

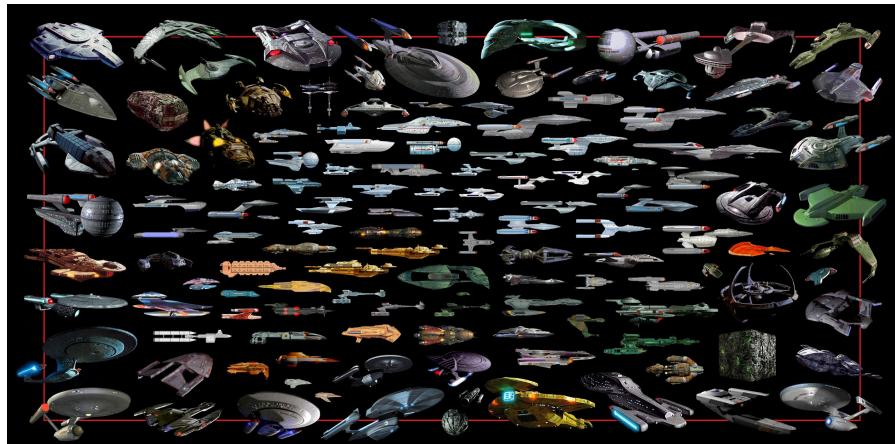
Peter Eijgermans

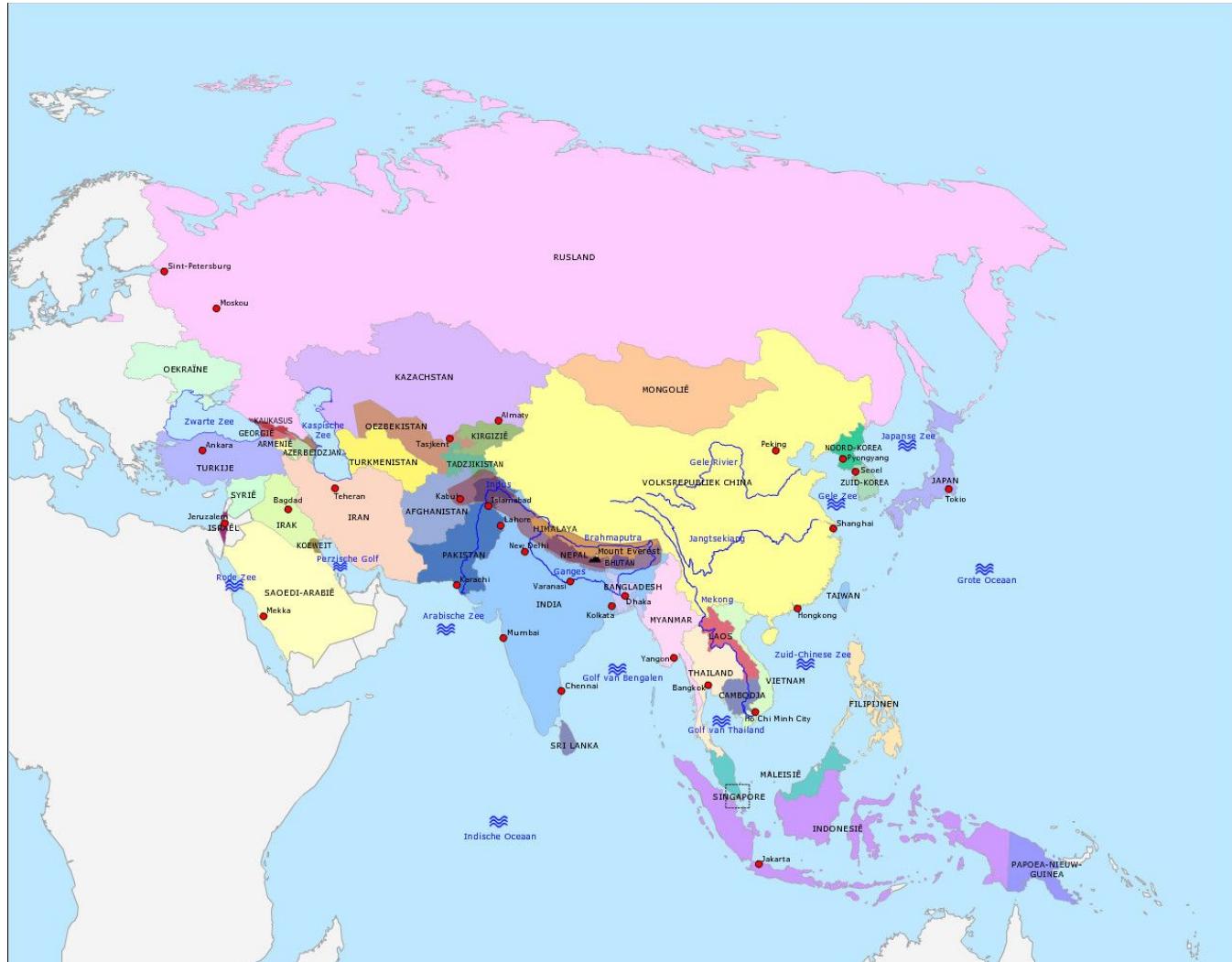
ORDINA



 @EijgermansPeter

Microservices or Monolith?











What's so bad about monoliths anyway...?!

Fewer Cross-cutting Concerns

Less Operational Overhead

Performance

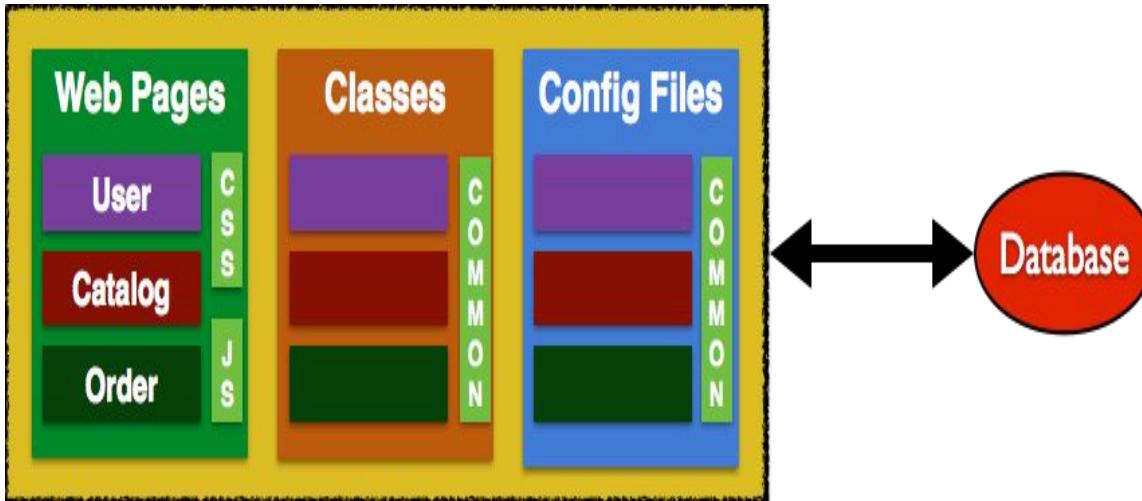
What's so bad about monoliths anyway...?!

If you can't build a monolith,
what makes you think microservices are the answer?



