Agenda.

→ Kafka.

→ Zookeeper.

Problem Statement

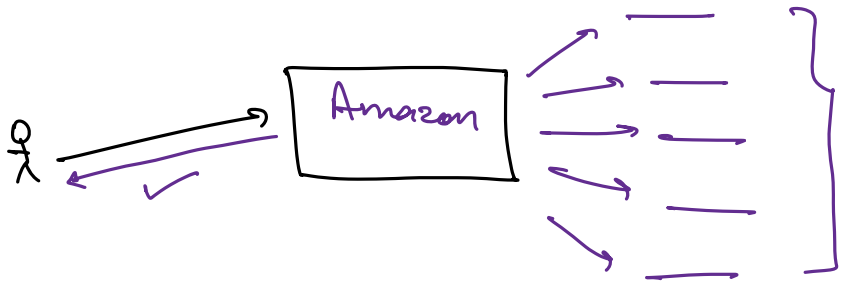
Video Upload on YT.

Order placement at Amazon.

→ When we place an order at Amazon :-

→ Amazon must be taking lot of actions once an order gets confirmed.

1) Inventory Service
2) Seller Service
3) Notification Service
4) InvoiceService
5) DeliveryService



→ Customer places an order.

→ Customer completes the payment

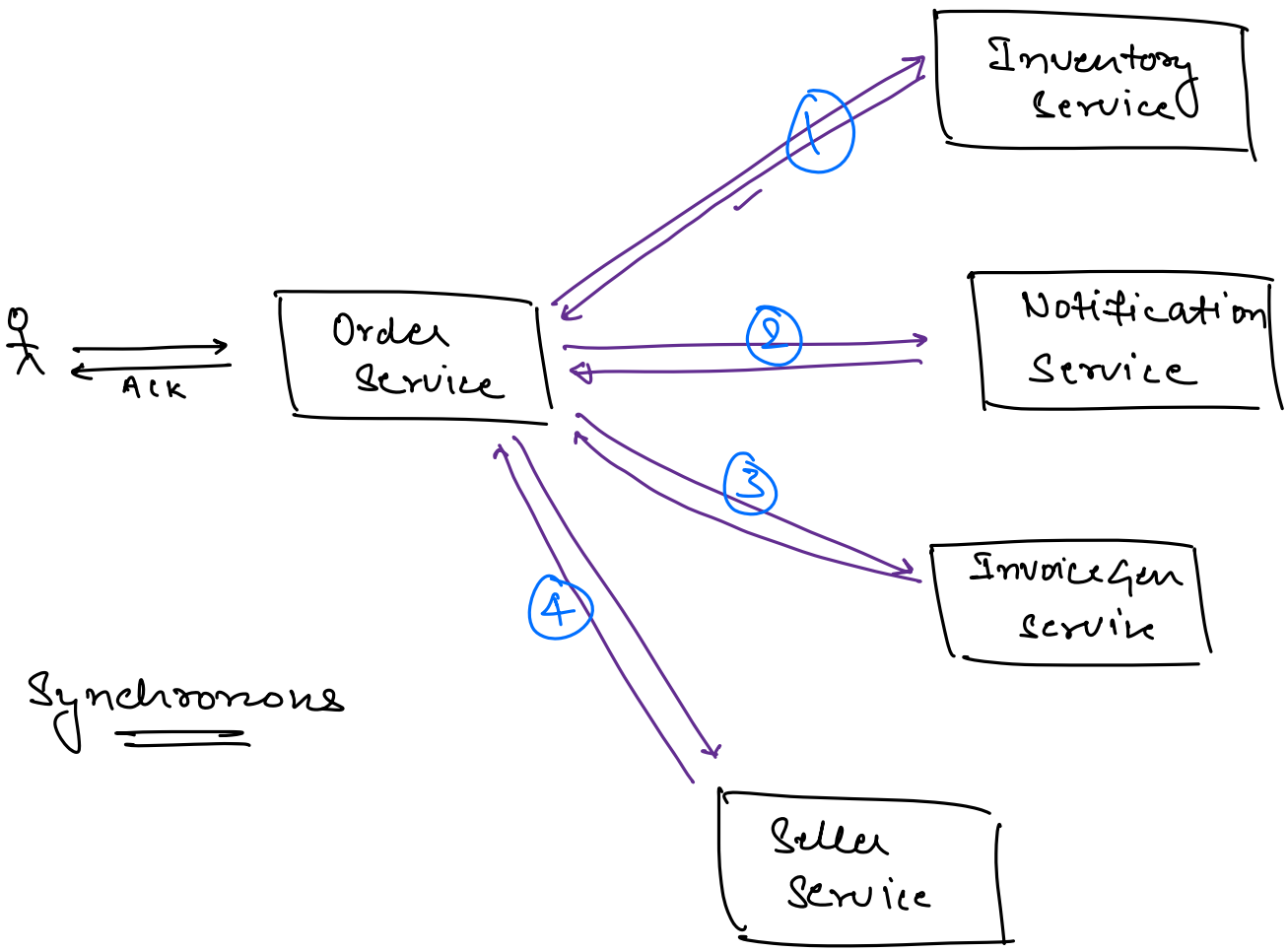
→ Customer gets order confirmation page.

