JMS
(Java Message Service)

By Kasun Dinesh Madusanke

Topics

- Message and Messaging
- What is JMS?
- Goals of JMS
- Benefits of JMS
- When to use JMS?
- JMS Message

Topics Cont.

- Java Mail vs JMS
- JMS Components
- JMS Application
- MOM Service Providers
- Messaging Models
- JMS API Programming Model

Topics Cont.

- Demo-JMS Pure Java
- Spring JMS
- JMSTemplate
- Demo-Spring JMS
- Summary

Message & Messaging

Message

- Bytes of data that is meaningful between the applications which use it.
- Used to transfer information form one application to others.

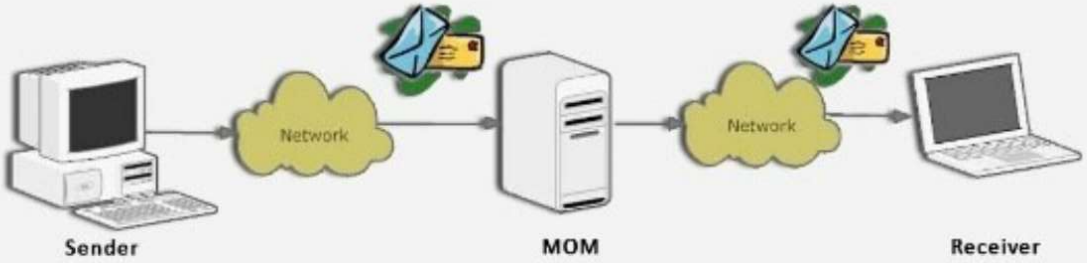
Messaging

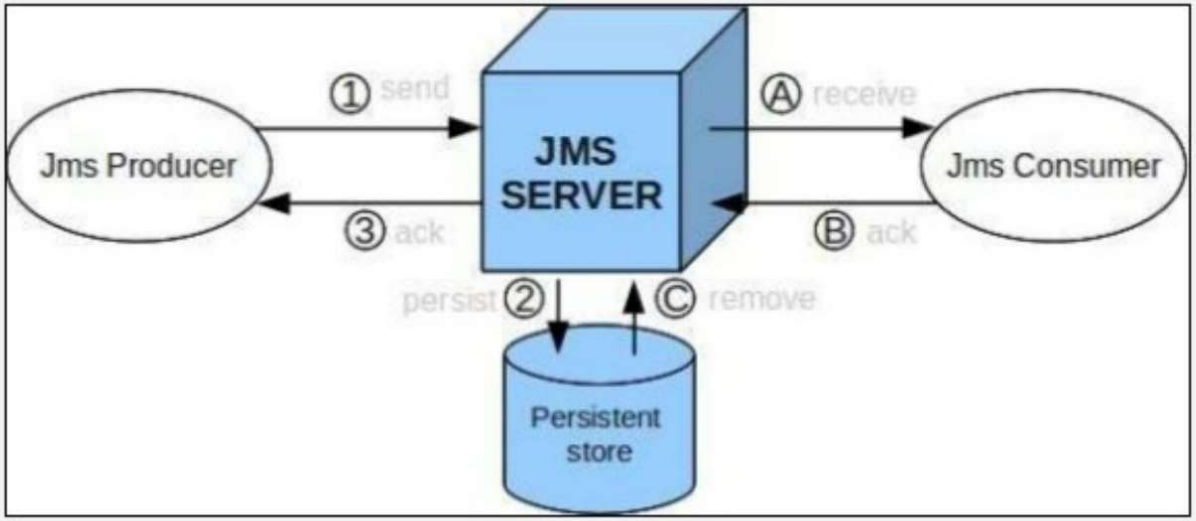
- Communication between system components or different applications.

What is JMS?

