CS-ELECTIVE -1

RCS-E12: WEB TECHNOLOGIES

Unit -4

Enterprise Java Bean: Preparing a Class to be a JavaBeans, Creating a JavaBeans, JavaBeans Properties, Types of beans, Stateful Session bean, Stateless Session bean, Entity bean

Java Database Connectivity (JDBC): Merging Data from Multiple Tables: Joining, Manipulating, Databases with JDBC, Prepared Statements, Transaction Processing, Stored Procedures.

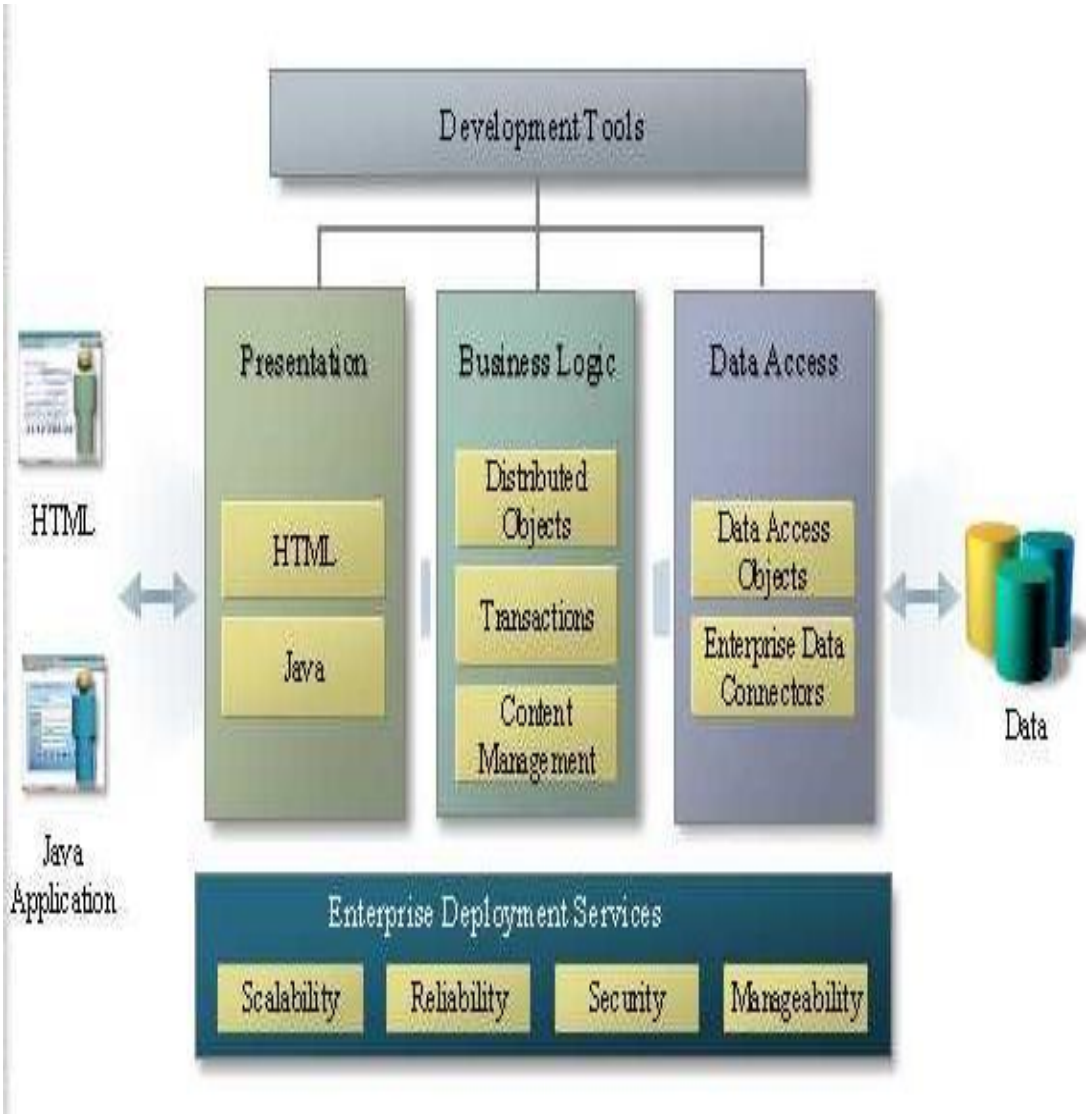
.

