CLOUD COMPUTING APPLICATIONS

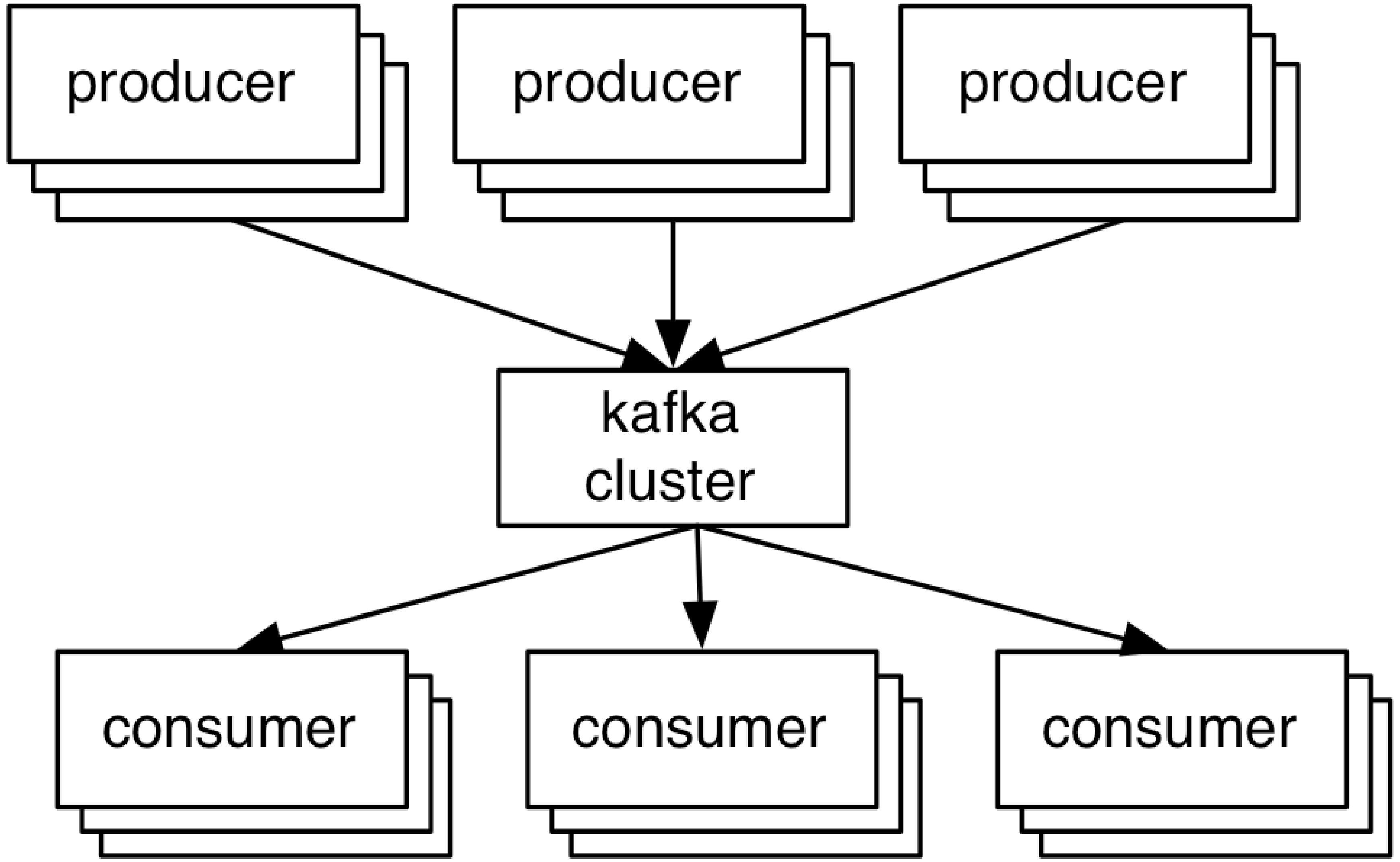
Kafka

Roy Campbell & Reza Farivar

Thanks to public domain slides Jiangjie (Becket) Qin

Contents

Kafka: a distributed, partitioned, replicated publish subscribe system providing commit log service