- JMS is a Message Oriented Middleware (MOM)
- Published and maintained by Sun Microsystems
- First published in August 1998
- Loosely coupled communication
- Asynchronous messaging
- Reliable delivery

What is JMS? Cont.





hSenid Lanka: JMS

7

Goals of JMS

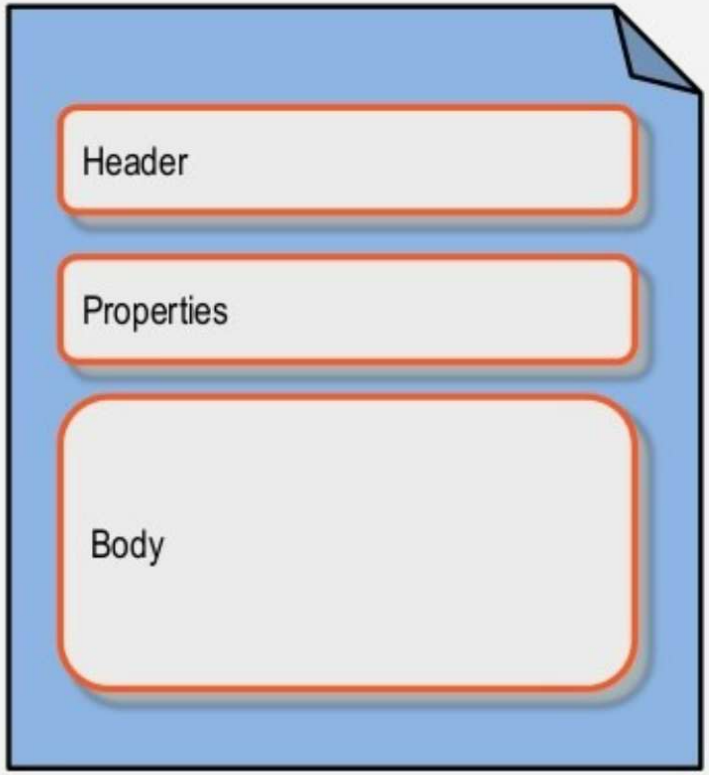
- Minimizes the set of concepts a programmer must learn to use messaging products (programmer friendly).
- Provides enough features to support sophisticated messaging applications.
- Maximize the portability of JMS applications across JMS providers in the same messaging domain.

When to use JMS?

- The provider wants the components not to depend on information about other components' interfaces, so that components can be easily replaced.
- The provider wants the application to run whether or not all components are up and running simultaneously.
- The application business model allows a component to send information to another and to continue to operate without receiving an immediate response.

JMS Message

This image shows the Structure of a JMS Message



Header, Properties and Body

Header	Properties	Body
Identify message	Added by the application developer	Message body
Destination	Application specific properties	Can contain arbitrary data types Eg: Text messages Map (key-value pairs) XML Serialized objects (Java) Binary data Empty
Routing Information	Key-value pairs	
Priority	Extensions for messaging systems	
Timestamp		
Message type		