Introduction

- Enterprise Java Beans (EJB) is
  - a *middleware component model* for Java and CORBA
  - a **specification** for creating server-side, scalable, transactional, multi-user and secure enterprise-level applications
- Presented by Sun in the 1999, they are easier than other technologies as RMI or Corba

Introduction

- This is the three level structure for Application Server



Applicaton Server

- Presentation
  - HTML Application
  - Java Application
- Business Logic
- Data Access

Presentation

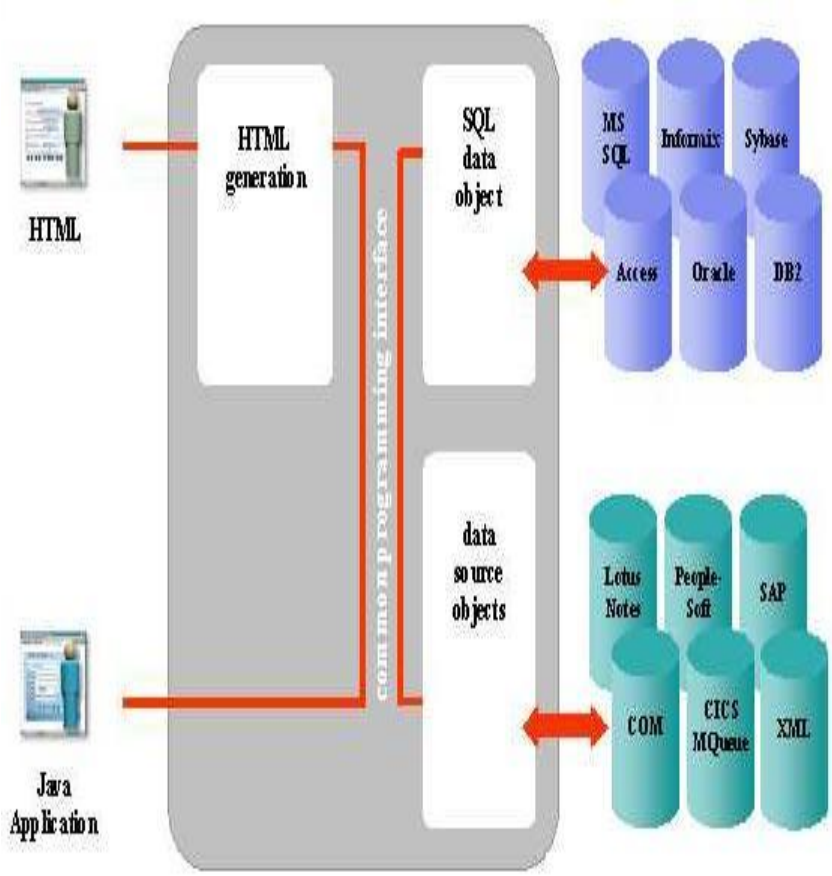
- HTML
  - Generated server-side HTML
  - Runs on any Web browser
  - Less client-side power
- Java
  - Required Java virtual Machine
  - More client side power
  - Runned on a page
  - Launched from a browser or a standalone application

Business Logic

- Implements the logic of the application defining all the function that may be used from a client
  - Change Business Rules Easily
  - Re-use components
  - Make complex applications manageable

