Microservices + Software Development Life Cycle

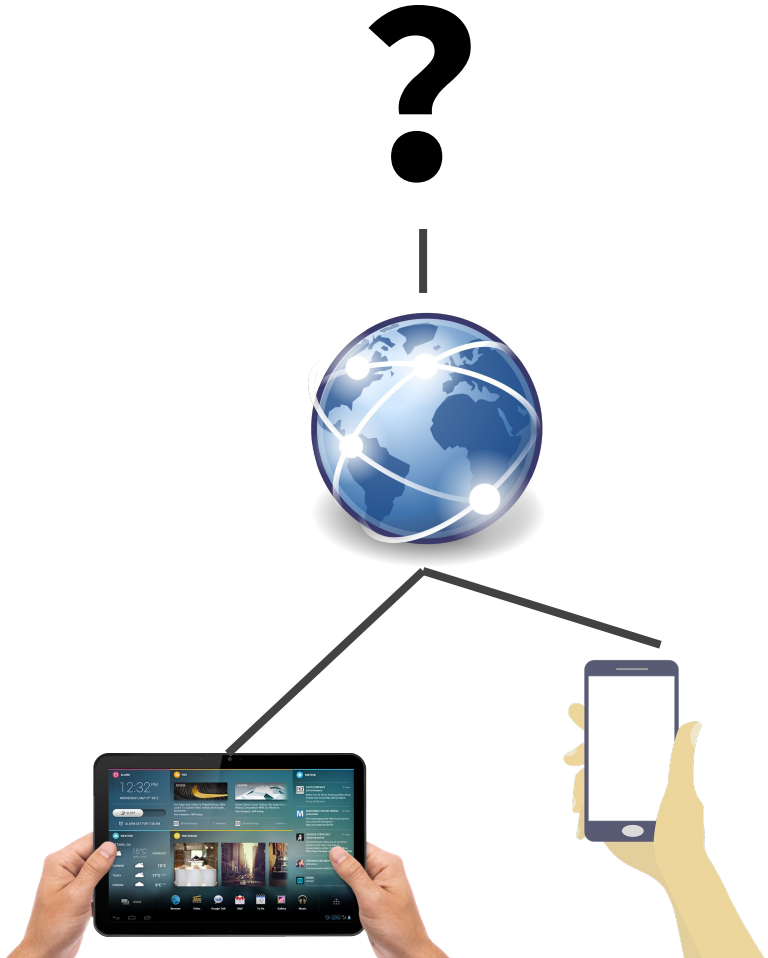
Team 3
Edgar Lopez-Garcia
Ashot Chobanyan

Overview

- Monolithic Architecture
- Microservices Architecture
- Benefits of Microservices
- SDLC Implications
- Tools, Frameworks, Platforms, and Services

Monolithic Architecture

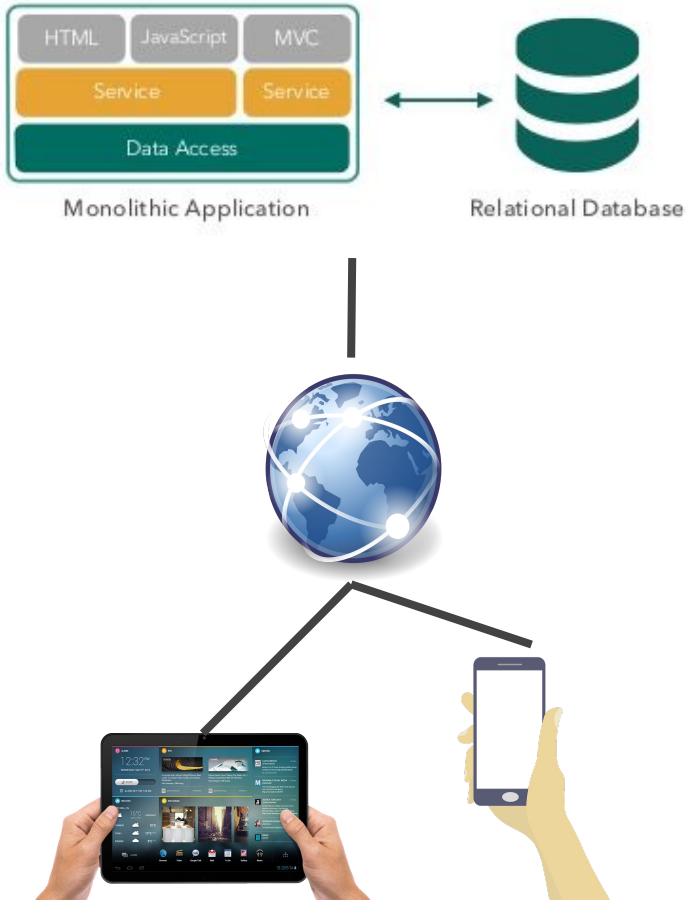
- Web-based enterprise application that supports thousands of users on desktop, mobile.
- Requires application on servers that holds business rules, data, etc.



