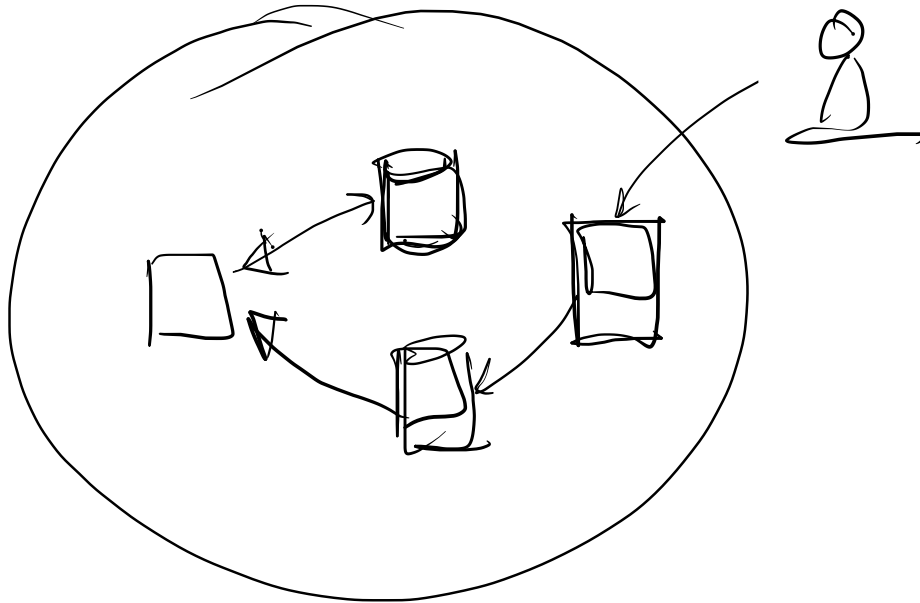
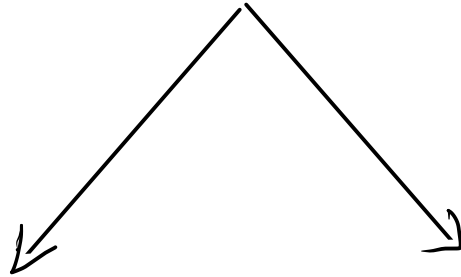


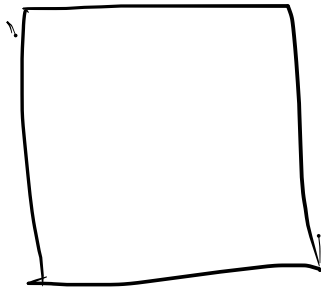
WEB APP ARCHITECTURE



WEB APP ARCHITECTURE



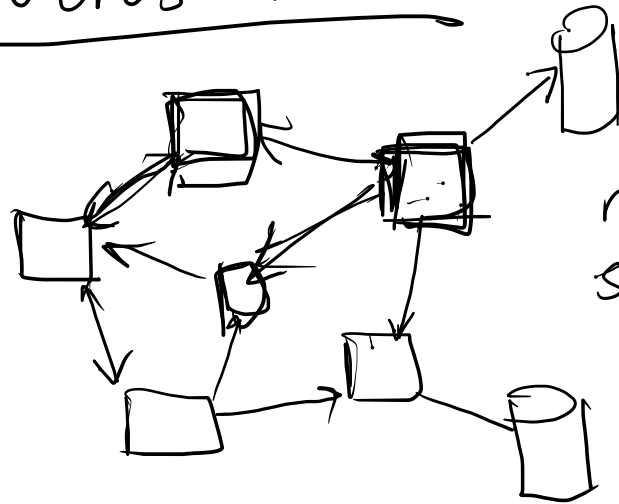
Monolithic



one

Single
code base

Microservices



multiple
smaller
services

Mono lithic :



full

Advantages

1.) Good for small teams

2.) Simplicity

3.) Less duplication

4.) Easier to deploy
as only single jar/war file

5.) Less problems of
network latency & security

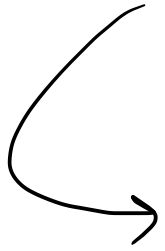
Disadvantages of monolithic:

