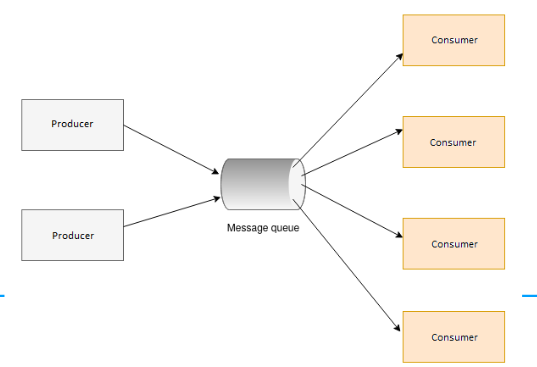
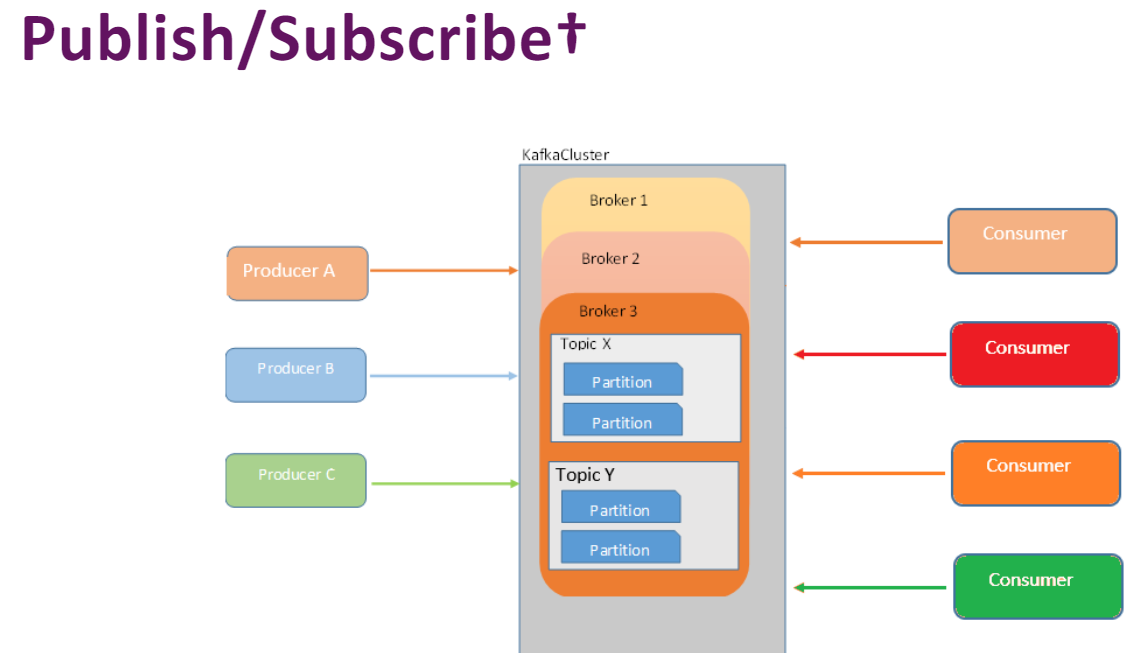
23/04/24 **Lecture 10: Apache Kafka**

Kafka, an Apache project, optimises stream processing on large data sets, facilitating efficient data delivery from sources to destinations, a practice utilised by tech companies like LinkedIn and Uber.

**Mechanism**

Kafka is a Distributed Streaming Platform that uses a publish-subscribe communication pattern between producers and consumers, enabling distributed applications and microservices to read, write, and process data streams in parallel.





**Kafka Topic Partitioning and Storage**

• Kafka partitions topics into disjoint lists of messages, distributed to brokers.

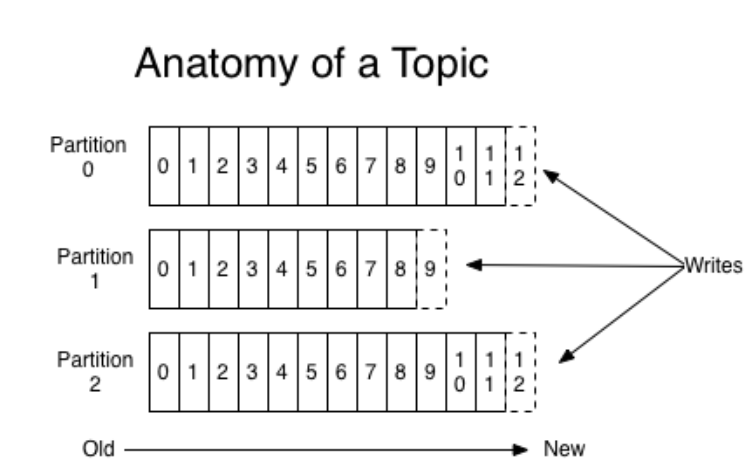
• Provides scalable storage for topics spanning multiple disks.

