Container Bootcamp

What are Microservices?

INNOQ

UNIX Philosophy

Unix

- Write programs that do one thing and do it well
- Write programs to work together
- Write programs with a common interface

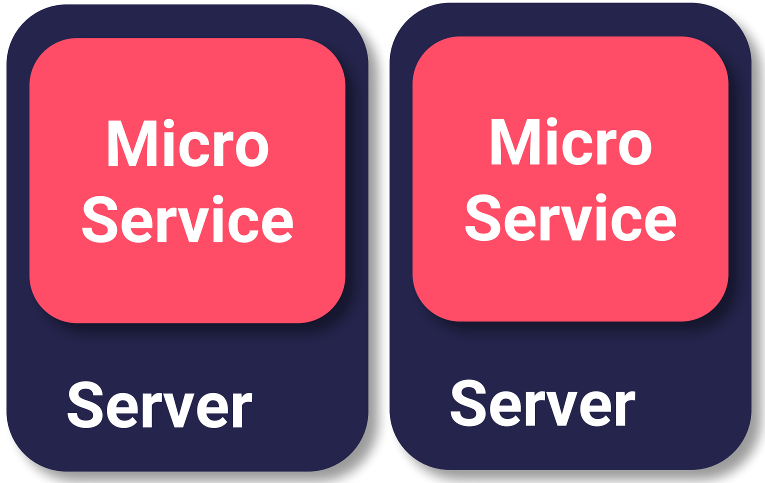
Microservices: Definition

- Technology for Modularization
- Module = independent deployment units

- Independent data handling
- i.e. no shared schema / data types

Microservices: Definition

- Any technology
- Any infrastructure
- Includes additional services (e.g. database)
- Communication via network

