Assignment 1

(Design and Development of Cloud Applications)

Phagun Jain, R171219022, CSE DevOps

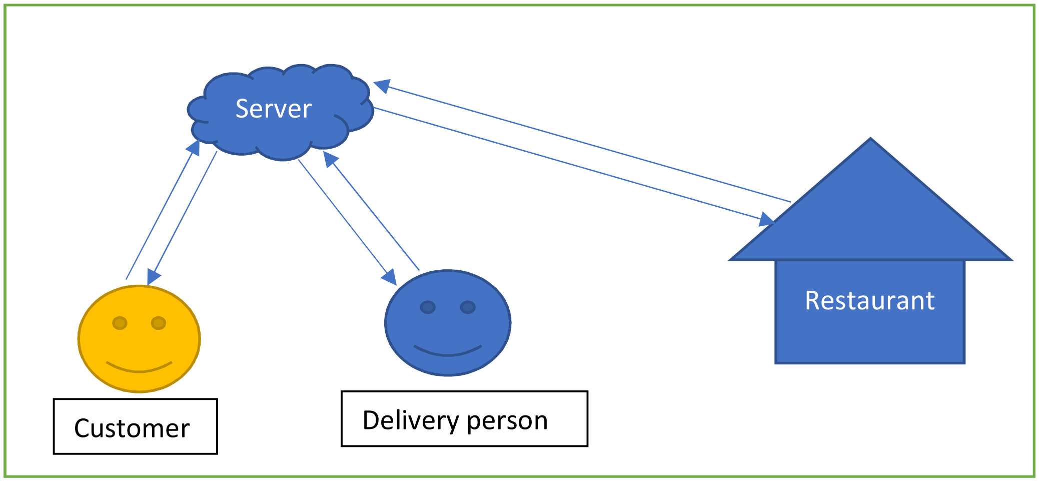

Fig.1 Blueprint of Request flow

A use-case diagram depicts the functional requirements or the activities that an actor can perform with the system. The main components involved in any use-case diagrams are Actors, Usecase, System, and Relationships. Actors are the person or automated process that interacts with the system; Usecase is that particular activity that the actor can perform with the system; System is the logical boundary within which all the use-case are enclosed; Relationships depicts the semantics between two use case or a use-case and an actor.

In the case of the food delivery app, various requirements from the system can be user management for authentication and authorization to define who can do what and this can be achieved with a Login and Signup facility; a basic requirement in any food delivery app is the search facility for the given keyword; an order placing system; a payment management mechanism; and last but not the least a push notification system to provide the user with a status update of their order, all these requirements are only customer-specific however some hidden requirements are used by the secondary actors on the system such as delivery guidance system for the delivery person, and an event-based detection system to update the status.

A use case diagram for a food delivery system specifying the functional requirement or the use cases expected from the application

