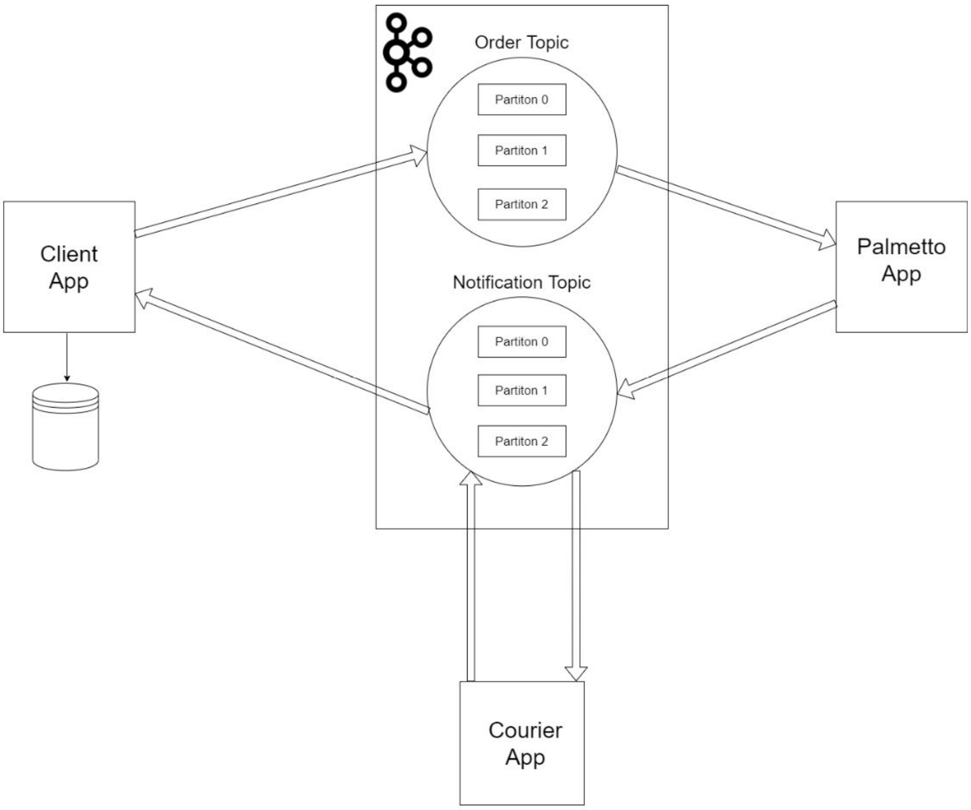
Apache Kafka

The task is to implement a system for creating online orders for pizzeria "Palmetto".



The architecture of the"Palmetto" online ordering platform

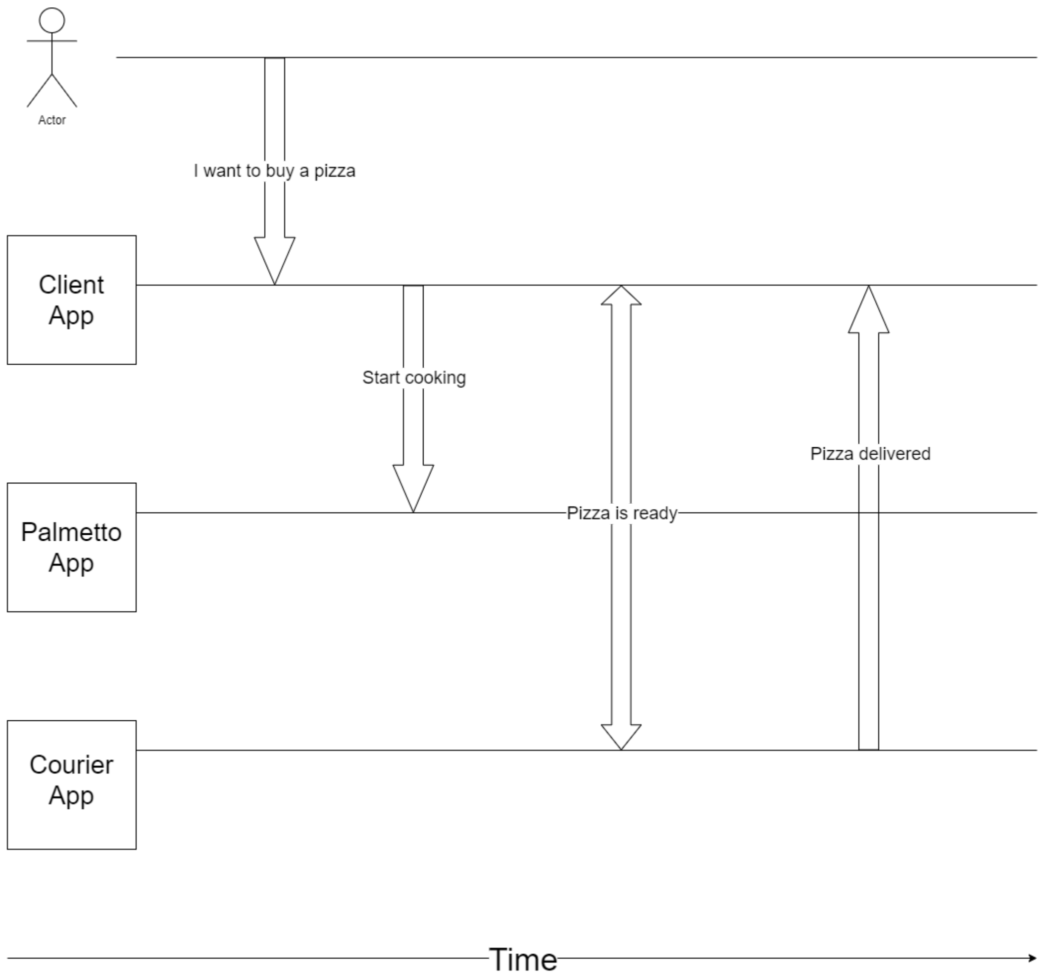
The online ordering system consists of the following components:

- There are 3 Spring Boot applications ("Client App", "Palmetto App", "Courier App"). "Client App" has a database for storing order information. You can use any preferred database;
- Apache Kafka is a message broker. It has 2 topics. The Order topic is used for sending events with customer orders to the pizzeria. The Notification topic is used for sending events with a new order status.

The Client App is a service for receiving and storing incoming orders from clients. It has a REST API with two endpoints. The first endpoint is used for receiving an order from a customer. The second endpoint is used for getting information about the order. The Client App is subscribed to the Notification topic for receiving messages with a new order status.

The Palmetto App is a pizzeria service for preparing customer orders. It is subscribed to the Order topic for receiving orders from the Client App and sends a new order status to the Notification topic.

The Courier App is a courier service that facilitates order delivery to customers. It is subscribed to the Notification topic for receiving orders from the Palmetto App and it sends a new order status to the Notification topic.



Process diagram

Non-functional requirements:

- each service has 3 consumers, one for each partition,
- use a correlation ID to distinguish orders,
- use “At least once” delivery semantic for each consumer and producer with Acks=1,
- each topic should consist of 3 partitions.

****Extra mile.***

Create integration tests using Kafka test containers.