- 1.) Becomes too large with time
 → difficult to manage
- 2.) need to redeploy the whole application, even for small change
 → deployment time increases
- 3.) difficult for new developer joining the team. (to understand the logic)
- 4.) Single fault in any module can bring down the whole application

Microservices

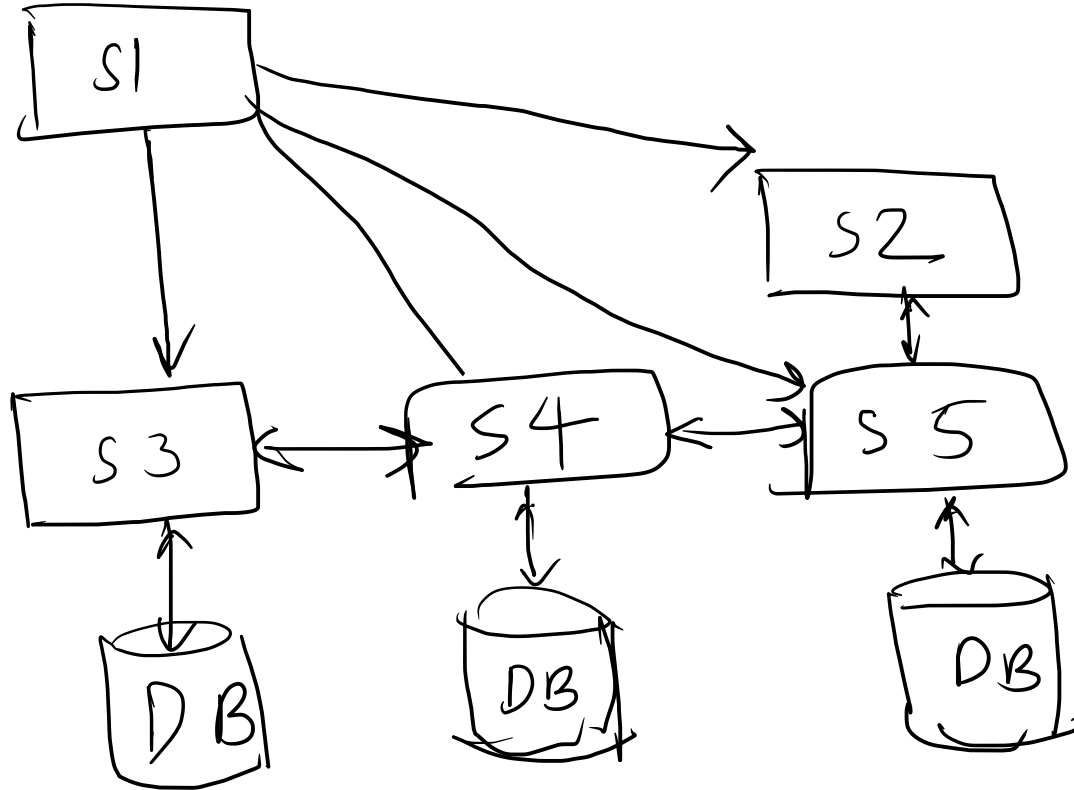
→ application is made up of smaller services that handle small portion of the functionality and data by communicating with each other directly.