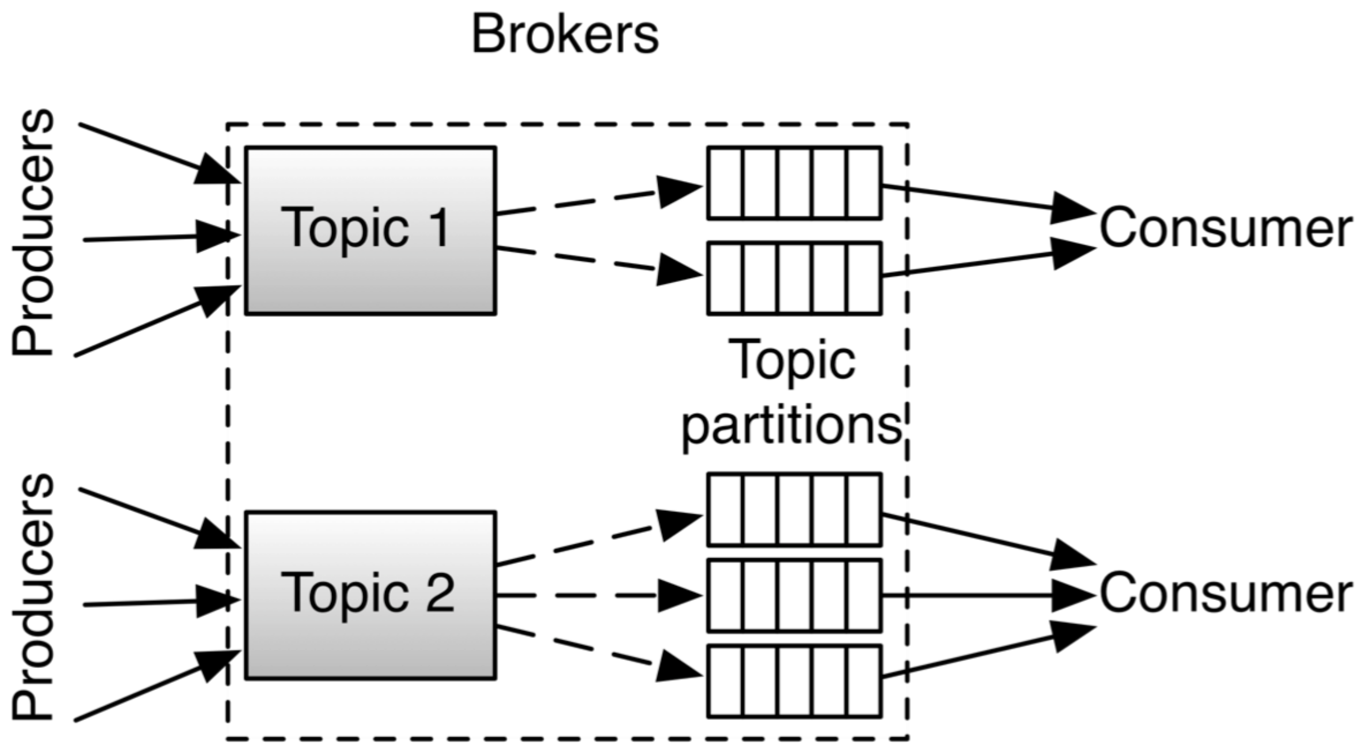
Description

- Kafka maintains feeds of messages in categories called *topics*.
- Processes that publish messages to a Kafka topic are *producers*.
- Processes that subscribe to topics and process the feed of published messages are *consumers*.
- Kafka is run as a cluster comprised of one or more servers each of which is called a *broker*.
- Communication uses TCP, Clients include Java

