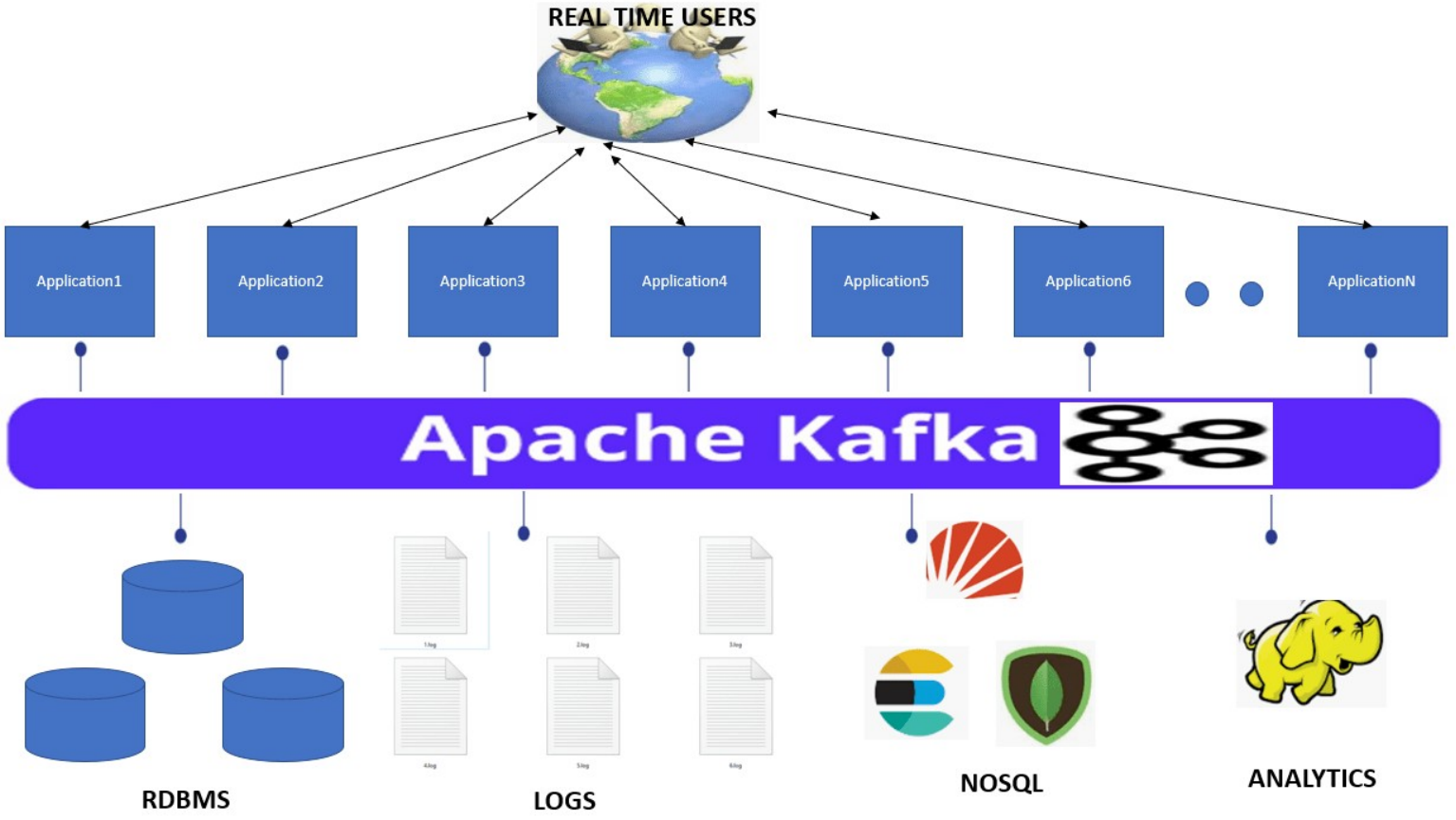
Introduction to Apache Kafka

Rajeev Gupta
Java Trainer & consultant

What is Apache Kafka?