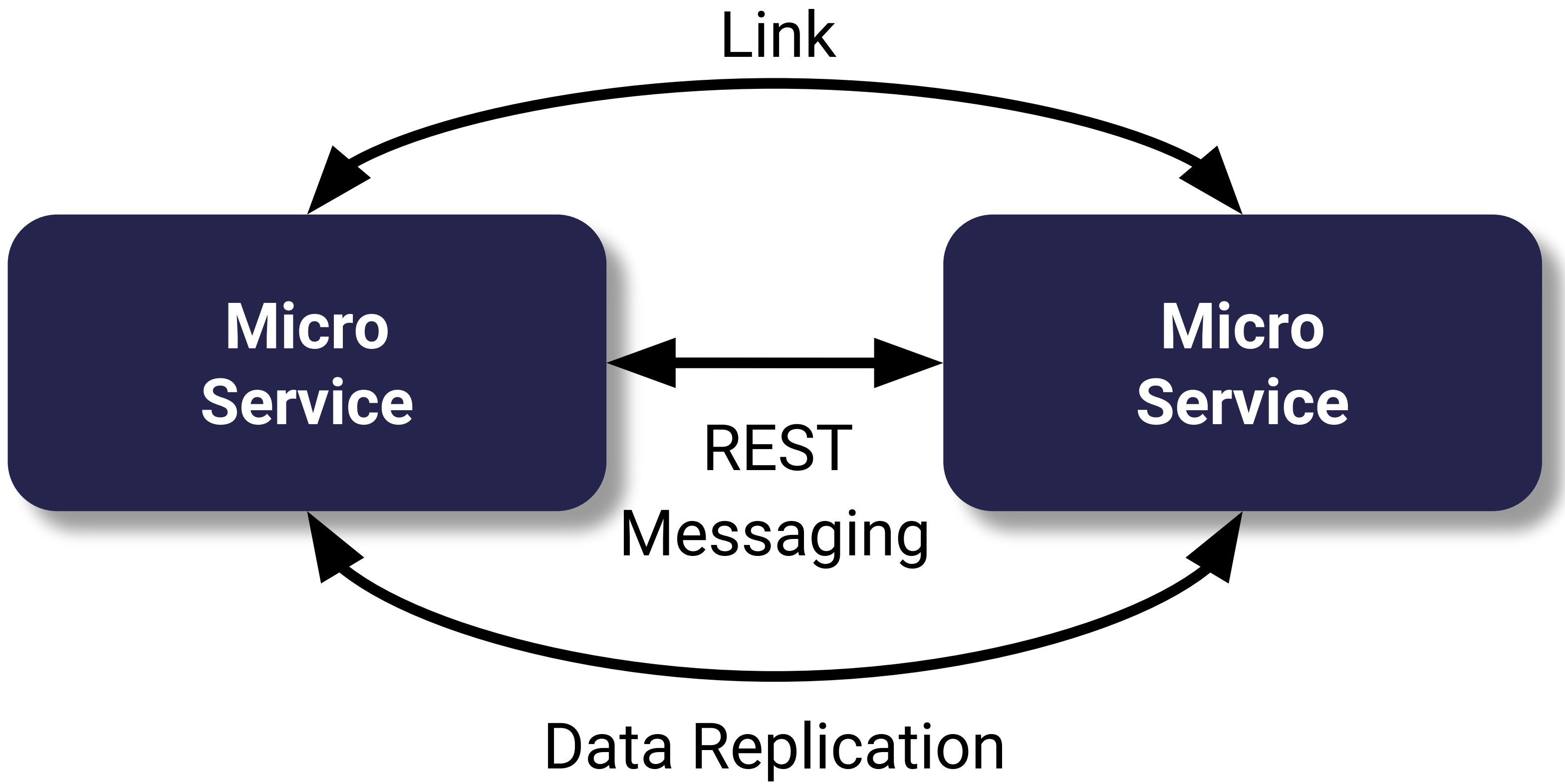


Microservices

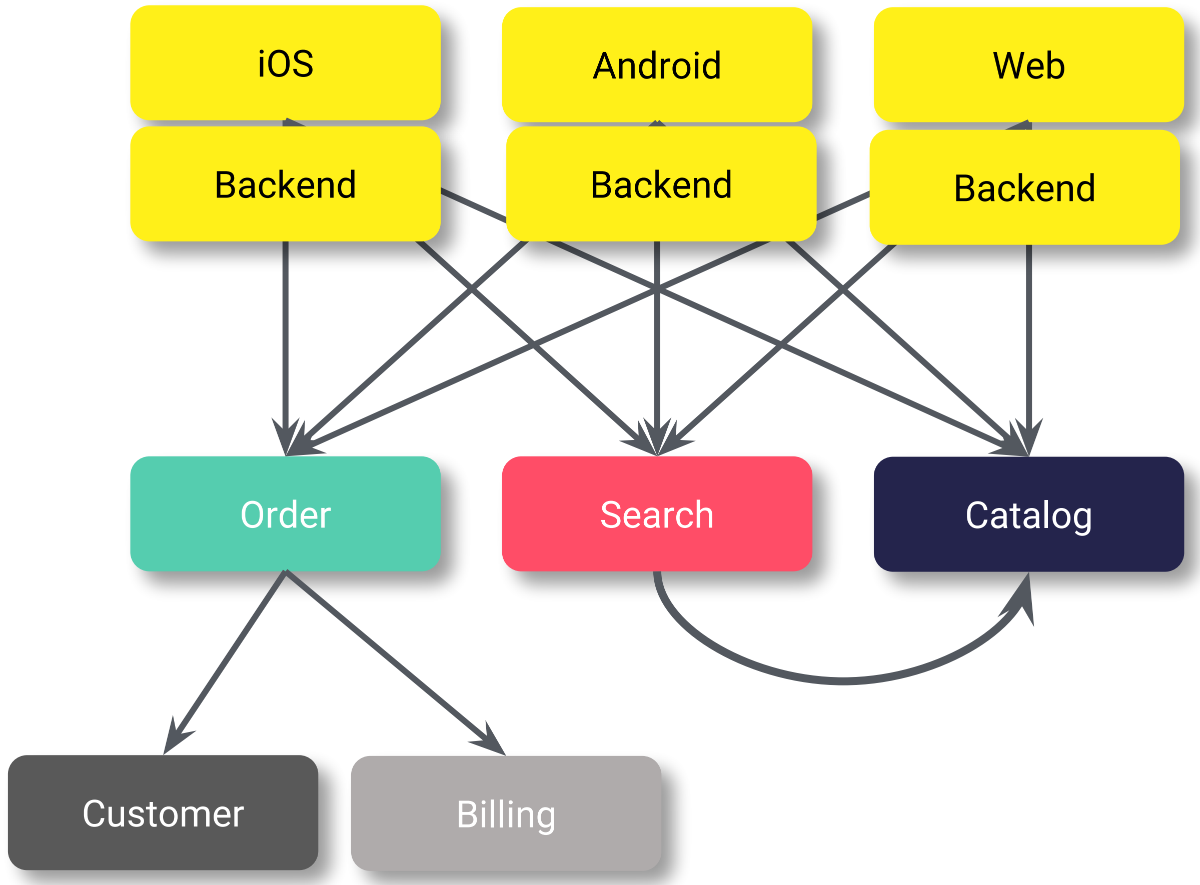
- **Component Model**

- **Component…**
  - Individual deployment unit
  - Separate process / VM
  - GUI+Logic