→ And all the above mentioned tasks will be executed by Amazon.

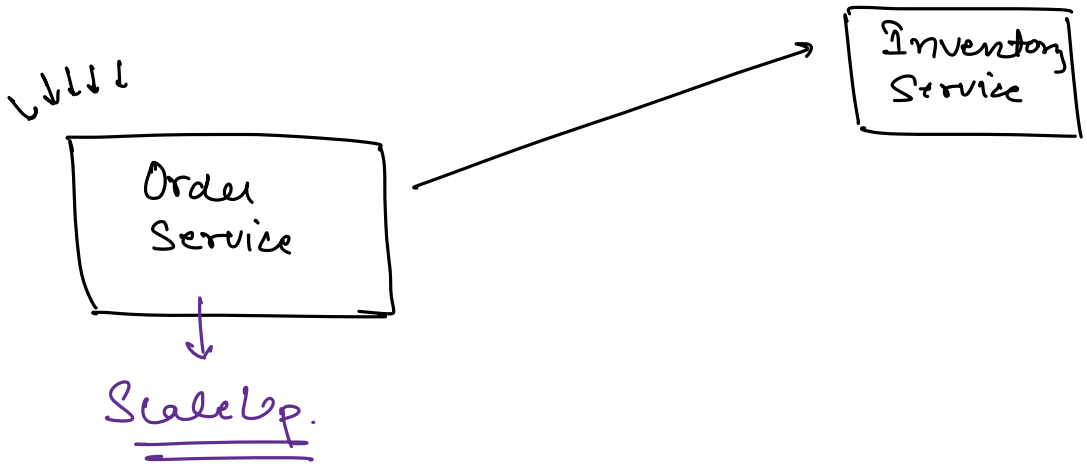
⇒ Asynchronous Task.

↓

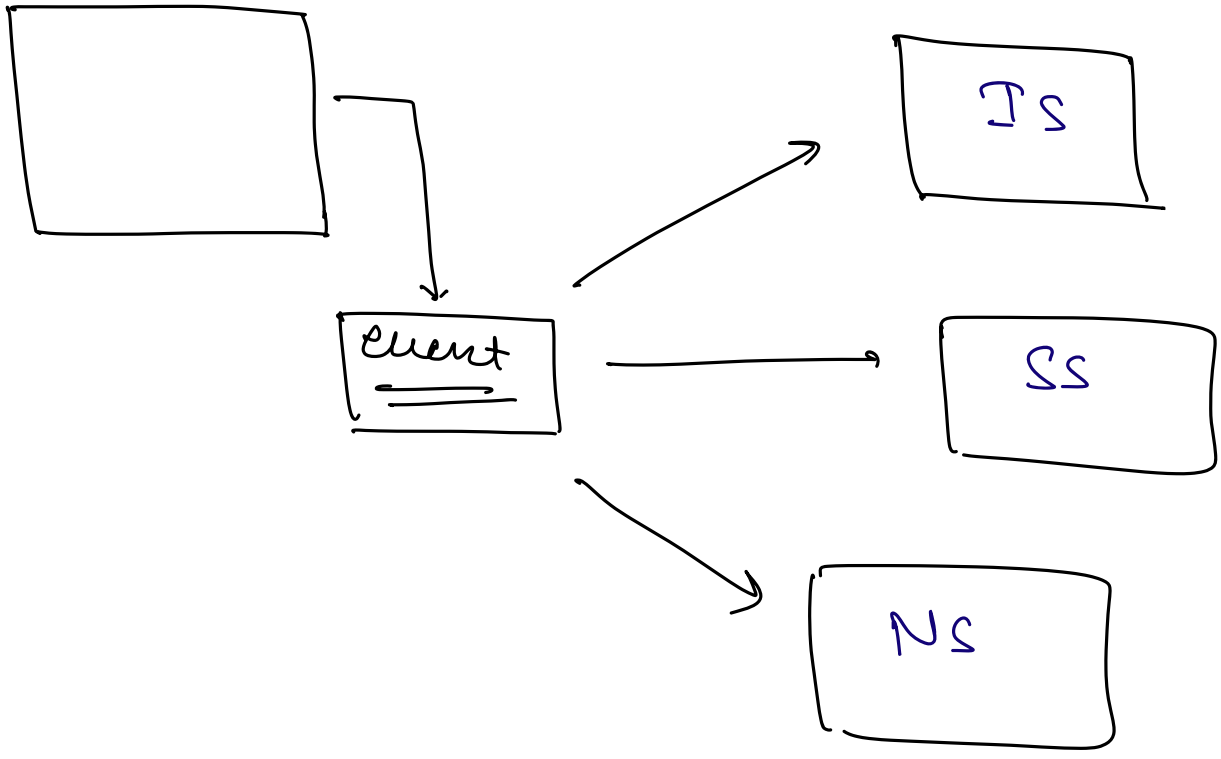
Parallely.