Microservices

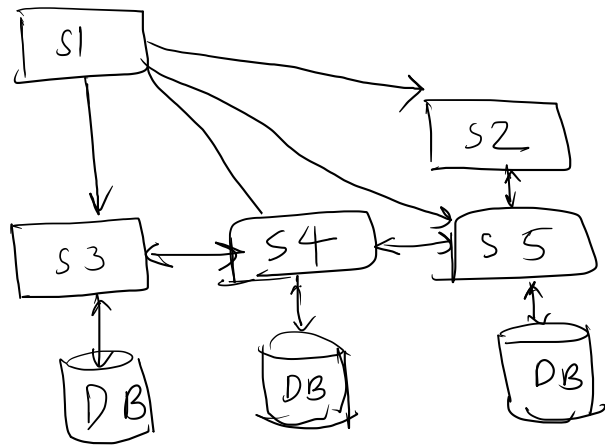


small services working together

Microservices



Microservices



1.) Each microservice has its own database

2.) duplication of some data

3.) Each service can be written in different programming language.

Principles of Microservices

- 1.) Single Responsibility 1 responsibility → 1 functionality
- 2.) Build around business capabilities
(→ each ms can use diff technology based on business requirement)
- 3.) Design for failure :
one ms goes down should not affect the whole system.

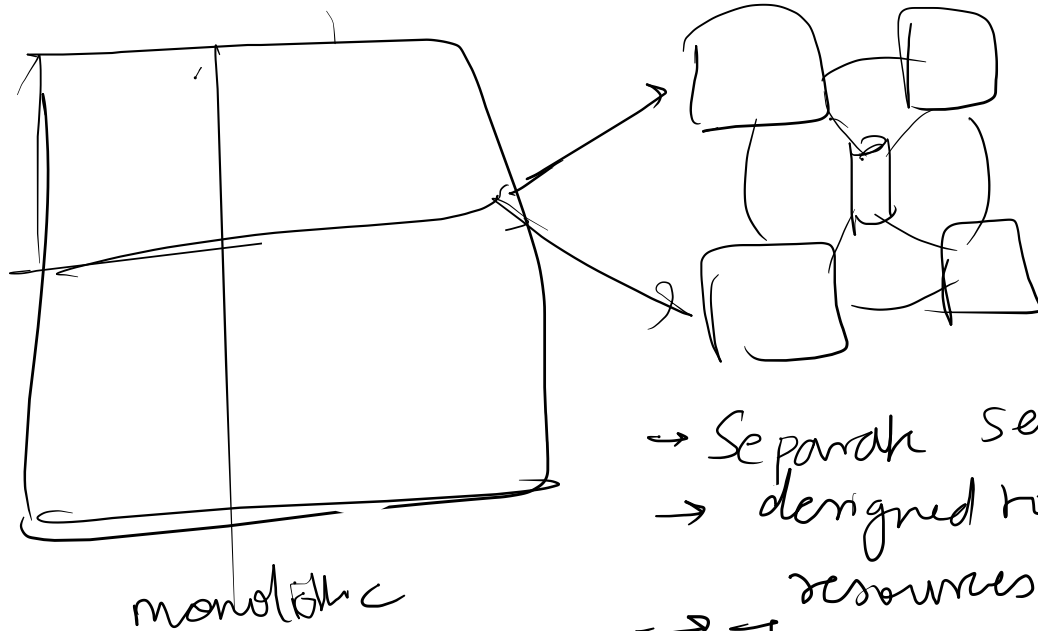
Advantages

- 1.) Easy to manage each service
- 2.) If any update in one MS, only redeploy that specific MS
- 3.) Easy for new developer to onboard.
- 4.) If one MS is facing heavy traffic, we can scale just that MS.
- 5.) Each service → diff. technology
- 6.) one service failure does not bring down the whole application

Disadvantages:

- 1.) Being distributed system, more complex.
→ Complexity
- 2.) Skilled developers are required to manage multiple services and their intercommunication.
- 3.) Network latency issues
Security issues due to inter-service communication
- 4.) Debugging is complex → difficult to find error cause
- 5.) Testing is complex → all services should work seamlessly

SOA (Service oriented Architecture)



- separate service
- shared resources
- Separate service modules
- designed to share resources across services.
- Involves component sharing