Monolithic Architecture

- Monolithic - formed of a single large block of stone.
- Application all in one place (server).

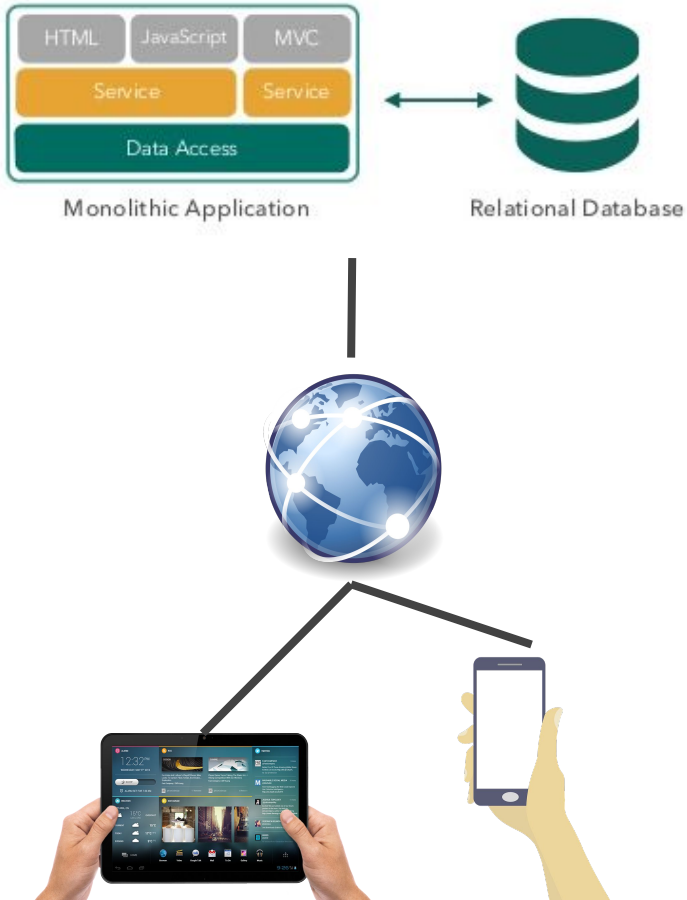
1. Users requests website via http.
2. Server picks up request and sends back html/js/css.
3. Any additional requests are processed by same server.



Monolithic Architecture

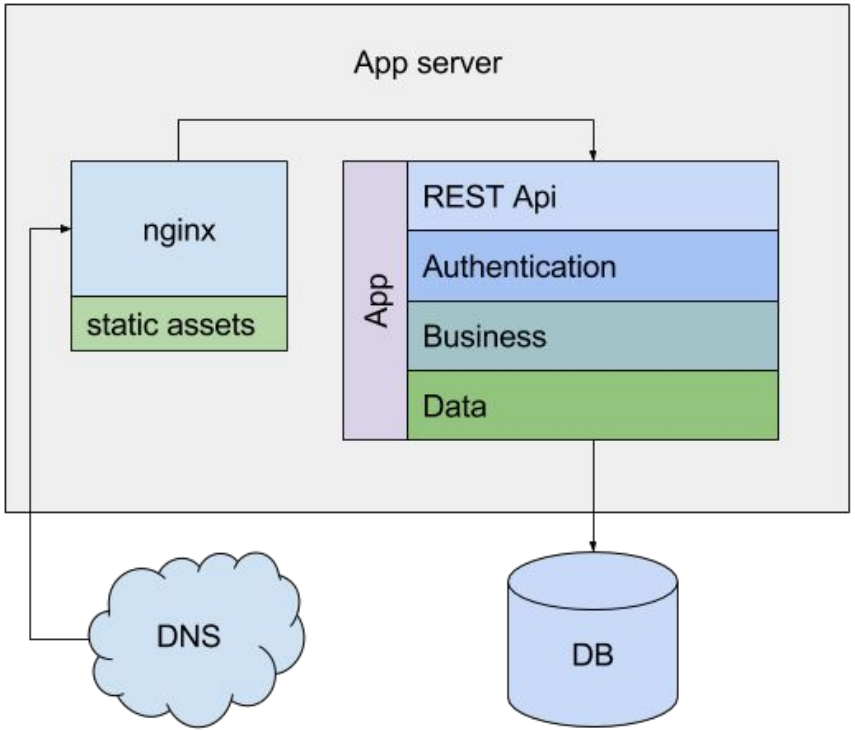
+ Easy deployment
+ Easy and fast development

