



**#ASLI ENGINEERING**

# Why microservices?



**BY**

**ARPIT BHAYANI**

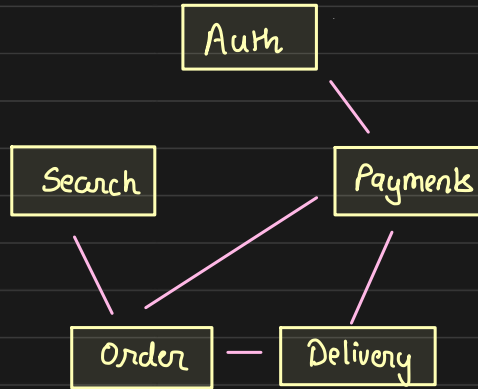
# Why Microservices?

What are microservices?

Small, autonomous, harmonic

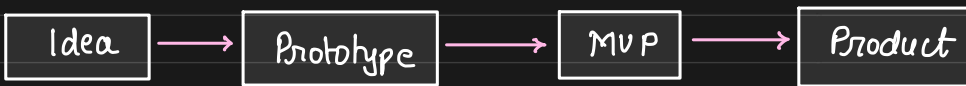
Core idea: Divide and Conquer

Break a product into manageable sub-problems and tackle each one of them separately.



\* Each sub-problem is solved separately and optimally

Why microservices?

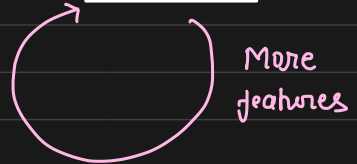


- codebase grows over time

↓  
- difficult to co-ordinate

- change in one module by one team can break others

- reduces product development velocity



- Scaling becomes predictable

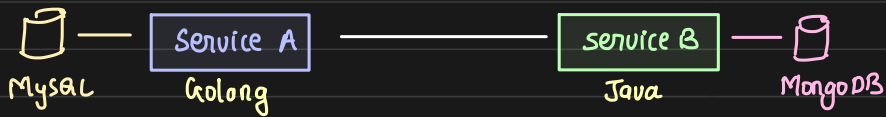
↳ Each service scales independently

Search might be on 10 instances

While Auth on 20 & payments on 3

- Autonomous & isolation

- your own tech stack, networking interface, protocols



- pure client-server relationship

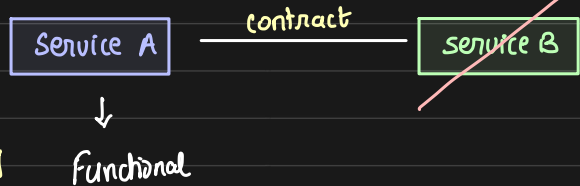


- You change your own code, deploy when you want

- Fault Tolerance

if one service is down.

you might still have a partially  
operational system



- Upgrades are simpler

You can evolve your service to  
transparently

Better Language

Better DB

Better infra

Java —————> Golang



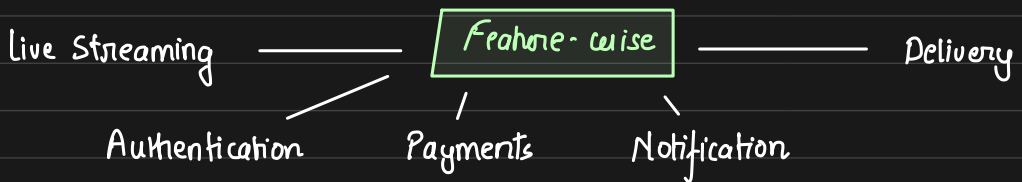
Seamless so long as API contracts are intact

How to fence Microservices?

We know it needs to have "small" set of responsibilities,  
but how small?

Too big is a problem → same as monolith

Too small is a problem → inter team, inter service dependency



Note: Each microservice may have  
sub-components / processes