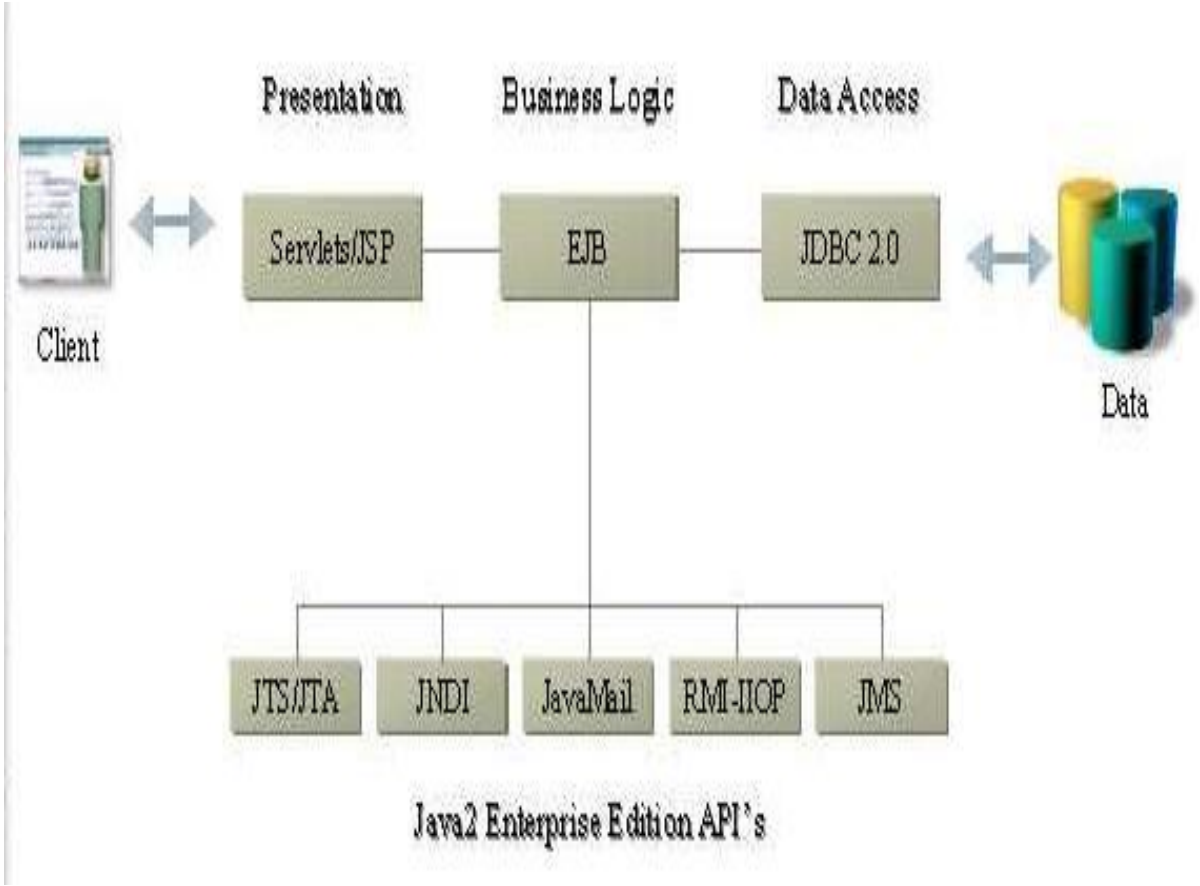
Data Access

- Utility to access external datas such as Database or other Web component



J2EE Application Server

- Java 2 Enterprise Edition standardizes interfaces for Application Server components



What is an Enterprise Bean ?

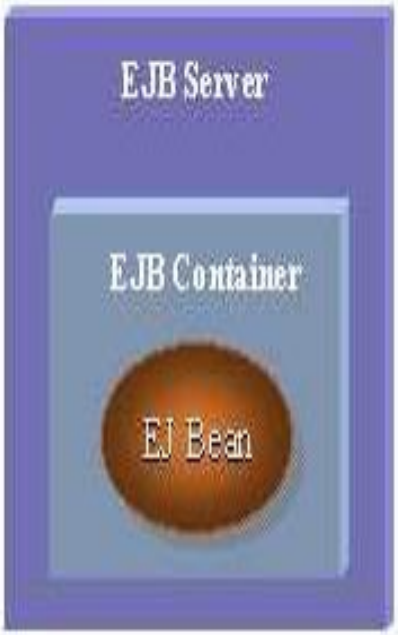
- Is a server side component written in Java Language
- Industry standard distribuited component model
- Incorporates the business logic of an application (the code that implements the purpose of the application)