- Tight coupling
- Single point of failure
- Hard to scale independent services



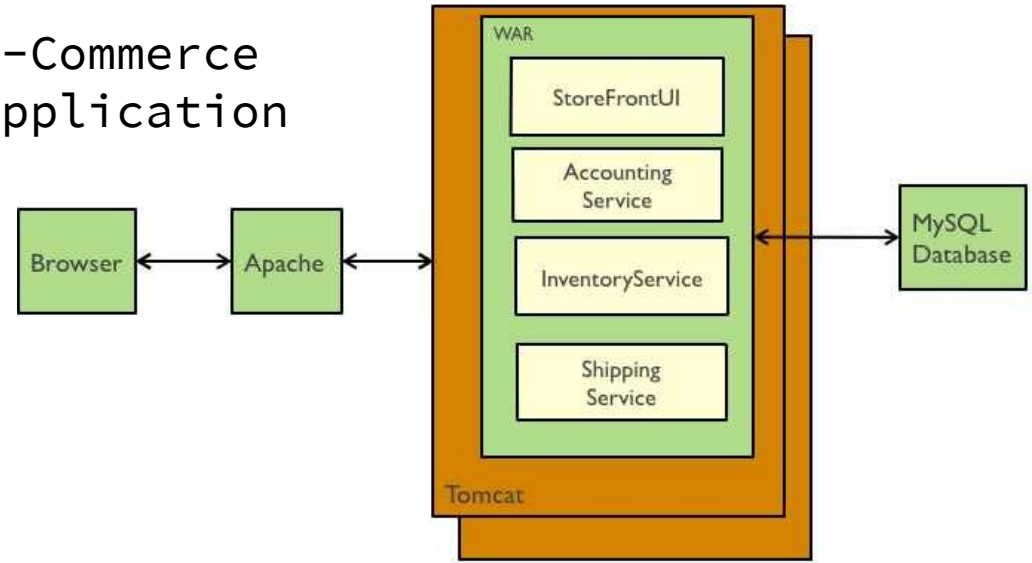
Monolithic Architecture

- Server contents
  - Nginx - http
  - Static assets
  - App binaries
- Example app services
  - Authentication
  - Business
  - Data
  - API



