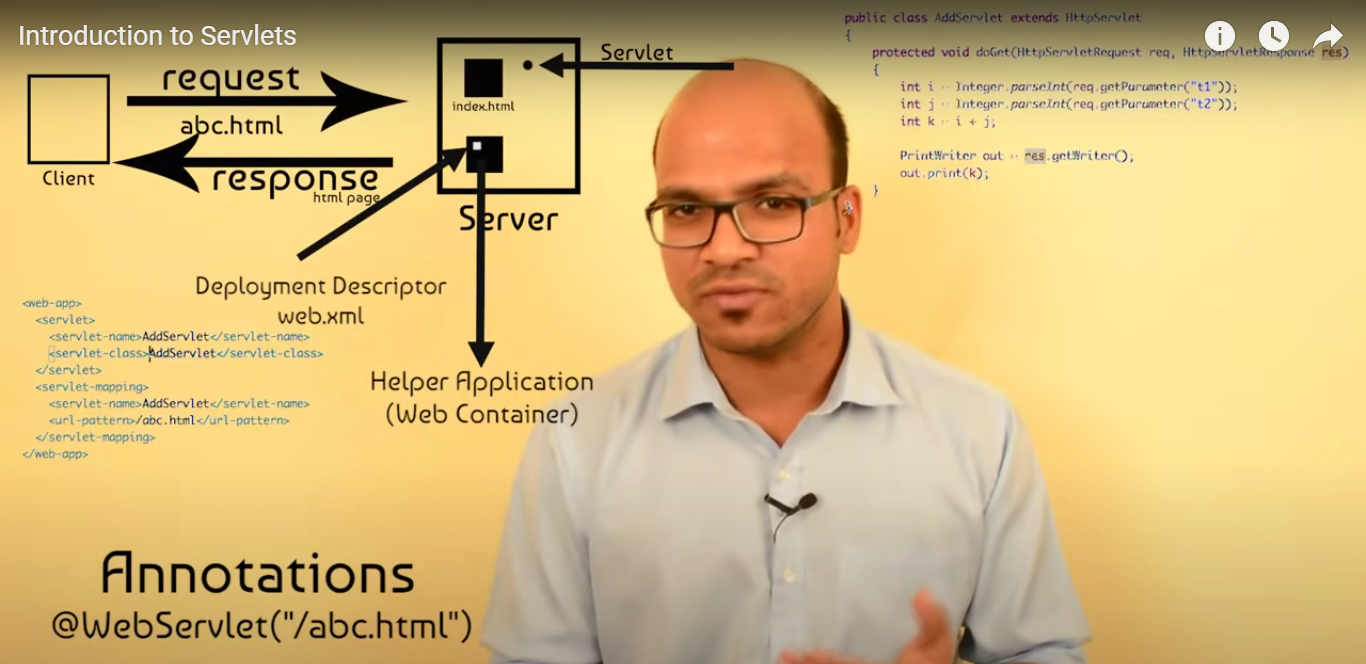
**Servlet:**

**1) Here Normal class extends HttpServlet.**

**2) Here a client sends request to servlet, servlet sends request to webContainer, webContainer uses a file called web.xml and detects which servlet to call as per servlet name, this servlet will process the response to the client machine. (Static and dynamic pages)**



**Java EE(Enterprise edition):**

facilitates development of large scale applications.

provides functionalities like web applications, and Servlets.

It provides user authentication.

JMS:

JMS (Java Message Service) is an API that provides the facility to create, send and read messages.

send message from one application to another.

Advantages :

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 is delivered.

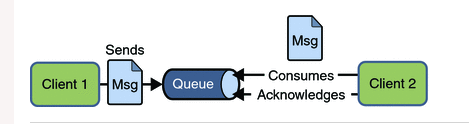
Messaging Domains :

## 1) Point-to-Point (PTP)

one message is **delivered to one receiver** only.