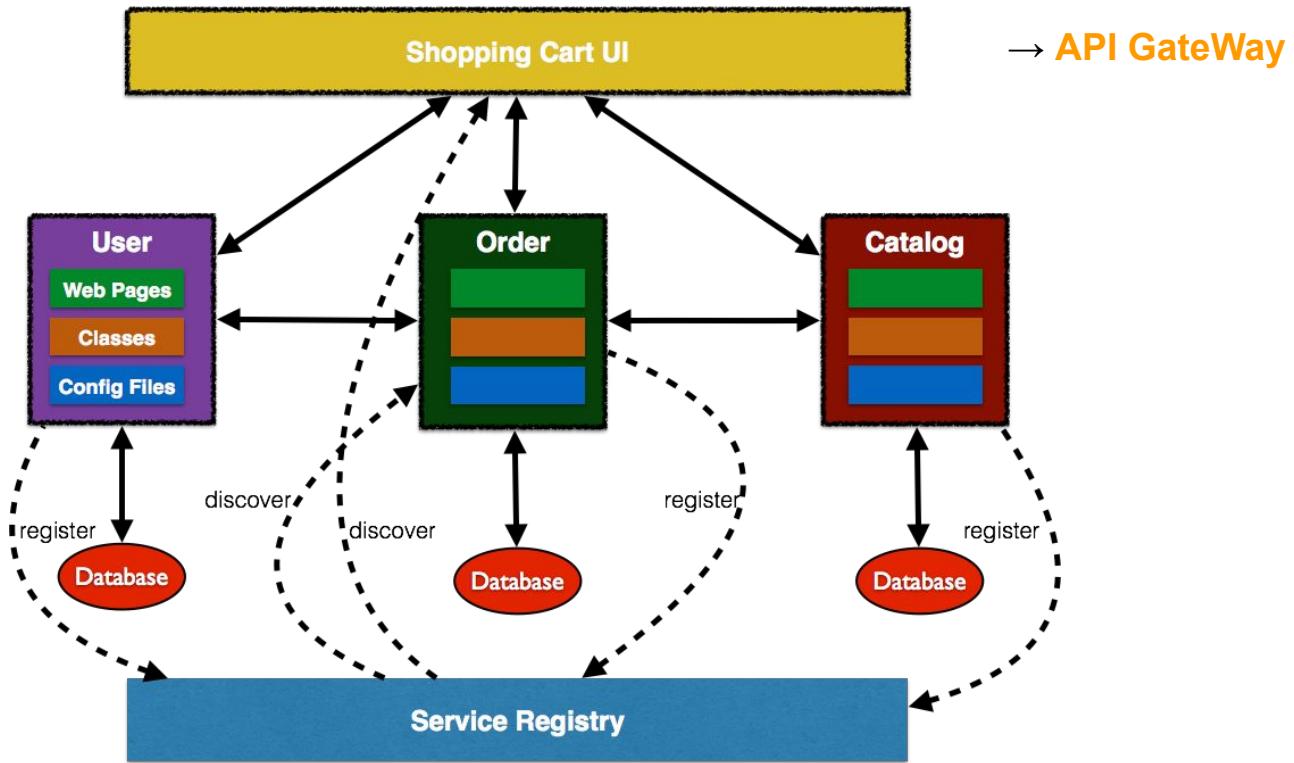
Monolith versus Microservice

Monolith



What are Microservices?

Microservice



What are Microservices?

Benefits

Capabilities

Benefits of Microservices?



Enables business agility, fast respond to markets and cost-efficiency

For what type of systems do you have to use microservices and when not?

Capabilities

Independent & Scalable



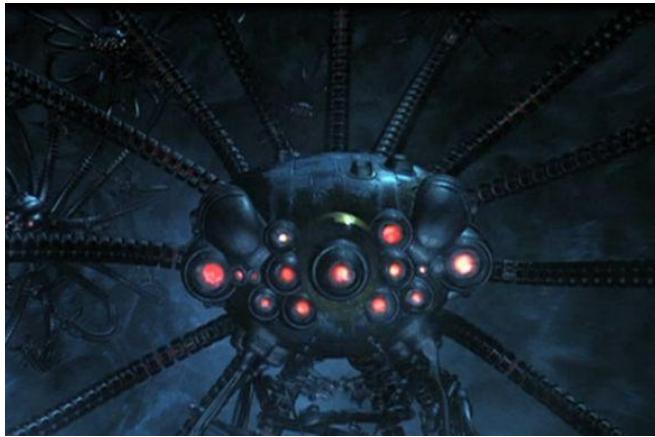
Capabilities

Monitorable & Fast delivery



Capabilities

Easy to maintain & Resilient



Statement

**The key concern of Resilience is not
“how to avoid failure” but “how to deal with failure”**

Tooling

Resilience: Netflix OSS tools, Hystrix, Ribbon, etc...

Platform: Zuul proxy, Kubernetes, Mesos, Marathon, Rancher, Puppet, Chef, Ansible, OpenStack...

Service Discovery: ZooKeeper, etcd, consul, Kubernetes, Netflix Eureka

Delivery: Jenkins, GITLab, Bamboo, TeamCity, GoCD, Docker, rkt (Rocket), ...

Logging Metrics: Prometheus, GrayLog, LogStash, Kibana, Grafana, ElasticSearch, Splunk

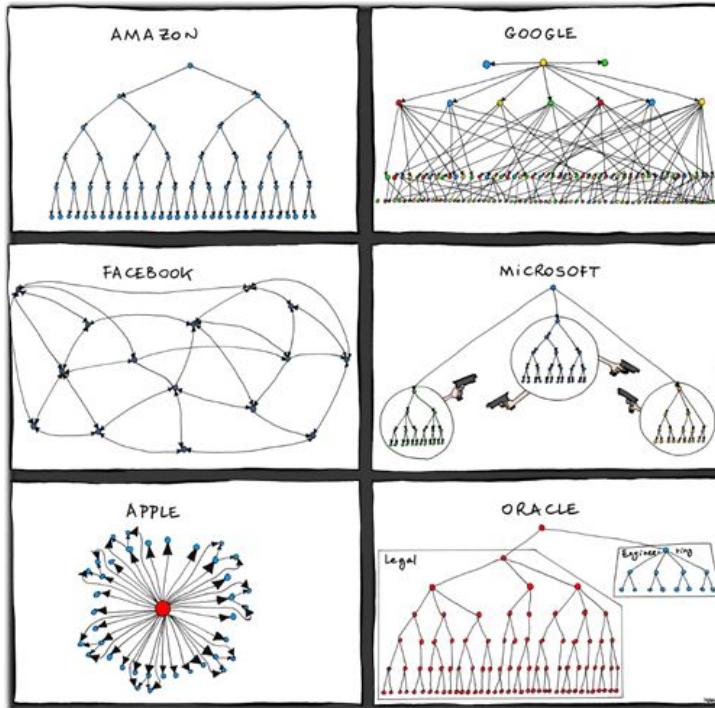
Messaging: Kafka, RabbitMQ, ...

(Distributed) Databases: MongoDb, CouchDb, Cassandra





Conway's Law



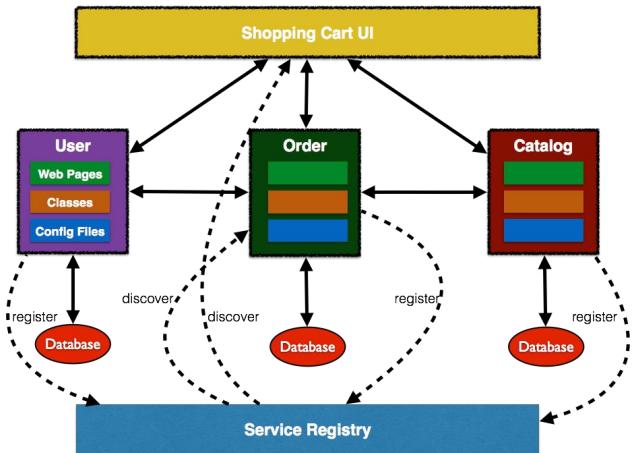
Statement

**If the organization does not change,
the benefits of agility are much less!**

Statement

Microservices have a much more complex ecosystem than monolithic applications.