Characteristics

- Scalability (Kafka is backed by file system)
  - Hundreds of MB/sec/server throughput
  - Many TB per server
- Strong guarantees about messages
  - Strictly ordered (within partitions)
  - All data persistent
- Distributed
  - Replication
  - Partitioning model

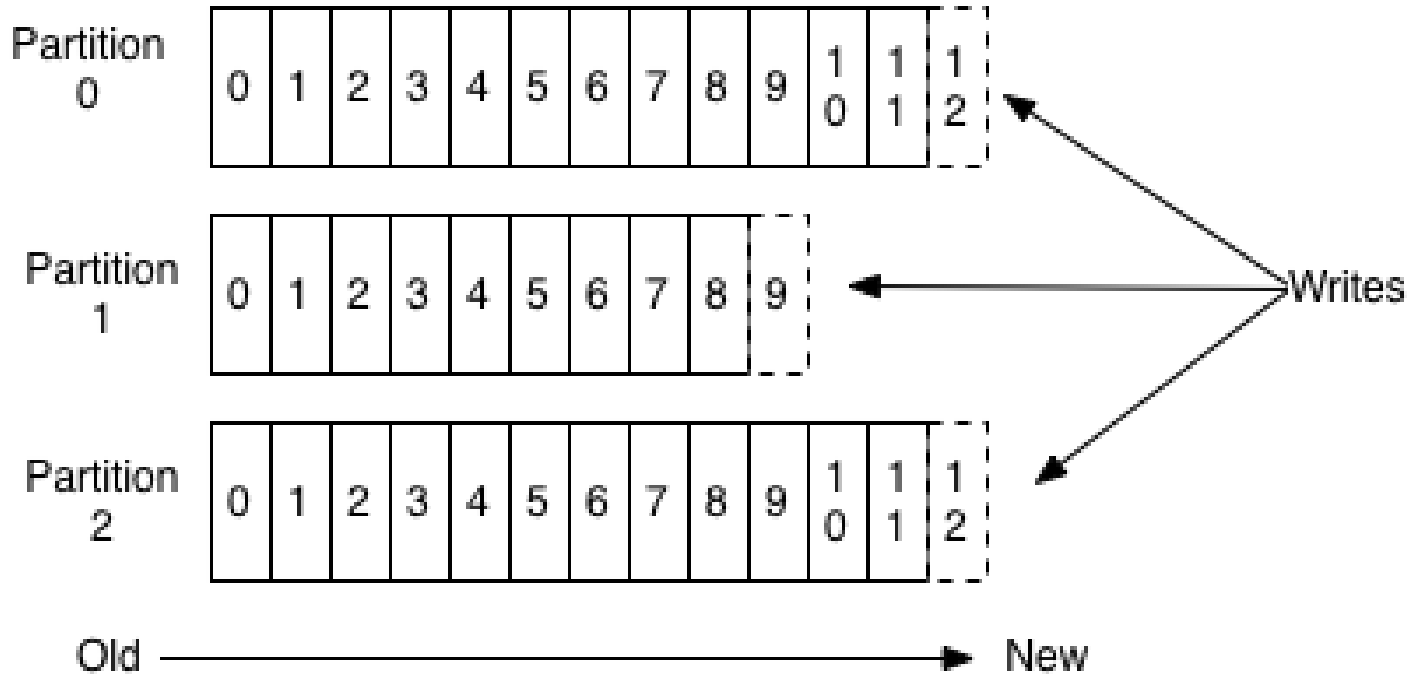
Topics



- A **Topic** has several **Partitions**
- **Partitions** of a **Topic** are distributed across **Brokers**

Topics and Logs

- Kafka store messages about a topic in a partition as an append only log.



Each partition is an ordered, numbered, immutable append only sequence of messages--- like a commit log.

Kafka Server Cluster Implementation

- Each partition is replicated across a configurable number of servers.

