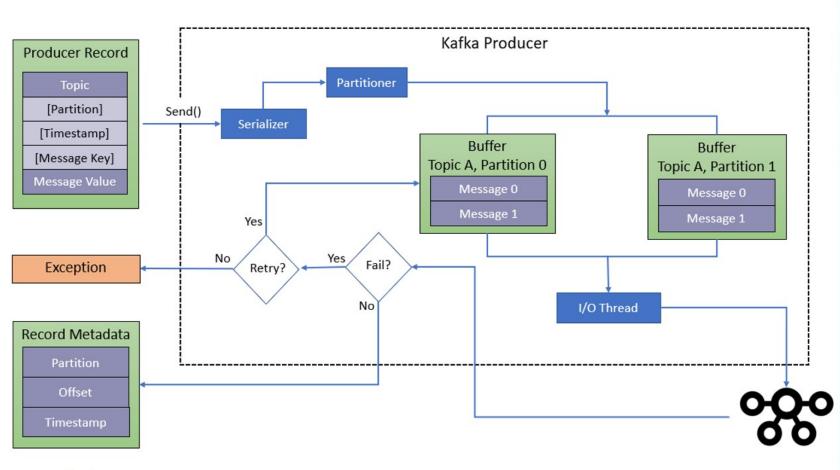# Kafka Producer
Final Note and References

Kafka Producer

1. Synchronous Send
2. Producer Callback
3. Custom Partitioner
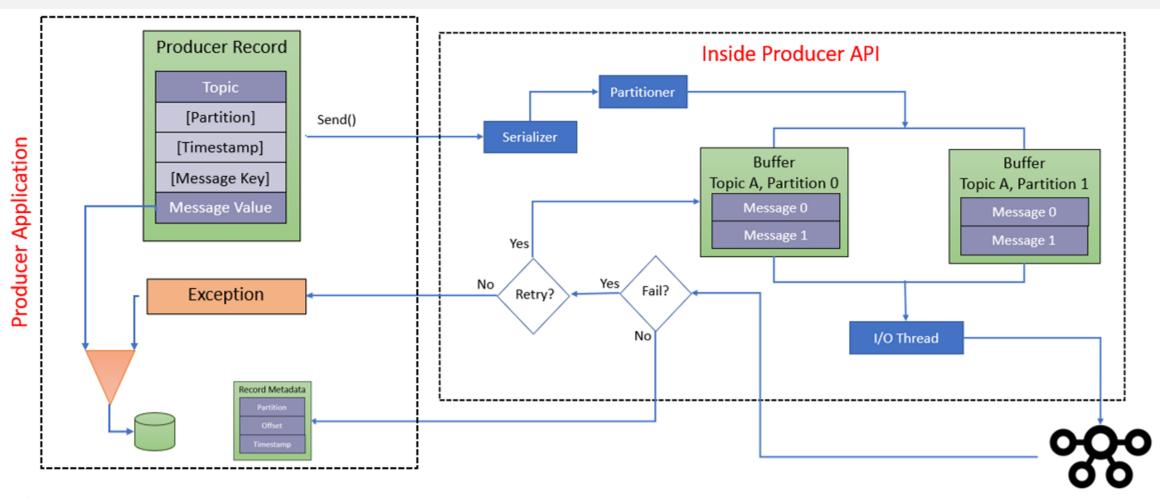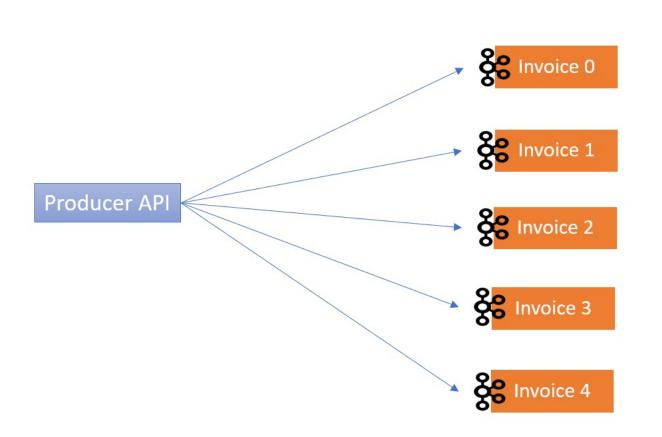4. Avro Serializer and Schema Registry

Handling Message Delivery Failure

Handling Message Delivery Failure

Handling Message Delivery Failure

```java
RecordMetadata metadata;
try (KafkaProducer<Integer, String> producer = new KafkaProducer<>(props)) {
    for (int i = 1; i <= AppConfigs.numEvents; i++) {
        Thread.sleep(1000);
        metadata = producer.send(new ProducerRecord<>(AppConfigs.topicName, i, "Simple Message-" + i)).get();
        logger.info("Message " + i + " persisted with offset " + metadata.offset()
            + " and timestamp on " + new Timestamp(metadata.timestamp()));
    }
} catch (Exception e) {
    logger.info("Can't send message - Received exception \n" + e.getMessage());
}
```

Handling Message Delivery Failure

```java
for (int j = 1; j <= numEvents; j++) {
    int i = j;
    producer.send(new ProducerRecord<>(topicName, i, "Simple Message-" + i),
            (recordMetadata, e) -> {
                if (e != null)
                    logger.error("Error sending message with key " + i + " Error - " + e.getMessage());
                else
                    logger.info("Message " + i + " persisted with offset " + recordMetadata.offset()
                            + " and timestamp on " + new Timestamp(recordMetadata.timestamp()));
            });
}
```

Handling Custom Partitioning

Producer API → Invoice 0
Producer API → Invoice 1
Producer API → Invoice 2
Producer API → Invoice 3
Producer API → Invoice 4
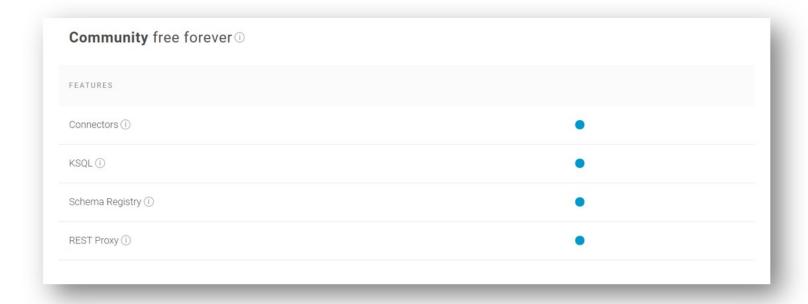
AVRO Serialization and Schema Registry

https://www.confluent.io/download/

Community free forever ⓘ

FEATURES

Connectors ⓘ   ●

KSQL ⓘ   ●

Schema Registry ⓘ   ●

REST Proxy ⓘ   ●

AVRO Serialization and Schema Registry

```
{     name: "John",
      age: 31,
      city: "New York"

}
```

```
{     name: "John",
      age: 31,
      city: "New York",
      phone" "XXXXXXXXXX"

}
```

AVRO Serialization and Schema Registry