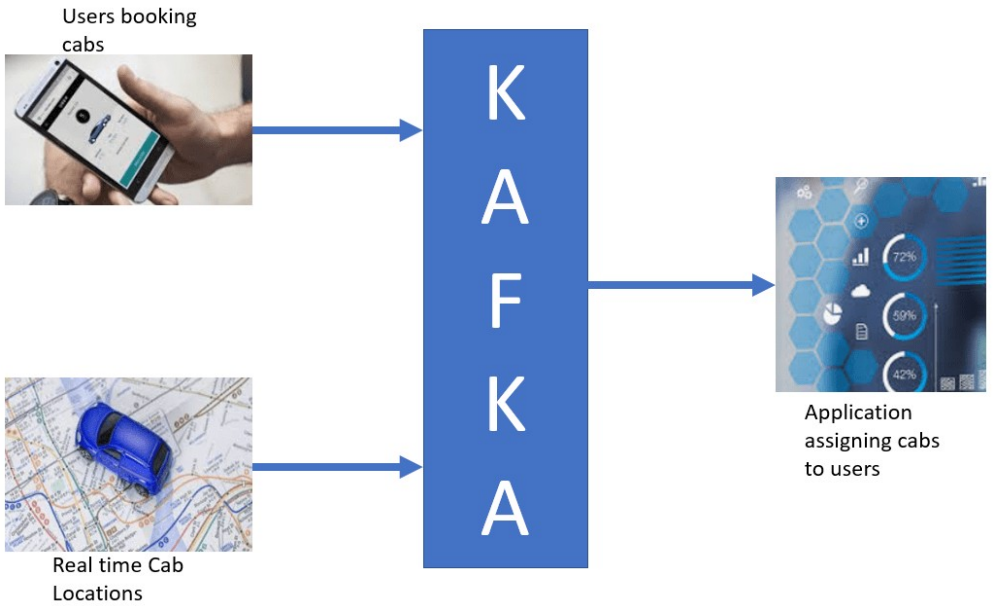
- Apache Kafka is an open-source stream-processing software platform developed by LinkedIn and donated to the Apache Software Foundation.
- It has been developed using Java and Scala.
- Apache Kafka is a high throughput distributed messaging system for handling real-time data feeds.

Traditional Messaging vs SystemApache Kafka



Apache Kafka Usages

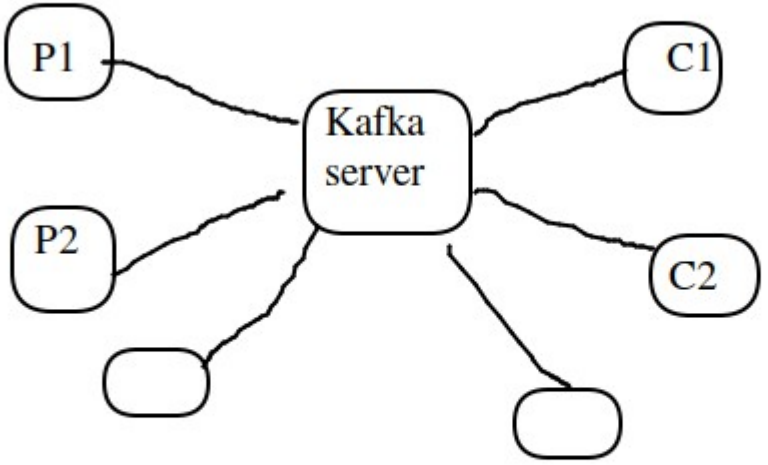
- Real time example of Apache Kafka is Uber cab booking service. Uber makes use of Kafka to send User and Cab information to Uber Cab Booking System.



What is Apache Kafka?