What are the issues in using REST over HTTP for Microservices?



container management

skilled teams

deployments to production

testing

debugging async messaging

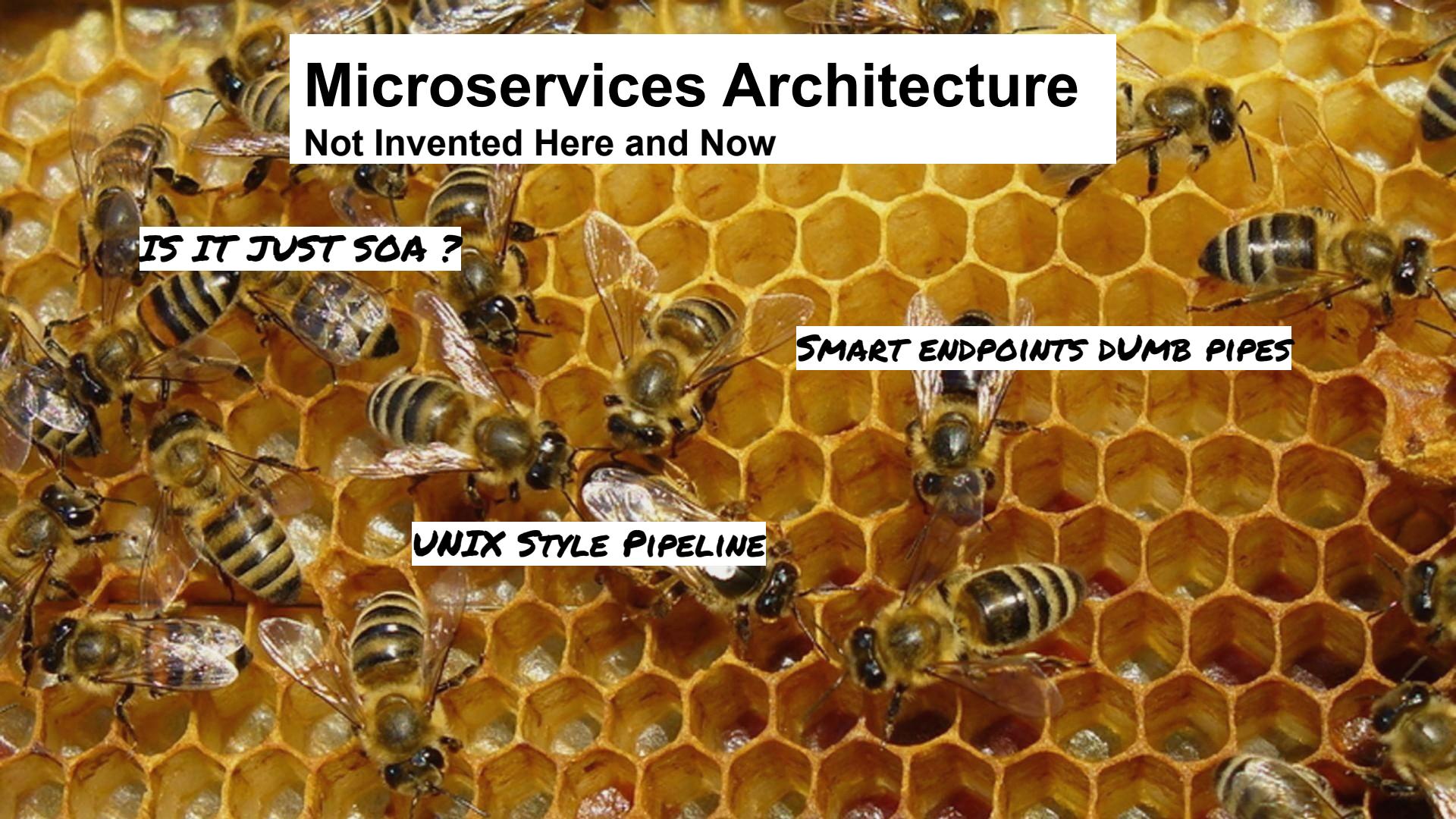
organisation change

Engineering Culture

A microservice is the responsibility of one team

Multidisciplinary teams





Microservices Architecture

Not Invented Here and Now

IS IT JUST SOA ?

SMART ENDPOINTS DUMB PIPES

UNIX STYLE PIPELINE

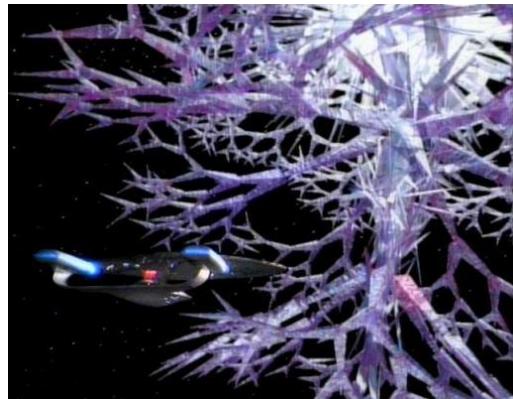
Why would microservices not be beneficial?

- A. If your organization is starting with microservices, rather than a monolith.
- B. Your organization is not successfully strategizing for such compartmental design.
- C. Members of a development team aren't as knowledgeable.
- D. All of the above.

Design characteristics of Microservices?

Design characteristics of Microservices

Single responsibility

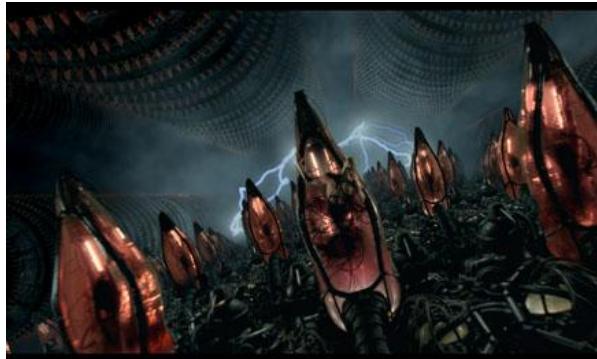


**Explicitly Published
Interface**



Design characteristics of Microservices

DURS and Lightweight Communication



Design characteristics of Microservices

Domain Driven Design



Polyglot

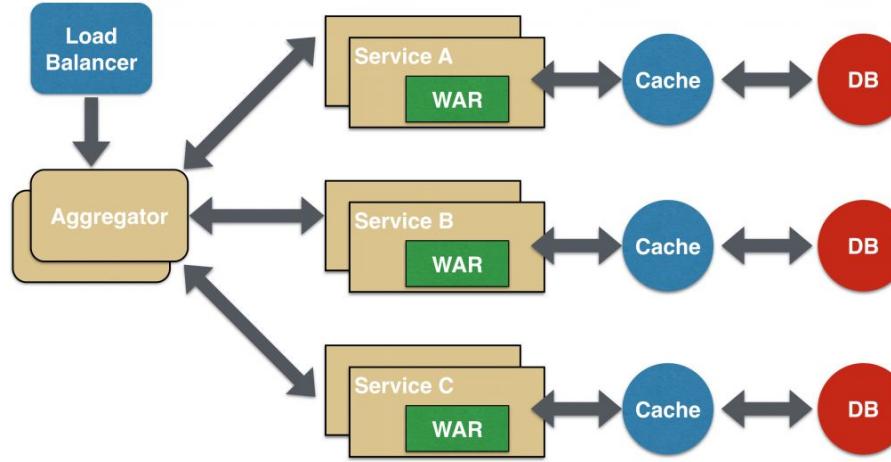


Why would a microservices architecture be beneficial?