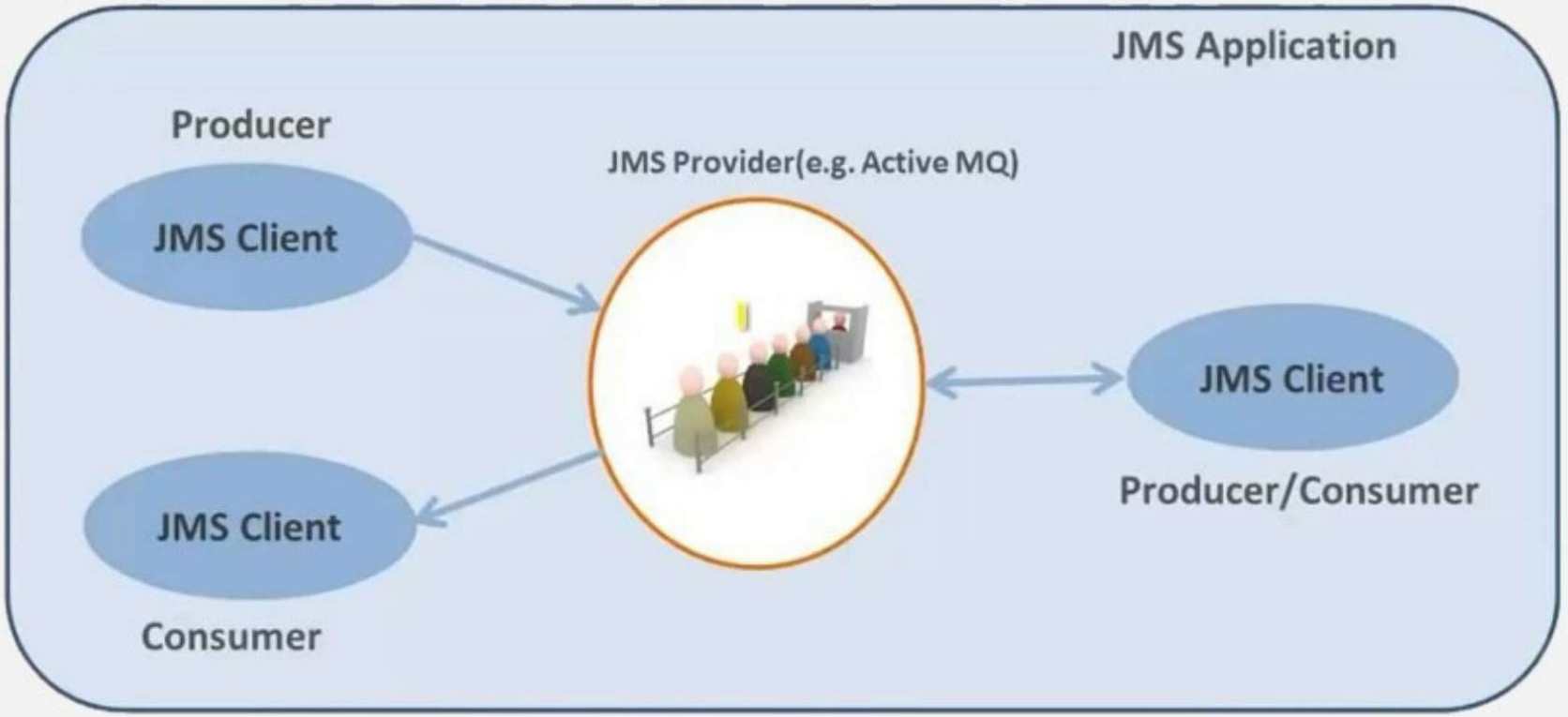
Java Mail vs JMS

Java Mail	JMS
API for sending emails (eg: with attachments).	Capable of exchanging messages between applications.
Mainly for human information exchange	Application/Human to Application/Human Messaging.
Delivery can be duplicated.	More like a database. Messages removed once it reads by the recipient system.
Delivery not guaranteed.	Delivery is guaranteed (Above reason).
Designed for connectivity on the web.	Bunch of different qualities of service (durable vs non-durable, queue vs topic).

JMS Components

- JMS Provider:
  - The messaging system (MOM) that implements JMS.
- JMS Clients
  - Java applications that produce/receive messages.
- JMS Producer/Publisher
  - A JMS client that creates and sends messages.
- JMS Consumer/Subscriber
  - A JMS client that receives messages.
- JMS Application
  - The system composed of JMS clients and a JMS provider.

