Introduction to Microservice

Hi, my name is Armand,

I am a Senior Software Engineer at 

Mandocornelis

- **Monolith** and **Microservice**
- Fallacies of **Distributed** System
- Principle of **Microservice**
- **Microservice** Chassis

Monolith

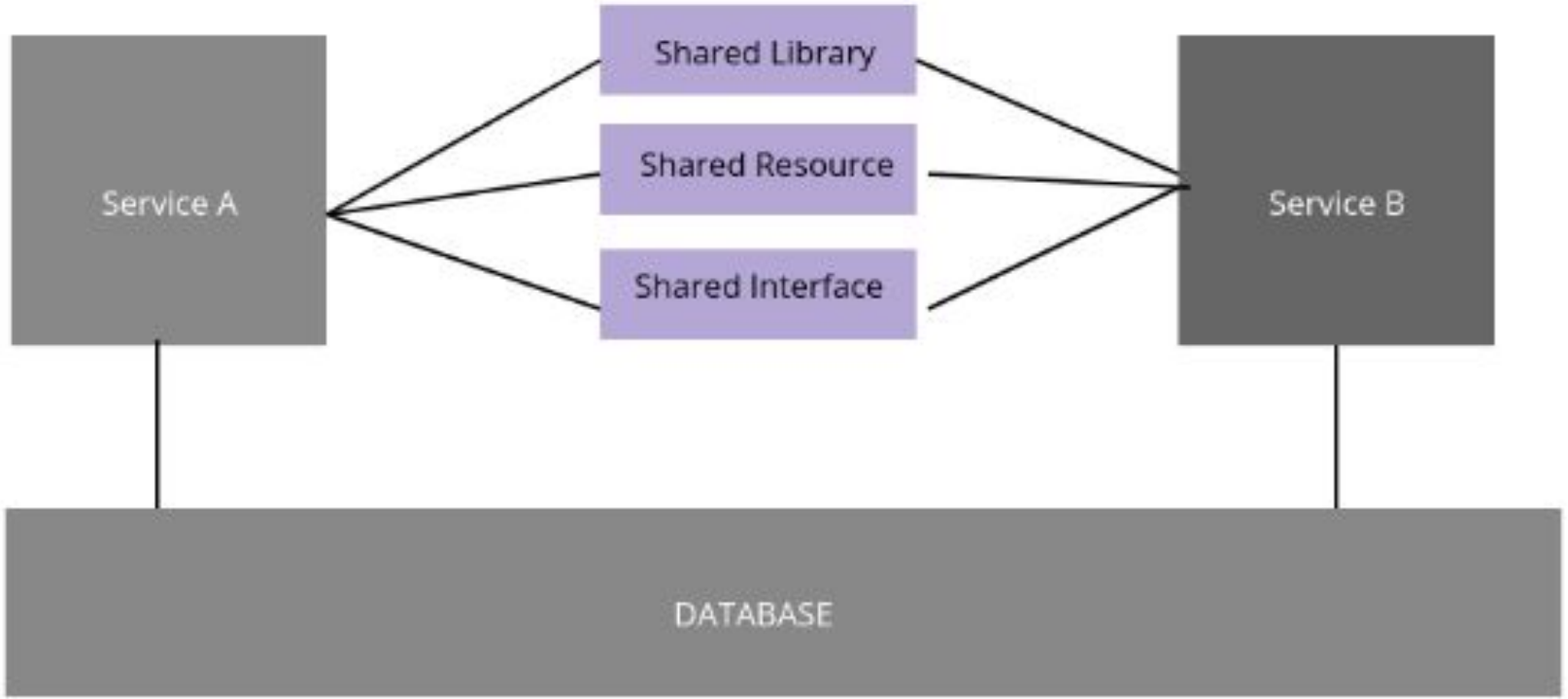
Service whose design, information model, and interface combine multiple competing and interfering domains into single big ball.

Monolith

- Quick and Easy :)
- Simple infrastructure, hooray!
- Glorious debugging!