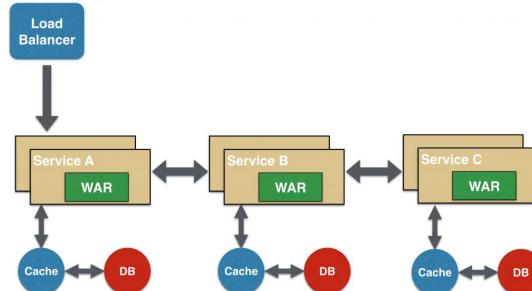
- A. A suite of services can be deployed and upgraded without a rebuild of other, functioning services.
- B. Less service interruption for fast-paced cloud computing.
- C. Different parts of a development team can manage individual parts of a microservice.
- D. All of the above.
- E. None of the above.

Microservice Design Patterns

Aggregator

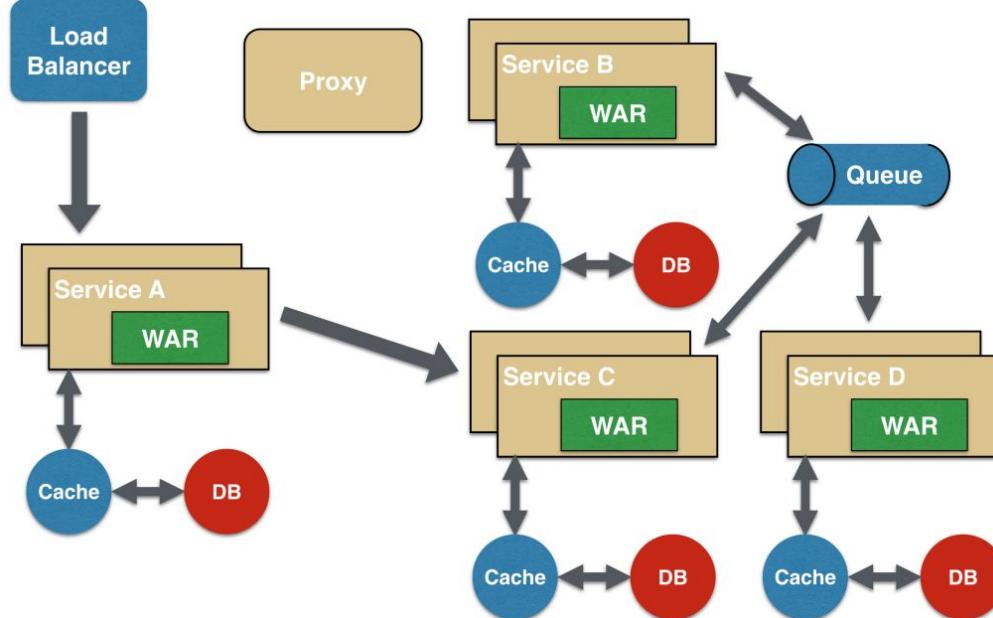


Chained



Microservice Design Patterns

Asynchronous Messaging

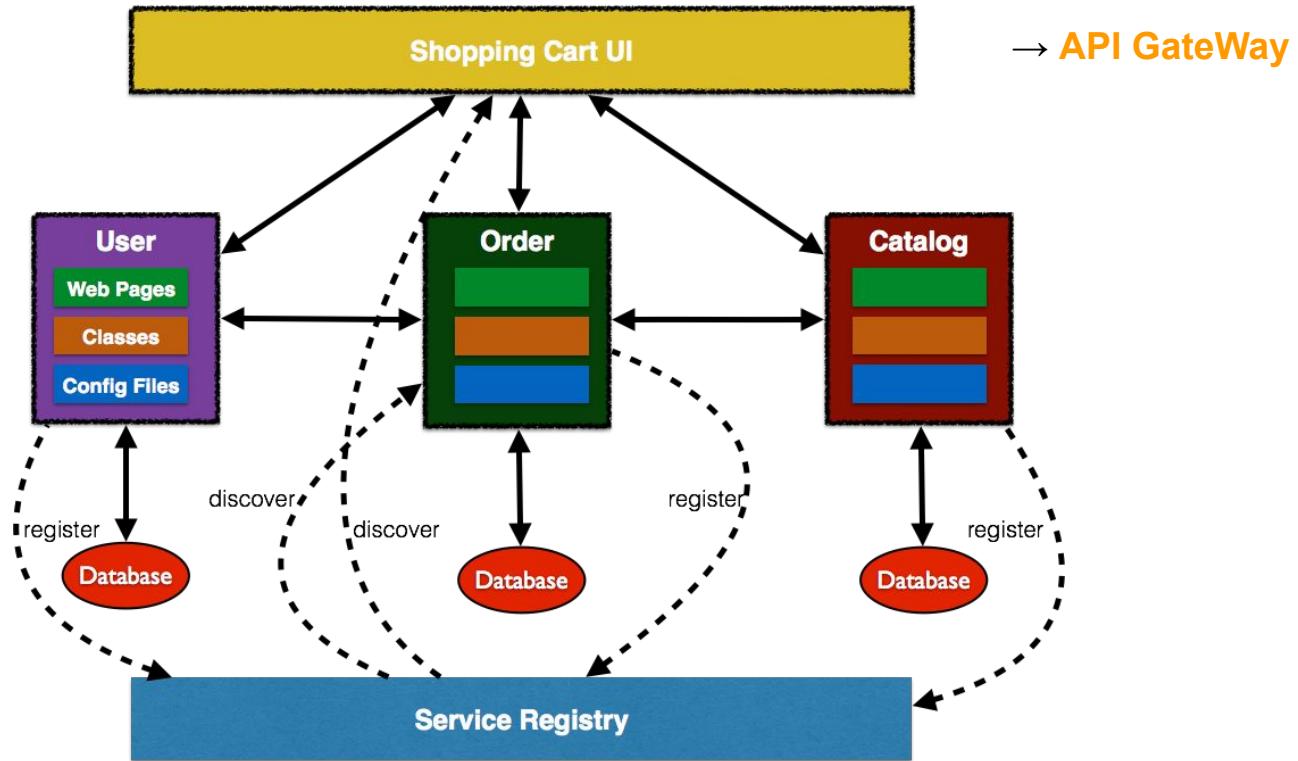




shutterstock.com • 248507947

Monolith versus Microservice

Microservice



How to convert a Monolith to Microservices?

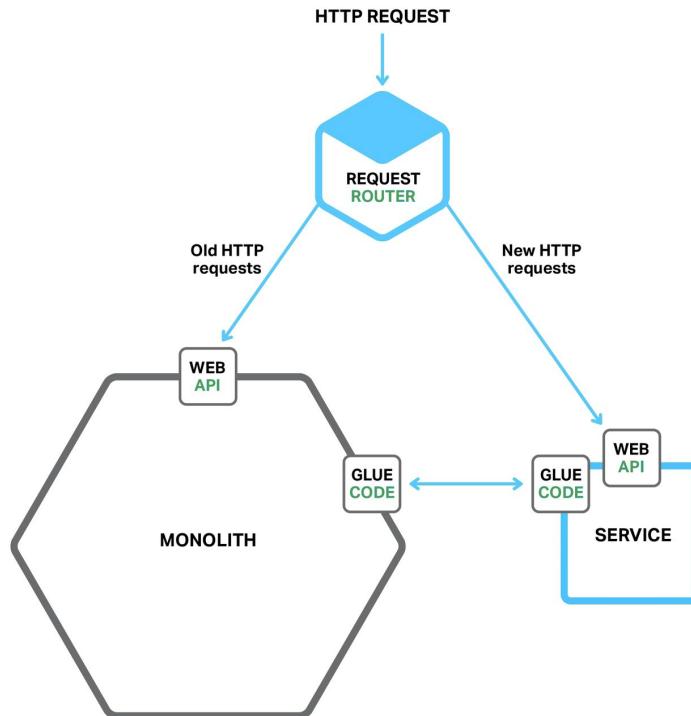
Strategies to convert a monolith to microservices