⇒ Synchronous



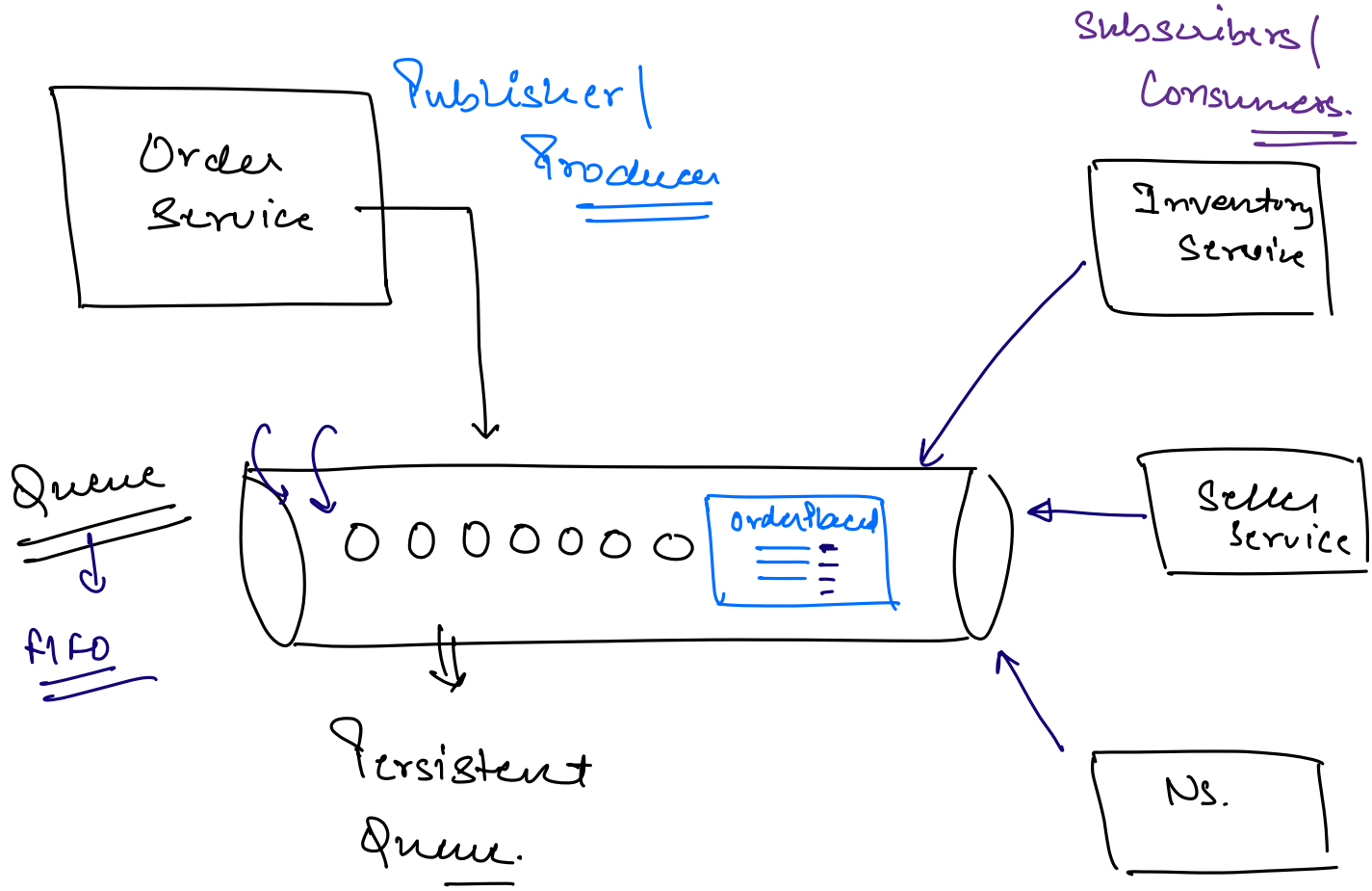
⇒ If we scale up OrderService, then we'll have to scaleup Inventory Service as well otherwise Inventory Service might not be able to execute all the requests.