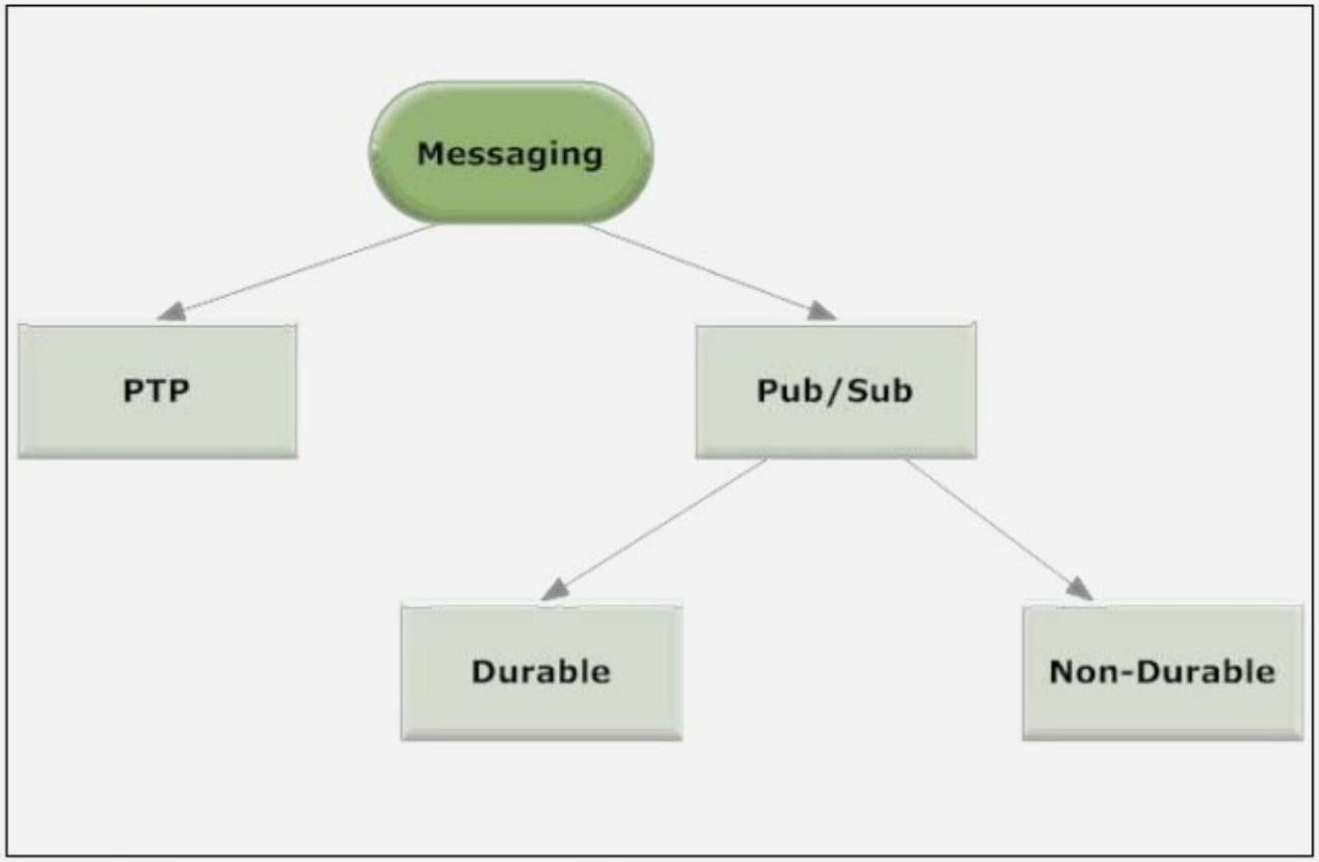
JMS Application



MOM Service Providers

Service provider Product	Company
Weblogic	Oracle
MQSeries	IBM
JBOSSMQ	JBOSS
SoniqMQ	Progress
TIBCO EMS	TIBCO
ActiveMQ	Apache