**Queue** is used as a message oriented middleware (MOM).

**no timing dependency** between sender and receiver.

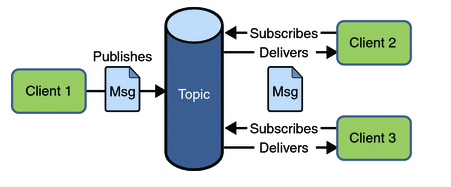


## 2) Publisher/Subscriber (Pub/Sub)

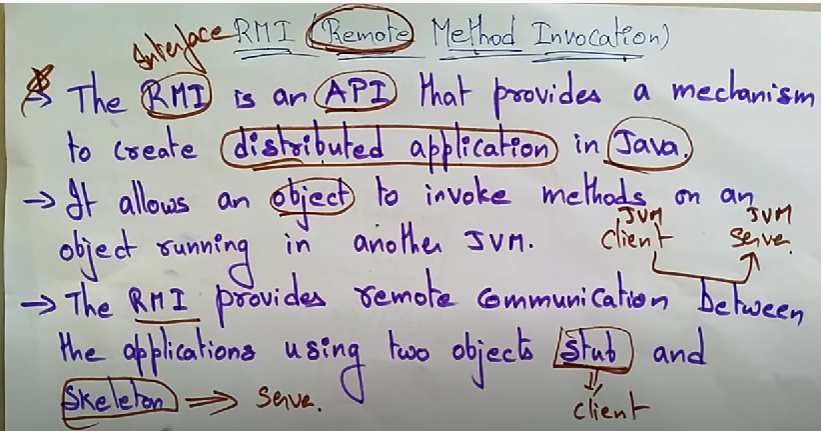
one message is **delivered to all the subscribers**.

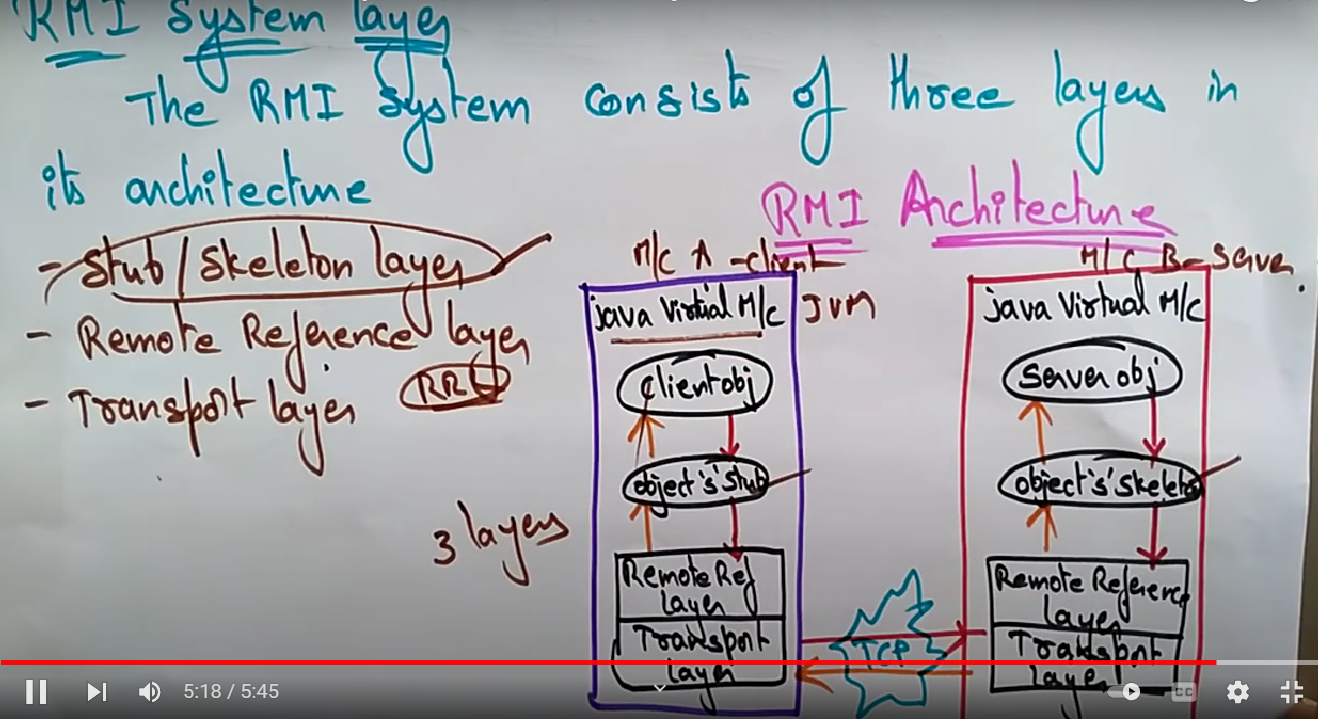
**Topic** is used as a message oriented middleware

**timing dependency** between publisher and subscriber.



**RMI:**





**Stub and skeleton are two objects that act as a gateway, request are routed through them.**

**JDBC : Java database connectivity.**

It basically acts as an interface or channel between your Java program and databases i.e it establishes a link between the two so that a programmer could send data from Java code and store it in the database for future use.