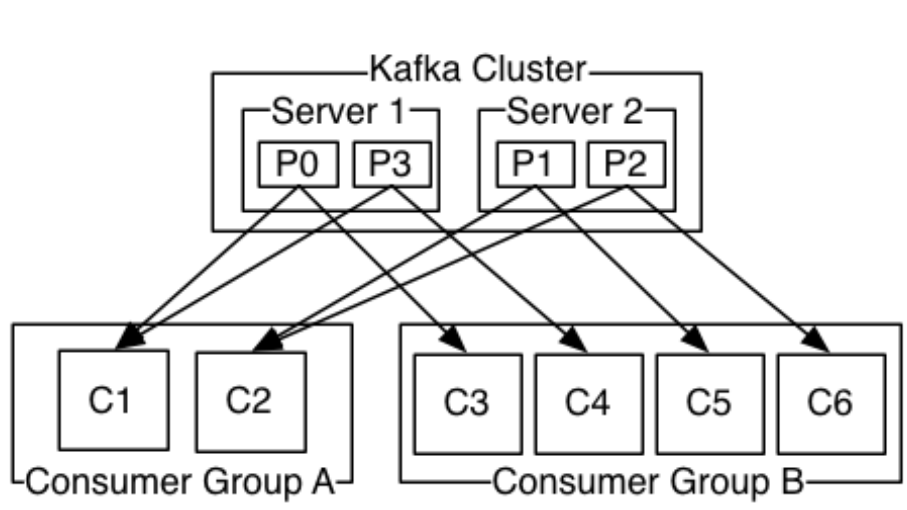
• Supports parallelism in topic reading.

• Allows concurrent reading of multiple partitions by a consumer group.

• Avoids bottleneck of all reading from the same broker.

• Consumer group could be a Spark Structured Streaming job distributed over multiple executors in a cluster.



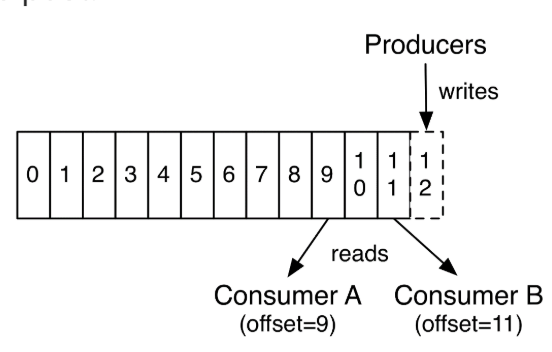


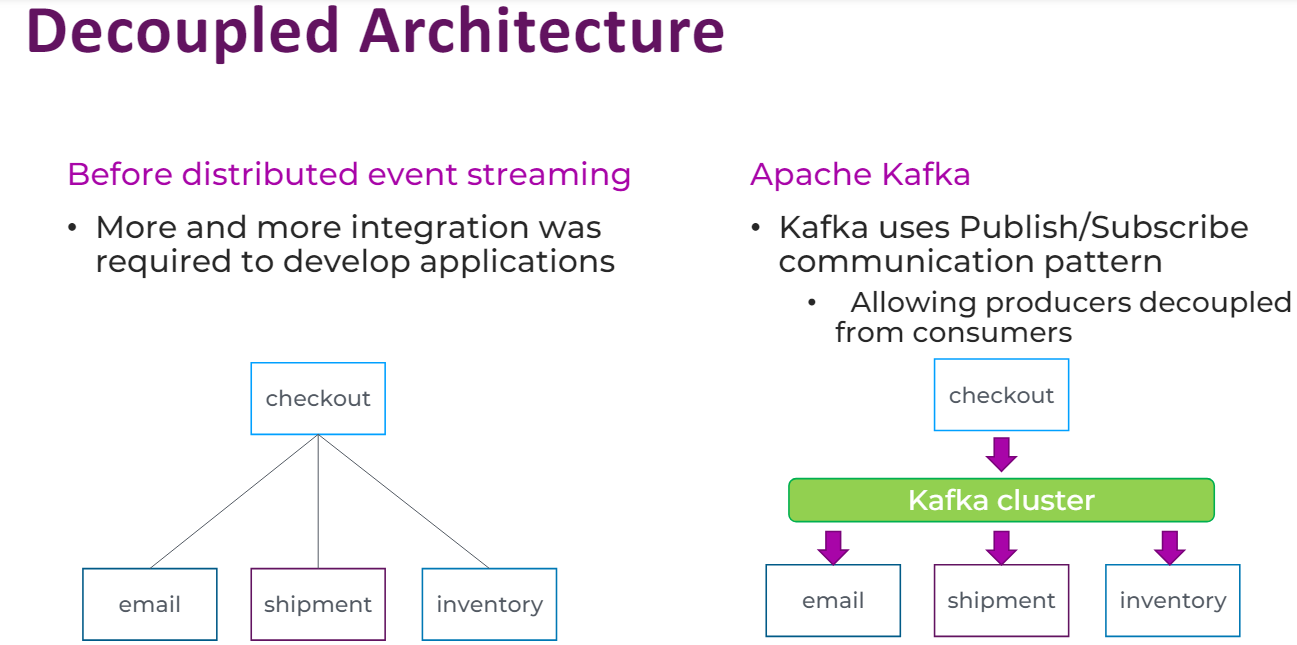
**Kafka Cluster's Durability and Retention**

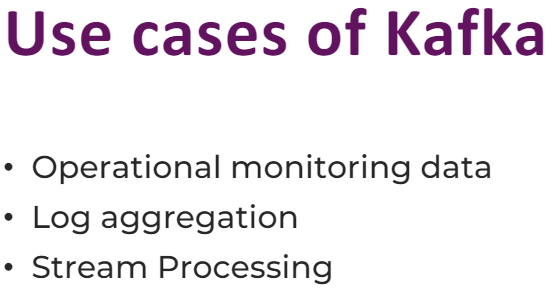
**•** Persistently persists all published records using a configurable retention period.

• Stores received messages even after consumption by one consumer.

• Allows other consumers to view past messages.







**Pros and cons**

Pros:

- Scalability.

- Fault tolerance

- High throughput

- Message retention

- Stream processing

- Ecosystem

Cons:

- Complexity

- Operational overhead

- Learning curve

- Storage requirements

- Configuration challenges