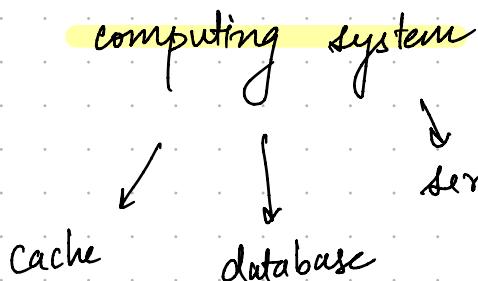


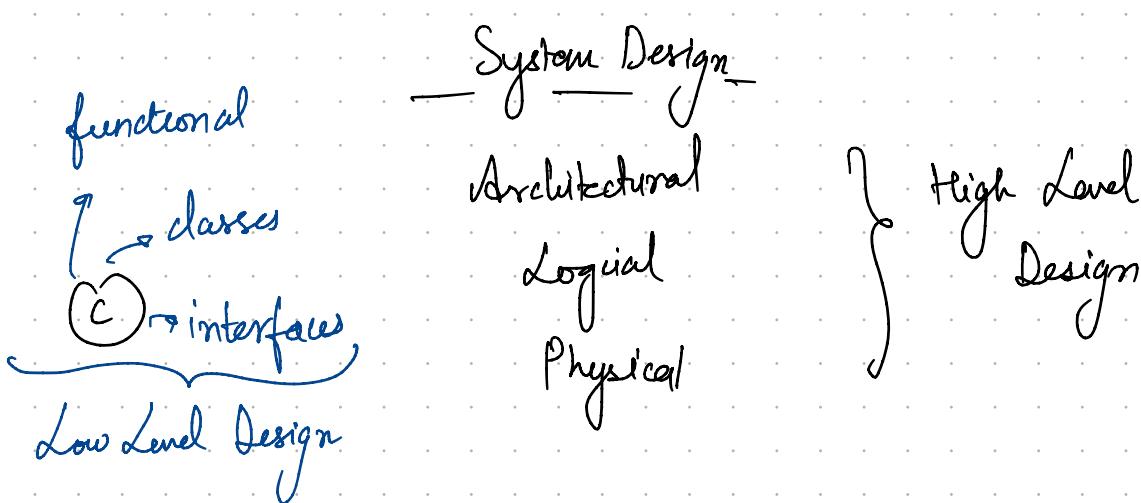
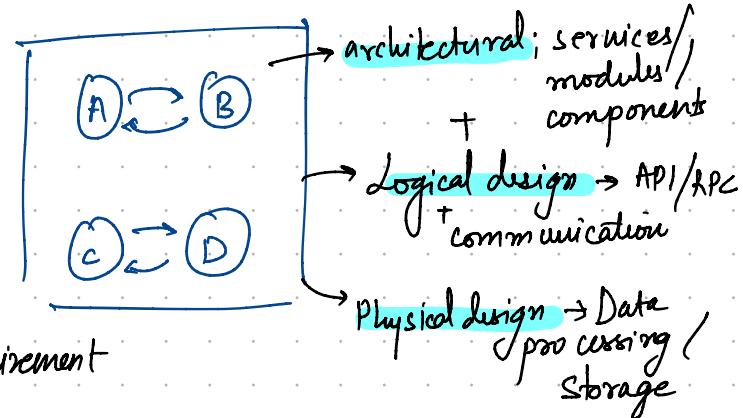
Introduction to System Design



- Design → process of understanding user requirement
 - design the components; make them communicate

