virtusa

# INVITO: Organize, invite and celebrate.

By,

N. Philip Niron

# 1. Abstract of the Project

Hosting an event is not that easy thing to do. The event host should consider the availability of the venue such as hotels and restaurants, conducting time, availability of invitees, arranging the food and beverages, quality of the venue, cost, etc. Mainly the venue, time, and facilities are the things to consider most. The availability of the venue in a particular time frame is the most important thing to count on. Even after doing all the arrangements, the host should care about the invitees to the event. The host may not know all the residence addresses of all the invitees. Sometimes invitees may not participate in the event so the arrangement costs and foods and beverages for them will be in vain. Invitees also face issues with their available time, finding the location of the venue, etc. So, the proposed system is capable of handling the above-mentioned problems and giving guided solutions to the users.

# 2. Description of the artifact

The proposed system is based on the microservice architecture. Each services will be deployed on top of the docker containers. Backend of the application will be developed using Spring boot, and for the front end, the developer will use angular and android.

## 2.1 Aim

- To implement a system to handle the event organizing and benefit both the host and invitees using microservice architecture.

## 2.2 Objectives

1. To implement microservices for users, events, venues, invitations, booking, catering, locations, Utilities and extras.
2. To develop backends of each services using Spring boot framework.
3. To implement invitation using Kafka message broker framework.
4. To implement service discovery using Eureka.
5. To deploy each services in dockers.
6. To provide good UI/UX to the user.

## 2.3 System functionalities

- Provide information about the availability of the venue in the allocated time.
- Provide ratings for every venue, so that the user can easily pick the highly rated venue.
- Invite the invitees.
- Invitees can accept or reject the invitation.
- Invitees can navigate to the location using location service.
- User can analyze the budget according to the response of the invitees.
- User can host multiple events in each venues on the same time.
- Invitees will get customizable alert reminders to the event.
- Host can select the buffets to be served in the event.
- Additional utilities like DJs, outdoor huts can be ordered by the host.

## 2.4 Benefits

1. Provide easiness to the host to check the availability of venues available in the time slot.
2. Easy to plan the foods and beverages according to the response of the invitees.
3. Invitees can accept or reject the invitation.
4. Host can pick the best spot according to the user ratings.
5. Wastage of the food and beverages can be reduced because of the prior response of the invitees.
6. Invitees can easily navigate to the location using location service.
7. Budget can be easily analyzed and planned.
8. Venues can get their events easily and make a good outcome.

## 2.5 Intellectual challenges

- To implement the microservice architecture preplanning is mandatory.
- Complex testing over a distributed environment.
- Interface control is critical to implement.
- Need to have a proper idea to handle it.
- Operations configuration is more complex.

# 3. Implementation

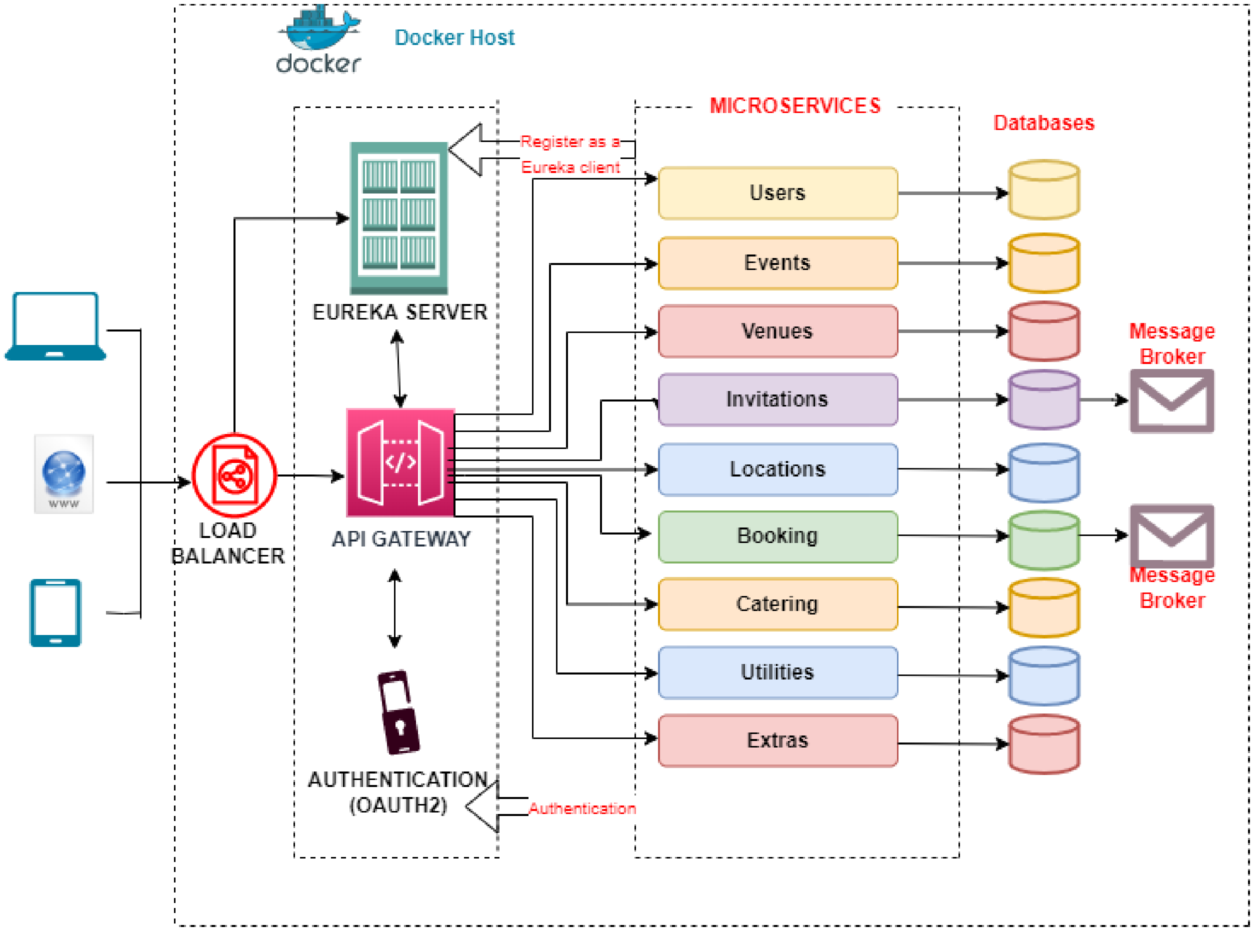

*Figure 1- Architecture Diagram*

The implementation of the proposed system is based on the microservices architecture. So, the entire proposed system is divided into nine microservices. Each microservices will separately have their databases. Invitation service and booking service require the help of the message brokers to perform their tasks. To validate the user authenticity OAuth2 will be used. Service discovery will be performed using Eureka server in the proposed system. Unit testing, Integration testing, load testing, stress testing, and longevity testing will be performed to ensure the quality of the system. System will be available for the end user after performing all the testing. Blue green deployment strategy will be implemented in case of updating the versions of the system. Finally, problems currently faced by the users will be resolved after the release of this proposed system.