Message Queue

- Kafka.
- Rabbit MQ
- Active MQ

| → SQS
| →

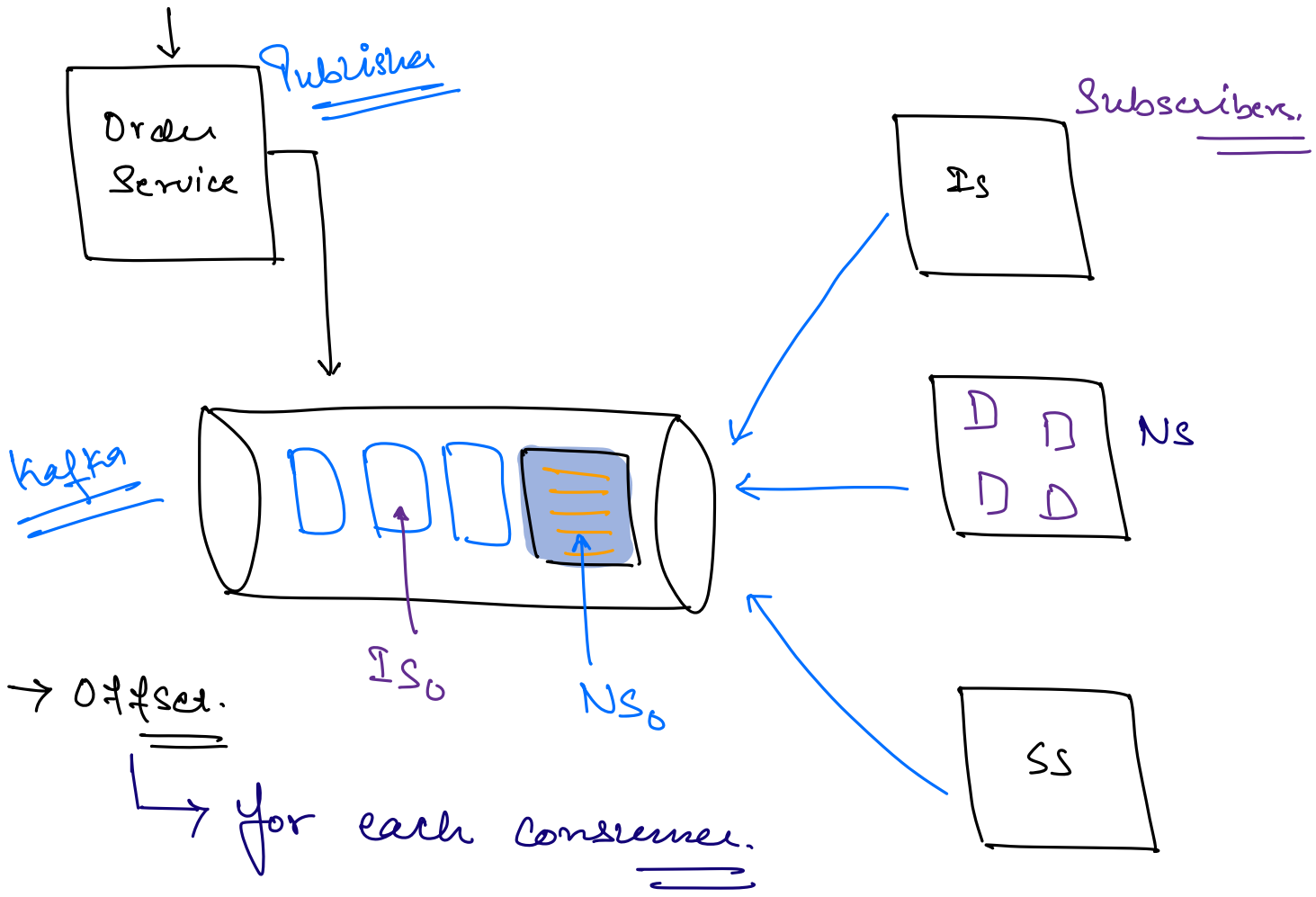


⇒ Timeout : Configurable.

↳ 24 hrs.

⇒ DLQ. : Dead letter Queue.

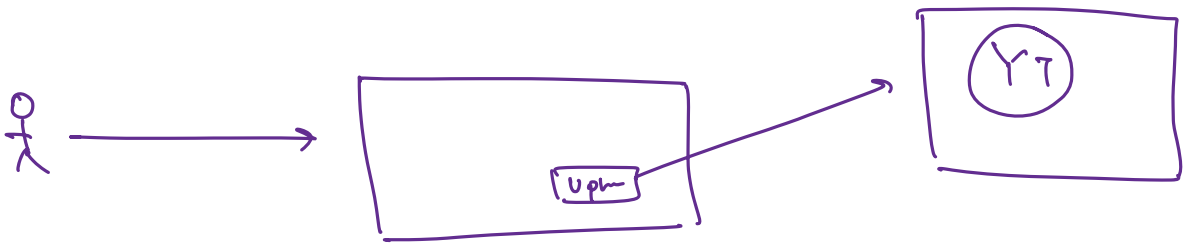
⇒ Asynchronous communication



→ Offset.