Event driven arch, millions of messages can be process per second

multiple producer

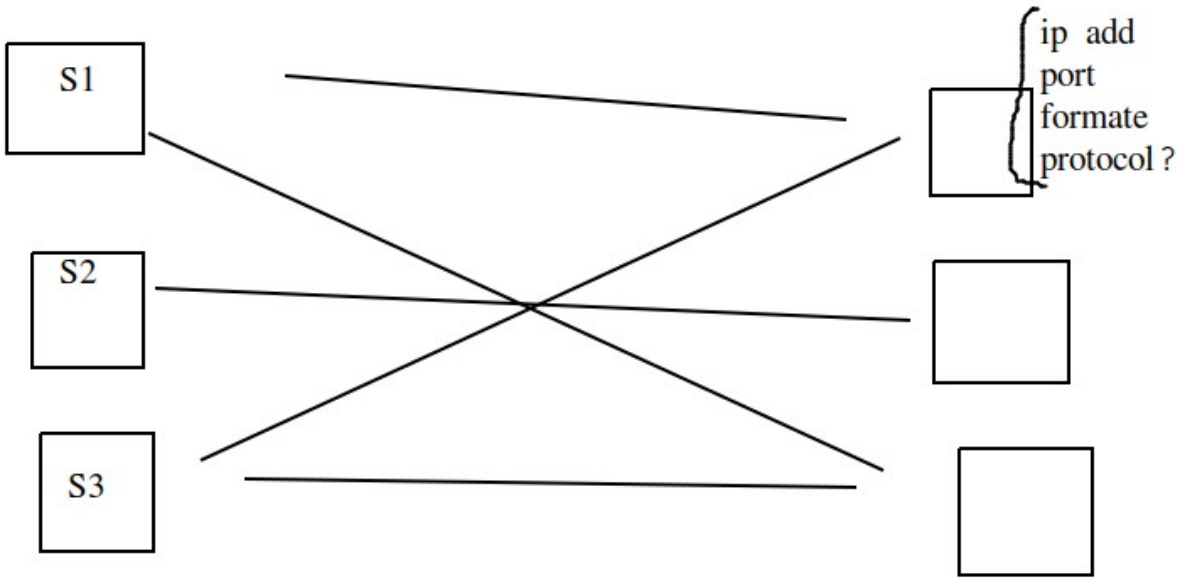


multiple broker and consumer

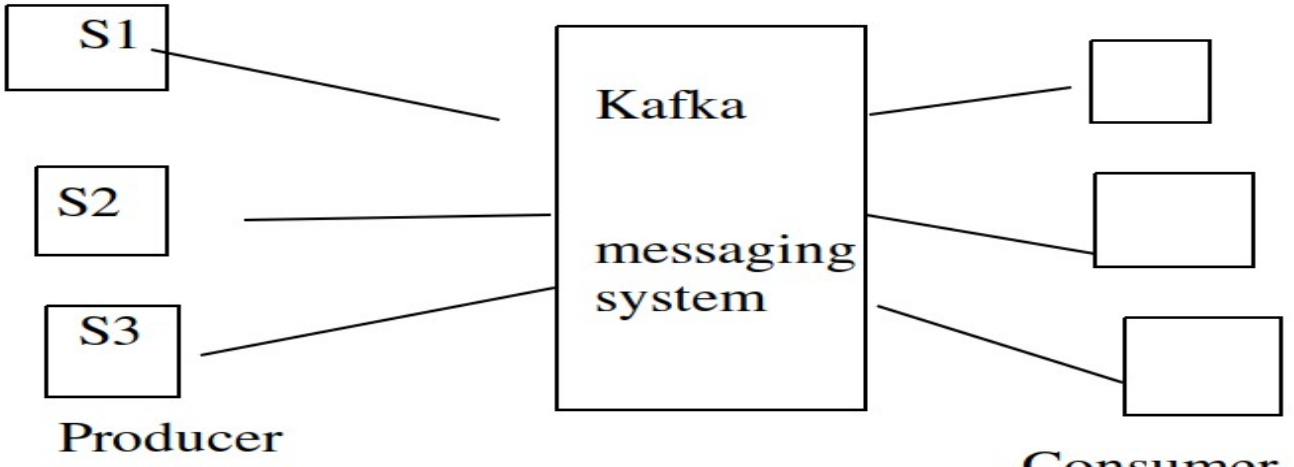
Apache Kafka is an open-source stream-processing software platform

In distributed env. kafka is reffered as kafka cluster made of more then one kafka server

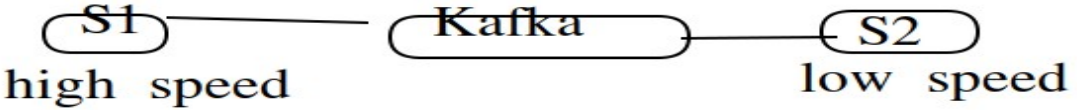
Need of Apache Kafka?



