Enterprise Java Bean

What is EJB

- EJB is an acronym for *enterprise java bean*. It is a specification provided by Sun Microsystems to develop secured, robust and scalable distributed applications.
- EJB application is deployed on the server, so it is called server side component also.
- EJB is like COM (*Component Object Model*) provided by Microsoft. But, it is different from Java Bean, RMI and Web Services.

It performs following task

- life cycle management,
- security,
- transaction management, and
- object pooling.

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.

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

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.

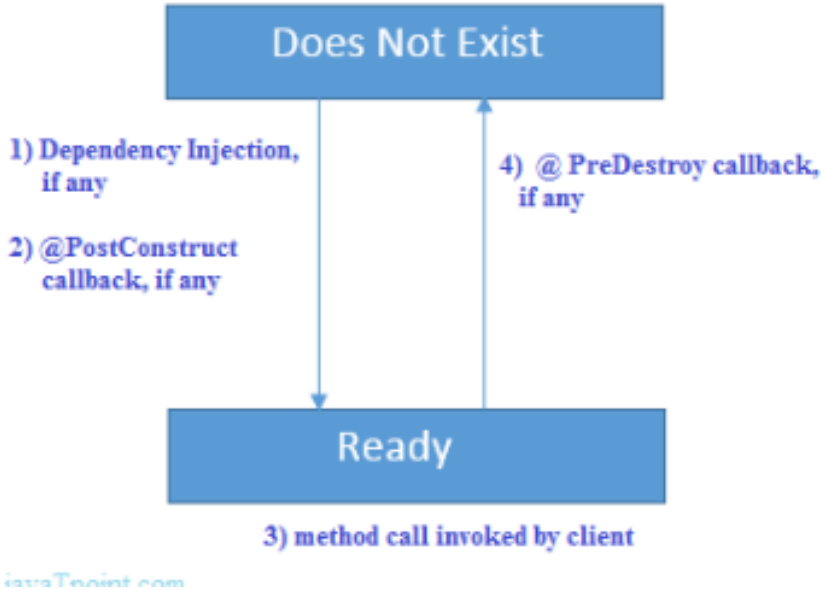
Stateless Session Bean

- **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

Annotations used in Stateless Session Bean

- @Stateless
- @PostConstruct
- @PreDestroy

Life cycle of Stateless Session Bean (does not exist and ready.)



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

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

JMS (Java Message Service)

- JMS (Java Message Service) is an API that provides the facility to create, send and read messages. It provides loosely coupled, reliable and asynchronous communication.
- JMS is also known as a messaging service.

Messaging

- Messaging is a technique to communicate applications or software components.
- JMS is mainly used to send and receive message from one application to another.

- Advantage of JMS

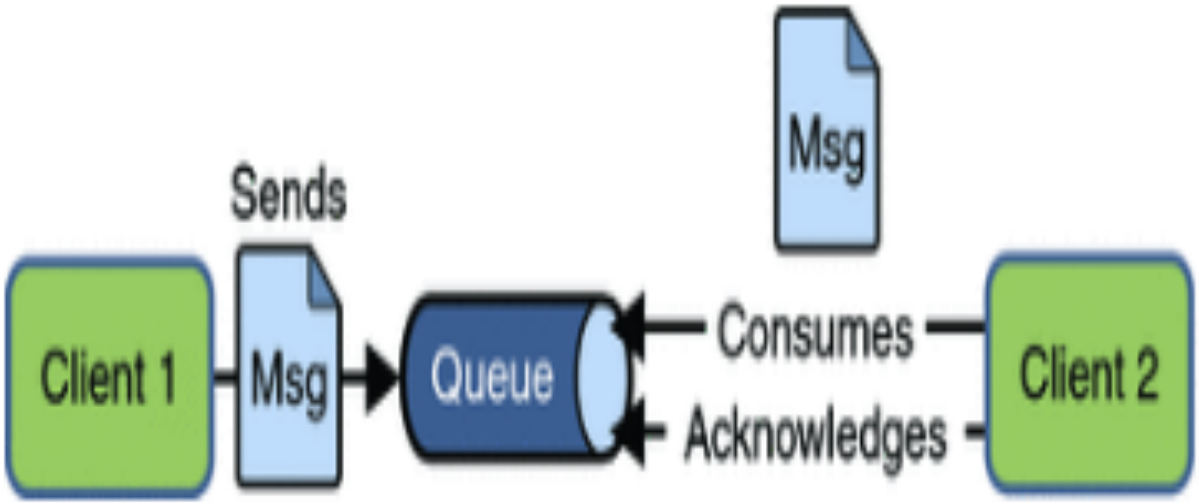
1) **Asynchronous:** To receive the message, client is not required to send request. Message will arrive automatically to the client.