Stop Digging

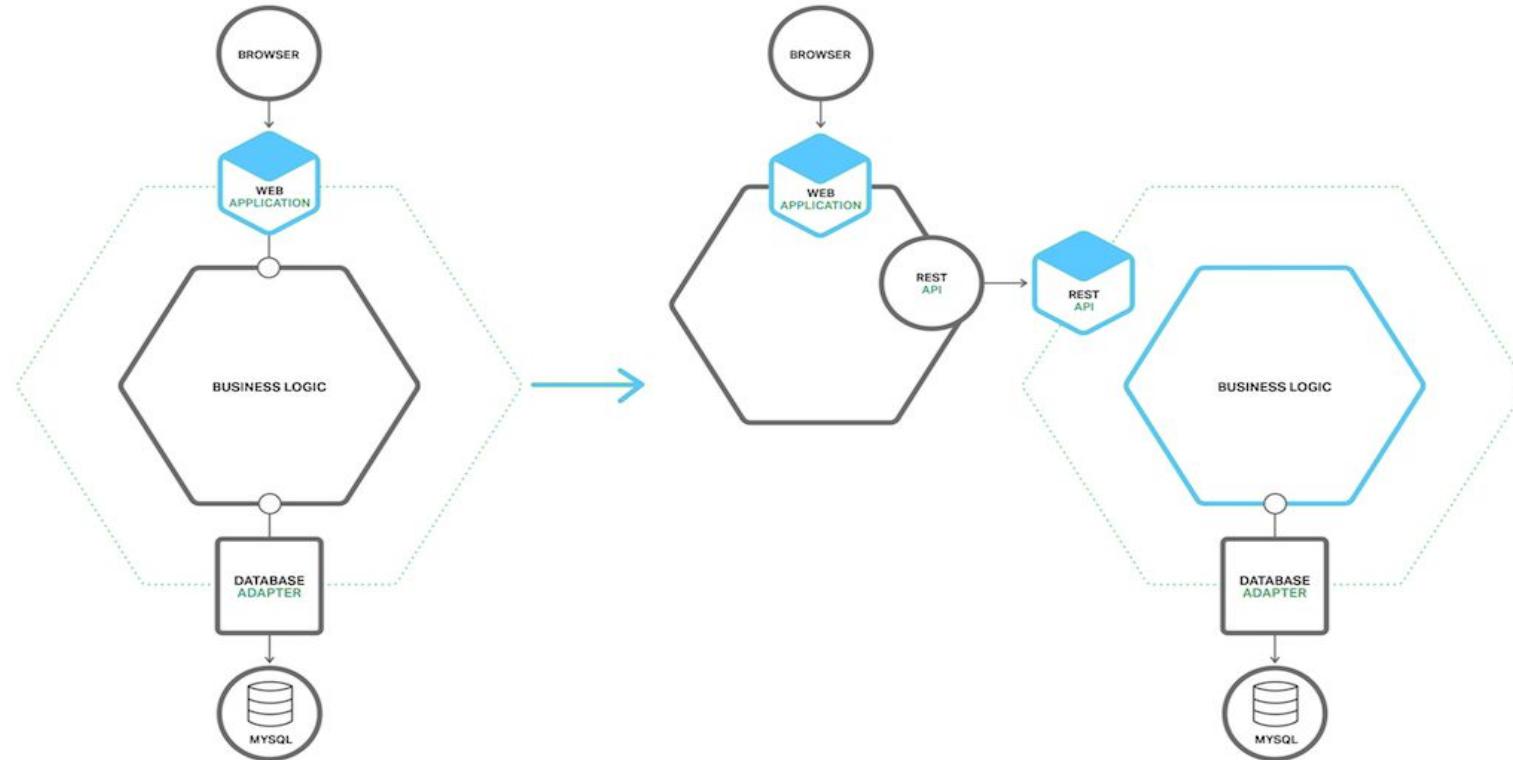
Split Frontend and Backend

Extract Services

Strategy 1 - Stop Digging

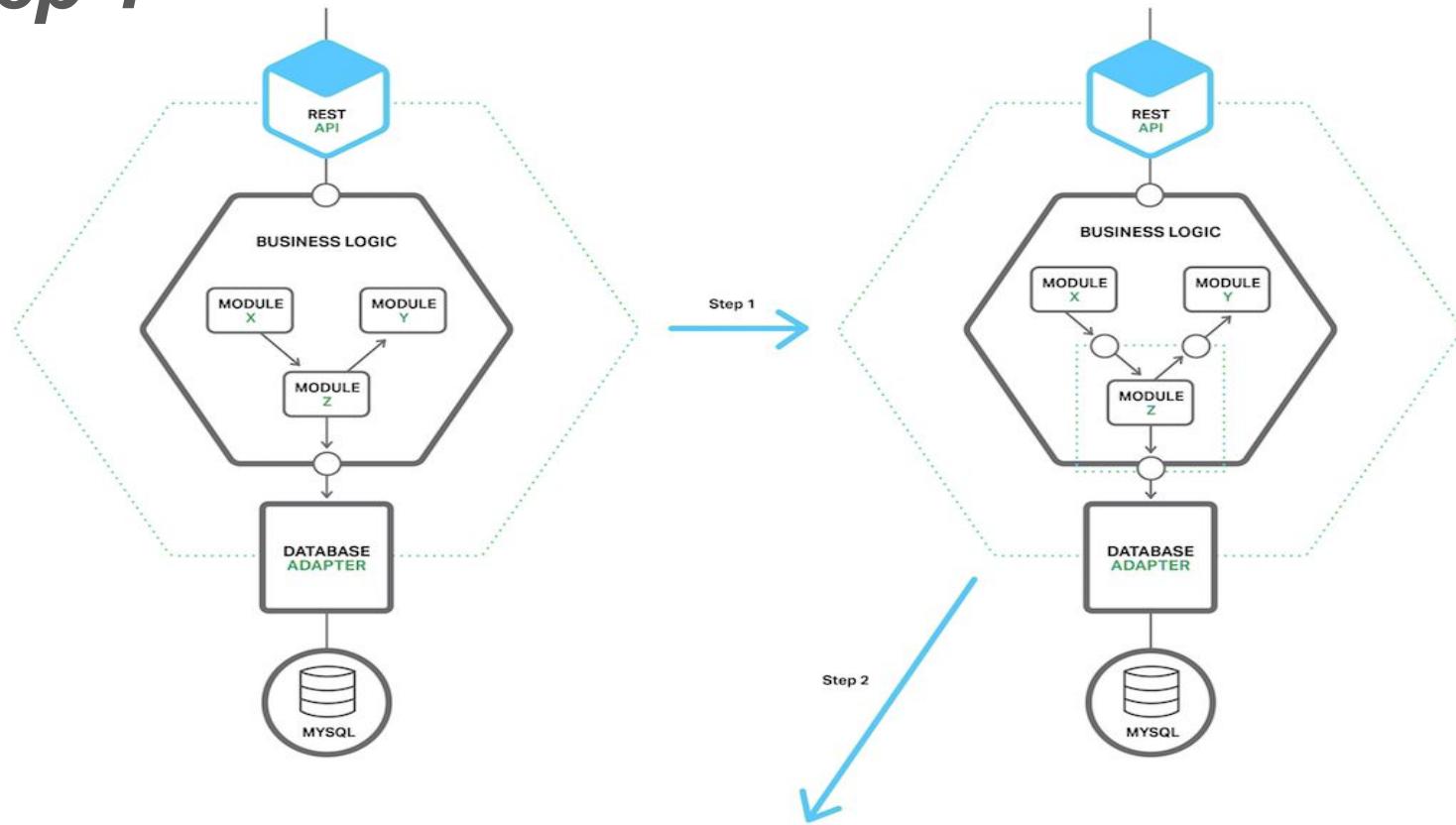


Strategy 2 - Split Frontend and Backend



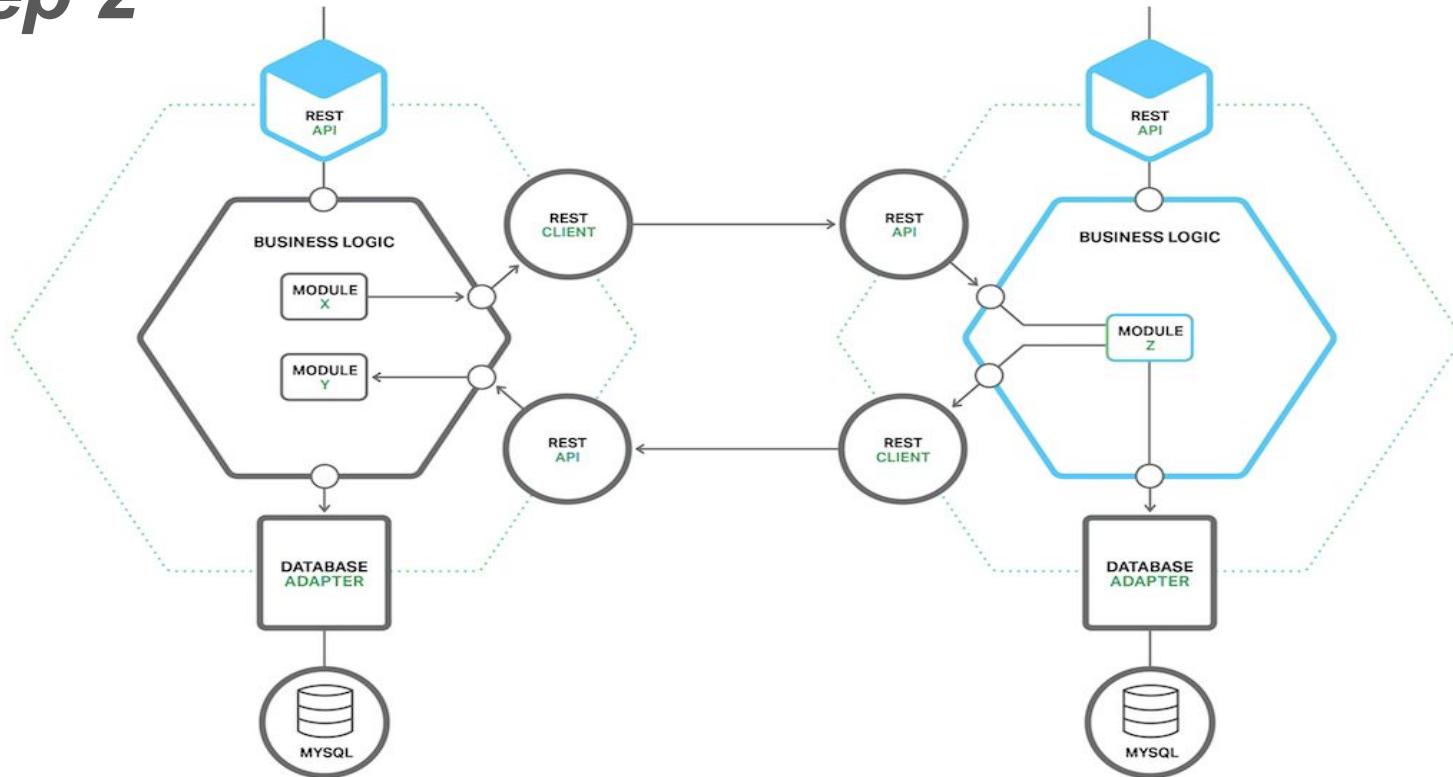
Strategy 3 - Extract Services

Step 1

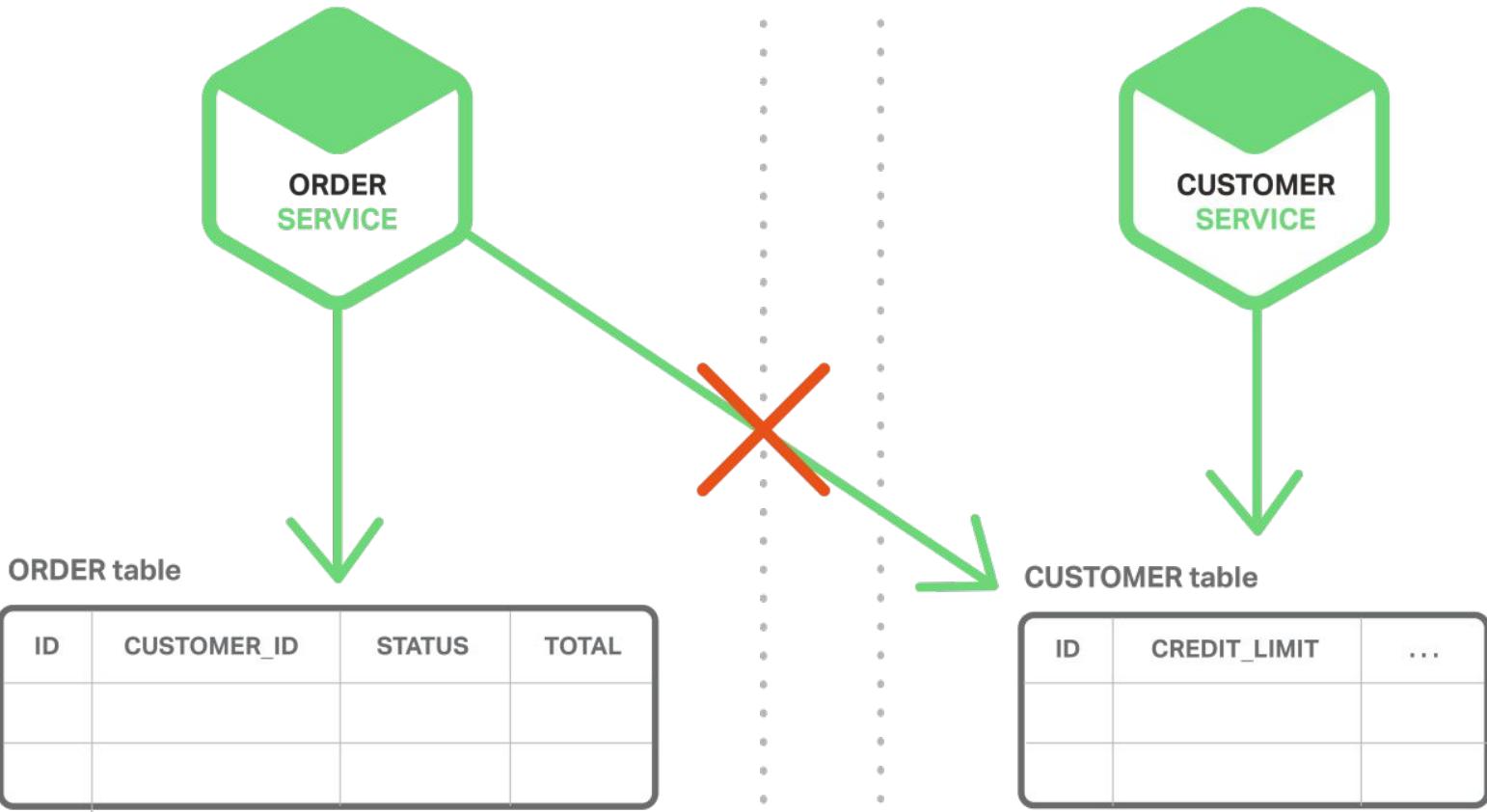


Strategy 3 - Extract Services

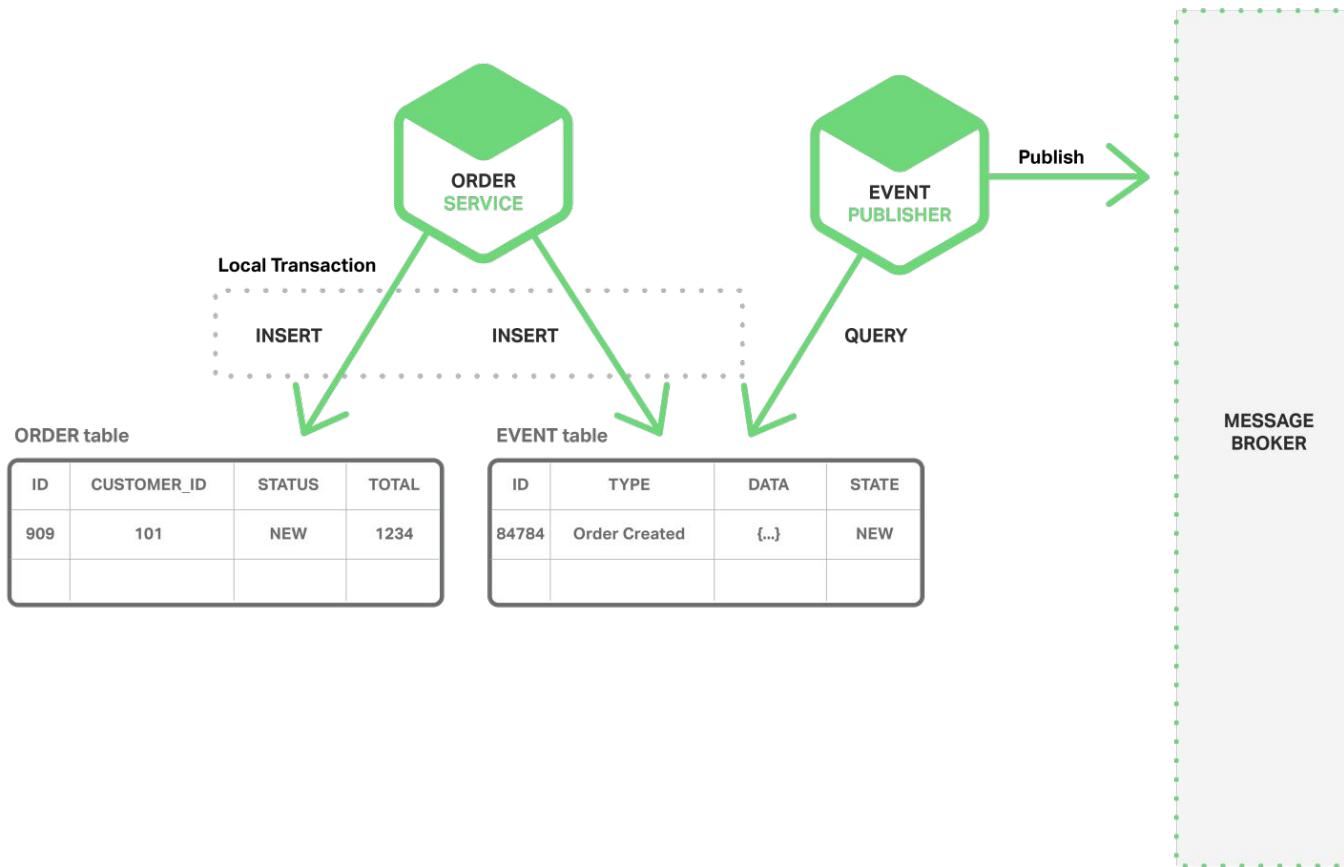
Step 2



Is it a good idea for Microservices to share a common database?



Publishing Events Using Local Transactions