EJB Properties

- Bean writers need not write
  - Remote access Protocols
  - Transactional Behaviour
  - Threads
  - Security
  - State Management
  - Object life cycle
  - Resource pooling
  - Persistence

EJB Overview

- Simple Code



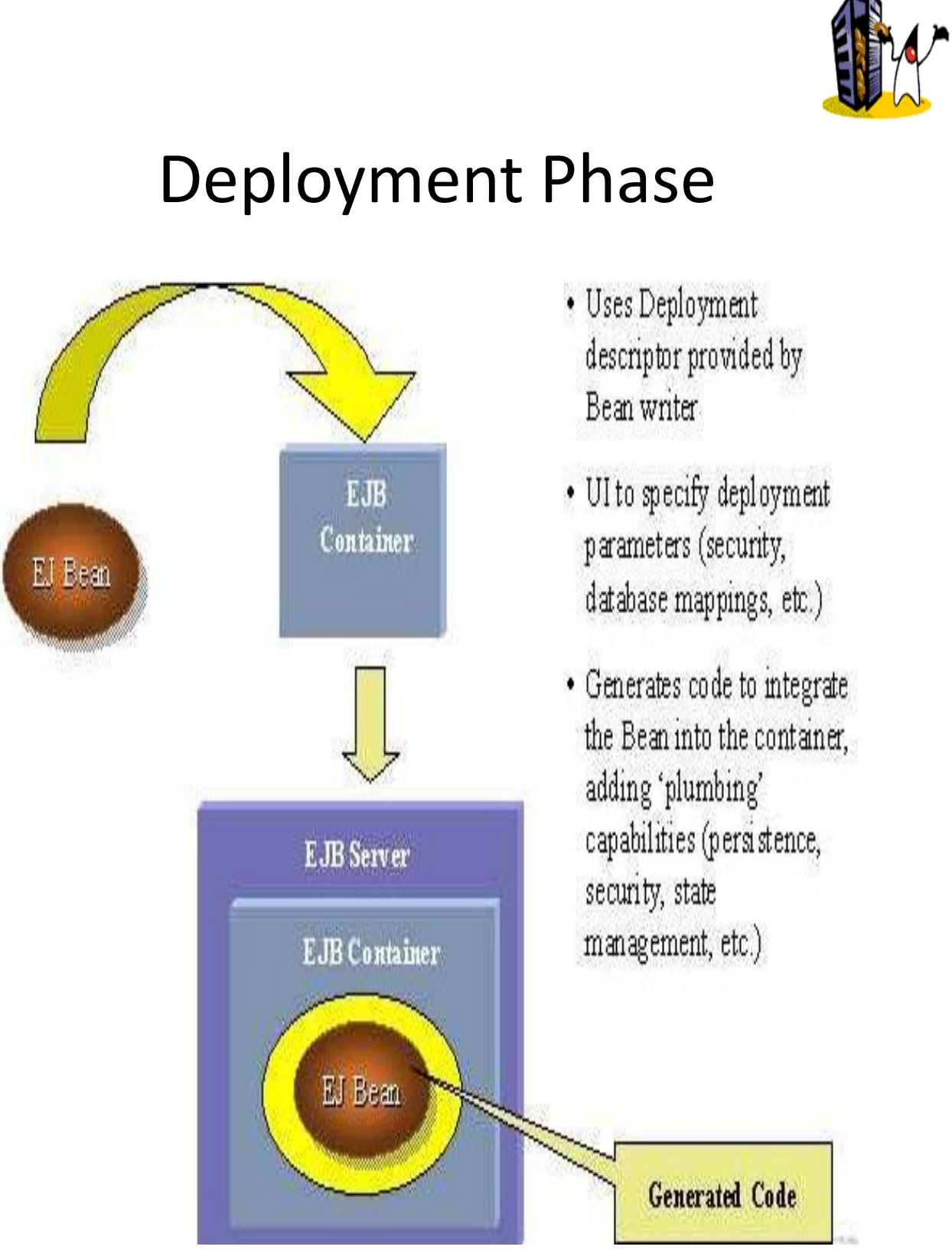
- Generates code to wrap the Bean

EJB Container

- Generated code provides Transactions, Security, Persistence, Remote protocols, resource pooling, etc.