- Concurrency hell
- Adding features take months
- Bugs galore

High Coupling Low Cohesion




I HAVE A BAD FEELING
ABOUT THIS MONOLITH.
imgflip.com
www.StrangeZoo.com

PROBLEM	SOFTWARE	TEAM
Complaints	Lengthy fixes and release	Frustrated last minute changes
Scale issue	Hard to maintain 1M+ line code	Lack of Agility
Release	Takes months to ship	Who is the owner? Fixer no Innovator
Solutions	Shared mess	Hack fix, unclear standard

Service Oriented Architecture

Service-oriented Architecture (SOA) is an architectural pattern in computer software design in which application components provide services to other components via a communication protocol.

SOA Principles

- Boundaries are explicit
- Services are autonomous
- Service shares schema and contract, not class
- Service compatibility is based upon policy

Fallacies of Distributed System

- The network is reliable.
- Latency is zero.
- Bandwidth is infinite.
- The network is secure.
- Topology doesn’t change.
- There is one administrator.
- Transport cost is zero.
- The network is homogeneous.

Distributed Monolith

A **change** to one microservice often requires **changes** to other microservices

Deploying one microservice **requires** other microservices to be **deployed** at the **same time**

Your microservices are **overly chatty**

Your microservices share **a lot** of the same **code or models**

What is microservice?

“Service oriented architecture

composed of

loosely coupled elements

that have

bounded context”

What is microservice?

“Service oriented architecture

Service communicate with each other over the network

What is microservice?

“Service oriented architecture

composed of

loosely coupled elements

You can update the services independently; updating one service doesn’t require changing any other service

What is microservice?

Self-contained; you can update the code without knowing anything about the internals of other microservice

that have

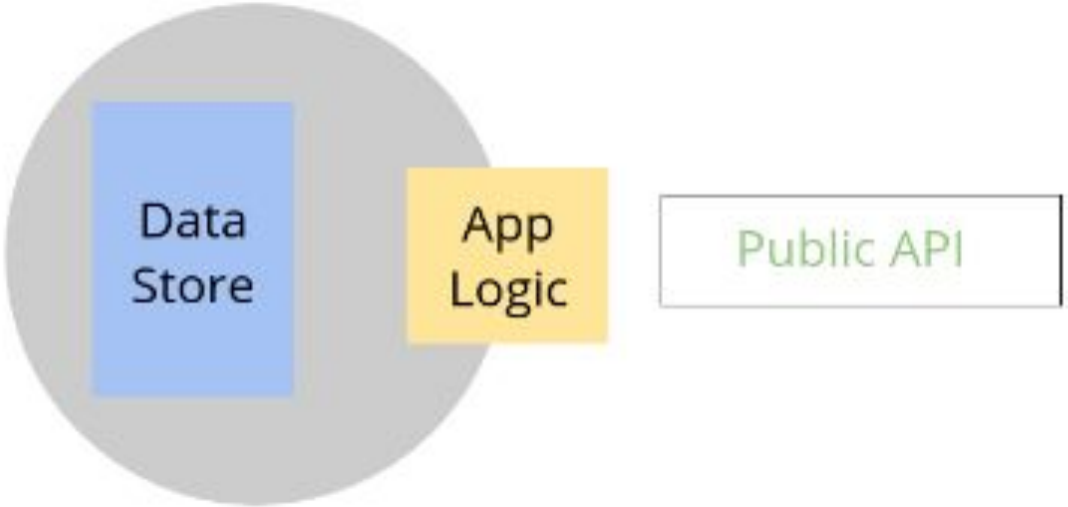
bounded context”

Why Microservices?

