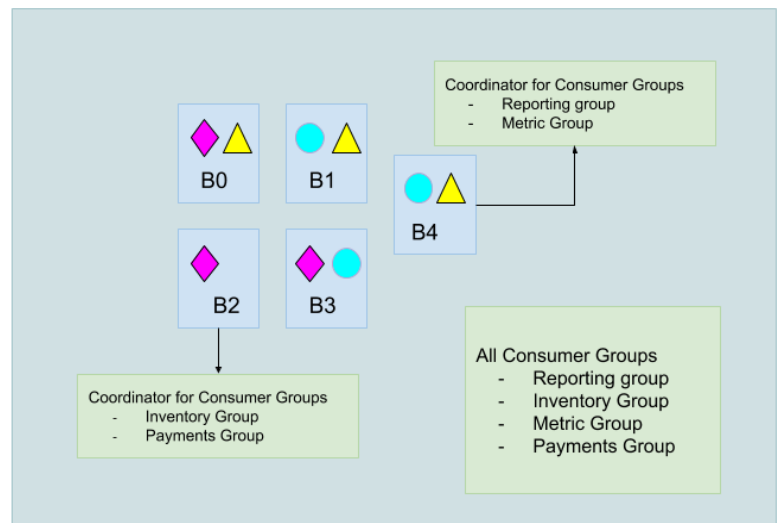
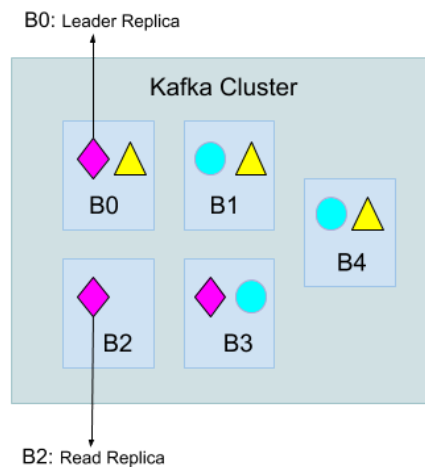


# Introduction to Consumer API

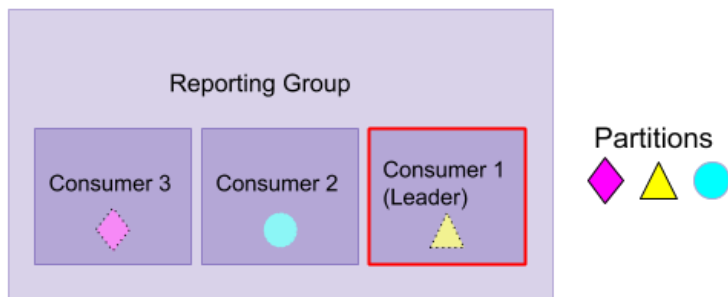
## Terminology

Leader replica/ Read replica

Coordinator



Group leader

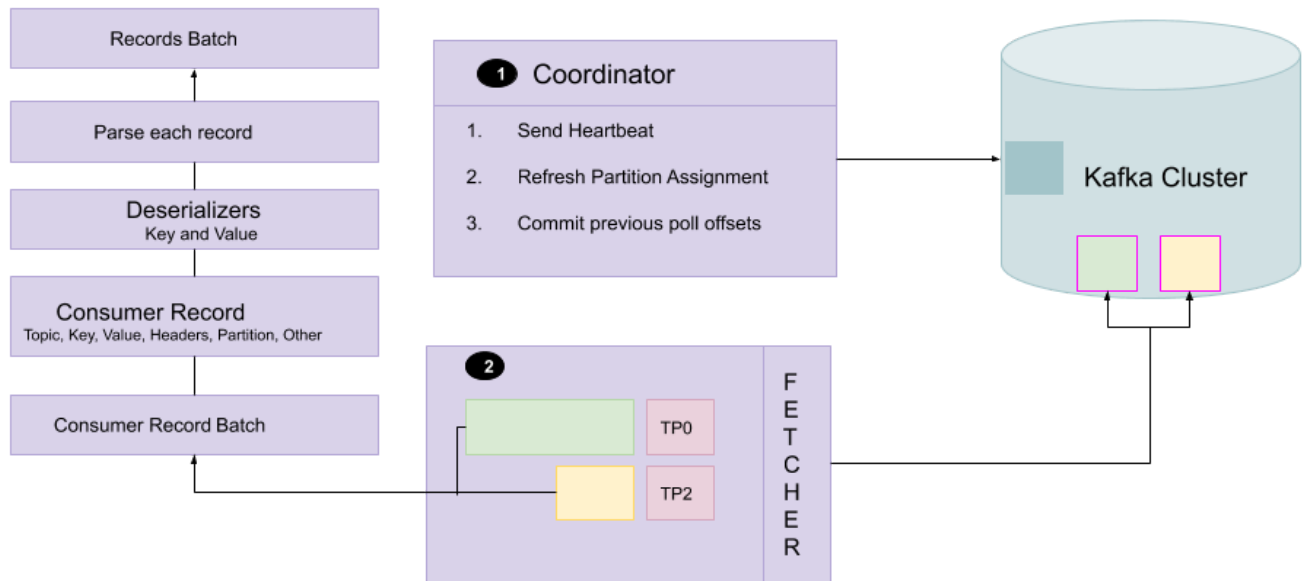


Leader replica/ Read replica: One of the replica brokers for partition is marked as Leader replica. Similarly for the Read replica.

