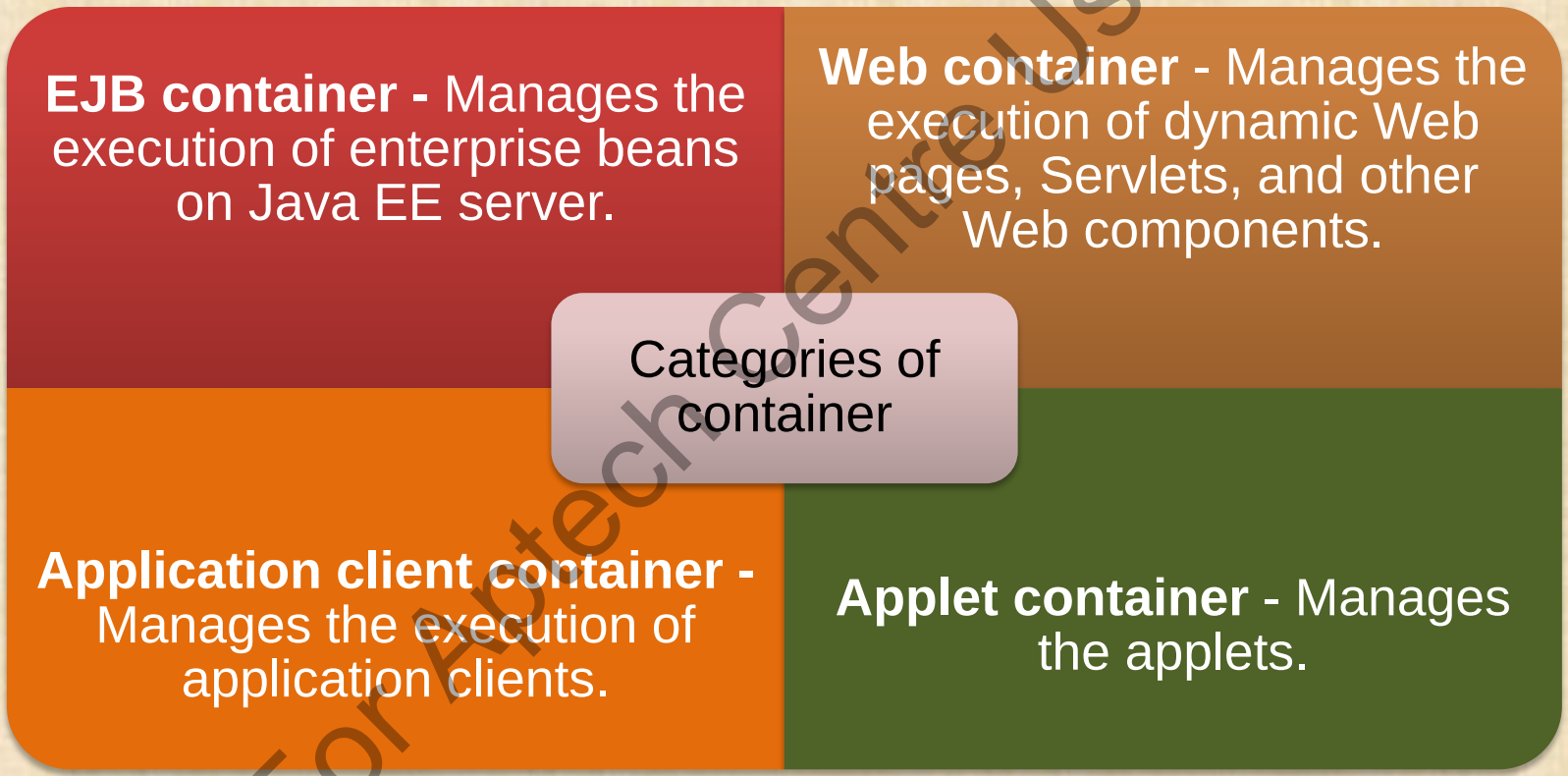
Java EE Containers 2-6

- Following figure shows the different types of containers:



Java EE Containers 3-6

- The container in an enterprise application can be categorized as:



Java EE Containers 4-6

- The services provided by the container to the application component are:

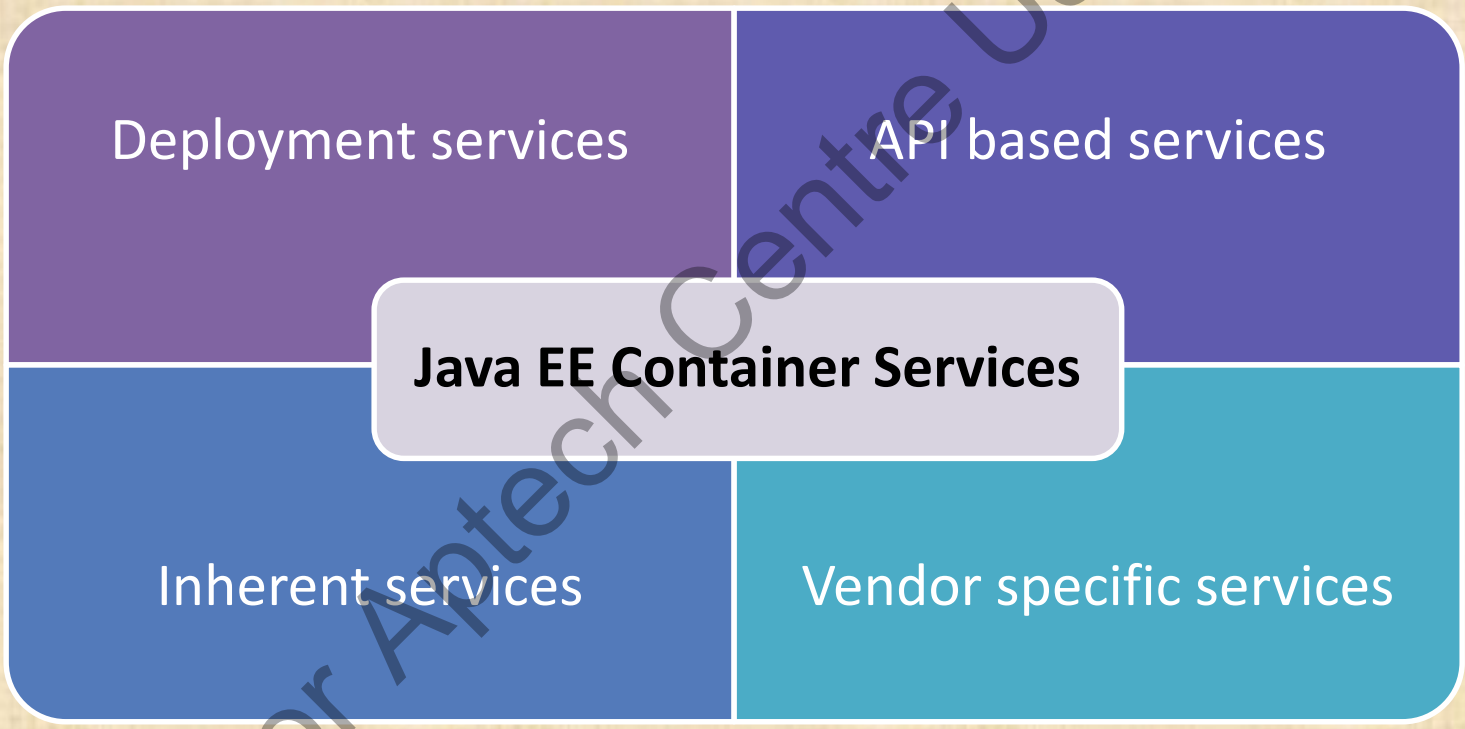
Security	• Provides security model by configuring the container.
Transaction support	• Provides transaction support to the application component.
Java Naming and Directory service	• Provides naming services to ensure that the objects of the application are appropriately accessed.
Java EE communication model	• Provides low level communication between clients and enterprise beans.

Java EE Containers 5-6

- The containers also manages enterprise bean and Servlet lifecycle, database connection, resource pooling, data persistence, and access to the Java EE platform APIs.