↳ for each consumer.

Video Upload on YT.



→ Copyright Check

→ Convert into multiple resolutions.

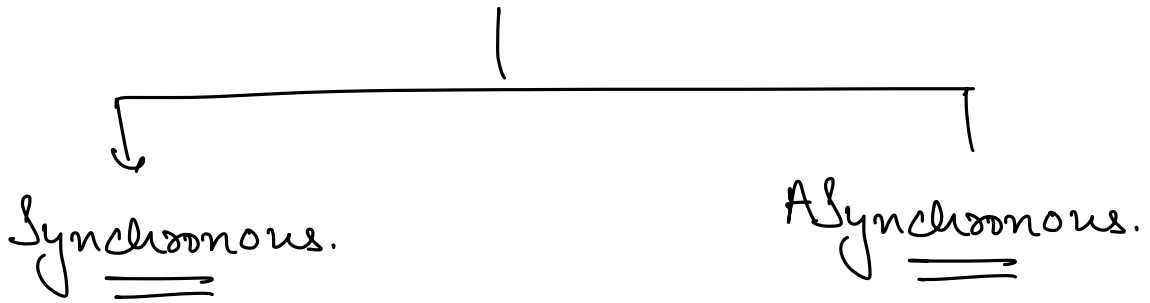
→ Tnc

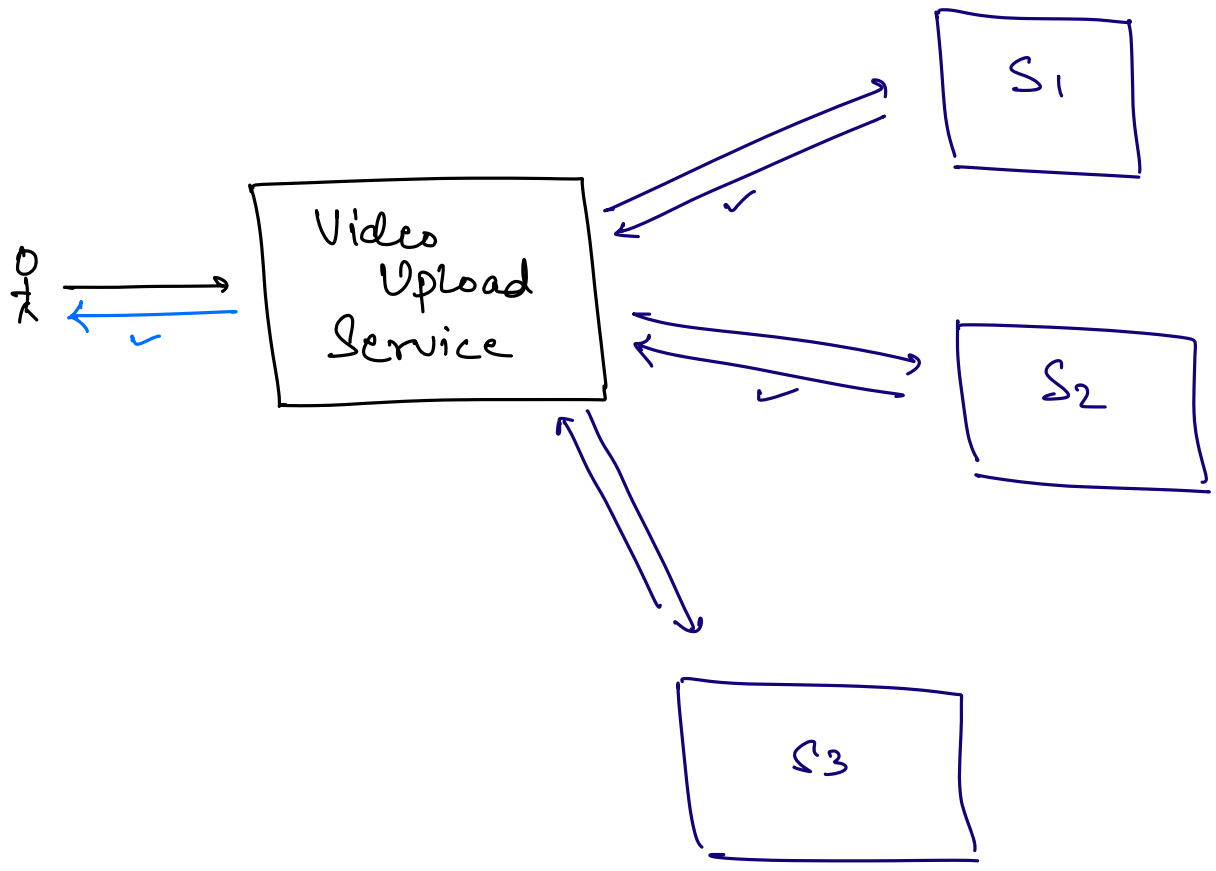
→ Captions