Coordinator: One of the brokers is elected as the coordinator for a subset of the consumer groups

Group leader: One of the consumers in a consumer group is marked as Leader of the group.

## Consumer API



### Consumer Record

Payload to be produced

### Deserializers

Converts raw byte stream into key and value

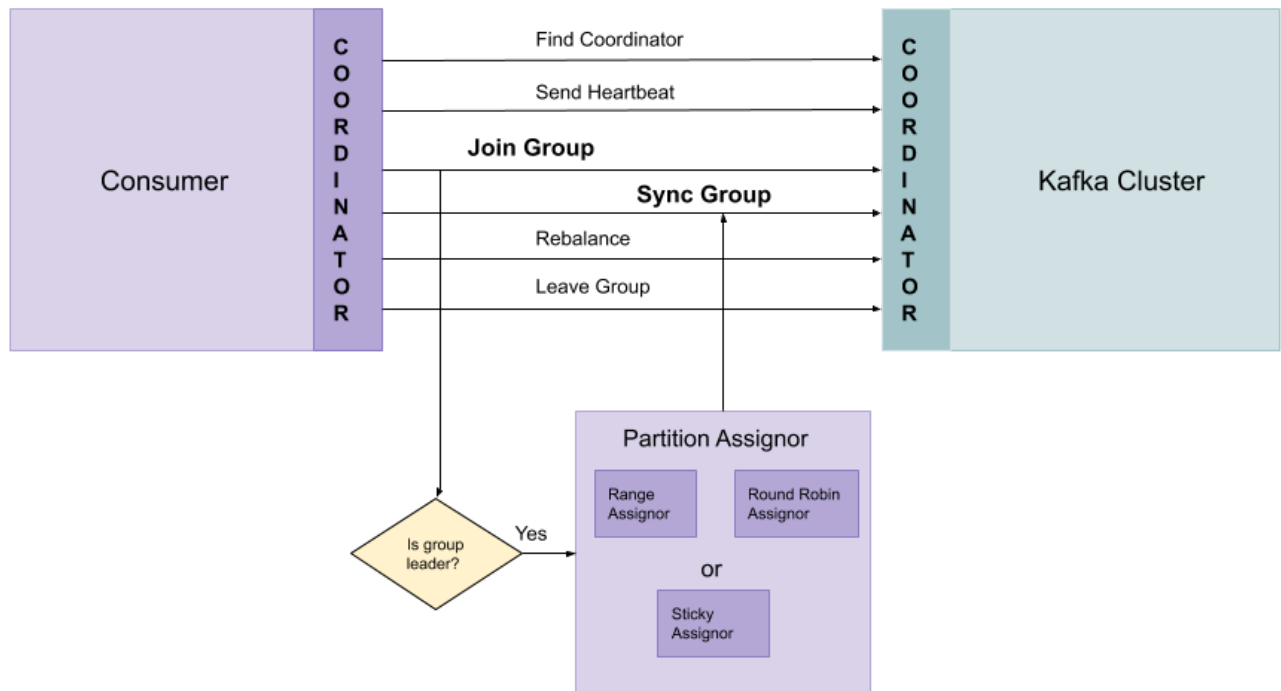
### Coordinator

Continuously poll Kafka side Coordinator for metadata updates

### Fetcher

Fetch data from partition's Leader replica / Read replica.

## Coordinator in detail



```
While ( Heartbeat ) {
    active group;
    continue;
}
While ( change in metadata version ) {
    JoinGroup / Rejoin Group
    if leader
        Sync group with new partition assignments
    if follower
        Wait for Sync group response for new partition assignments
    Rebalance; //Pause the consumption from revoked partitions or Start consuming from new Partitions
}
```

### Responsibilities

- Find Coordinator
- Send Heartbeat
- Join Group
- Sync Group
- Rebalance
- Leave Group

## Partition Assignor

When a consumer joins a group as a leader, it generates the partition assignments. Using one of the partition assignor strategy

## Exercise

Read about KafkaConsumer class in detail

- <https://kafka.apache.org/0100/javadoc/index.html?org/apache/kafka/clients/consumer/KafkaConsumer.html>

Read about partition assignor strategies

- Range Assignor
- Round Robin Assignor
- Sticky Assignor

Read about below classes

- KafkaConsumer
- ConsumerRecord
- Fetcher
- ConsumerCoordinator

## Hands-On

Set up the below project

<https://github.com/swathi-kurella/kafka-workshop-series.git>

Prerequisites:

Language: Java

Build: Gradle

Create a sample topic

```
bin/kafka-topics.sh --create --topic kafka-workshop-eg --partitions 3 --bootstrap-server localhost:9092 --replication-factor 1
```

Run the sample Consumer

Start SampleProducer class and provide desired messages to be produced in the input console

Start console producer to produce messages

```
bin/kafka-console-producer.sh --topic kafka-workshop-eg --broker-list localhost:9092
```

Observe console consumer console to view messages