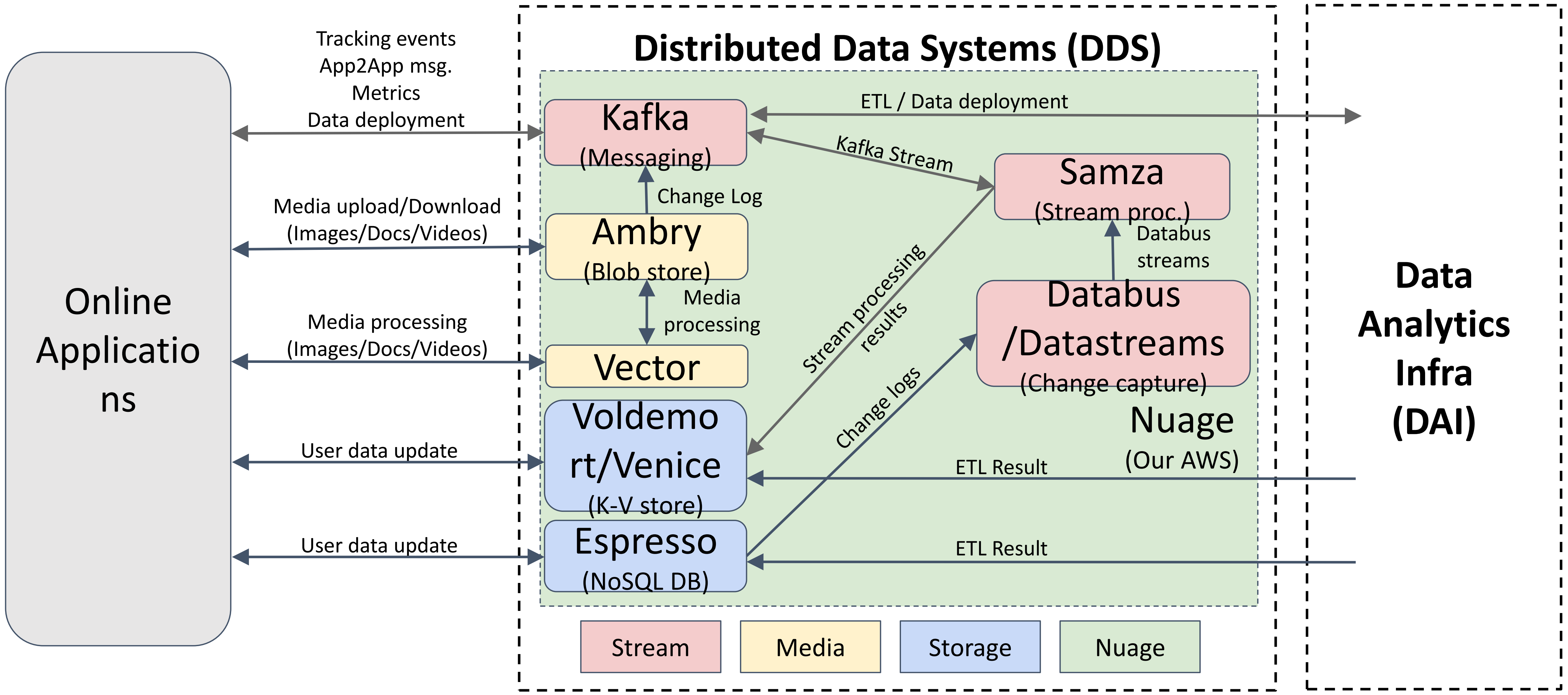
- Each partition has one “leader” server and 0 or more followers.
- A leader handles read and write requests

- A follower replicates the leader and acts as backup.

