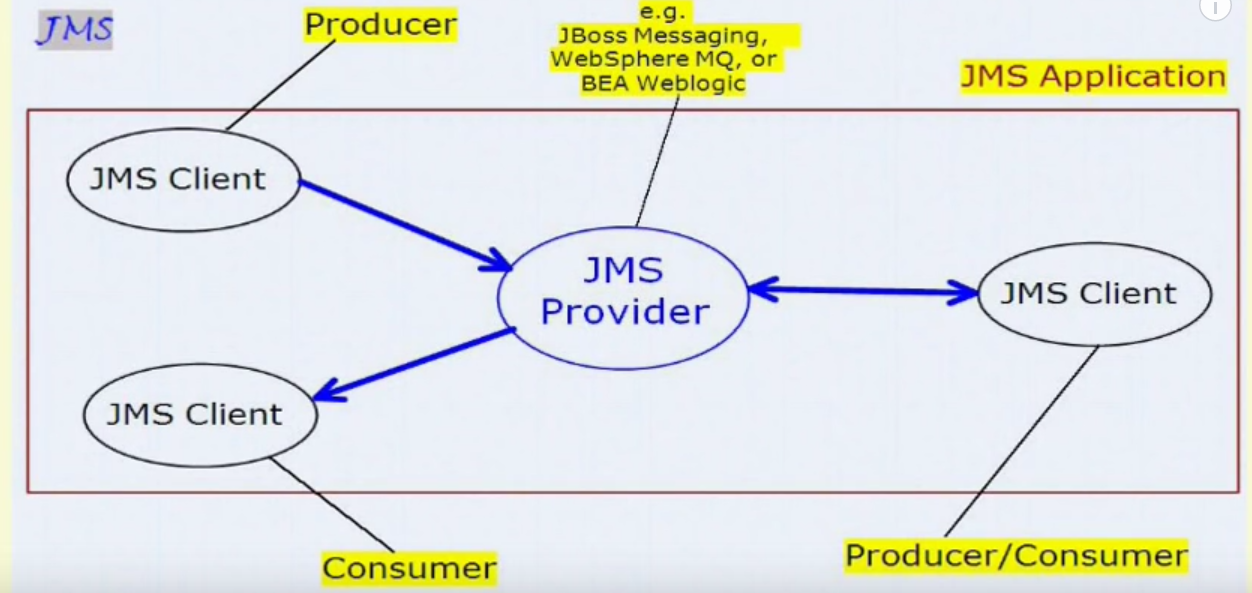
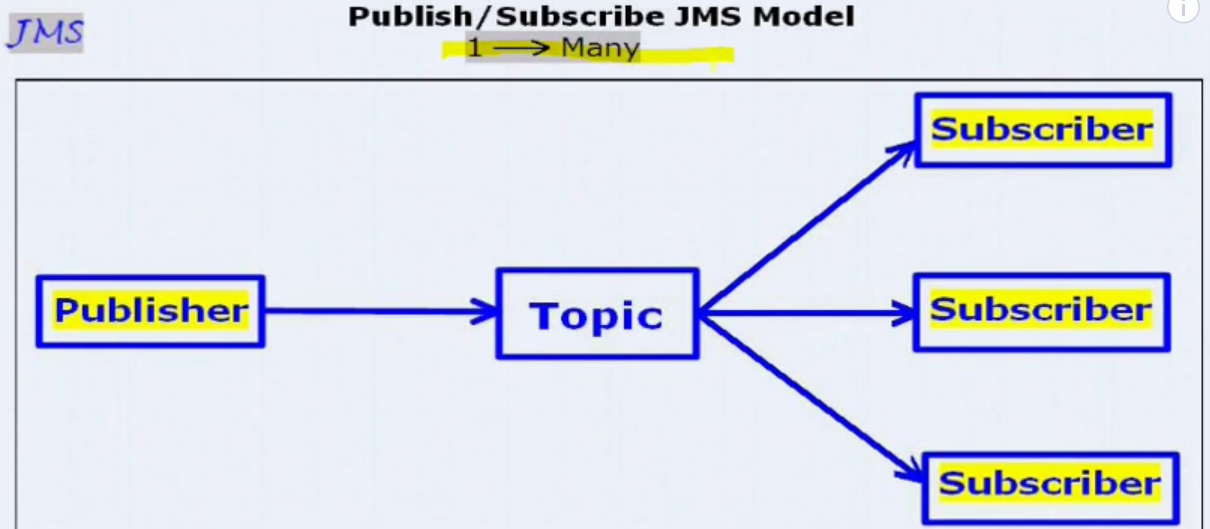
**JMS – Java messaging service:**

* **JMS** is like JDBC – Like the JDBC API that can be used to access different relational databases.
* JMS API provides vendor independent access to enterprise messaging systems
  + JBoss Messaging
  + Websphere MQ
  + BEA Weblogic
* **JMS Diagram:**



* **Producer** – JMS client that sends a message
* **Consumer** – JMS client that receive a message.
* **JMS Provider** – Messaging system that handles routing and delivery of messages.
* **JMS Application** – Business systems composed of many JMS clients and usually one JMS provider.
* **JMS Client** - Java application using JMS. It can be both a producer and a consumer of messages.
* **Advantage of JMS**
* **Asynchronous:** To receive the message, client is not required to send request. Message will arrive automatically to the client.
* **Reliable:** It provides assurance that message is delivered.
* **JMS has 2 models**
  + Point to point queues
  + Publish/subscribe topics
* **Publish/Subscribe topics:**
* One too many model - one message is **delivered to all the subscribers**. It is like broadcasting. Here, **Topic** is used as a message oriented middleware that is responsible to hold and deliver messages.



* **Point to Point queues:**
* One message is **delivered to one receiver** only. Here, **Queue** is used as a message oriented middleware (MOM).
* The Queue is responsible to hold the message until receiver is ready.