Messaging Models

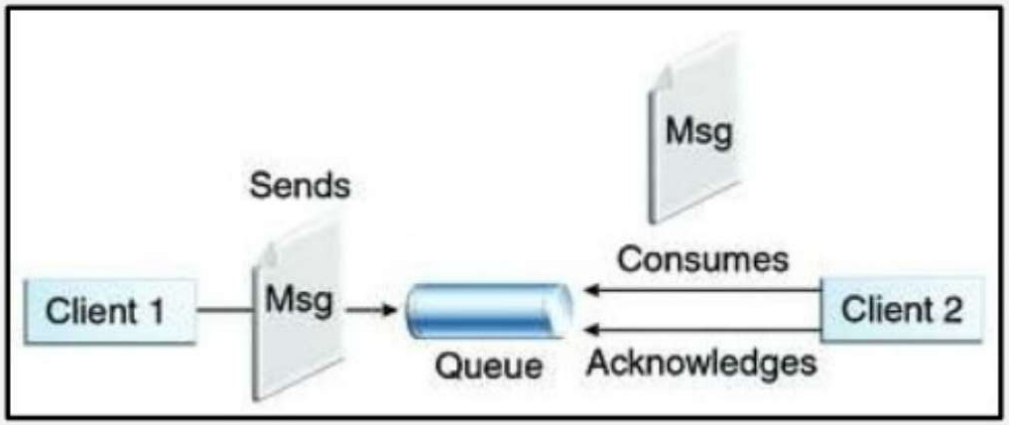


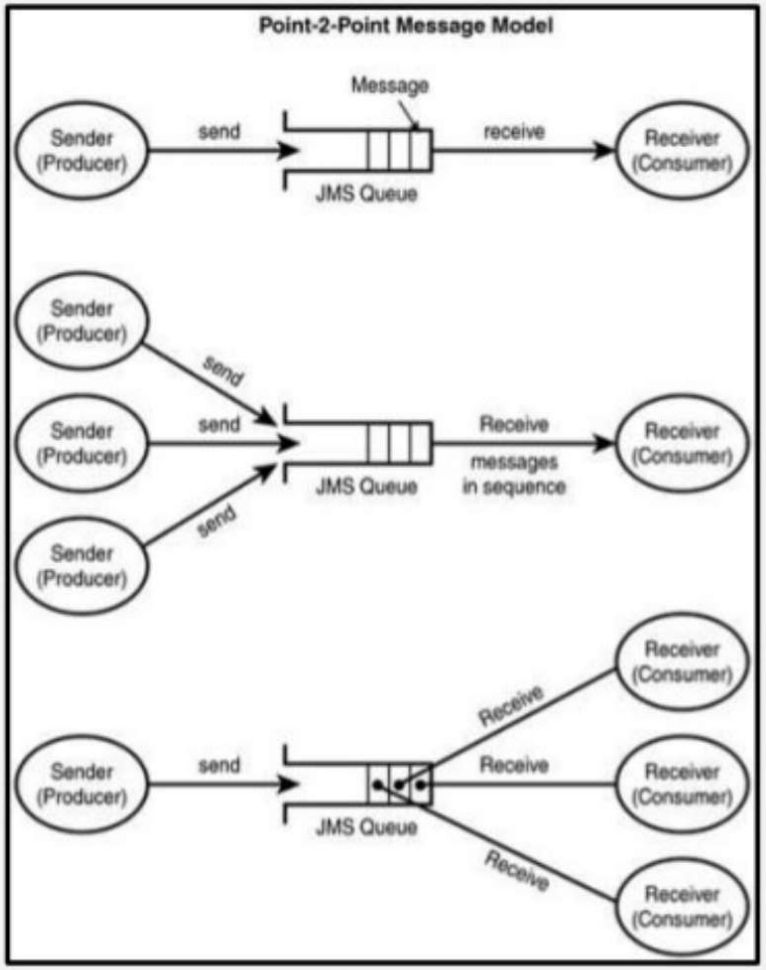
Messaging Models Cont.

- Point to Point
  - Message queues, senders and receivers.
  - Message is sent to a queue.
  - Each message has only one consumer.
  - Queue may be configured to persist messages.
- Publish-Subscribe
  - Publishers, subscribers, topics.
  - Message may have multiple consumers, or no consumer at all.
  - Each message is delivered to every client subscribed to a topic.

Messaging Models Cont.