Decoupling data processing pipeline

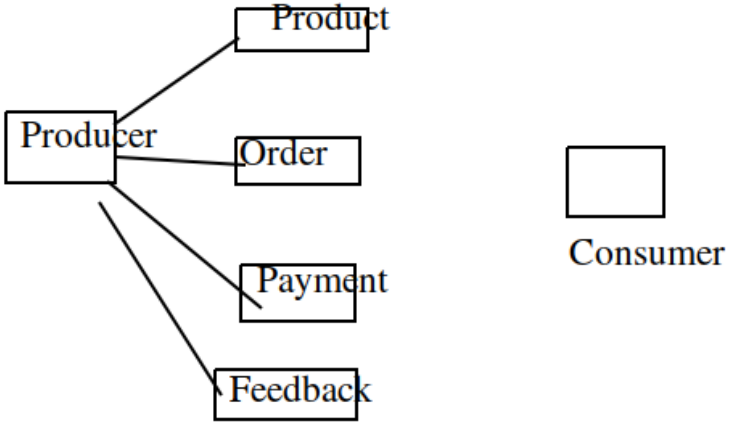


It solve Complex communication problem
it solve speed mismach problem

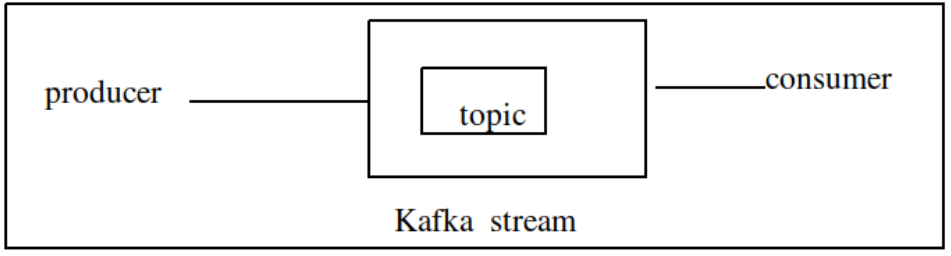


What is topic

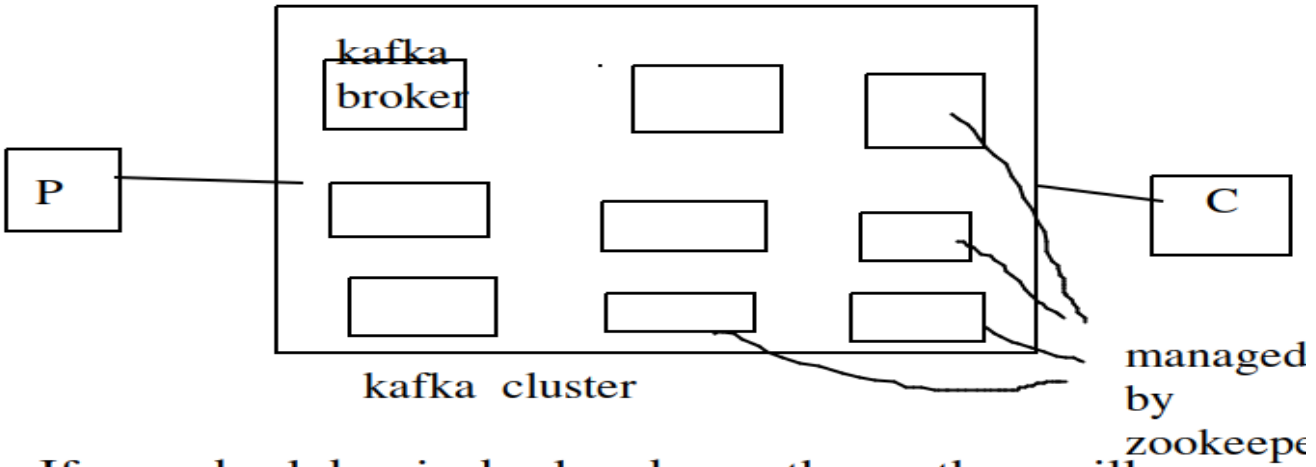
What if producer is sending 4 type of data?
Consumer get confused if he is only intrested on product data



Solution: segrate the data streams=> 4 topics for each category of data
similer to database table: related to one type of data



Scalability and fault tolerence(Zookeeper)

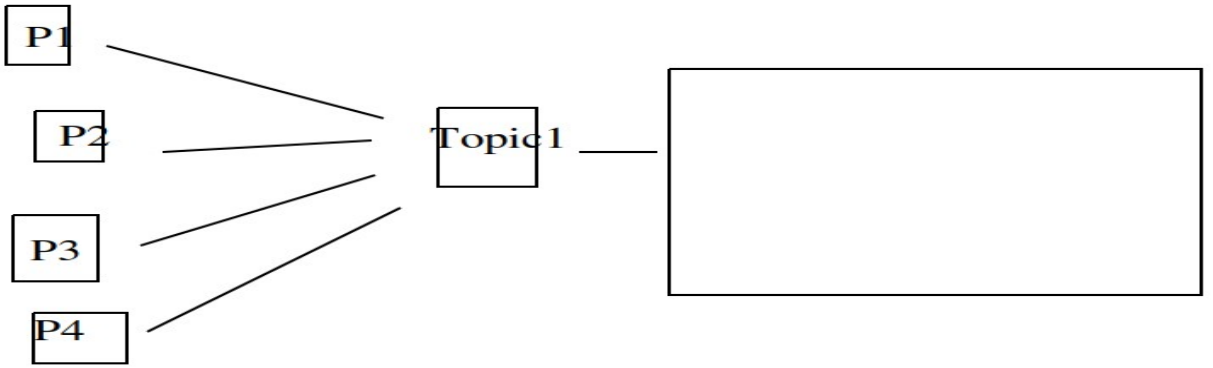


If any brokder is broke down then other will take care

How to manage it?
if one broker is broken down then who coordinate
Distributed service to manage lare amout of host
Focus on BL and not on distribution of logic
First we have to start the zookeeper and then kafka broker

Horizontal scaling: add new kafka broker if required

Problem of parallelism



If P1 is sending data to topic1 then other producer p2 p3 etc can not send at the same time=> how to handle ?