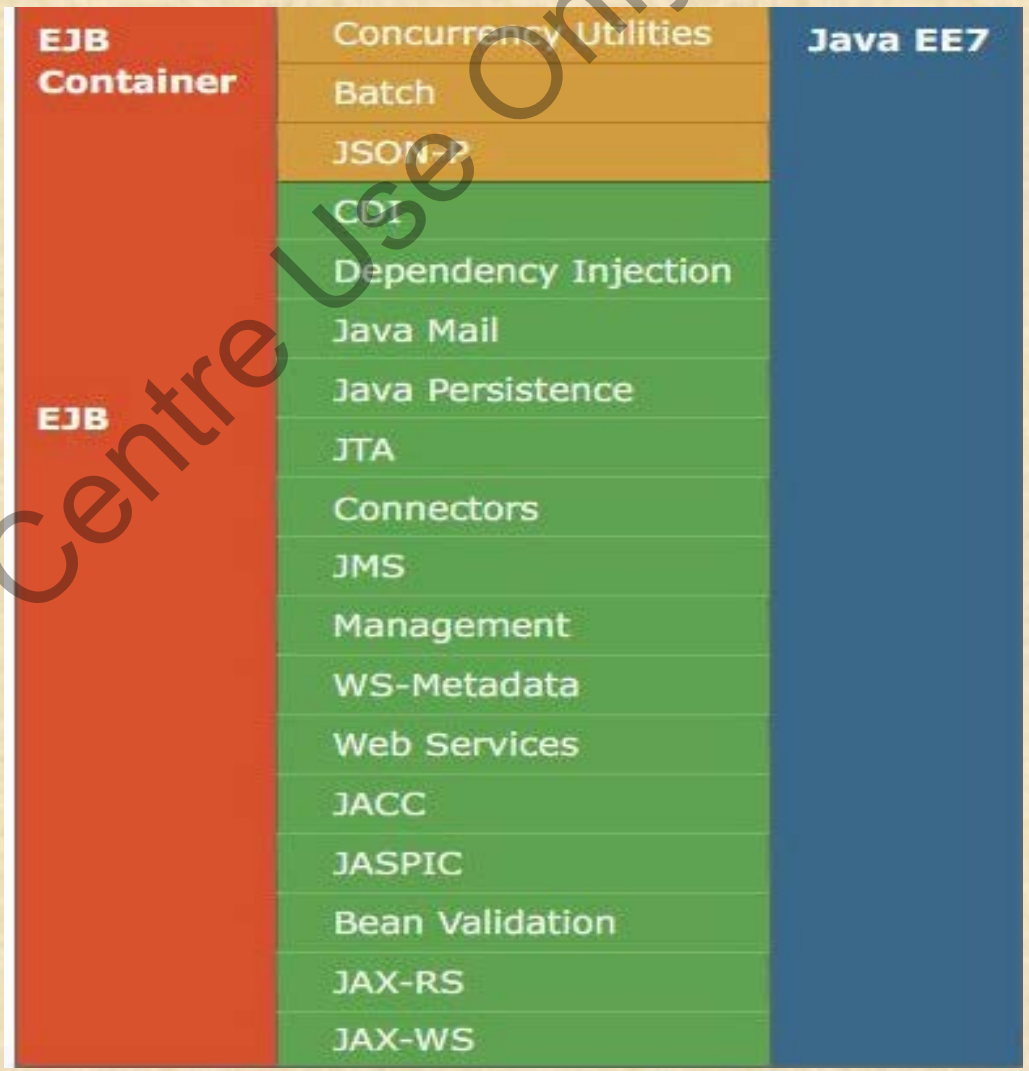
Java EE Containers 6-6

- Apart from these services, J2EE container also provides the following services to the applications it hosts:



Java EE APIs and Services 1-3

- Following figure shows the APIs provided by Java EE 7. The APIs highlighted in yellow are the new ones introduced in Java EE 7.



Java EE APIs and Services 2-3

- Following are the commonly used APIs are as follows:
  - **Java Persistence API** – Uses an object-relational mapping approach to enable interaction between the beans and the database.
  - **Java Transaction API** – Sets the boundaries of the transactions in the application and manages multiple transactions among the components of the application.
  - **Java API for RESTful Web services** – Defines APIs for the development of Web services based on Representation State Transfer architectural style.
  - **Contexts and Dependency injection** – Defines a set of services provided by the container to enable the enterprise beans to function along with the JSF pages.