Monolithic Architecture

E-Commerce application

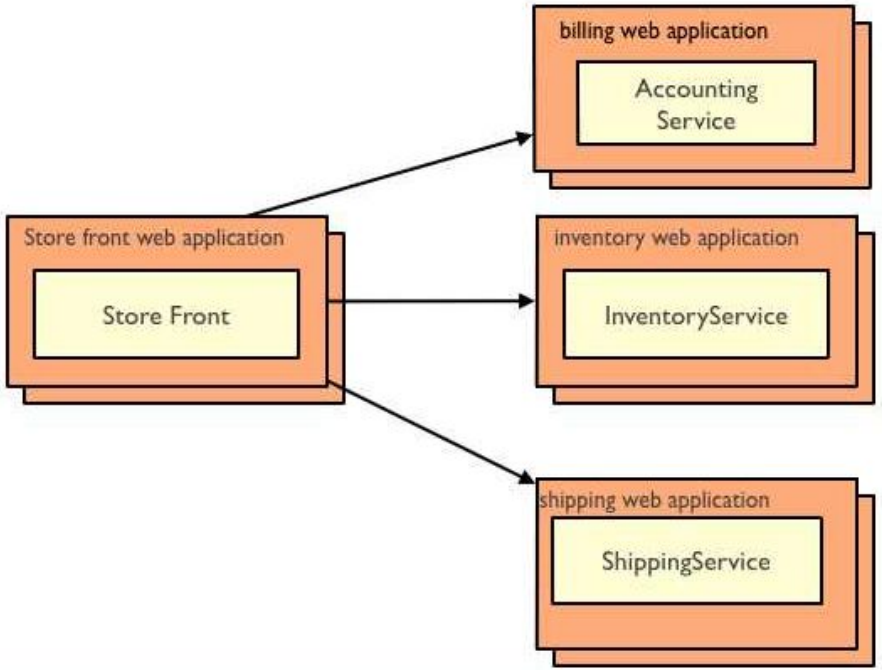


Apache - HTTP Server
Tomcat - Serverlet container
- Process requests via Java Serverlet
- Dynamic content
- Java runtime environment

Web **A**pplication **R**esource (**WAR**) - File format
- Contains JAR, JavaServer Pages (JSP), Java Serverlets, Java classes, XML, etc.

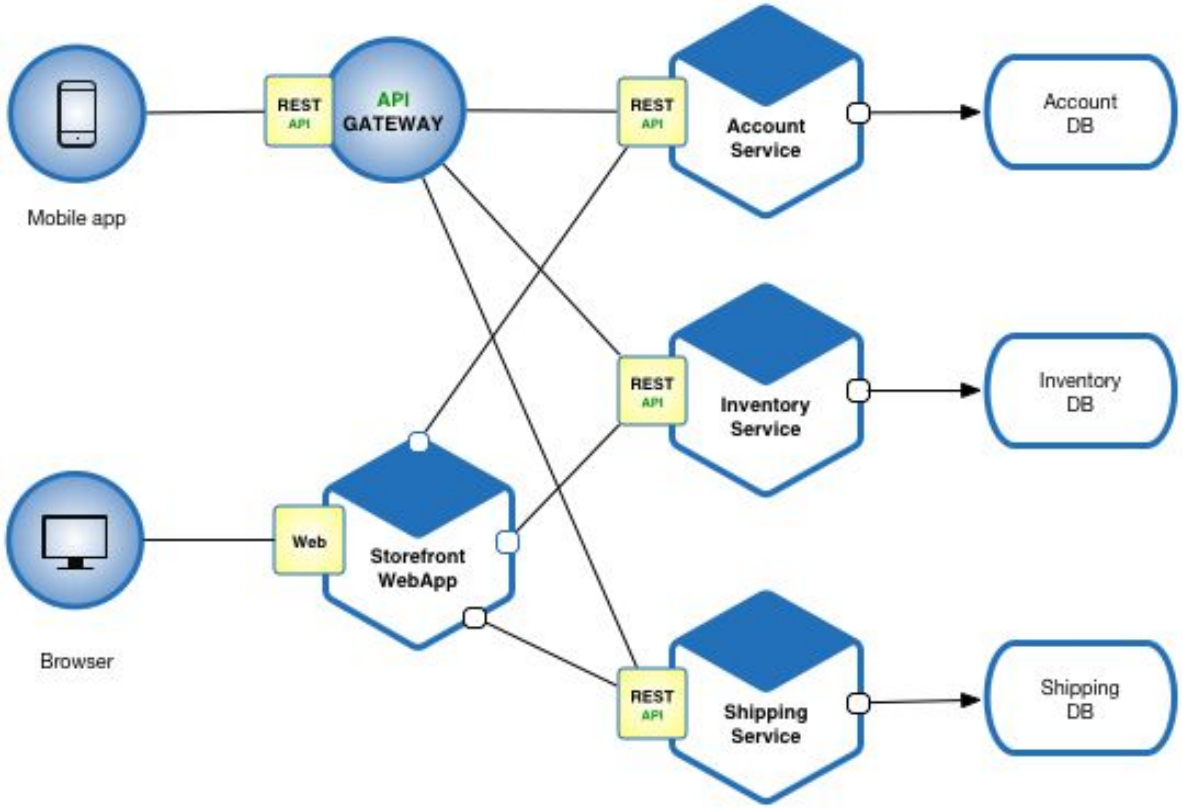
Microservices Architecture

- Microservices - break into smaller processing units
- Application spread out into many serving environments
  - Application is the microservices communicating with each other