To solve this issue we have another concept partition

You divide the topi into multiple partion

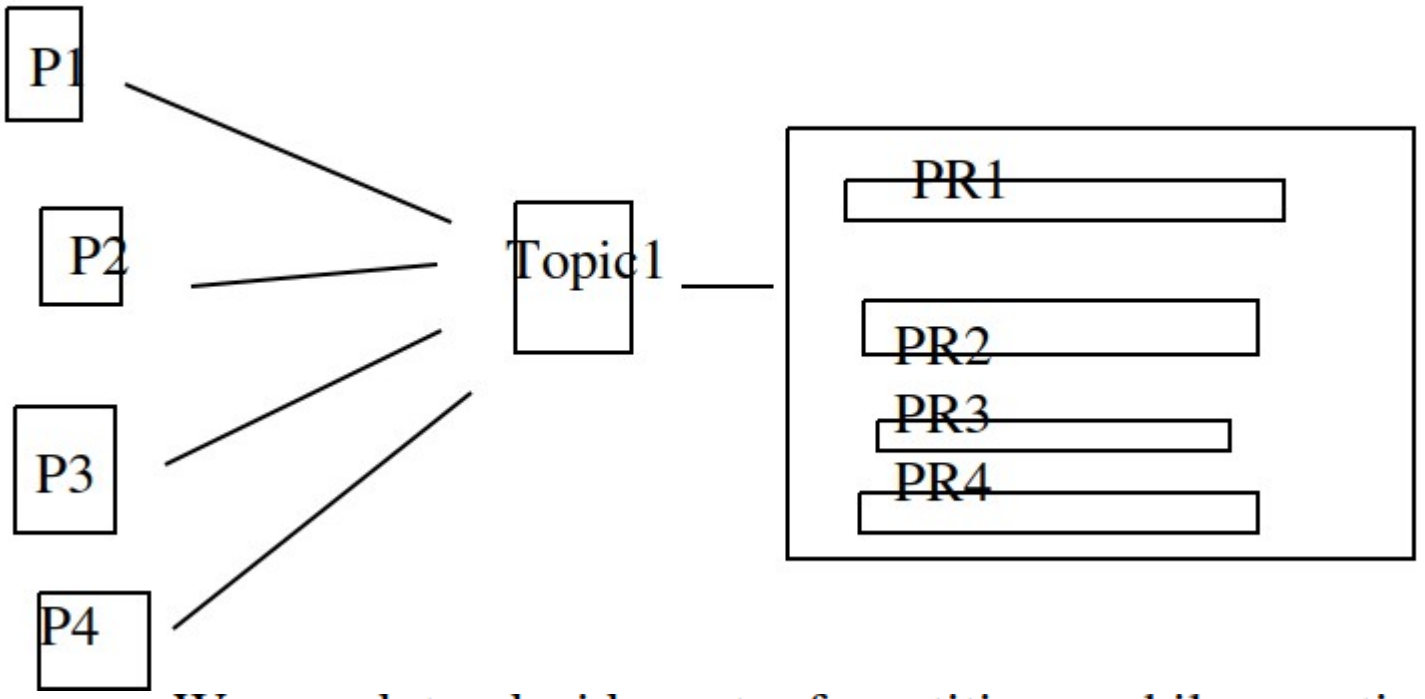
PR1

PR2

PR3

PR4

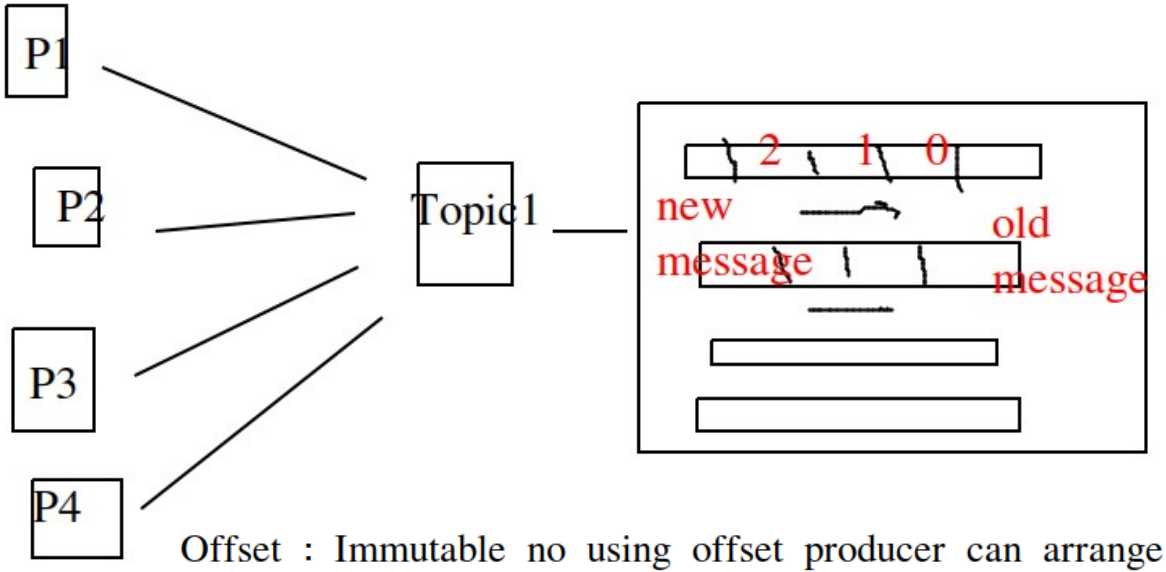
Problem of parallelism: Partitions



We need to decide not of partitions while creating topics we need to tell no of partitions