- Each server is a leader for some of its partitions and a follower for others to load balance

- Zookeeper is used to keep the servers consistent

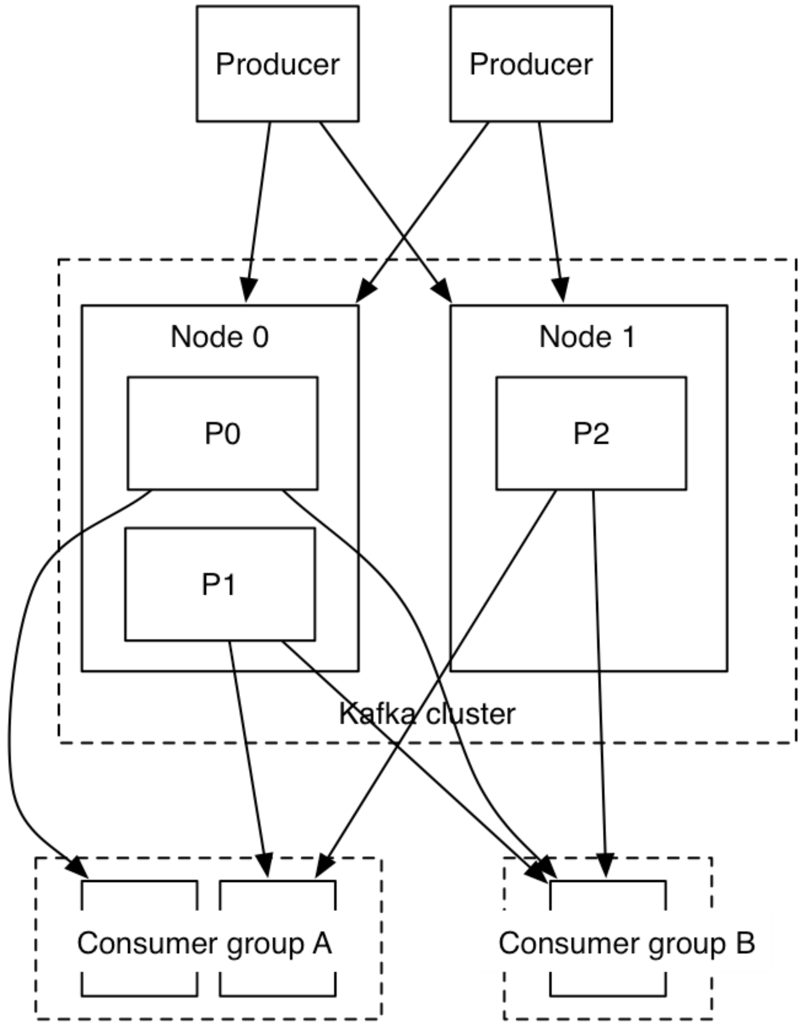
Kafka in a big picture (Linked In)



Producer in Kafka

- Send messages to Kafka Brokers
- Messages are sent to a Topic
    - Messages with same **Key** go to same partition (so they are in order)
    - Messages without a key go to a random partition (no order guarantee here)
    - Number of partitions changed? - Sorry…Same key might go to another partition...

Consumer in Kafka



- A consumer **can** belong to a **Consumer Group (CG)**
- Consumers in the same **CG**
  - Coordinate with each other to determine which consumer will consume from which partition
  - Share the **Consumer Offsets**