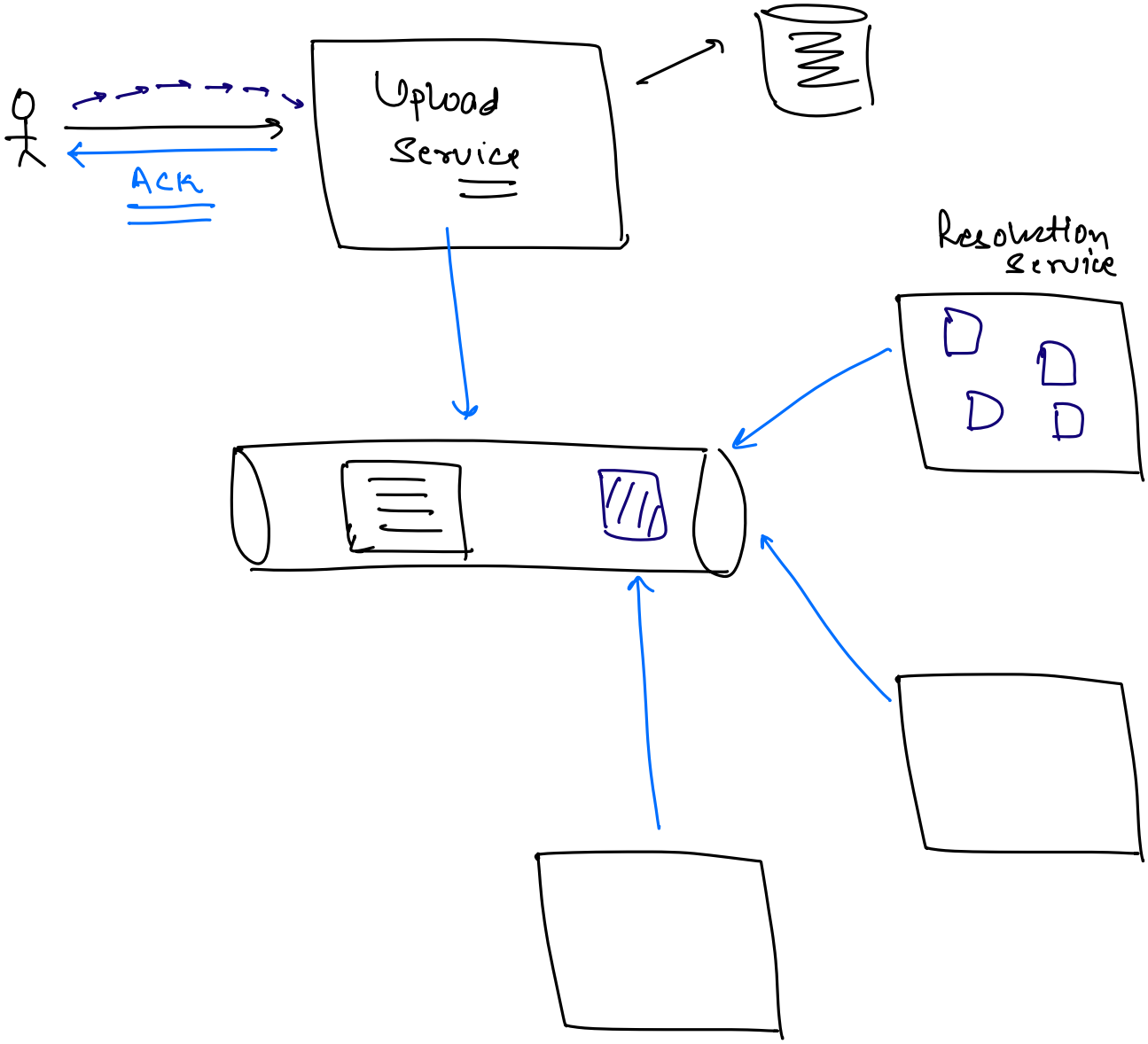
→ —

—

—







Consumer Group.

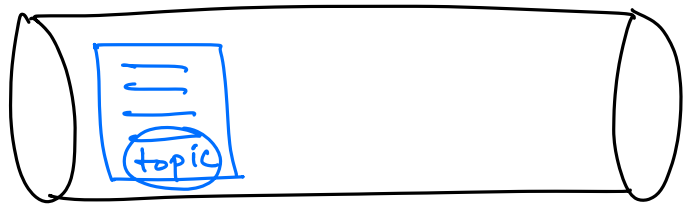
⇒ Kafka makes sure that only one consumer from a consumer group will be able to read a msg from the Queue.

