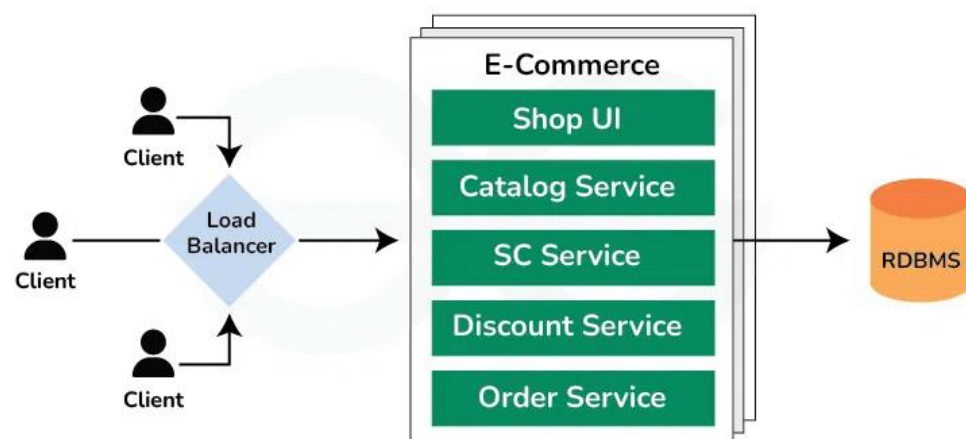


Monolithic Architecture Vs Microservices based Architecture

Monolithic Architecture:

- It has single code base with multiple modules.
- Working: Request-> Presentation layer (Controller)-> services (business logic)-> pojo_class/dao_class, it can communicate Database. (single Database)
- We can create Single jar file for one container container+ different modules which contains service_class +pojo_class)
- Disadvantages: difficult to manage, for single change redeployment of whole application needed, Difficult to adapt new technology for single functionality, single bug may down whole application
- Advantages: simple to develop. Simple to deploy because of one jar file.
Problem of network latency are relatively less (all modules run on same server, so network delay is less)



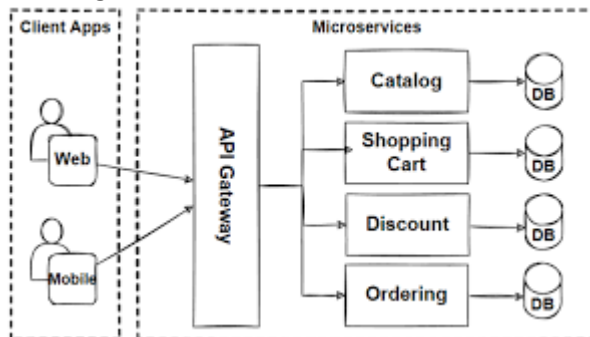
Monolithic Architecture



Microservices Based Architecture:

- Microservices are the small services that work together.
- Loose coupling
- These smaller services communicating with each other directly using light weight protocols like HTTP.
- Different microservices can have different databases and microservices can communicate each other

- Different code base
- Advantages: it is possible to change or upgrade each service individually rather than upgrading in the entire app, one service may down without impacting to others. Easily use different technology for building different microservices. Less dependency. Loosely coupled.
- Working of microservice architecture:



- user client (request using API port number)-> Entry point-API gateway (single end point) ->microservices
- API using spring cloud gateway
- Hystrix dashboard: work on fault tolerance, working on circuit breaker design pattern
- Eureka services discovery: service registry: we can register all microservices on these server (Eureka client: register microservices) and from here only discovery will start.
- By using Config server, we can access global configuration from GitHub (can store configuration on these server)
- API gateway: gateway dependency & eureka discovery client