2) **Reliable:** It provides assurance that message

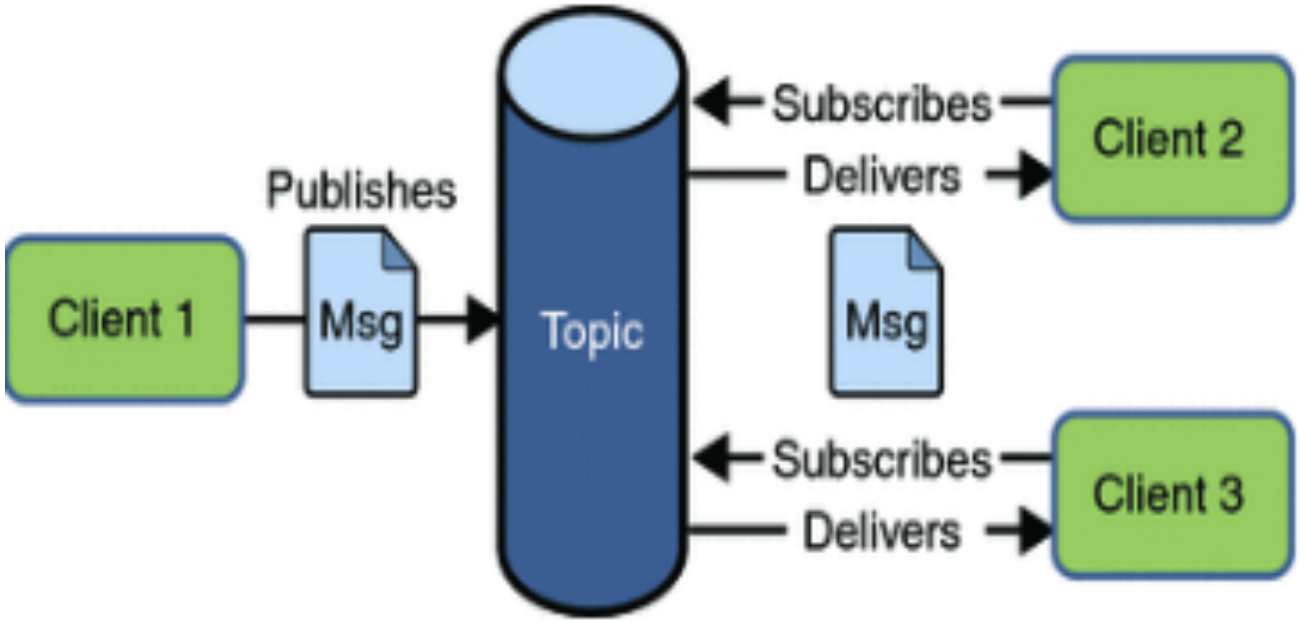
Messaging Domains

- Point-to-Point Messaging Domain
- Publisher/Subscriber Messaging Domain

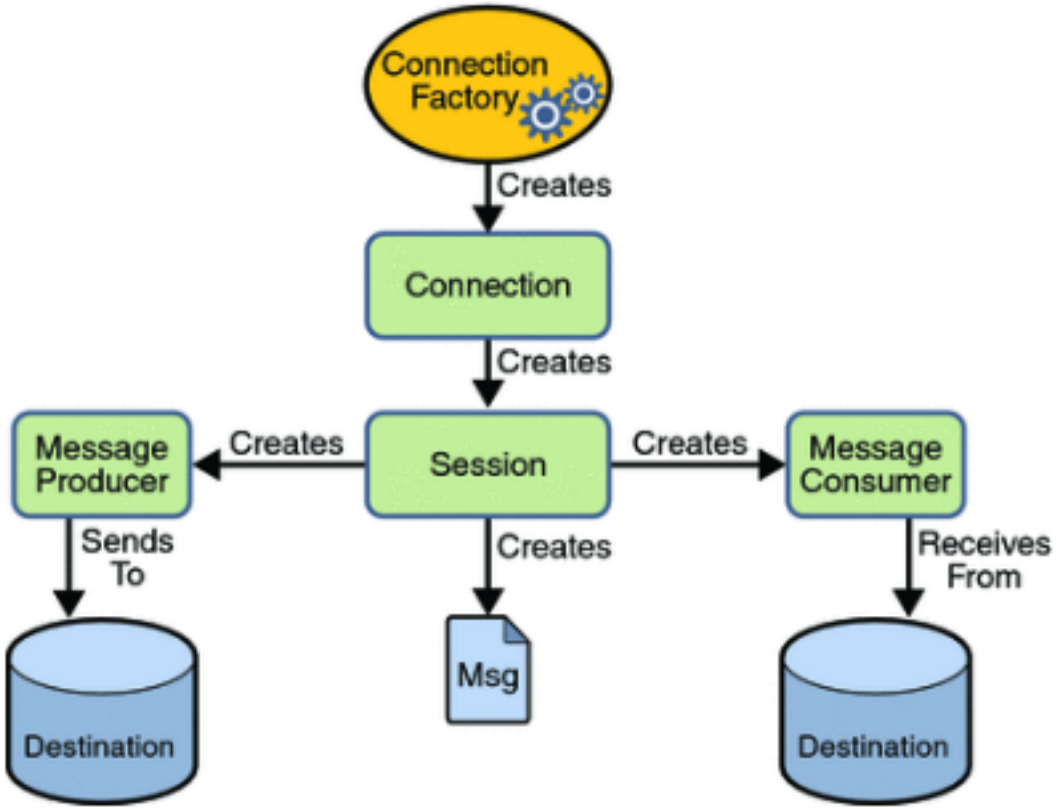
1) Point-to-Point (PTP) Messaging Domain



