Day 3
Monday, October 2020 10:00

# Monolith

## Design Zerodha/Upstox

Monolith | Microservices

→ Sign up
→ Market data
→ Buy/Sell
→ Portfolio - reports
→ Payment
→ Portfolio
→ Notification
→ Logging, Monitoring (Analytics)

Part
→ Portfolio
Portfolio
Buy/Sell
Monolith

Monolith
→ Single codebase
→ Easy Development
→ Easy testing
→ Easy Deployment
→ Easy Debugging
→ Easy to track performance

Drawbacks
→ Scaling
→ Deployment
→ Constraint in Tech (everyone uses one single box)
→ Single point of failure
→ Slower development speed

Microservices
→ collection of small self contained modules

Pros
→ Independent
→ Scaling
→ Tech flexibility (each module can be in its own best language & tech)
→ No Single point of failure
→ Rapid development

Cons
→ Higher Cost
→ Hard - Multiple source
→ Organization Coordination & costly
→ No standardization

Microservices → Monolith

Pros
→ Lower
→ Speed
→ Netflix (2008) → Amazon Prime Conference (2018)

How?
Read code modules of component existing from Microservice to Monolith

Microservices
Buy/Sell → Buy
→ Sell
Portfolio → Login
→ Signup
Payment → Pay
→ Refund
Analytics → Logging
→ Monitoring
Order management → Order Backing
→ Order Payment

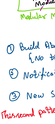
How? Micro to Monolith

Monolith → Microservices

Patterns (Guidelines)

① Identify parts of existing system that we want to migrate
② Implement new functionality into new microservice
③ Redirection {{Strangler Fig Pattern}}

Zerodha




① Build Abstraction for functionality (no tight coupling)
② Notification Module
③ New Service

Microservice pattern is called Branch by Abstraction

## Microservice Communication

Monolith can just invoke function for communication

But for microservices,
→ call function/procedure in a remote process as if they were local
ie. RPC → Remote Procedure Call

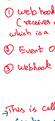
gRPC → open source
→ built on top of HTTP2
Multiplexing capability of HTTP2
gRPC uses Protocol buffers → Protobuf

Protobuf
→ Language agnostic
Binary Serialization Format
→ Higher performance
→ Deal with high volume of data
→ Smaller Latency
Communication between microservices } → gRPC

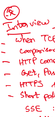
JSON
{
name : "",
age : 25,
Active : true,
}
→ Binary
So the binary lightweight

3rd party

Payment




Goals




webhook




when the event occurs, send HTTP POST to this URL.

① Web hook setup → Unique URL (receiver will send URL to 3rd party, which is a unique URL)
② Event Occurred
③ Webhook Execution → HTTP POST request to receiver with data on that unique URL

→ This is called Event Driven Architecture
→ Can be used in Automation
→ Integration with third party APIs

## Interview Week 1

- When TCP, UDP to use
- Component TCP vs UDP trade off
- HTTP Communication
- Get, Post, Put, Delete
- HTTPS secure
- Short polling, long polling, webhook, SSE, Message
- Monolith vs Microservice
- gRPC
- Web hook