Partitions offset

Producer send the data into message offset system



Offset : Immutable no using offset producer can arrange data into accending /decending order

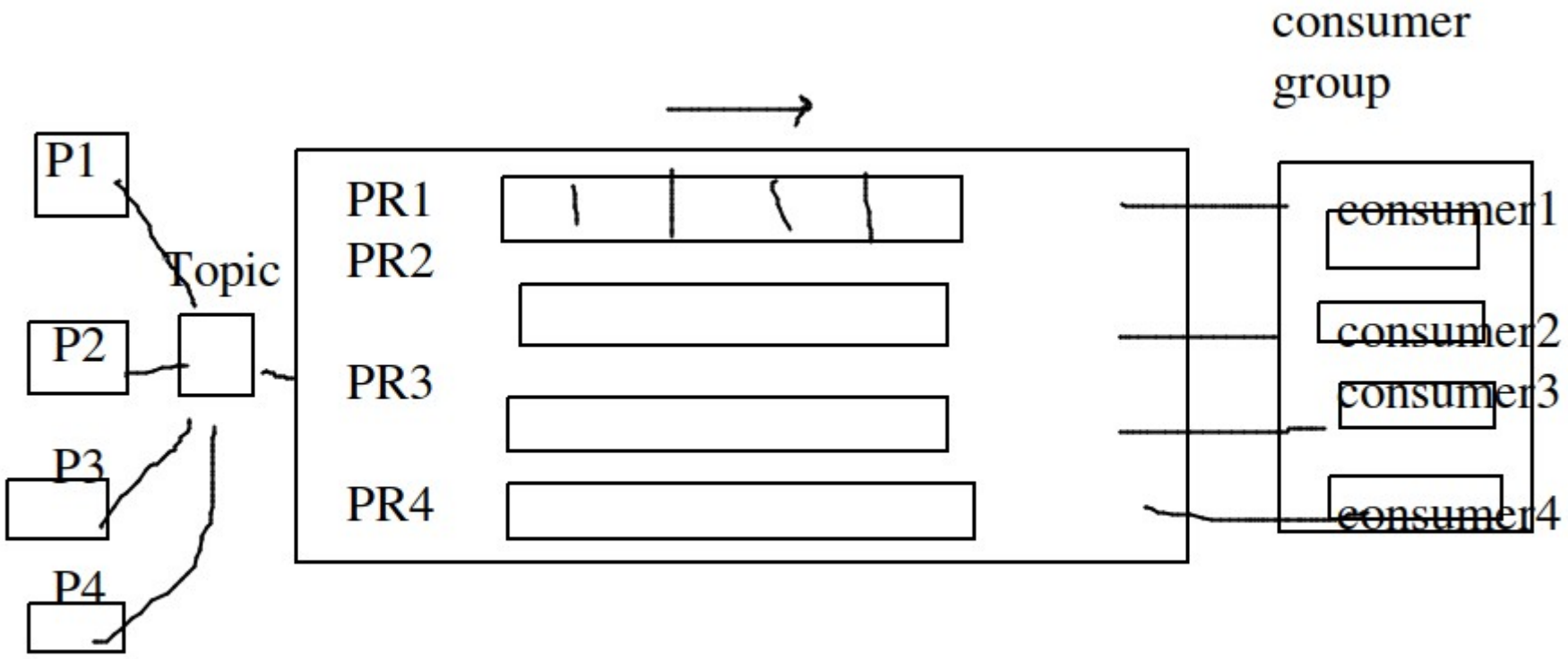
Partition 1 data is not same as partion 2 data