**What are the drawbacks you face while
working with Microservice?**

Drawbacks of Microservices



www.r2-d2.de



Write code to handle partial failure

Update multiple databases owned by different services

Drawbacks of Microservices



Testing a microservices

Implementing changes

Drawbacks of Microservices



API Changes of a service affects all callers of this service

Deploying multiple services

Drawbacks of Microservices



Complex eco-system

Reorganization

Need Micro Frontends?

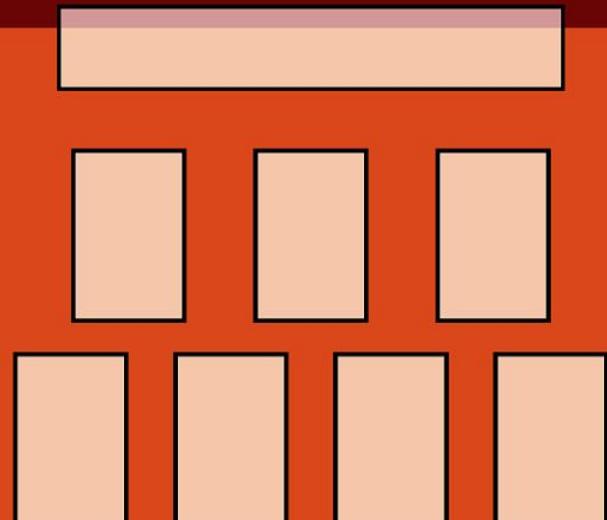
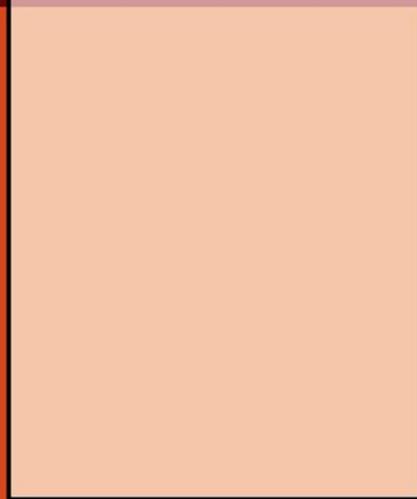