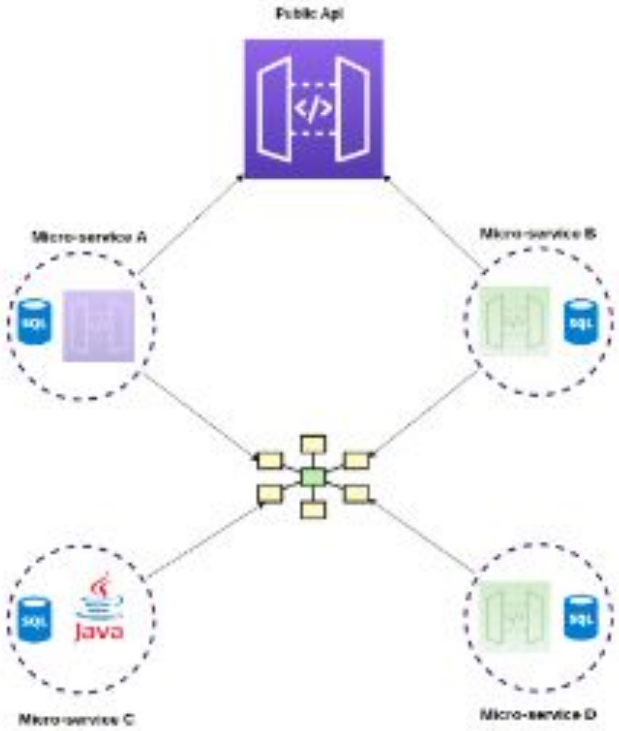
- Clear responsibilities and dependencies
- Small & clean packages
- Easy to scale
- Easier deployments
- Faster delivery
- Open up language agnostic possibilities