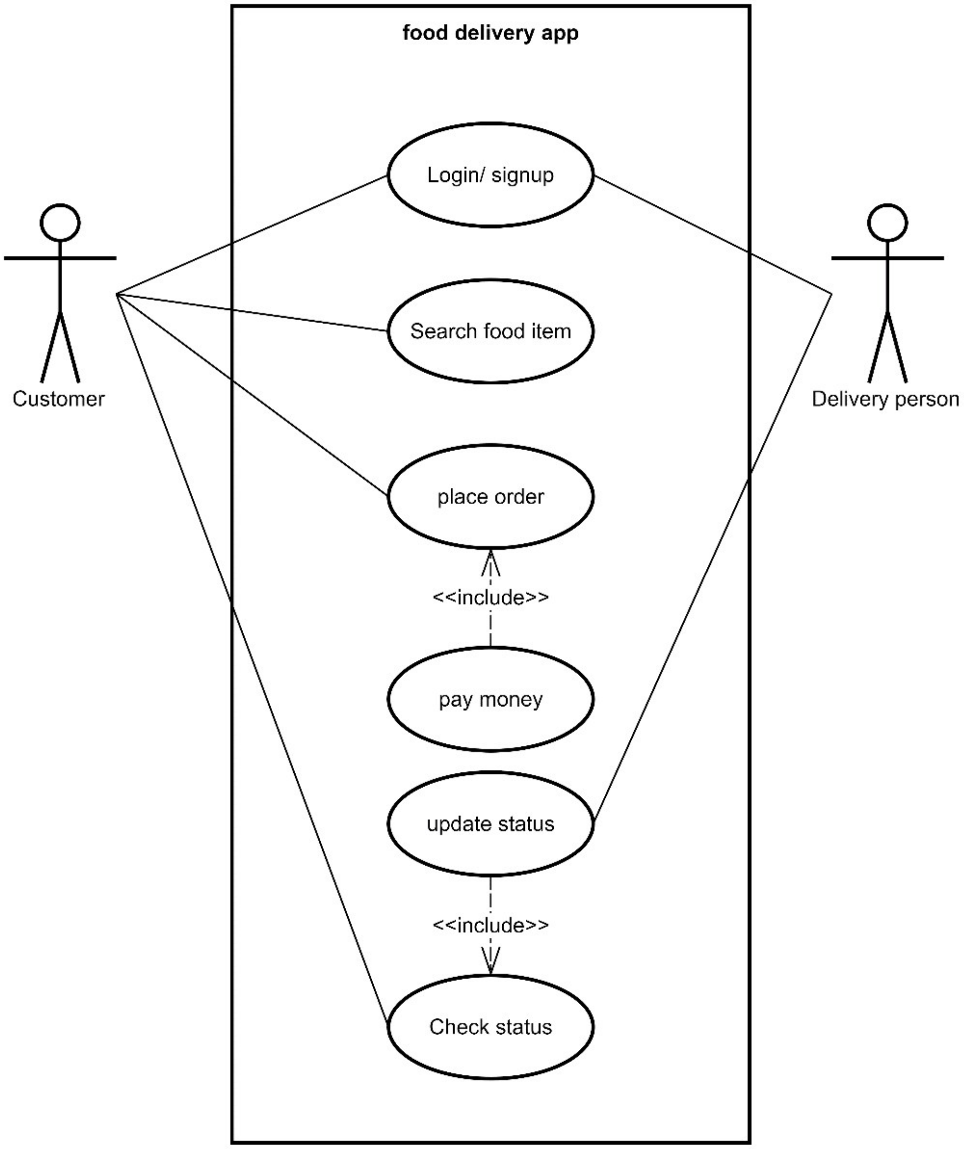

Fig.2 Use casediagram for the food delivery app

State machine diagrams are made to describe the significant behaviors of the objects involved in the system. The behavior is described by various states, it captures the dynamic aspect of the system, it also models the interactions that take place between the outer and the system objects.

State diagram for the food delivery system

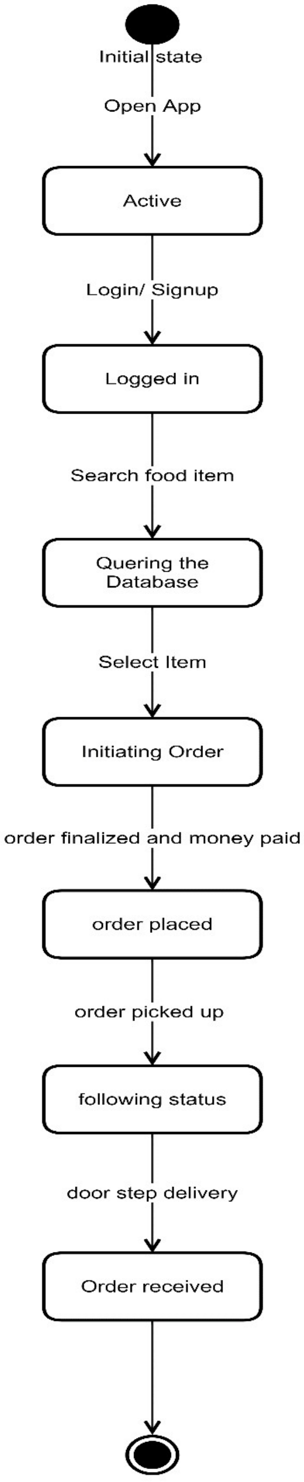

Fig.3 State machine diagram for the food delivery app

The two main components of greater interest in the architecture of our food delivery app i.e., the order sync module; and the push notification system. The modules work on the event-driven approach. Whenever the event and state transition takes place there is some corresponding action performed for that specific event like for order received event the corresponding action would be publishing status to the restaurant and so on.

The framework most widely used to achieve the streaming of data smoothly is apache Kafka, it supports the data streaming in a hybrid approach i.e. in the queue method and pub-sub method; it allows to visualize the system as a sequence of events and those events are immutable and therefore impossible to change. The main components involved in the Kafka ecosystem are Kafka cluster, producer, consumer, topic, partition.