Frontend Monoliths

growing complexity
no scaling

Frontend



Backend



The Monolith

Front- & Backend

Microservices

Micro Frontends Definition

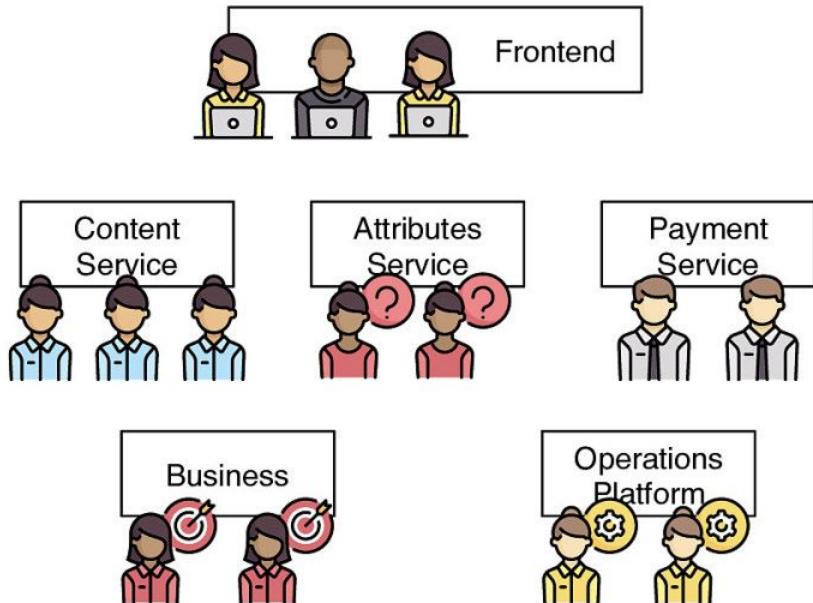
... Our preferred (and proven) approach is to split the browser-based code into micro frontends. In this approach, the web application is broken down into its features, and each feature is owned, frontend to backend, by a different team. This ensures that every feature is developed, tested and deployed independently from other features. ...