To recognize message
which topic id?
which partition id?
which offset id?

Consumer Group

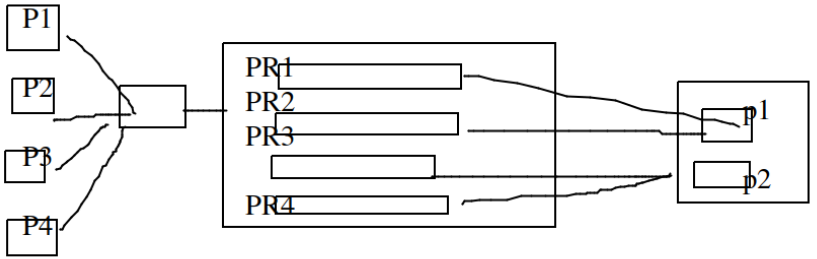
- We have only one consumer how data processing happens
- You create many consumer and connect one partion
- We can get all the data into one shot from 4 partitions, that is called consumer group
- Consumer group: single logical using they can share the work

Consumer Group



Consumer Group

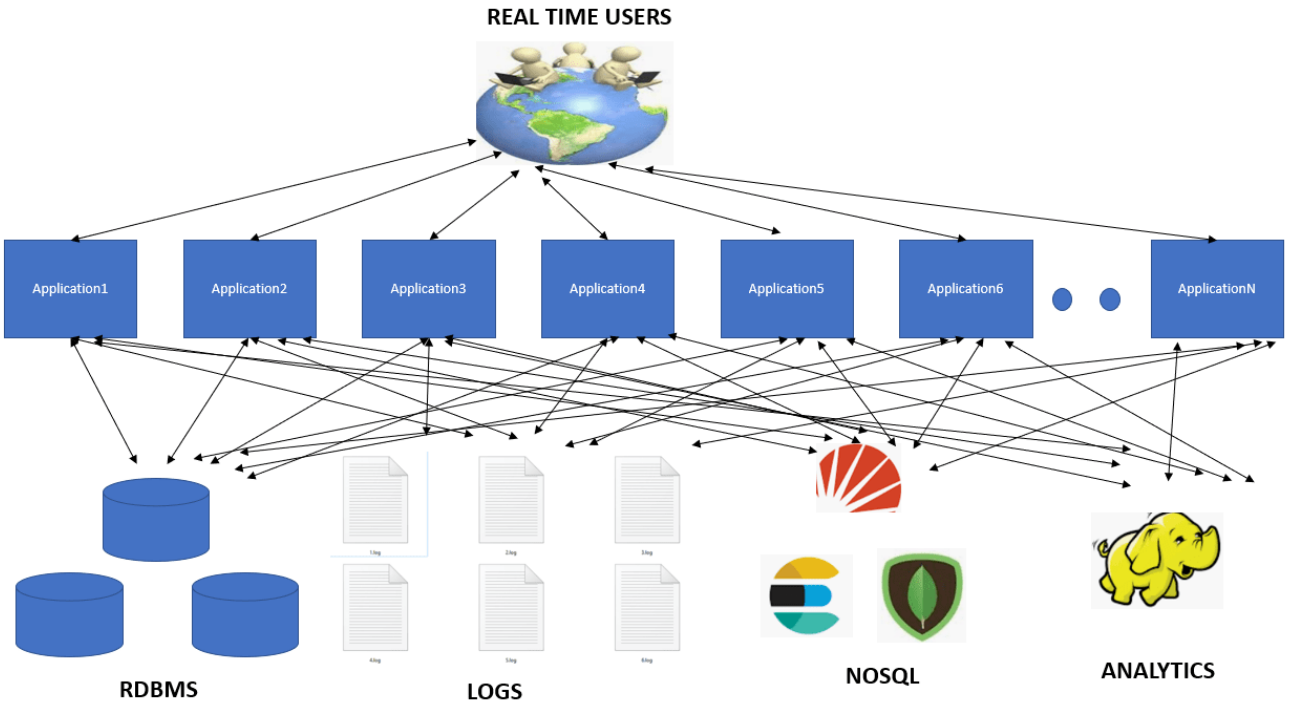
- Case I: if only 2 consumer is there in consumer group



- Case 2: if 4 consumer is there , each one take data from each partition
- Case 3: if we have 5 consumer group, 1 will be idel
- Case 4: if one consumer then all data send to that

What is need of Apache Kafka?