EJB Server

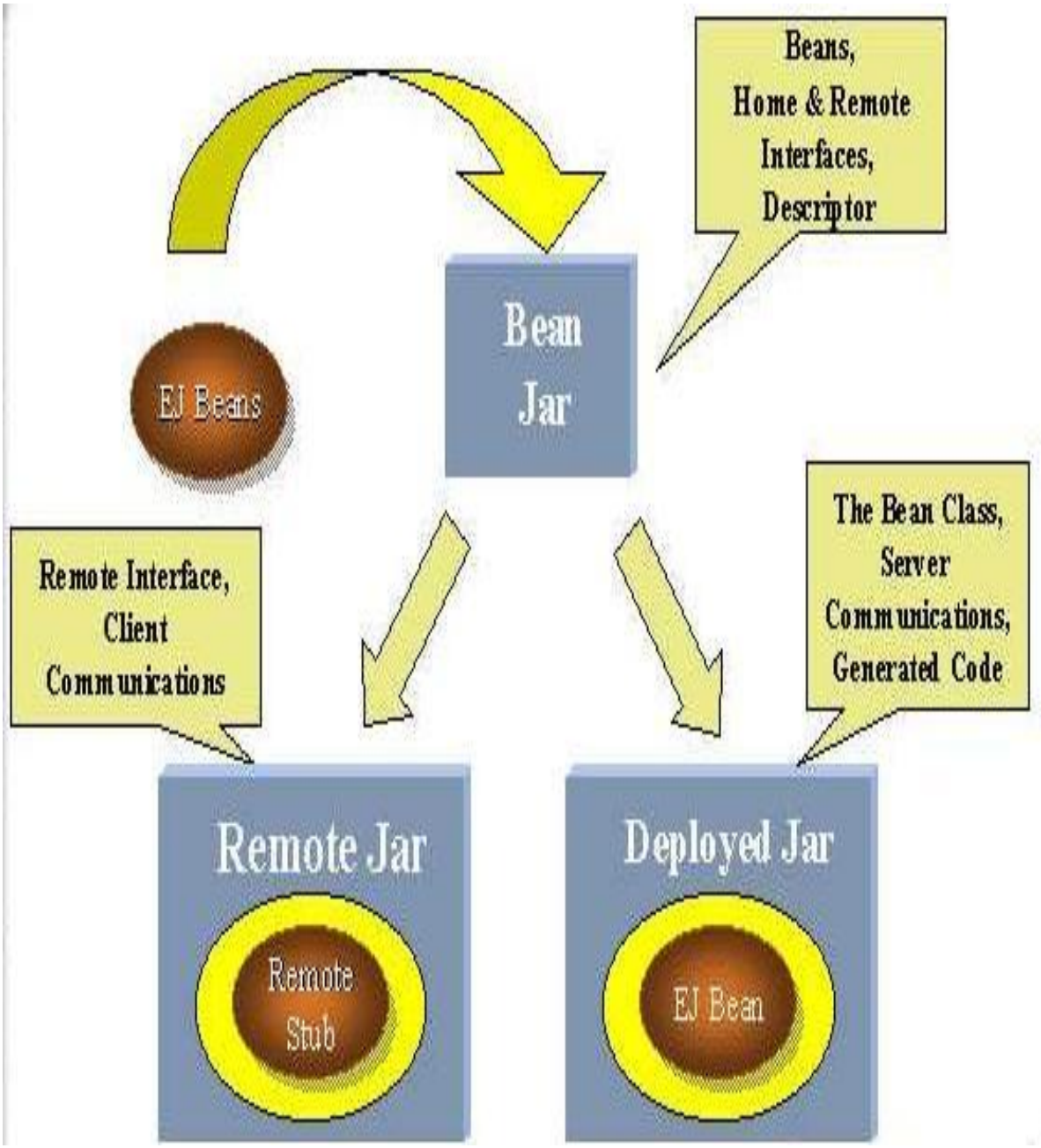
- Provides Application Services used by the container