ThoughtWorks Technology Radar

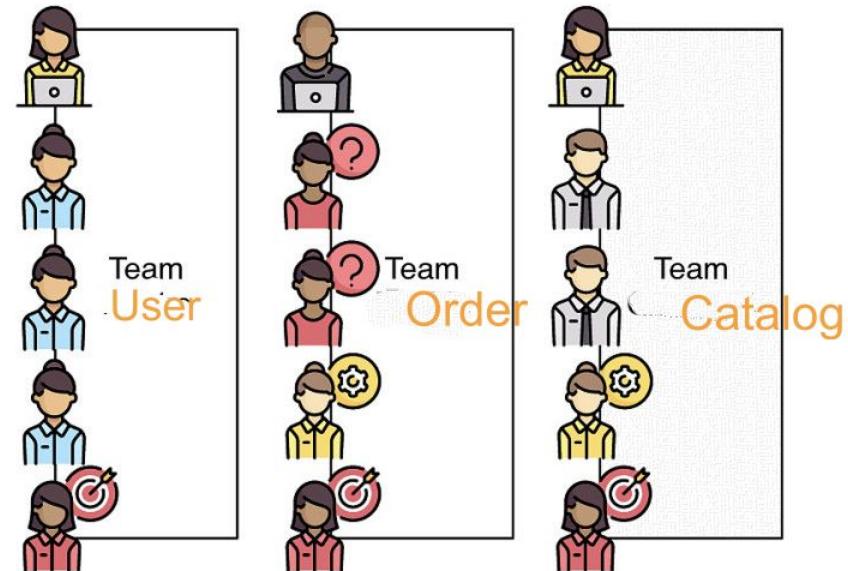
<https://www.thoughtworks.com/de/radar/techniques/micro-frontends>

Stages: Access (2016) > Trial (2017) > Adopt ★ (2019)

Specialist Teams



Cross Functional Teams



grouped around a **skill or technology**

grouped around a **use case or customer need**

Development speed



Development speed

Customer focus

Direct feedback loop

Cross functional teams

Specialist teams per feature

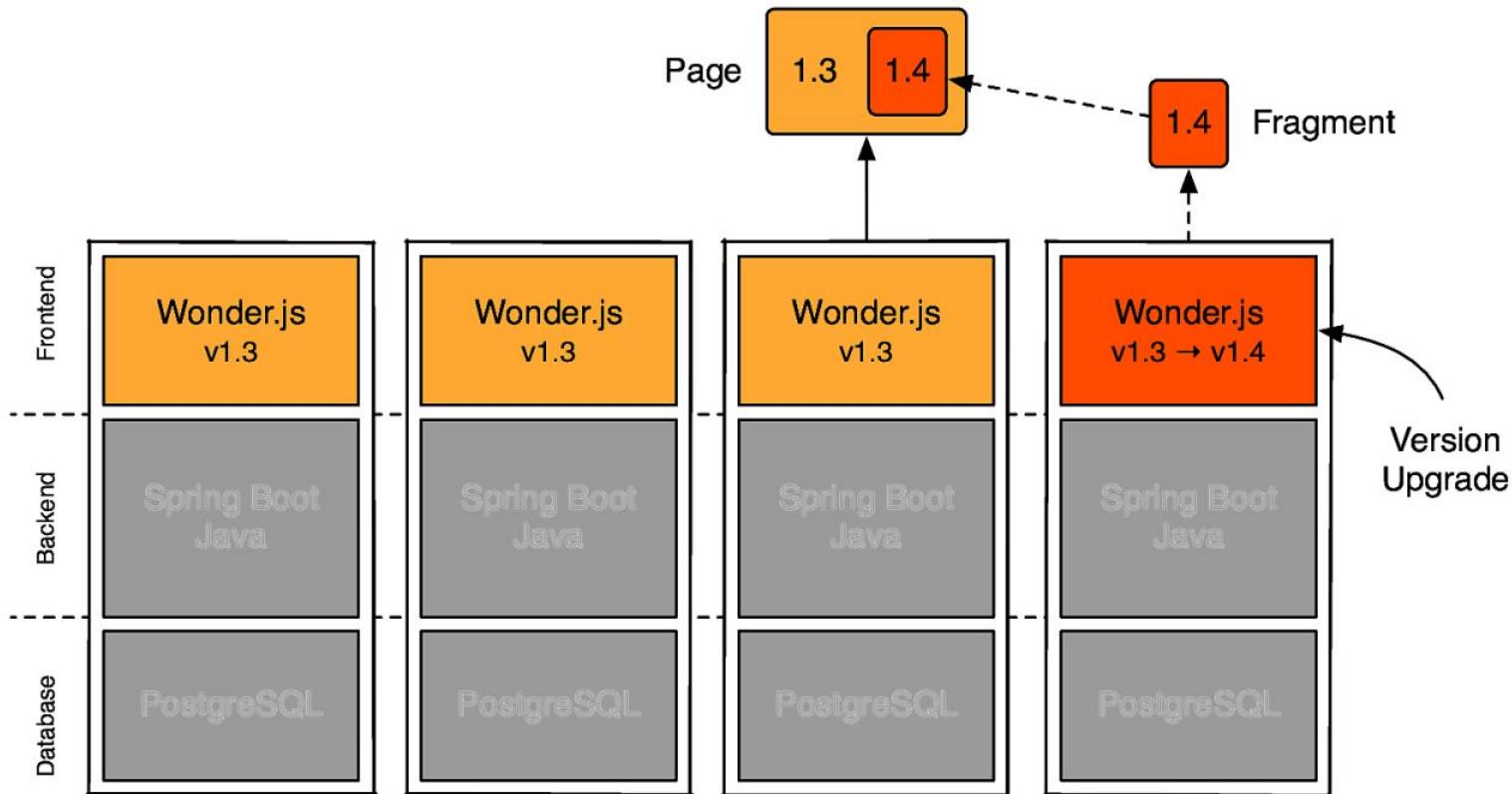
Reduced scope: *everything fits into my memory*

Ability to change frameworks FE and BE

Change versions feature

Independent deployments

... or upgrade at your own pace.





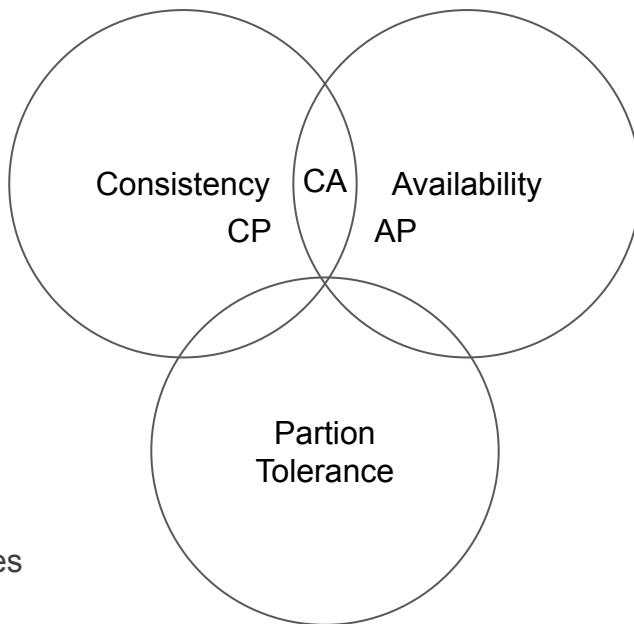
shutterstock.com • 41227285



Questions?



Data Consistency



The [CAP theorem](#) requires you to choose between availability and ACID-style consistency, and availability is usually the better choice.