- **Point to Point**



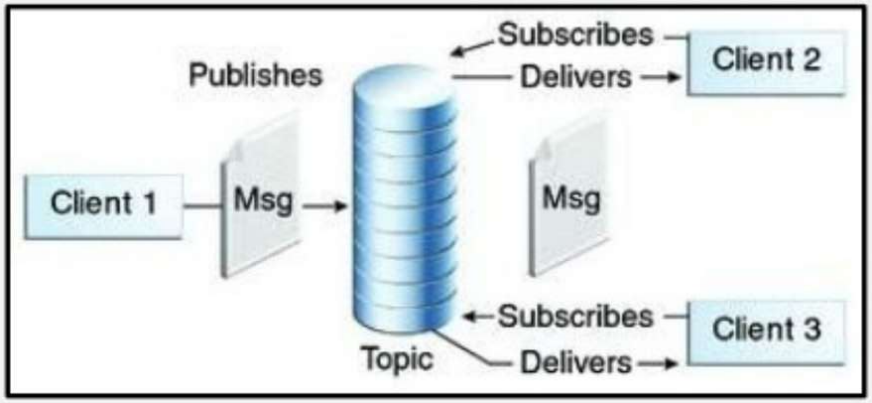


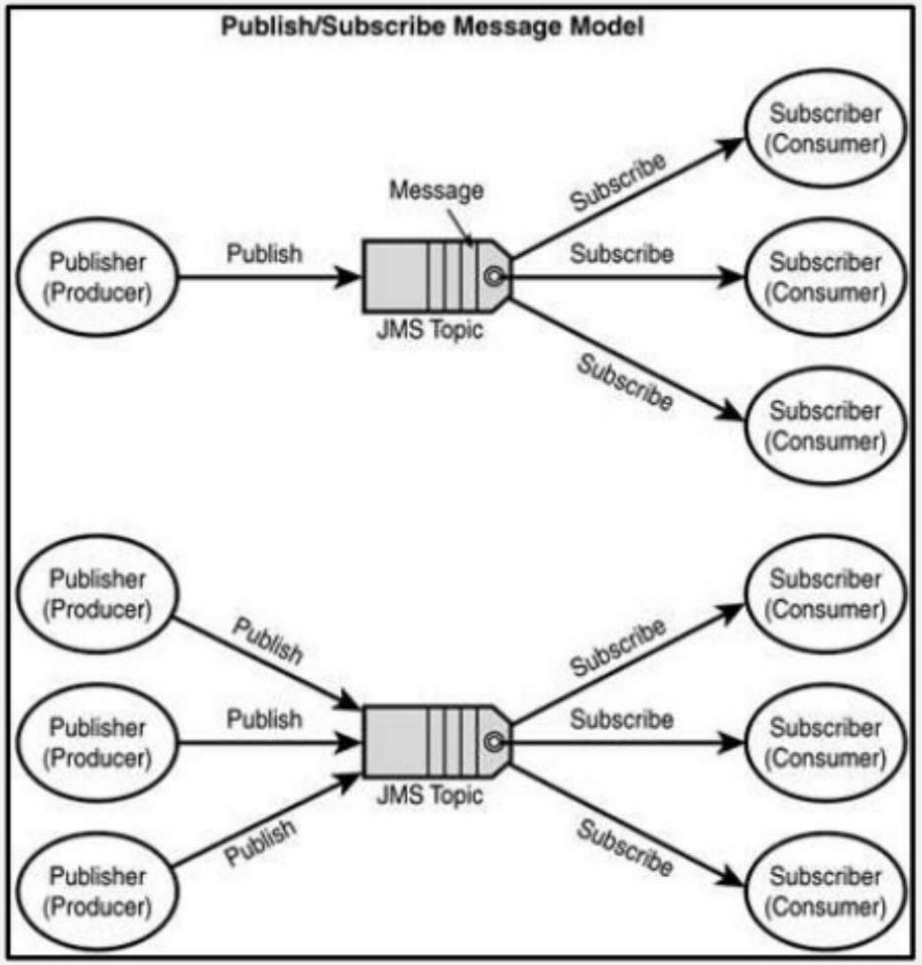
hSenid Lanka: JMS

18

Messaging Models Cont.

