Low Level Design.

1. Architecture

Layered Architecture based on Spring Framework.

* Presentation Layer[ Application Controllers]
* Business Layer [Service I/F and ServiceImpl]
* Persistance Layer [ Repository JPA]
* Database layer[ DB]

1. Modules/ Components

Inventory Service

Order Service

These two services needed/ figured out based on Use cases.

Pattern style: Bounded Context Pattern

1. Microservice communication:

Use HTTP RestFul API for Internal Microservice communication.

1. Microservice Discovery using Eureka Service Discovery:

Use api gw Routing pattern for Service Discovery and Routing.

1. Functional Requirement
2. List Stock
3. Add/Modify Delete inventory
4. Add/Modify delete supplier of inventory
5. Place order
6. Cancel Order
7. Automatic order initiation based on Threshold qty and Supplier type (local/international)
8. Non-Functional Requirement
9. High Scalability
10. High Availability
11. Data Isolation
12. Independently deployable
13. Technology agnostic
14. Observability
15. Monitoring
16. Patterns and Principles
17. Database per service pattern
18. Decompose service by scalability
19. Microservice Decomposition pattern:

-Context pattern

1. Architecture

Microservice Atchitecture with Dockerized containers.

1. Architecture Diagram

Microservice Atchitecture with Dockerized containers.

Microservices

Publisher-Consumer modes

RabbitMQ

Inventory

API Gw

Web Application

Android Application

SQLDB

Order

SQLDB

Publisher-Consumer modes

Functional Flow Explanation:

Client Side : Android or Web Application used by Owner to Update Parts/ Supplier. It connects over HTTPS to Server via GW

Serverside:

Inventory service and Order service are two microservices can be reached by API GW over RESTFUl APIs.

Order and Inventory Services communication:

1. Phase-1 Design: uses Cron job scheduled at every 1 hr for Local Supplier based items and Scheduled at 12am for International supplier
2. Phase-1 Design : After every successful order , Inventory is updated by Order Microservice using REST calls, After cancel order the inventory is again updated by Order service to Inventory Service.

Phase-2 Design: Asynchronous Inventory Update Or Order placement should be done by using RabbitMQ as enhancement.