2) Publisher/Subscriber (Pub/Sub) Messaging Domain



JMS Programming Model



Message Driven Bean

- A message driven bean (MDB) is a bean that contains business logic. But, it is invoked by passing the message. So, it is like JMS Receiver.
- MDB asynchronously receives the message and processes it.
- A message driven bean receives message from queue or topic, so you must have the knowledge of JMS API.


Java EE Server
EJB Container
MDB
Instances
Msg
Queue
Msg
Application
Client
Sends
Delivers

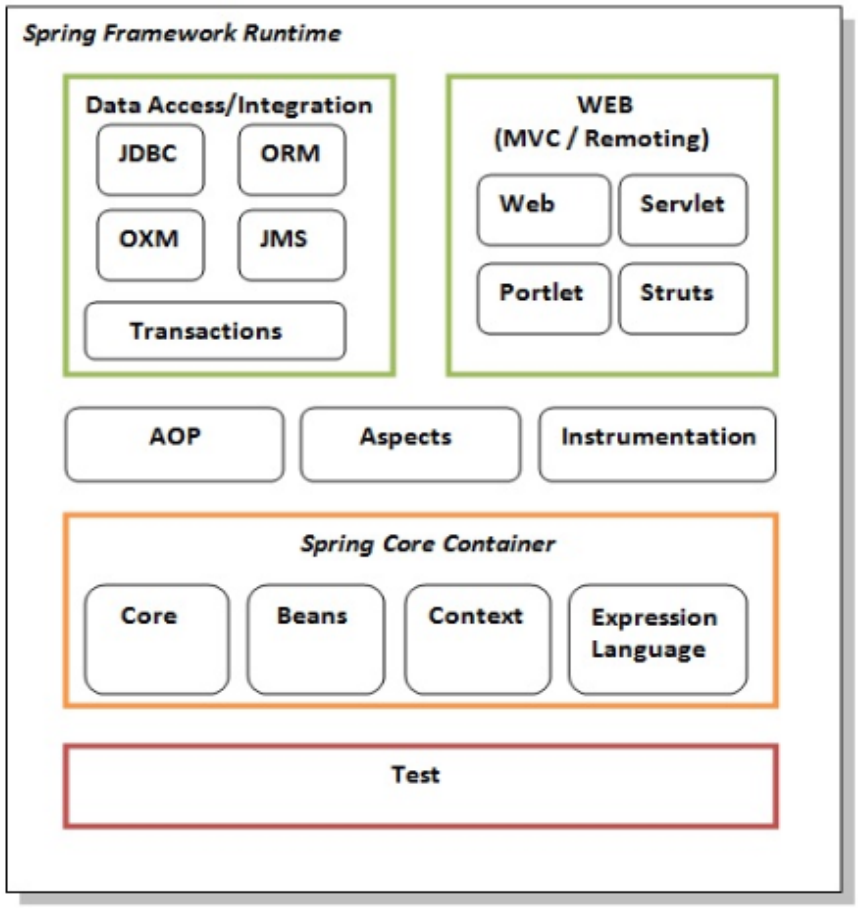
Entity Bean

- 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.

Spring Framework

- Spring is a *lightweight* framework. It can be thought of as a *framework of frameworks* because it provides support to various frameworks such as Struts, Hibernate, Tapestry, EJB, JSF etc. The framework, in broader sense, can be defined as a structure where we find solution of the various technical problems.

Spring Modules



STRUT

- The Struts 2 framework is used to develop MVC (Model View Controller) based web applications.
- struts2 = webwork + struts1

- Configurable MVC components
- POJO based actions
- AJAX support
- Integration support
- Various Result Types
- Various Tag support
- Theme and Template support