- **Publish-Subscribe**



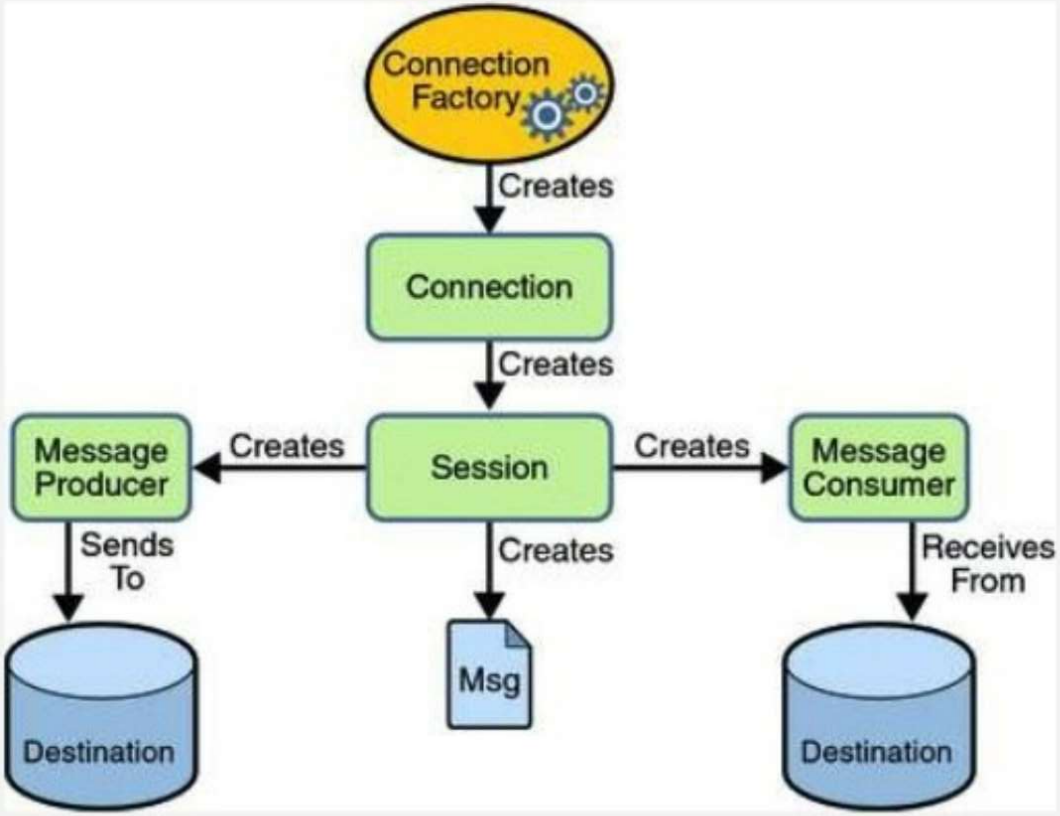


Messaging Models Cont.

PTP vs PUB/SUB

Point to Point	Publish-Subscribe
Each message has only one consumer.	Each message can have multiple consumers.
Messages first sent to a destination called Queue.	Messages first sent to a destination called Topic.
A sender and receiver of a message have no timing dependencies (receiver can fetch the message later).	Publishers and subscribers have a timing dependency (client need to be subscribed and active).
The receiver acknowledges the successful processing of a message.	Does not provide acknowledgement.

JMS API Programming Model



JMS API Programming Model Cont.

- **Connection Factory**
  - The client uses an object which is a connection factory used to **create a connection to a provider**.
  - It creates connection between JMS Provider and JMS Client.
- **Connection**
  - A JMS connection **encapsulates a virtual connection with a JMS provider**.
  - it is an open TCP/IP socket between a client and a provider service daemon.

JMS API Programming Model Cont.

- Session
  - The session is a **single threaded context** which is used for **producing and consuming messages**.
  - The sessions are used to create the following:
    - Message Producers
    - Message Consumers

JMS API Programming Model Cont.

- **Message Producer**
  - A JMS message producer object is created by a session.
  - It is used for **sending messages to a destination.**
- **Message Consumer**
  - A JMS message consumer object is created by session.
  - It helps for **receiving messages sent to a destination.**