Microservices Architecture

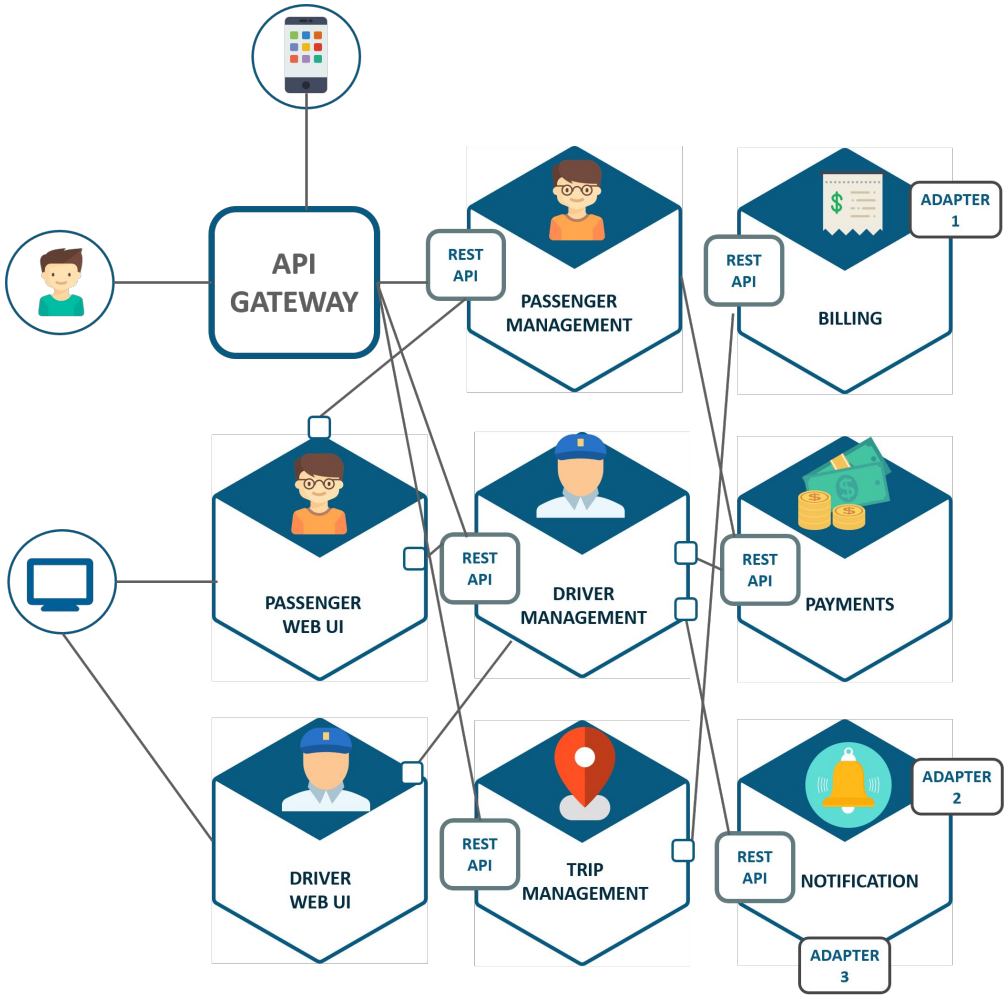
- Separate concerns
  - Decoupled & modular
  - Mobile & Browser
- Glued together by REST API