Java EE APIs and Services 3-3

- **Bean Validation** – Defines a meta data model and an API for validating Java bean components.
- **Java Message Service API** – Enables communication between application components in a loosely coupled, reliable, and asynchronous manner.
- **Java EE Connecter Architecture** – Used for system integration to create resource adapters.
- **Java Mail API** – Provides interfaces for sending and receiving e-mail notifications.

Java EE 7 Software Development Kit

Acts as a base for Java application development.

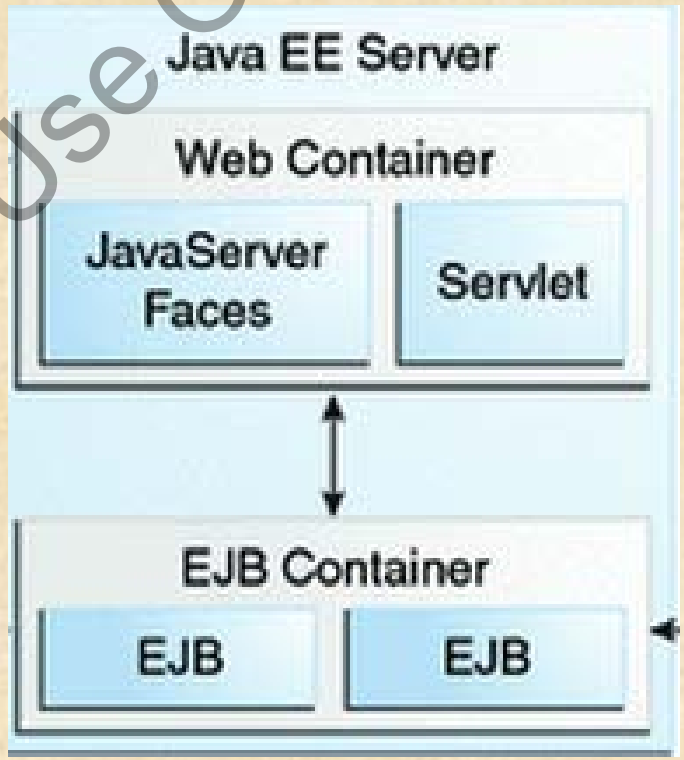
Is a pre-requisite for installation of GlassFish server and NetBeans IDE.

Has debuggers and compilers for java program execution.

Has runtime libraries and basic components for executing Java programs.

Java EE Application Servers 1-3

- Application servers:
  - Are entities of applications on which enterprise applications are deployed and run.
  - Consists of components such as database connectors, Web server connectors, runtime libraries, and so on.
  - Comprise an EJB container and Web container.
    - All beans of an application are deployed in the EJB container.
    - Web components are deployed in the Web container.