Offset

From Brokers' View

- The Index of a message in a log
- **Message Offset** does not change

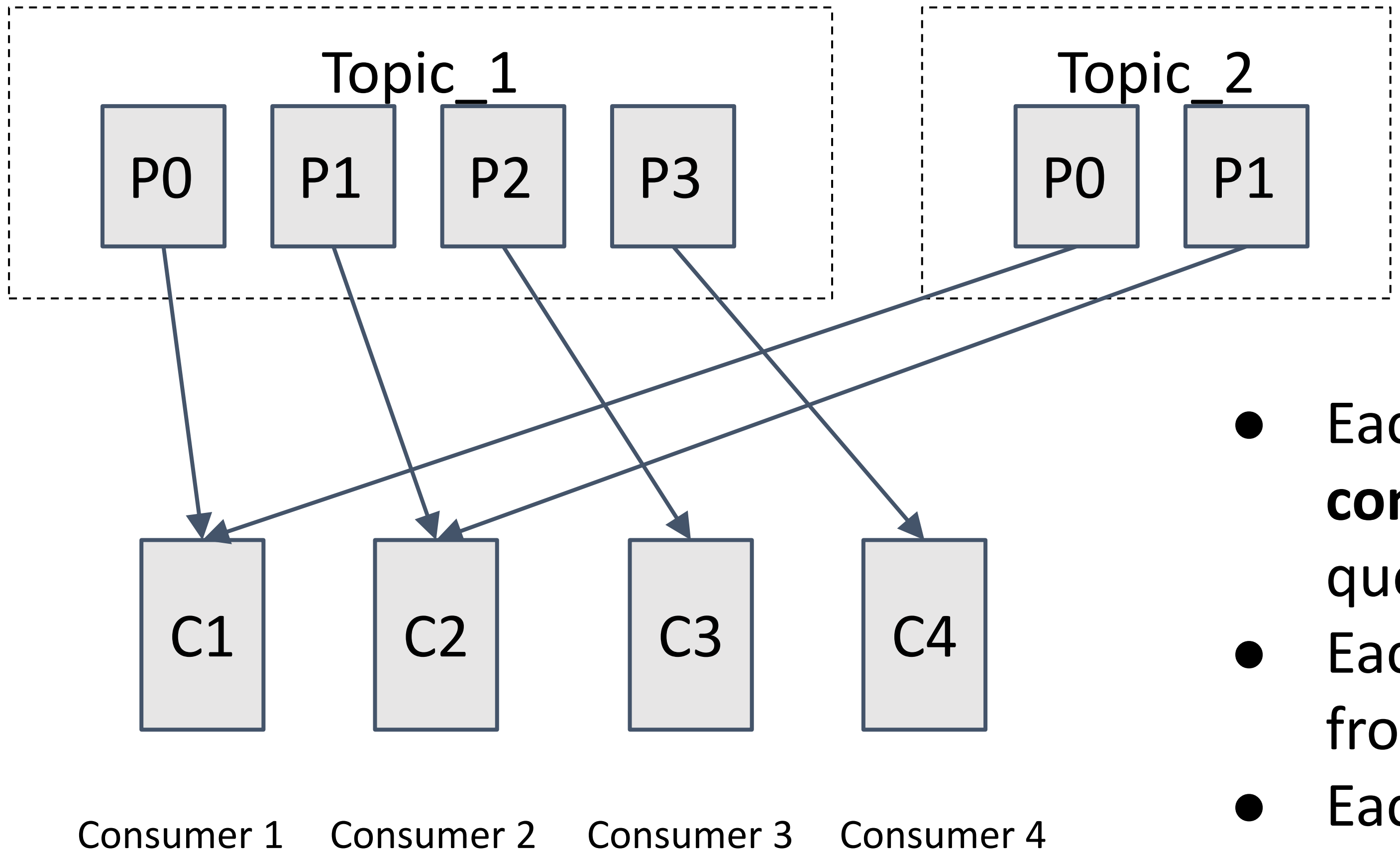
From Consumers' View

- **Consumer Offset**
- The position from where I am consuming
- Consumer Offset can change