Microservices Architecture

Another Example - Ridesharing application

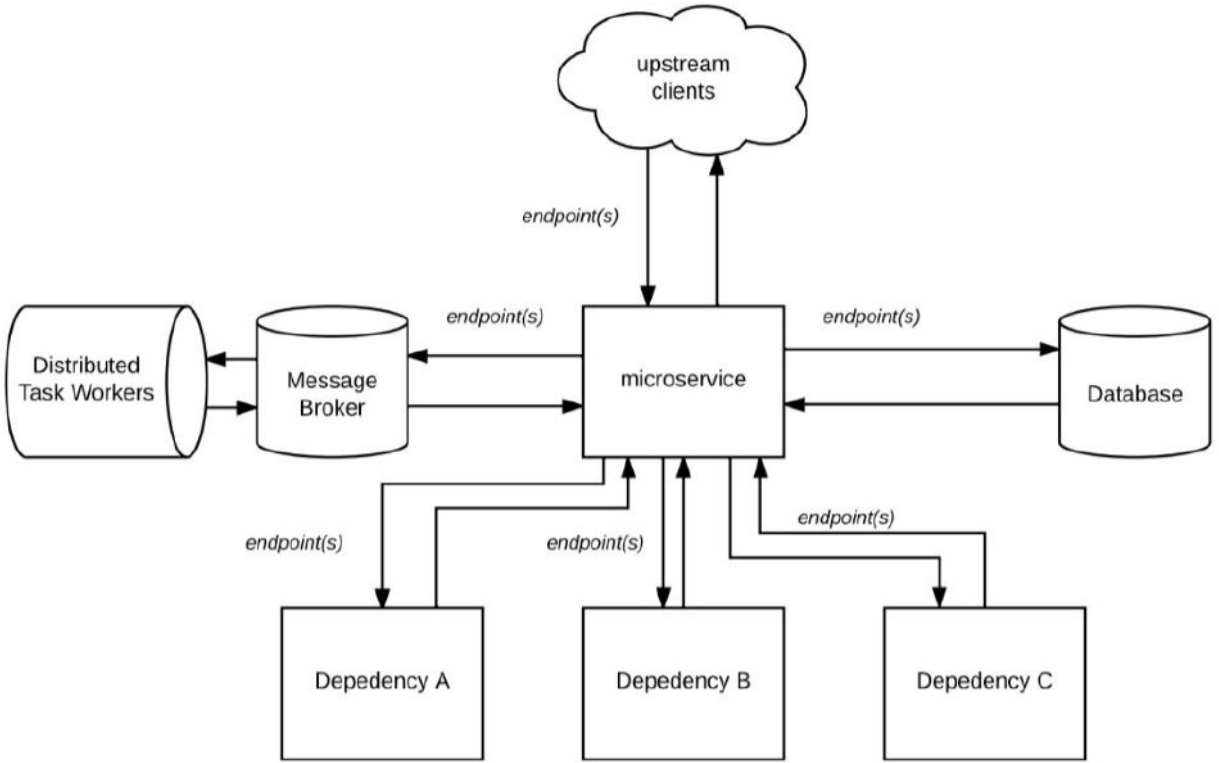
- Decoupled = Web & Mobile
- Each services only does one thing
- Application consists of inner-microservices communication