Do one thing and do it well ~ Separation of concern

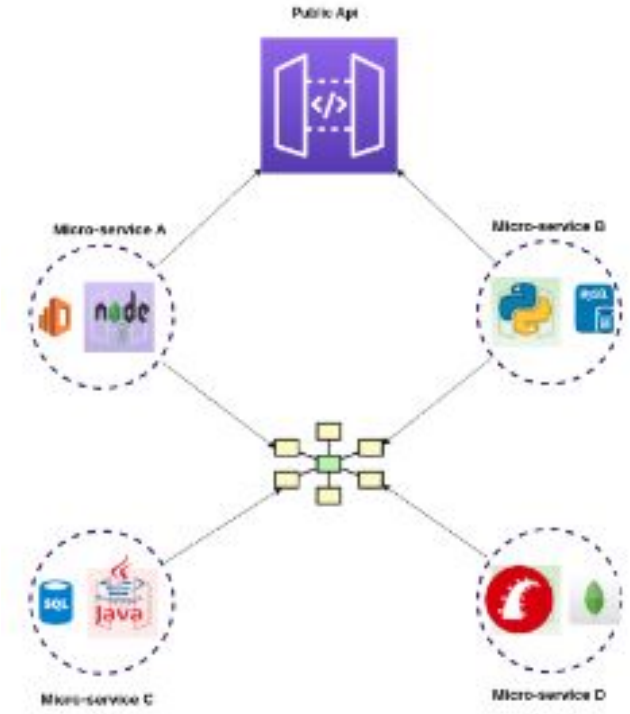
Microservice anatomy



Principle 1: Build clean and clean public APIs



Principle 2: Use the right tool for the right job

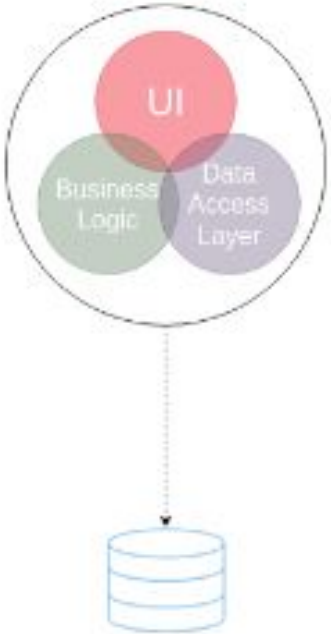


Principle 3: Ensure backward compatibility

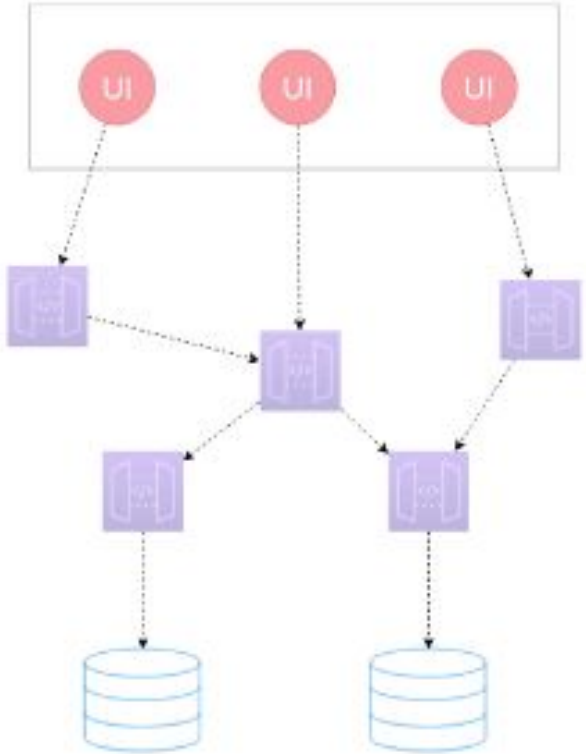
Principle 4: Secure everything!

- Network
- Container/Servers
- API Throttling
- Authentication
- Secret Management