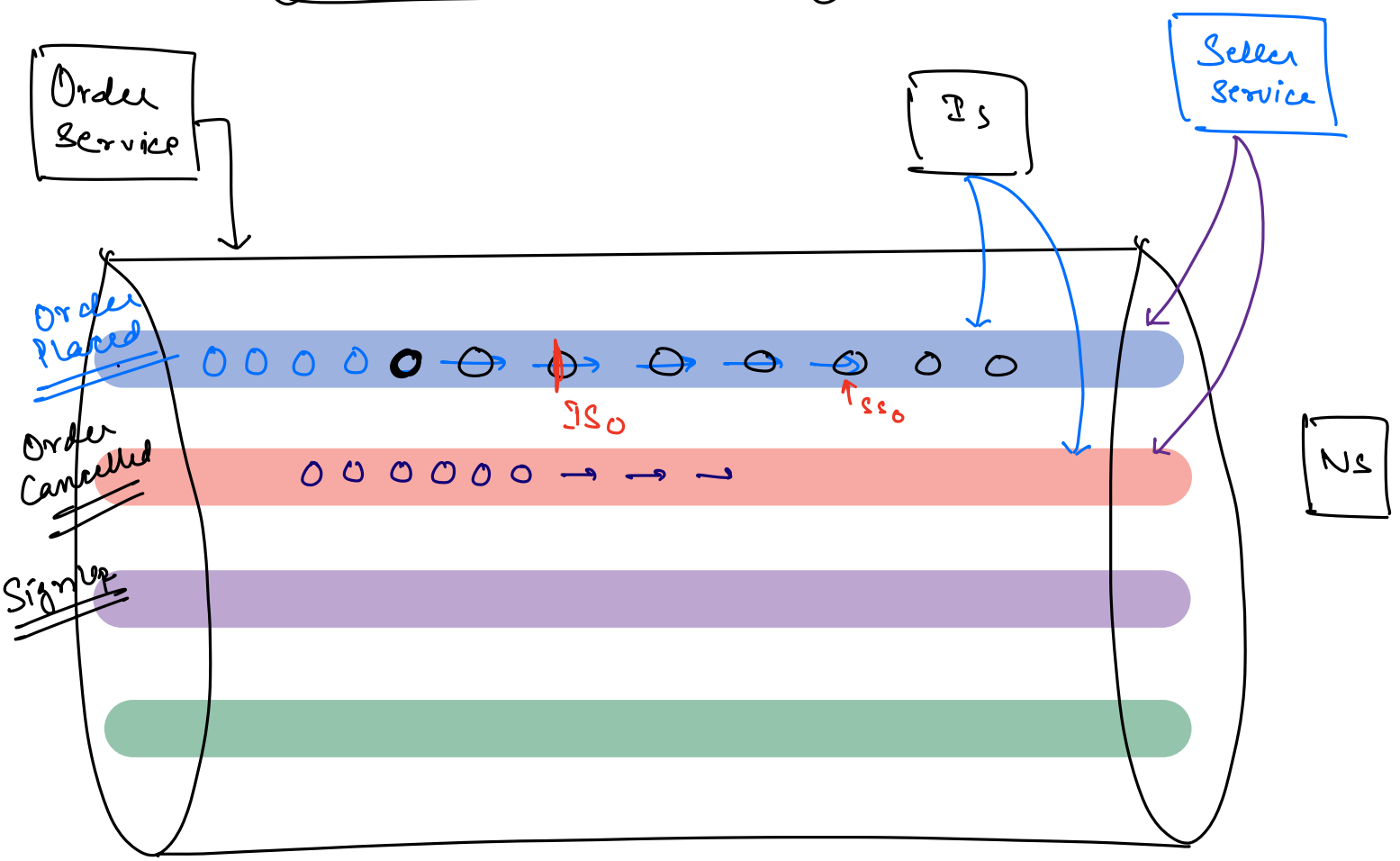
=> Internal architecture of kafka.

↓

SHARDING.

Topic : Type of Event.





@KafkaListener(....., topic = "orderPlaced")

