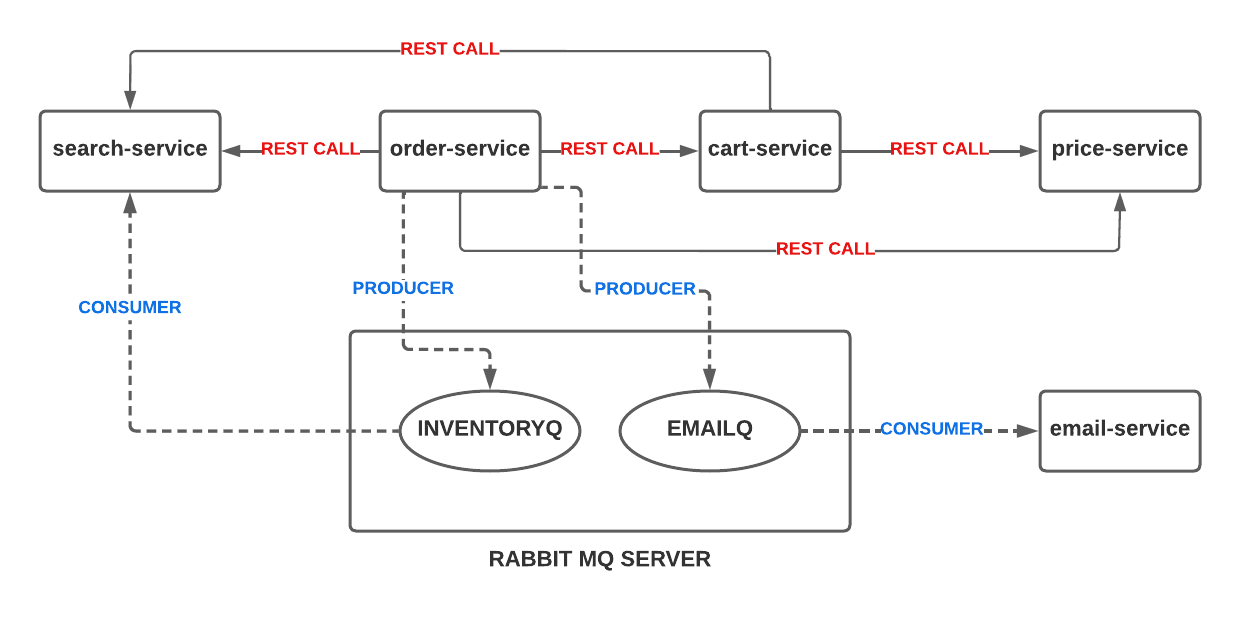
**CASE STUDY ON ECART APPLICATION USING MICROSERVICES**

**Develop Rest API Shopping cart microservice application for the following requirements**

* View hottest selling and newest listings on frontpage.
* Search listings on any category or a particular category of your choosing.
* Browse detailed product pages.
* Add products to cart, view and update your cart.
* Create and login account.
* Buy products and Email purchased products information
* Seamless transition between guest cart and logged in member cart.
* View your orders and their statuses.



Create services with following port numbers

1. price-service : 8081
2. search-service: 8082
3. cart-service : 8083
4. order-service : 8084
5. email-service : 8085
6. config-server : 8888
7. eureka server : 8761
8. api-gateway : 8090
9. user-service. : 8086 (optional)
10. payment-service: 8087 (optional)

Must use following technologies

**microservice communications**

* 1. RestClient / RestTemplate
  2. RabbitMQ

**Microservice cloud**

* 1. Config server
  2. Config client
  3. Feign
  4. Load balancer
  5. Eureka service discovery
  6. Routers / API gateway
  7. Fall back operations
  8. JWT (optional)
  9. Swagger API (optional)

**Software**

1. Java 8 and above (17)
2. Spring tool suite or Eclipse
3. Mysql / Mongo db / Redis
4. Rabbit MQ server

From above diagram,

1. Search service able to search for any products, and filters by category, brands and price
2. Cart service able to make a REST CALL to the search service to add product in the cart for a specified user
3. Order service able to make REST CALL to the cart service to order product or products available in the cart for a specified user
4. If order is placed successfully, then Order service should able to send order information to the EMAILQ Queue using Rabbit MQ server.
5. Order service should able to send ordered product quantity to the INVENTORYQ Queue using rabbit MQ server
6. Search service able to read INVENTORYQ data from Rabbit MQ server and should update the product inventory
7. Email service should able to read EMAILQ queue data from Rabbit MQ server and send email
8. If user service is implemented, user service is responsible for user profile update, password change , forget password options
9. If payment service is implemented, payment service is responsible for reading latest price of a product which is going to be purchased, and send this price to the selected payment method / bank
10. Implement load balancing using Eureka service discovery or API gate way
11. Configuration information like database or any service configuration should be provided in a centralized version controlling system like GIT hub using config server. And all services should able to read configuration information from GIT hub using config client
12. Create a API gateway service to route all the services from a single port number and implement load balancer using annotations.
13. Implement fall back operations using any circuit-breaker technique

Database tables for this application can be accessed from the following URL. If you want to create and implement your own database, you can implement your own db

<https://github.com/praveencloudlab/adm-microservices-b1/blob/main/db-dump/ecartv1.sql>