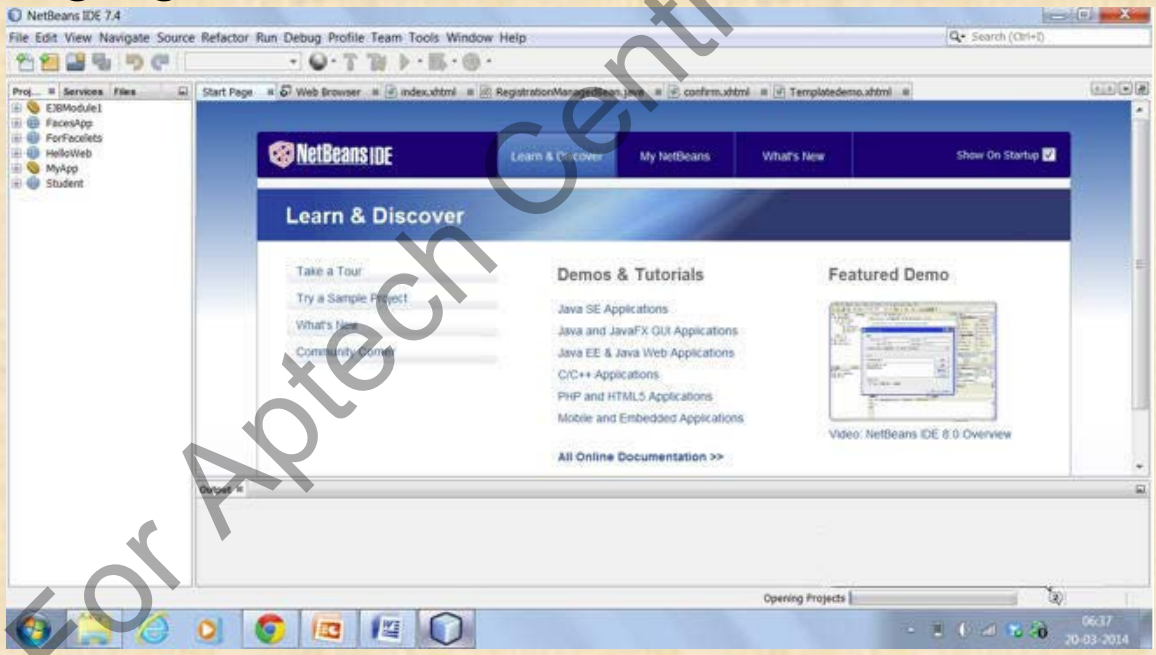


Java EE Application Servers 2-3

- Following are the Java EE application servers:

Java EE Application Servers 3-3

- Java EE application is done in NetBeans IDE.
- NetBeans IDE is an integrated development environment used for development in Java.
- Following figure shows the NetBeans IE interface:



Java EE Profiles 1-4

- A profile is a subset of the libraries of the Java Runtime Environment and requires less compute resources.
- EJBLite is a Java EE profile defined on Java EE7 platform.
- EJBLite has the following features implemented:
  - It implements all the three variants of session beans.
  - It implements local EJB interfaces or no interfaces.
  - Interceptors are part of EJBLite.
  - It supports container managed and bean managed transactions.
  - The security mechanisms are declarative.
  - Supports an Embedded API.

Java EE Profiles 2-4

- Following table shows the comparison of the components in EJB 3.1Lite and Complete EJB 3.1:

Components	EJB 3.1Lite	EJB 3.1
Session	Yes	Yes
Message-Driven	No	Yes
2.x/1.x	No	Yes
Java Persistence 2.0	Yes	Yes

Java EE Profiles 3-4

- Following table shows the comparison of the Session Bean Client Views in EJB 3.1Lite and Complete EJB 3.1:

Session Bean Client Views	EJB 3.1 Lite	Complete EJB 3.1
Local	Yes	Yes
No-interface	Yes	Yes
3.0 Remote Interface	No	Yes
2.0 Remote Interface	No	Yes
Web Service End point	No	Yes

Java EE Profiles 4-4

- Following table compares the services provided in the EJB 3.1Lite and Complete EJB 3.1:

Services	EJB 3.1Lite	EJB 3.1
Timer	No	Yes
Asynchronous Session Beans	No	Yes
Interceptors	Yes	Yes
RMI-IIOP interoperability	No	Yes
Transactions	Yes	Yes
Security	Yes	Yes
Embeddable API	Yes	Yes

Summary

- An enterprise application is a large business application. It is usually hosted on servers and simultaneously provides services to a large number of users over a computer network.
- Application architecture divides application components into tiers or layers based on their functionality in the environment.
- Distributed application architecture divides the large monolithic application into layers.
- As enterprise application execution is distributed across multiple tiers, there are various issues or concerns with respect to the design of these applications.
- The purpose of Java EE platform is to build large scalable, multi user applications.
- The business logic of an application is implemented by EJBs by using various supporting APIs provided by Java EE.
- EJB is based on component framework model where the application comprises communicating components.
- Enterprise application components are deployed into containers that provide supporting services such as transaction management, security, and so on to the application.
- Application servers are entities of applications on which enterprise applications are deployed and run.
- EJBLite is a Java EE profile, which is a compact runtime environment including all the essential application entities.