Date: __/__/____

#3: Monolithics Vs Microservices :

- Microservice and its Patterns (Introduction) and Decomposition Pattern.

→ Monolithics vs Microservices:

i) Monolithics is also called Legacy application (when all functionalities are present in one application)

Disadvantages:

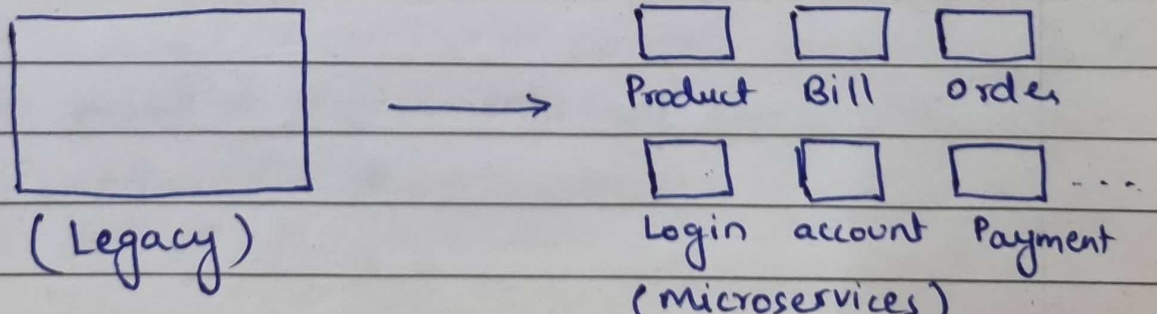
a) Overload IDE (large size application)

b) Scaling is hard

c) Tight coupled application
(If we change even one line, then it can affect many functionality and need to test whole application), also need to deploy whole application.

d) Difficult debugging

- To overcome all the disadvantages of monolithic we have introduced microservices.



#learnthesmarterway

Date: __/__/____

a) Managing the components becomes easy for developer
b) Scalability is easy

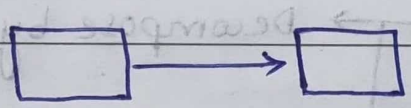
** Interview Question:

→ Advantage and Disadvantage of Microservice

Disadvantages:

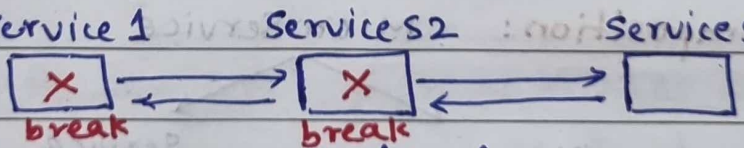
→ Properly decompose monolithic system, there should not be much dependency of components to each other

→ System should not be tightly coupled.



e.g. If product has high dependency on order service, then if we want to scale product then order should also be scaled.

→



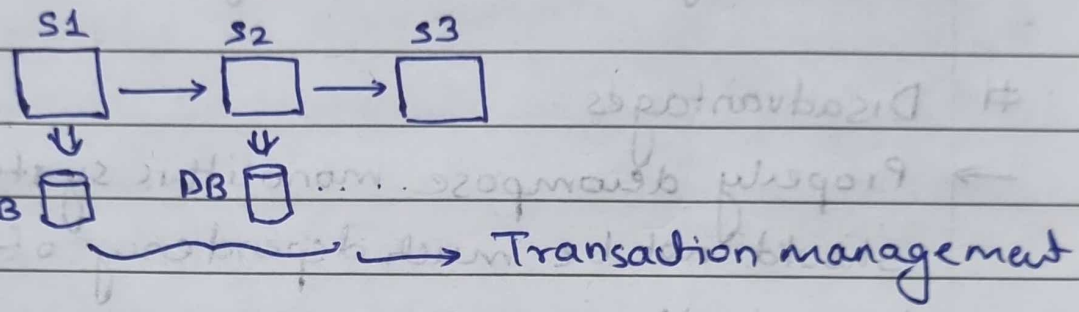
If there are some code changes in service S3, and it will go live. Now response of S3 changed and it is used by S2 (previously response), then S2 will break. If S1 uses S2, then S1 also breaks.

Date: __/__/____

→ Monitoring each components is difficult. Also debugging is diffult → e.g. the code breaks due to its own service or other service.

→ Transaction Management / consistency of data in Database.

→ Query



→ **Phases of Microservices:**

a) Decomposition → Decompose by business capability
b) Database → Decompose by subdomain
c) Communication
d) Integration
e) Deployment

a) **Decomposition:** MicroService

Micro (How much small?) | Service

⟹ Decomposition Pattern

a) Decompose by business capability
b) Decompose by subdomain

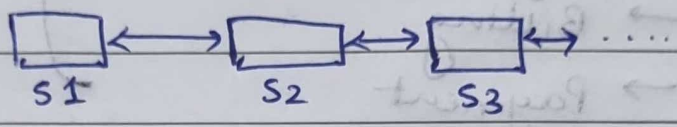
Date: __/__/____

b) **Database:** do we require separate DB for each service or a single DB to maintain all services (common DB)

→ Database per service

→ Shared database

c) **Communication:**



→ API communication

→ via Events

d) **Integration:** Microservice needs to be integrated with UI or other applications.

→ API gateway or other patterns.

e) **Deployment and Monitoring:**

→ So all these phases/patterns makes one microservice. From each phase, we need to decide one pattern based on requirements.

Then after the phases requirements / patterns were decided, then actual implementation starts.

Date: __/__/____

a) Decomposition :

i) Decompose by business capability :

e.g. Online order application

→ Order management
→ Product management
→ Account management
→ Login
→ Billing
→ Payment

} ⇒ all are microservices

Business functionality

Services are based on business functionality

Challenge : Good knowledge of business functions.

ii) Decompose by Subdomain (DDD) / Domain Driven Design