DEMO
(JMS Pure Java)
Download project files

Spring JMS

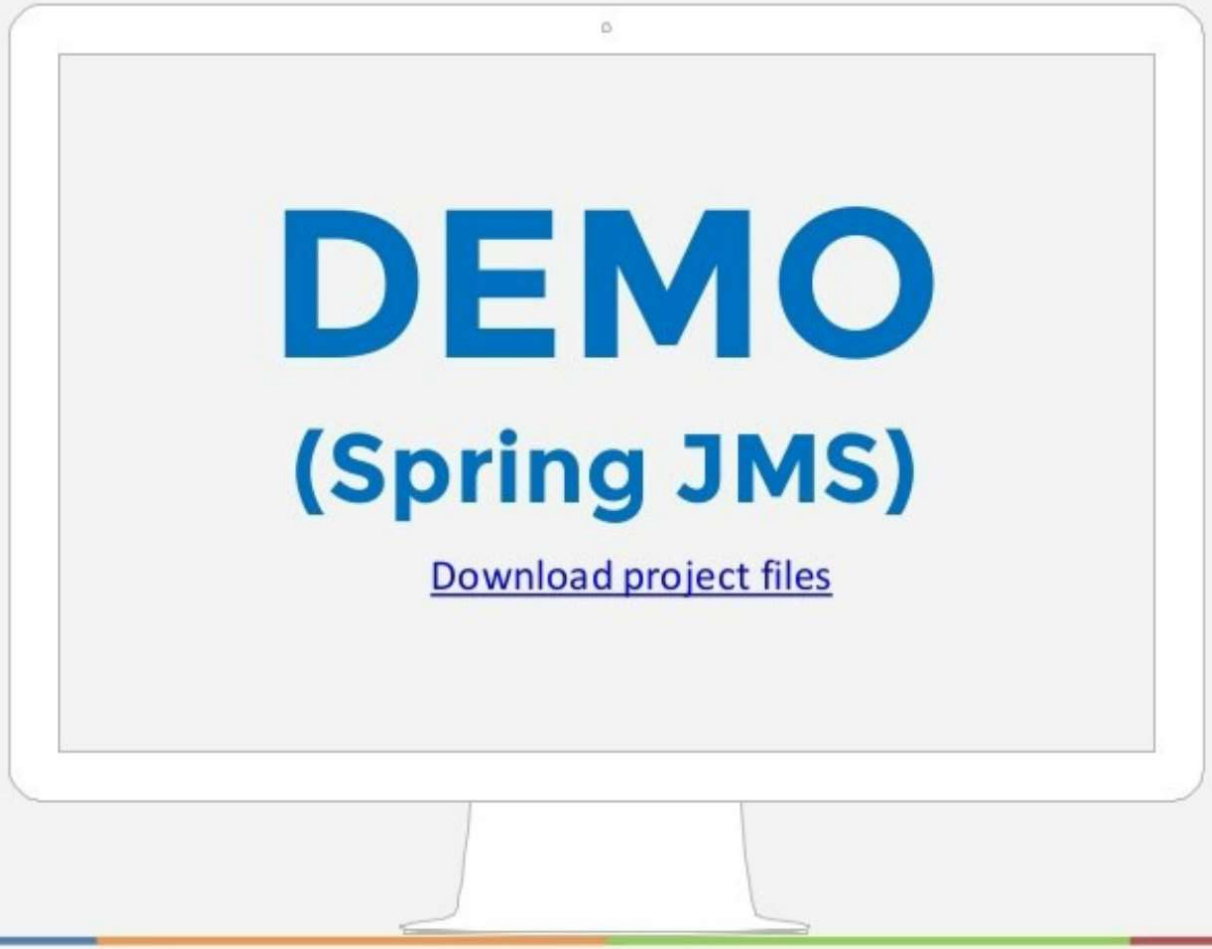
- Advantages
  - No EJB container required (no JEE container)
  - Simplified resource management
  - Connection Factory, Connections, Destinations
  - Simplified concurrency management
  - Simplified transaction management

Spring JMS Cont.

- JMS Template
  - Send and receive messages synchronously
- Message Listener Container
  - Receive messages asynchronously
  - Message-Driven POJOs (MDPs)

JMSTemplate

Method	Description
browse()	Browse messages in a queue
convertAndSend()	Send messages synchronously Convert a Java object to a JMS message
execute()	Provides access to callbacks for more complex scenarios
receive() receiveAndConvert()	Receive messages synchronously
receiveSelected() receiveSelectedAndConvert()	Receive filtered messages synchronously
send()	Send a message synchronously using a MessageCreator


DEMO
(Spring JMS)
Download project files

THANK YOU!

hSenid
MAKE IT HAPPEN

By Kasun Dinesh Madusanke