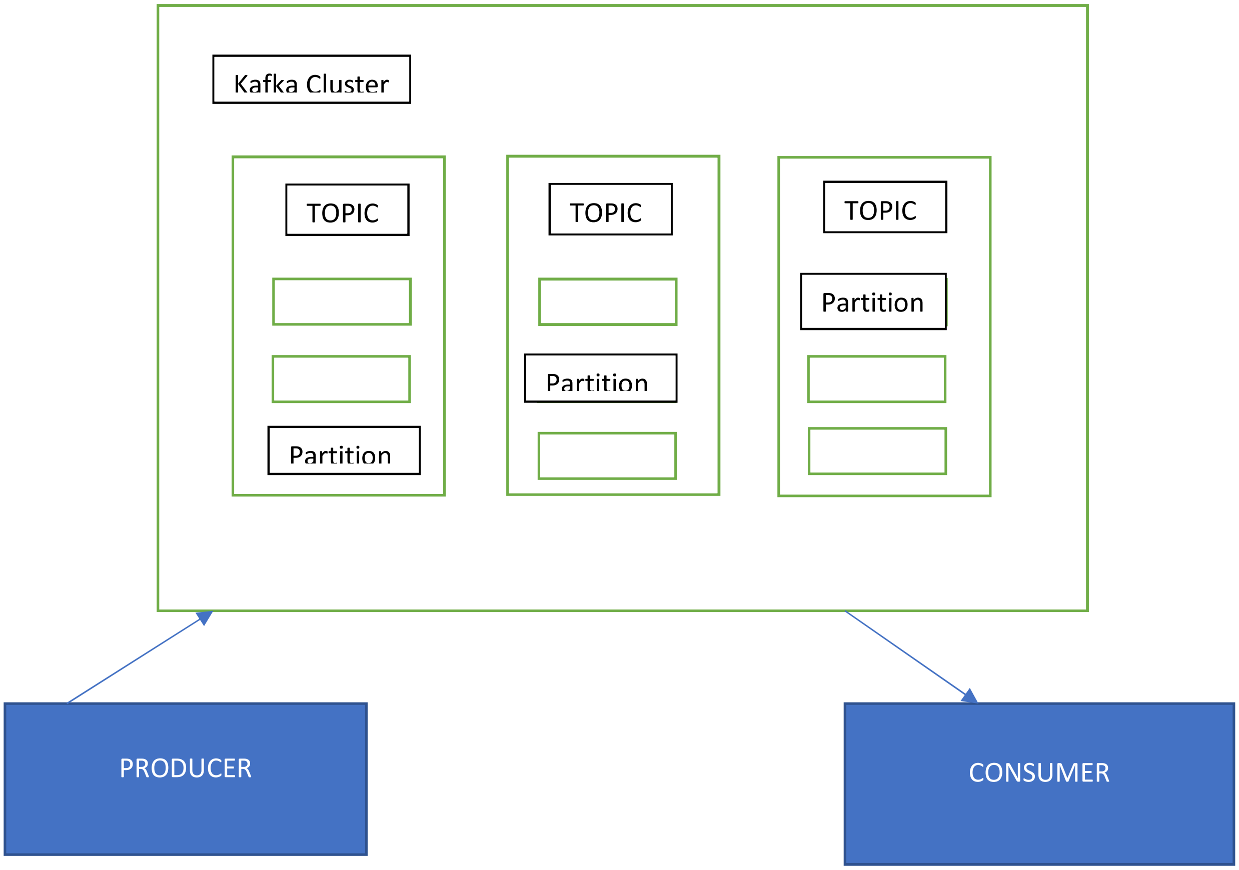

Fig.4 Typical Kafka framework

REFERENCES

[1] Push Notification Services: Google and Apple http://csce.uark.edu/~cwt/COURSES/2014-01--CSCE-4543--SW-ARCH/03--CHAPTERS/Chapter%2004--Push%20Notification%20Services--Cleaver.pdf

[2] Intro to Apache Kafka: How Kafka Works; https://www.confluent.io/blog/apache-kafka-intro-how-kafka-works/

[3] Architecture and Design Principles Behind the Swiggy's Delivery Partners app; https://bytes.swiggy.com/architecture-and-design-principles-behind-the-swiggys-delivery-partners-app-4db1d87a048a