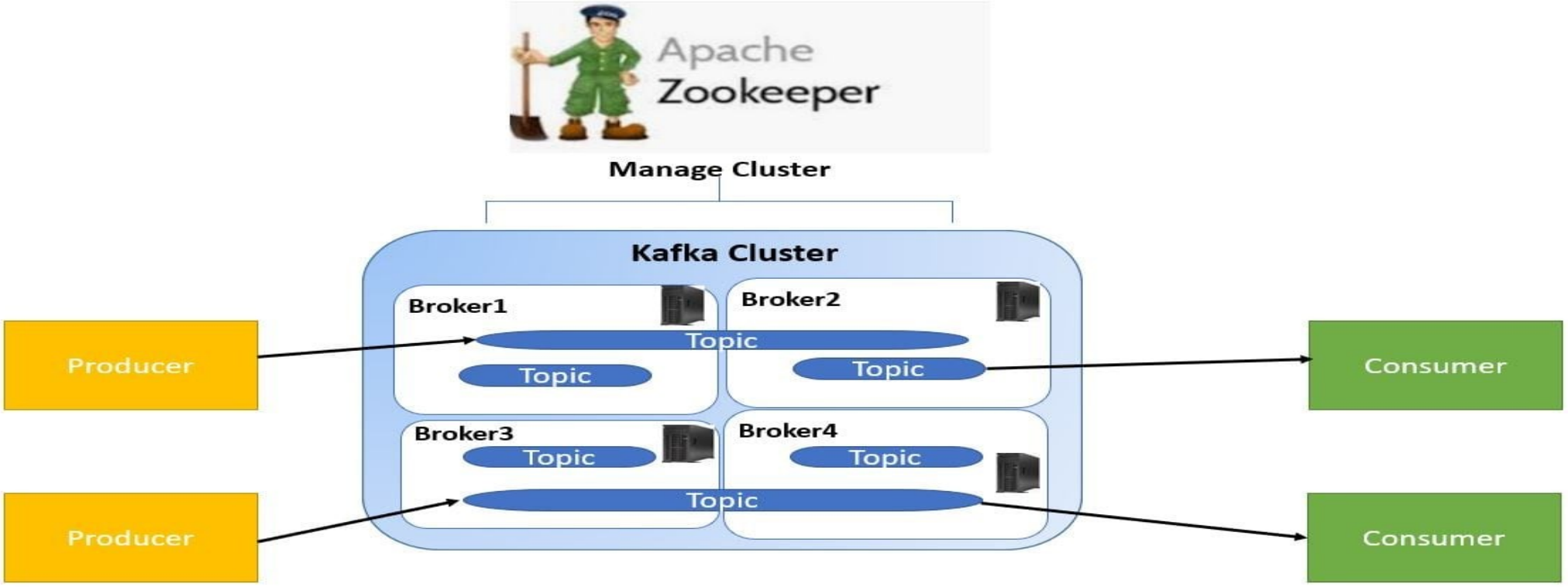
- Complex web of applications involving point to point data movement. This involves moving large amount of data from one point to another.



Replication factor

- Consider topic1 with 4 partition, then not all partition go into the broker
- Partition will be distribtued into multiple broker
- What if broker 2 is gone?
- How to solve the issue => replication copy
- If RF =3 then T1P1 should be replicated to 3 places in different brokers
- Although T1P1 is at 3 places one of them is called leader
- Kakfa zooker send data to the leader and then leader distributed/ replicate to others

Zookeeper


Apache
Zookeeper
Manage Cluster
Kafka Cluster
Broker1
Topic
Topic
Broker2
Topic
Broker3
Topic
Topic
Broker4
Topic
Producer
Producer
Consumer
Consumer

Zookeeper

- To manage the cluster we make use of Apache Zookeeper. Apache Zookeeper is a coordination service for distributed application that enables synchronization across a cluster.
- Zookeeper can be viewed as centralized repository where distributed applications can put data and get data out of it.
- It is used to keep the distributed system functioning together as a single unit, using its synchronization, serialization and coordination goals, selecting leader node.