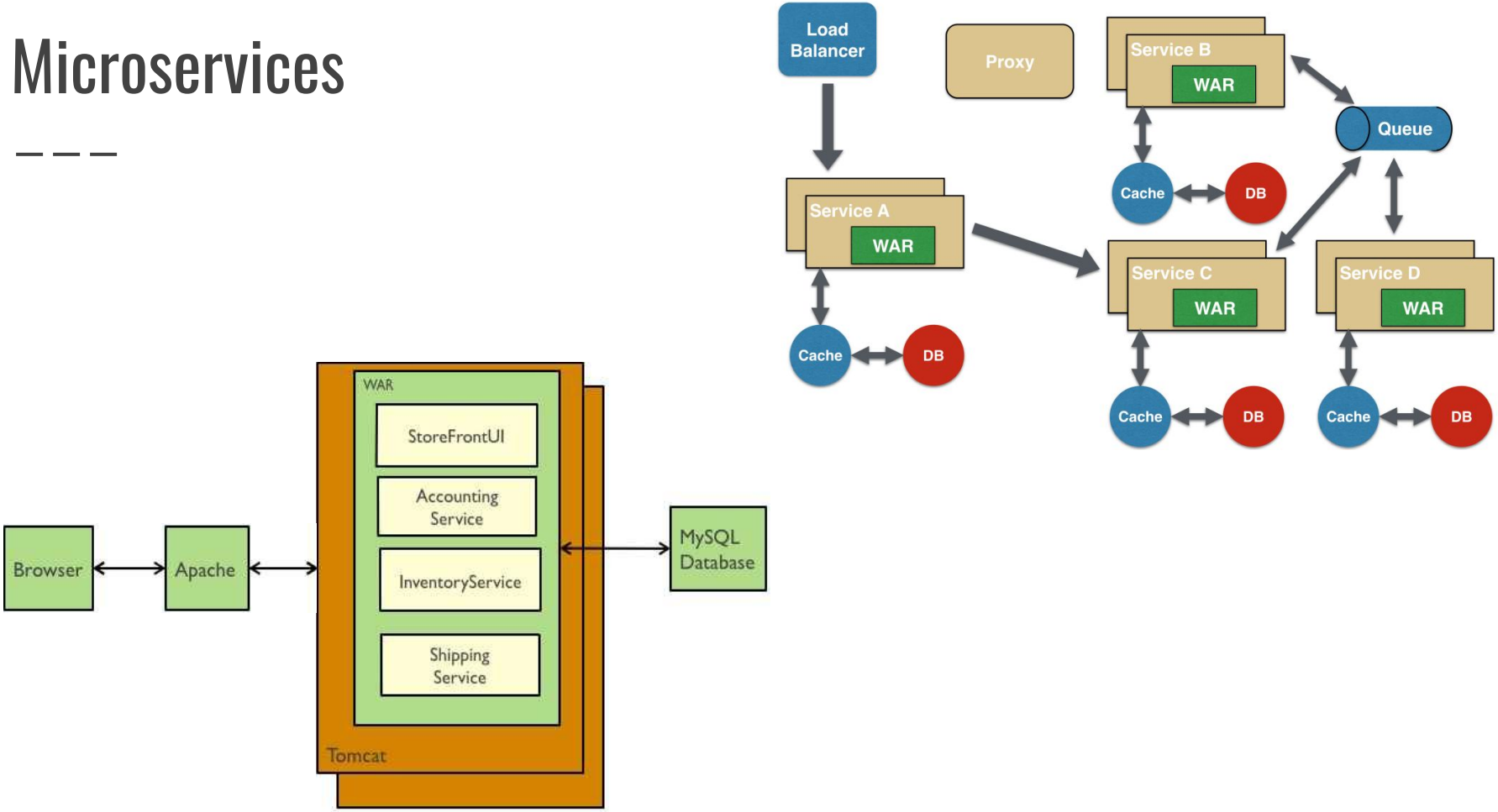
Microservices Architecture

Detailed look

- Message Broker (queue, pubsub,etc)
- Task workers
- All communication via endpoints