More about Consumer Offsets

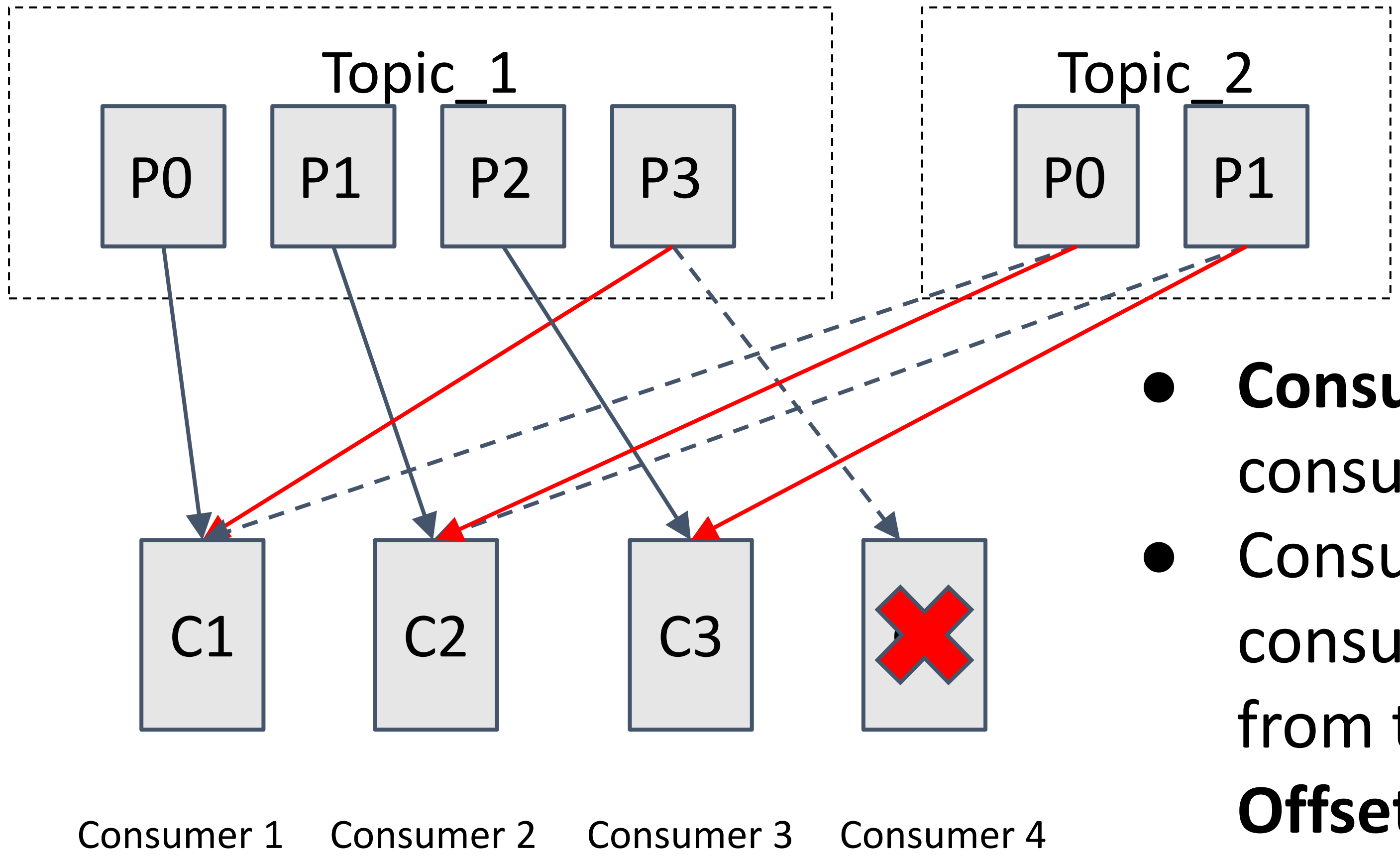
- Consumer Offsets are per
  Topic/Partition/ConsumerGroup
  (For a given group, look up the last consumed position in a topic/partition)
- Consumer Offsets can be **committed** as a checkpoint of consumption so it can be used when
  - Another consumer in the same CG takes over the partition
  - Resuming consumption later from committed offsets

Consumer Rebalance



- Each consumer can have several **consumer threads** (essentially one queue per thread)
- Each consumer thread can consume from multiple partitions
- Each partition will be consumed by **exactly one** consumer **in the entire group**

Consumer Rebalance



- **Consumer rebalance** occurs when consumer 4 is down
- Consumer 1, 2, 3 takes over consumer 4’s partitions and **resume** from the last committed **Consumer Offsets of the CG**
- **Transparent to user**