**JAAS :**   
[**Java Authentication And Authorization Service**](https://docs.oracle.com/en/java/javase/11/security/java-authentication-and-authorization-service-jaas-reference-guide.html) **:**

* Authentication: Identifying the entity that is currently running the code
* Authorization: Once authenticated, ensure that this entity has the required access control rights or permissions to execute sensitive code

**UML : unified modelling language :**

general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

Relational vs non-relational

**relational databases store data in rows and columns like a spreadsheet while non-relational databases store data don’t, using a key and value.**

**Git :**

**Git init**

**Git add sum.java (Staging area)**

Git status

Git commit –m “first commit”

Git log : previous all commits

Git branch

Git branch saurav

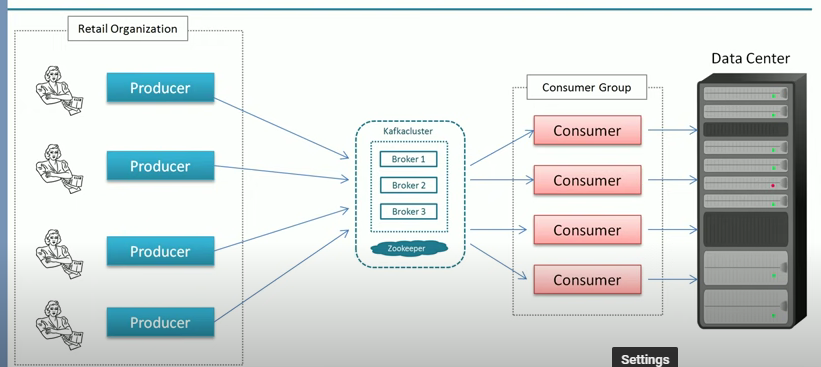
Git checkout Gaurav

Git merge Gaurav

Git remote add origin <https://dasd>

Git push –u origin master

**Kafka: Is a distributed system the runs on a cluster of computers**

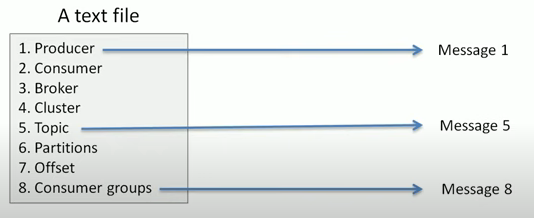




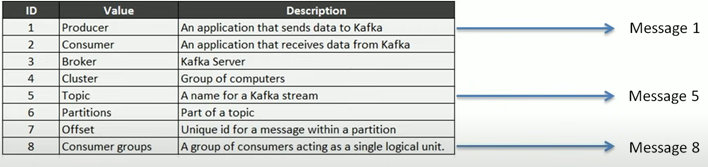
Producer: An application the sends message to Kafka.

Message: Small to medium sized piece of data.

**How producer sends message to kafka**: From a text file, each line is sent to kafka as a message.



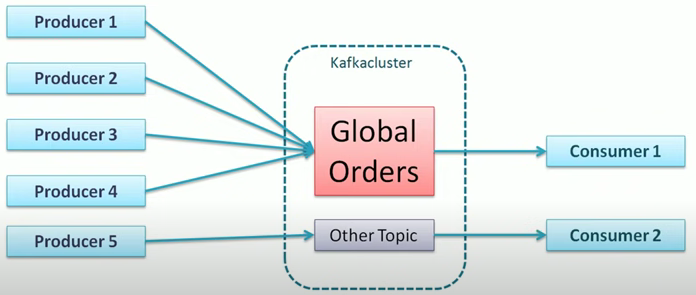
If a table is there (as a result of query), then each row will be sent as a message.



Consumer : Application that reads data from Kafka

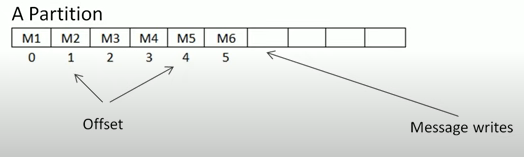
Cluster : A group of computers sharing workload for a common purpose. In this, each computer executing on instance of kafka.

Topic : unique name for a Data stream. (Immutable)



Partition: Breaks the TOPIC into partition and store each partition in a single computer. Number of partition is decided by us when we create a topic.

Offset: A sequence number given to a message when they arrive in a partition.



**Global unique identifier of a message:**



Consumer group **:** A group of Consumer acting as a single unit to share the work. Each Consumer will take one partition. Make sure that not more than 2 consumer reads the same partition to avoid double reading of records.

**Best practices for designing REST API’s :**

* Accept and respond with JSON
* Use nouns instead of verbs in endpoint paths
* Use logical nesting on endpoints
* Handle errors gracefully and return standard error codes
* Allow filtering, sorting, and pagination
* Maintain good security practices
* Cache data to improve performance
* Versioning our APIs

JDK : provides the environment to **develop and execute(run)** the Java program.

JRE : installation package that provides an environment to **only run(not develop)** the java program(or application)onto your machine.

JVM : executing the java program line by line.