Deployment Phase

- Uses Deployment descriptor provided by Bean writer
- UI to specify deployment parameters (security, database mappings, etc.)
- Generates code to integrate the Bean into the container, adding 'plumbing' capabilities (persistence, security, state management, etc.)

Deployment Phase



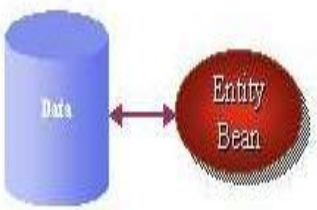
When use Enterprise Java Bean?

- **Application needs Remote Access**. In other words, it is distributed.
- **Application needs to be scalable**. EJB applications supports load balancing, clustering and fail-over.
- **Application needs encapsulated business logic**. EJB application is separated from presentation and persistent layer.

Types of Enterprise Java Bean

- There are 3 types of enterprise bean in java.
- **Session Bean**
- Session bean contains business logic that can be invoked by local, remote or webservice client.
- **Message Driven Bean**
- Like Session Bean, it contains the business logic but it is invoked by passing message.
- **Entity Bean**
- It encapsulates the state that can be persisted in the database. It is deprecated. Now, it is replaced with JPA (Java Persistent API).

- Business logic
- Transactions



- Represent a row in the database
- Properties are "mapped" to fields in the database

Session Bean

- Session bean encapsulates business logic only, it can be invoked by local, remote and webservice client.
- It can be used for calculations, database access etc.
- The life cycle of session bean is maintained by the application server (EJB Container).

Types of Session Bean

- There are 3 types of session bean.
- **1) Stateless Session Bean**: It doesn't maintain state of a client between multiple method calls.
- **2) Stateful Session Bean**: It maintains state of a client across multiple requests.
- **3) Singleton Session Bean**: One instance per application, it is shared between clients and supports concurrent access.

Stateless Session Bean

- **Stateless Session bean** *is a business object that represents business logic* only. It doesn't have state (data).
- In other words, *conversational state* between multiple method calls is not maintained by the container in case of stateless session bean.
- The stateless bean objects are pooled by the EJB container to service the request on demand.
- It can be accessed by one client at a time. In case of concurrent access, EJB container routes each request to different instance.

Annotations used in Stateless Session Bean

- There are 3 important annotations used in stateless session bean:

- @Stateless
- @PostConstruct
- @PreDestroy

Life cycle of Stateless Session Bean

- There is only two states of stateless session bean: does not exist and ready. It is explained by the figure given below.

Life cycle of Stateless Session Bean

EJB Container creates and maintains a pool of session bean first. It injects the dependency if then calls the @PostConstruct method if any. Now actual business logic method is invoked by the client. Then, container calls @PreDestory method if any. Now bean is ready for garbage collection.

Stateful Session Bean

- **Stateful Session bean** *is a business object that represents business logic* like stateless session bean. But, it maintains state (data).
- In other words, *conversational state* between multiple method calls is maintained by the container in stateful session bean.