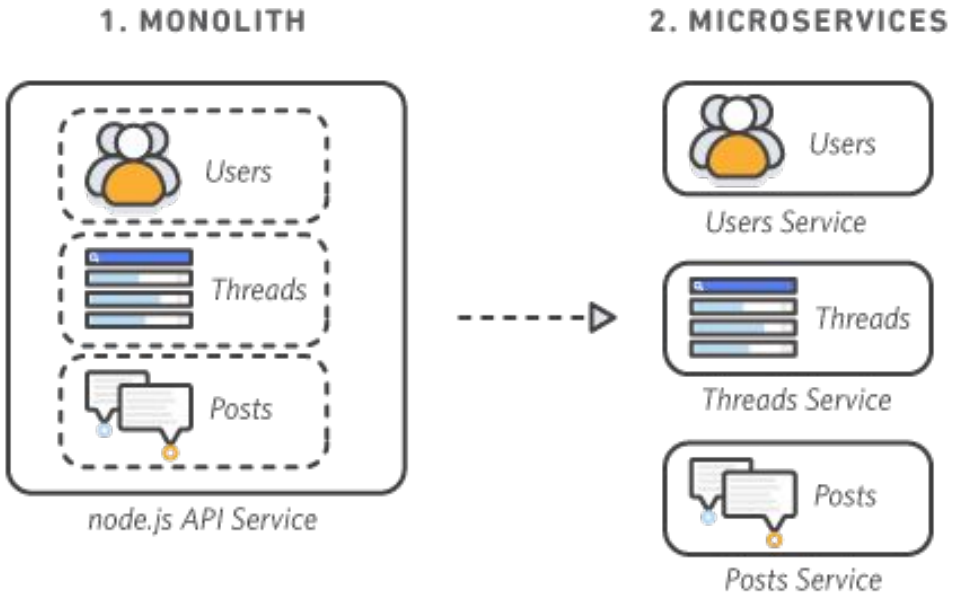
Microservices



Benefits of Microservices

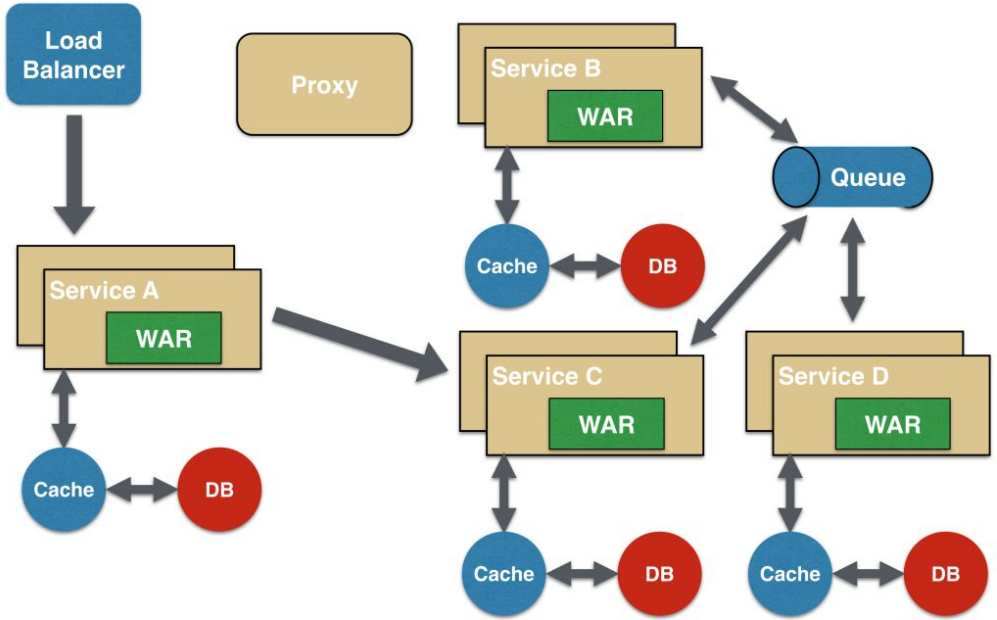
Social media app example

- Services are independently managed
- Each has specific job
- Fault doesn't crash all of app



Benefits of Microservices

- Independently deployable
- Specialized languages (performance)
- Better organized (one job)
- Decoupled



Software Development Lifecycle (SDLC) Implications

Purely Monolithic

- Single long SDLC (1 per application)
- Traditional techniques

Purely Microservices

- Distributed SDLCs (1 per feature)
- Favors AGILE workflows

SDLC Implications

Separate independent lifecycle per function in MSs

Creation, Release, Monitoring, Depreciation

SDLC Implications

Extra overhead due to microservices:

- Maintaining security
- Integration testing + Designing for failure
- Monitoring/operational overhead
- Ensuring consistency

SDLC Implications

Additional overhead in microservices can result from:

- Diminished code reuse
  - Addition of multiple languages can also affect this
- Interdependent/coupled microservices
- Expanding service boundaries (ex. where should new logic live?)

SDLC Implications

Overhead of Monolith SDLCs

- Heavy documentation needed
- Time to facilitate changes
- Difficulty adding new developers
- Tests need deployment of full stack

SDLC Implications

Benefits of Microservice SDLCs

- Overall reduction of overhead at scale
- Faster in organized large teams

Benefits of Monolith SDLCs

- Reduction of operational overhead
- Easier in general, especially for small teams

Mixed approaches are possible too!

Tools, Platforms, Frameworks, and Services

Needs: Gateway

Automated deployment

Accessible endpoints (eg Service Discovery)

Events

Tools, Platforms, Frameworks, and Services

Devops needs:

- Logging
- Config Management
- Monitoring

More robust tools are needed for MS based apps

Tools, **Platforms**, Frameworks, and Services

Containers

- Docker
- CoreOS Rocket

Container Orchestration

- Docker Swarm
- Kubernetes

Work for both Microservices and Monoliths

Tools, Platforms, **Frameworks**, and Services

Example Frameworks

- Dropwizard
- Vertx
- Spring Boot
- Restlet

Tools, **Platforms**, Frameworks, and **Services**

Serverless/Anonymous

```
    Serverless Microservices

    (PaaS Providers: Amazon, Azure, Google, IBM)

 +  Serverless Monitoring Tools

    (Dashbird, IOPipe, DataDog)
```

Not possible with Monoliths

Q & A