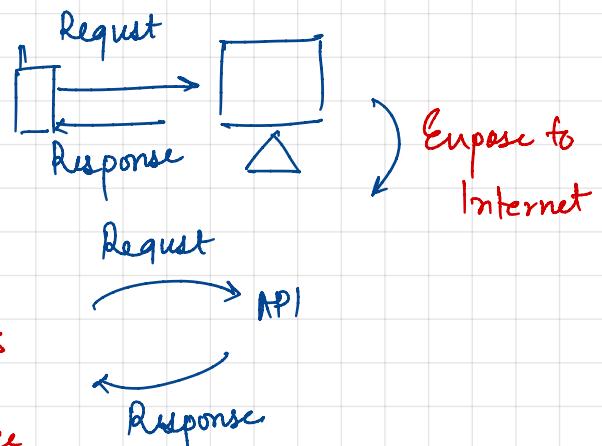
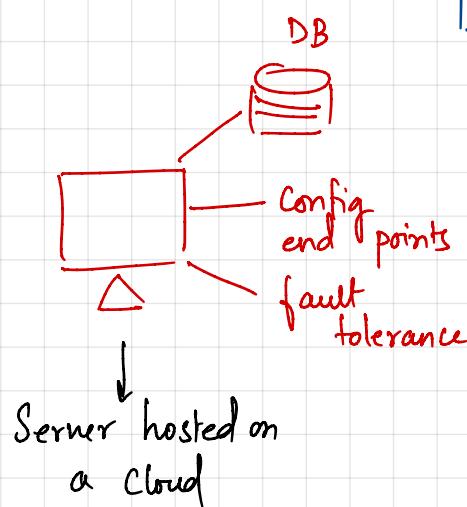
Process of defining building blocks

- components & integration
- communication
- Data functions for user requirement



System Design components

Horizontal and Vertical Scaling



SCALABILITY → process more requests

vertical scaling
Buy bigger machine

- * → N/A
- Single point failure
- * → Interprocess communication FAST
- * → Data consistency
- hardware limitations

horizontal scaling
buy more machines

- load balancer needed
- * → resilient
- Network calls (Remote procedure calls)
SLOW
- Data inconsistency
- * → Scales well as users increases

Distributed Systems

↓ HIGH LEVEL DESIGN

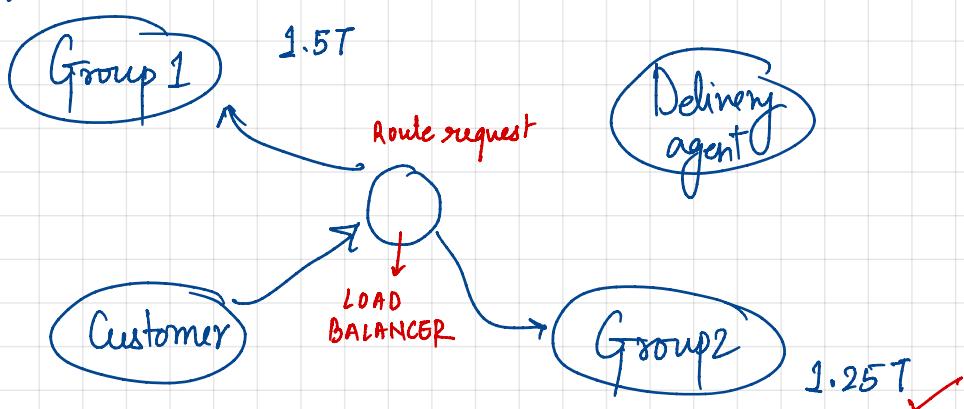
- ① Vertical scaling technique → optimize processes & increase throughput using the same resource
- ② Preparing beforehand during non-peak day
- ③ Keep backups & avoid single points of failure
- ④ Hire more resources — Horizontal Scaling

⑤ MICROSERVICE Architecture

Dividing & grouping resources for efficiency

- ⑥ Distributed System (Partitioning) ↗ fault tolerance
↳ for quick response → having local service

⑦ Load Balancer



How much time does it take to send response to customer?

- ⑧ Decoupling system — separating out concerns to handle separate systems more efficiently

⑨ Logging and Metrics ① Analytics

② Auditing

③ Reporting

④ Machine Learning

⑩ Extensible → designed to allow addition of new capabilities and functionalities

LOW LEVEL DESIGN

code implementation } process of specifying &
Classes defining the
functions detailed design of a
Objects software system