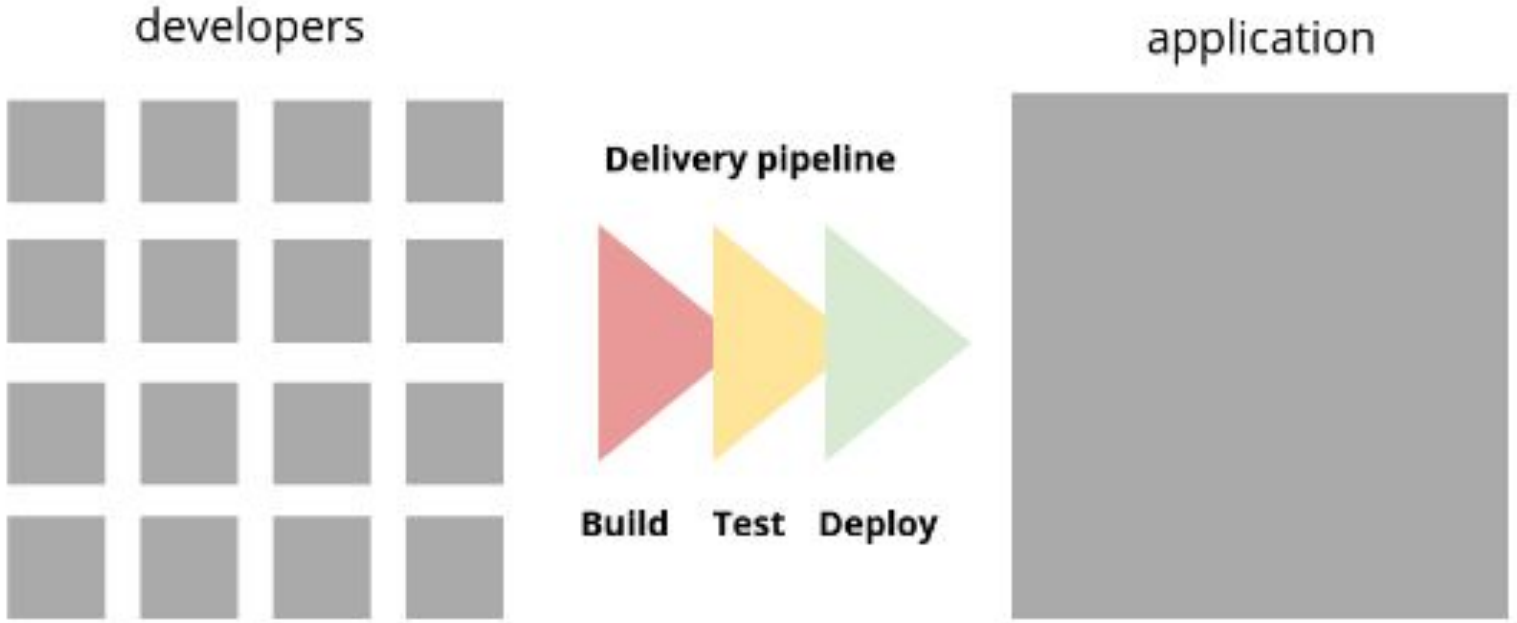
Monolitic



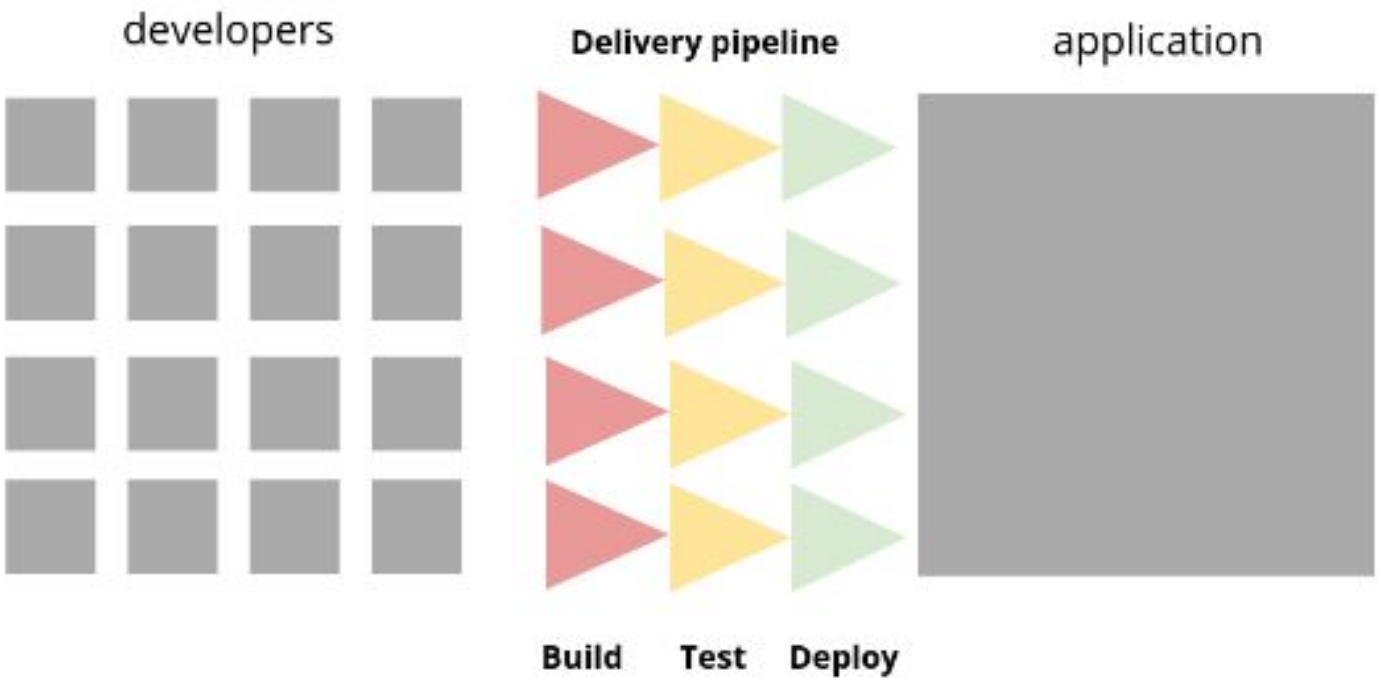
Microservice



Monolith’s development cycle



Microservice development cycle



Microservice Chassis

1. Scheduler
2. Logging
3. Service Discovery
4. Communication (REST vs Messaging)
5. Authentication and Authorization
6. Distributed Tracing
7. Externalized Configuration
8. Services Health Monitoring
9. Metrics

That’s all Folks!