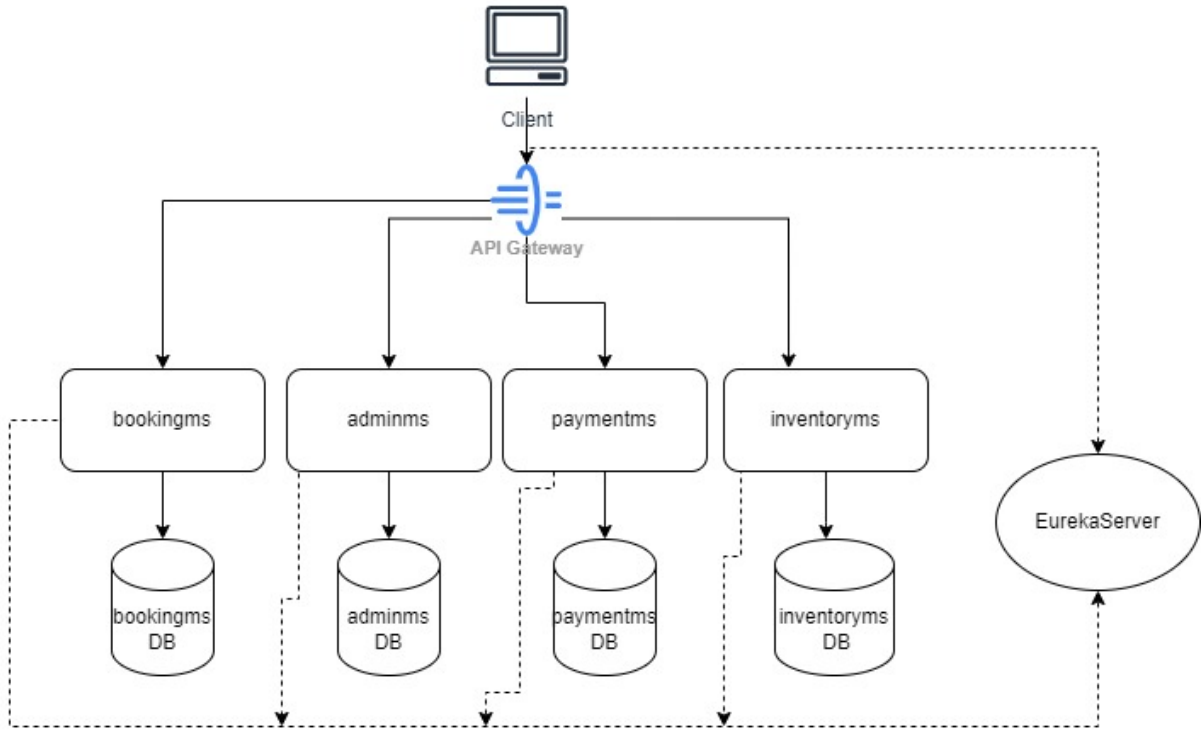
Assignment – online Bus reserve Application

Below is the architecture diagram for this microservice application



1. First run Docker RabbitMQ brocker via below command

latest RabbitMQ 3.13

docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3.13-management

2. Run EurekaServer. It will run on Port 8761.
3. Run bookingms(Port 9081),adminms(Port 9084) , inventoryms (port 9083),paymentms(port 9082), APIgateway (Port 8080)

4. Run http://localhost:8080/bookings/book with below data

Input :

```
{
 "busNumber": 1234,
 "bookingDate": "20/08/2024",
 "source": "kolkata",
 "destination": "durgapur",
 "numberOfSeats": 2
}
```

Output:


```
{
   "bookingNumber": 1,
   "busNumber": 1234,
   "bookingDate": "2024-08-20T03:01:07.956+00:00",
   "source": "kolkata",
   "destination": "durgapur",
   "numberOfSeats": 2,
   "status": "PENDING"
}
```


Booking service will call InventoryMS through method checkSeatAvailability to check the available seats . In case Seats are available , booking service will create a pending booking and send the event “bookingEvent” to paymentms. Event processing is happening via RabbitMQ. paymentMS will do do internal processing and on successful processing ,it will send an “paymentEvent” to inventoryMS . inventoryMS will deduct the number of available seats and send the “inventoryEvent” to Bookingms .BookingMS will mark the booking as confirmed once this event is received from inventoryms.

5. Call below URL , where 1 is the booking number , to get the booking status
   http://localhost:8080/bookings/1

output :


```
{
   "bookingNumber": 1,
   "busNumber": 1234,
   "bookingDate": "2024-08-20T03:01:07.956+00:00",
   "source": "kolkata",
   "destination": "durgapur",
   "numberOfSeats": 2,
   "status": "CONFIRMED"
}
```


6. All project has dockerfile present .So these microservices can be deployed through docker container .