High Level Design: -

Auxiliary services to check if the order is still pending after time as elapsed, if yes then mark them failed. Notify the user by email.

Third Party payment service.

Paytm, credit card, net banking etc.

Calling third party payment API with a callback method to update success or failure to the SQL table.



API Servers clusters with Leader Election for high availability.

Messaging topis Apache kafka, Google pub/sub



API servers in one region getting request from load balancer for order processing API.

Caching layer for payment rules for fast latency



Key Value Store

Storing business rules for payments

Zookeeper, Etcd etc.

Order Service



SQL Database

storing order information

Postgres, Oracle, DB2 etc.



Order Service



Load Balancers clusters specific to different regions in the world, like India, US, Europe, etc. They will use round robin strategy .

DNS Load Balancers clusters sending request as per the location

Users around the globe calling order service application API through various clients like mobile app. Etc.



Requirement Gathering: -

Assumptions for order processing for large scale company: -

1. Highly Available, fast latency
2. Global presence
3. Millions of users
4. 10 orders/second
5. Skipped auxiliary services like authorization and authentication.

APIs: -

1. processOrder(String customerID, String orderID, List<Products>,Enum paymentType,Enum paymentInfo)
2. callOrderService(String customerID, String orderID,List<Products>,Enum paymentType,Enum paymentInfo)
3. sendMessagesThroughTopic(String orderID,String customerID)
4. callThirdPartyPayment(String customerID,String orderID,Function callback)
5. callbackFunction(String customerID,String orderID,Boolean success,String reason)

SQL DB: -

1. Orders
2. Order\_ID
3. Customer\_ID
4. paymentType(paytm,creditcard etc)
5. paymentInfo(payment for subscription, or book etc.)
6. success
7. reason
8. createdAt

mapping each order to a particular row

1. Prodcucts
2. Order\_ID
3. Product\_ID

For mapping multiple products to a particular order.