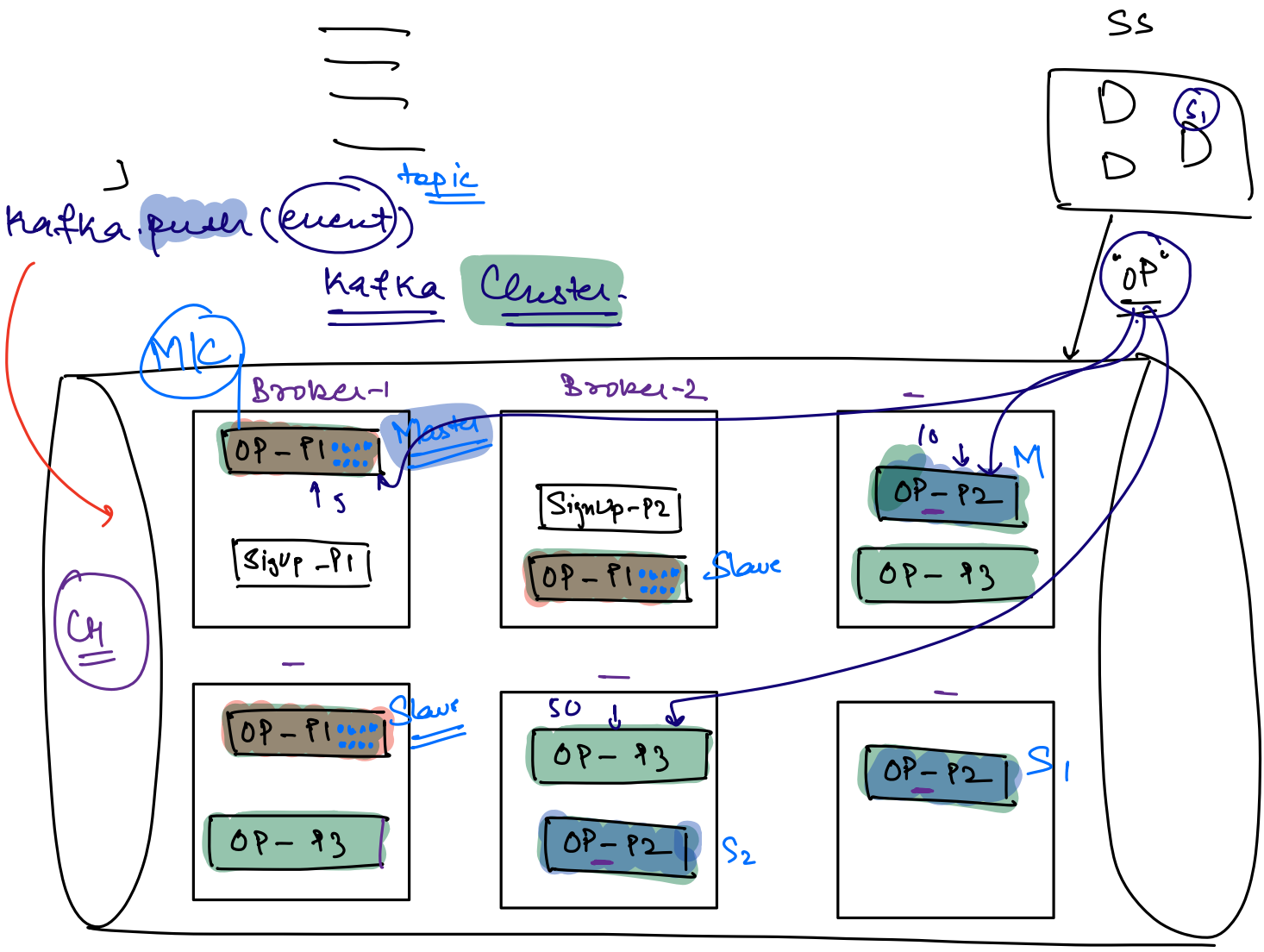
handleOrderPlacedEvent(=) {

}

@KafkaListener(....., topic = "orderCancelled")

handleOrderCancelled(=) {

}



Topic

↳ Order Placed

Master Slave

⇒ Partition.

⇒ Order Placed → 3 Partitions.

SignUp ⇒ 2 partitions.

Kafka Cluster

↳ Brokers ≡ M/c

↳ Partitions.

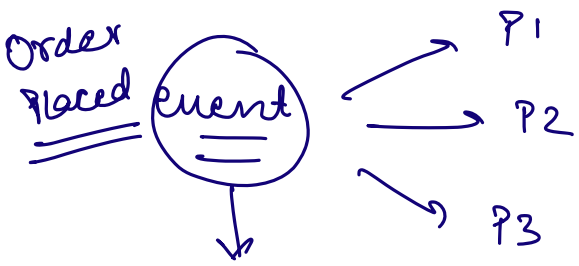
Topic

≡ Category of events.

event

↓

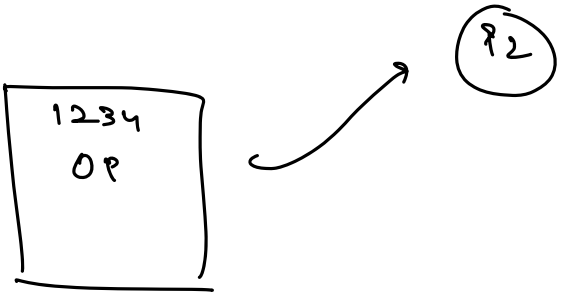
topic



≡, topic

+

SHARDING KEY (orderid)



⇒ Replicas.