Annotations used in Stateful Session Bean

There are 5 important annotations used in stateful session bean:

- @Stateful
- @PostConstruct
- @PreDestroy
- @PrePassivate
- @PostActivate

Entity Bean in EJB 3.x

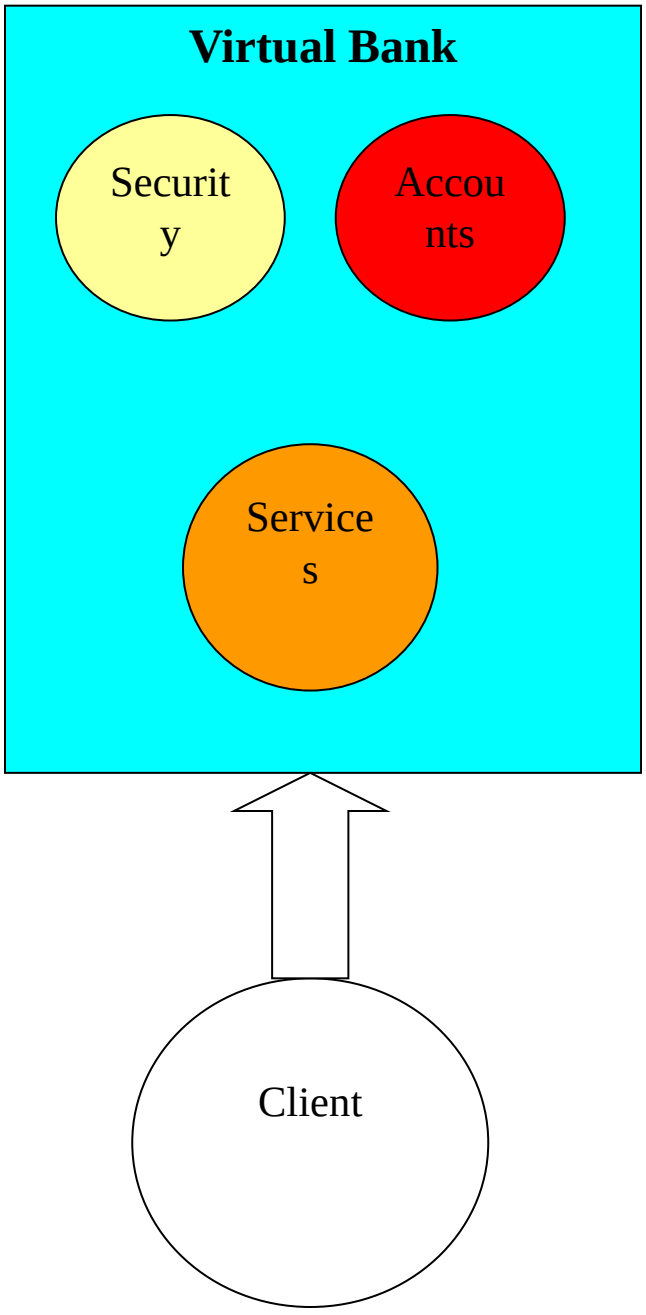
- Entity bean represents the persistent data stored in the database. It is a server-side component.
- In EJB 2.x, there was two types of entity beans: **bean managed persistence** (BMP) and container managed persistence (CMP).
- Since EJB 3.x, it is deprecated and replaced by JPA (Java Persistence API) that is covered in the hibernate tutorial.
- In hibernate tutorial, there are given hibernate with annotation examples where we are using JPA annotations. The JPA with Hibernate is widely used today.

EJB Example

- **The OnLine Bank**

  We will take a not completed system to give an idea to how choose if a component is an entity, session or message driven bean.

EJB Example



EJB Example

The example has three component:

- Services: what the client can do in the system such as see the foreign currency , listed shares or make operations on his hown account.
- Accounts: a database containing the accounts of all the clients of the bank with information about credit,debit,access etc..
- Security: is a subsystem that receives all the alarm caused from wrong access and performs action about the situation

  (calls police and stops operation of that client keeping information about him)

EJB Example

- In this example is easy to create an EJB structure.
  - Client will have a web page at client side to insert values and connect the system.This will be done using JSP (Java Servlet Pages)
  - Services will be a Statefull Session Bean and it will be different for each client connecting the system mantaining data about the client connected.
  - Accounts will be formed by an Entity Bean for each account in the system with a code-account as primary key.
  - Security will be a Message driven bean and will be called only from container if some operation are abnormal for result or the autentification for the same client fails too much times.