Components Collaborate



Layered


iOS
Android
Web
Backend
Backend
Backend
Order
Search
Catalog
Customer
Billing

Layered

- **Reusable Backend Services**
- **Mobile client / Web App as frontend**
- **Backend for frontend (BFF): Custom backend services**
  - ...to implement frontend specific logic
- **E.g. Netflix**

Layered: Benefits

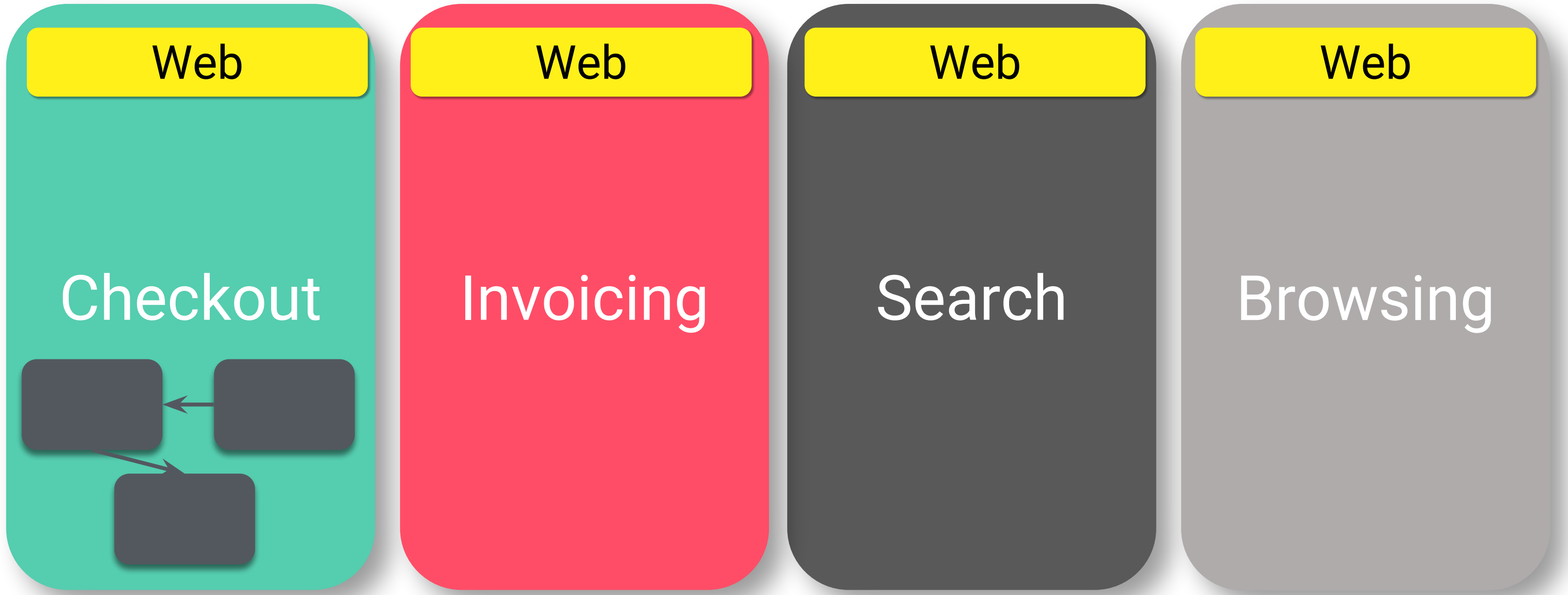
- Good way to build an API
- E.g. for mobile

- Might be easier to migrate into
- …if existing architecture is similar

Layered: Issues

- BFF might contain the same logic – same processes
- Processes are the most relevant logic
- Changing a business process cause changes in many services – BFF, Frontend, backend
- Lots of communication between teams and components

Self-contained Systems



Self-contained Systems (SCS)

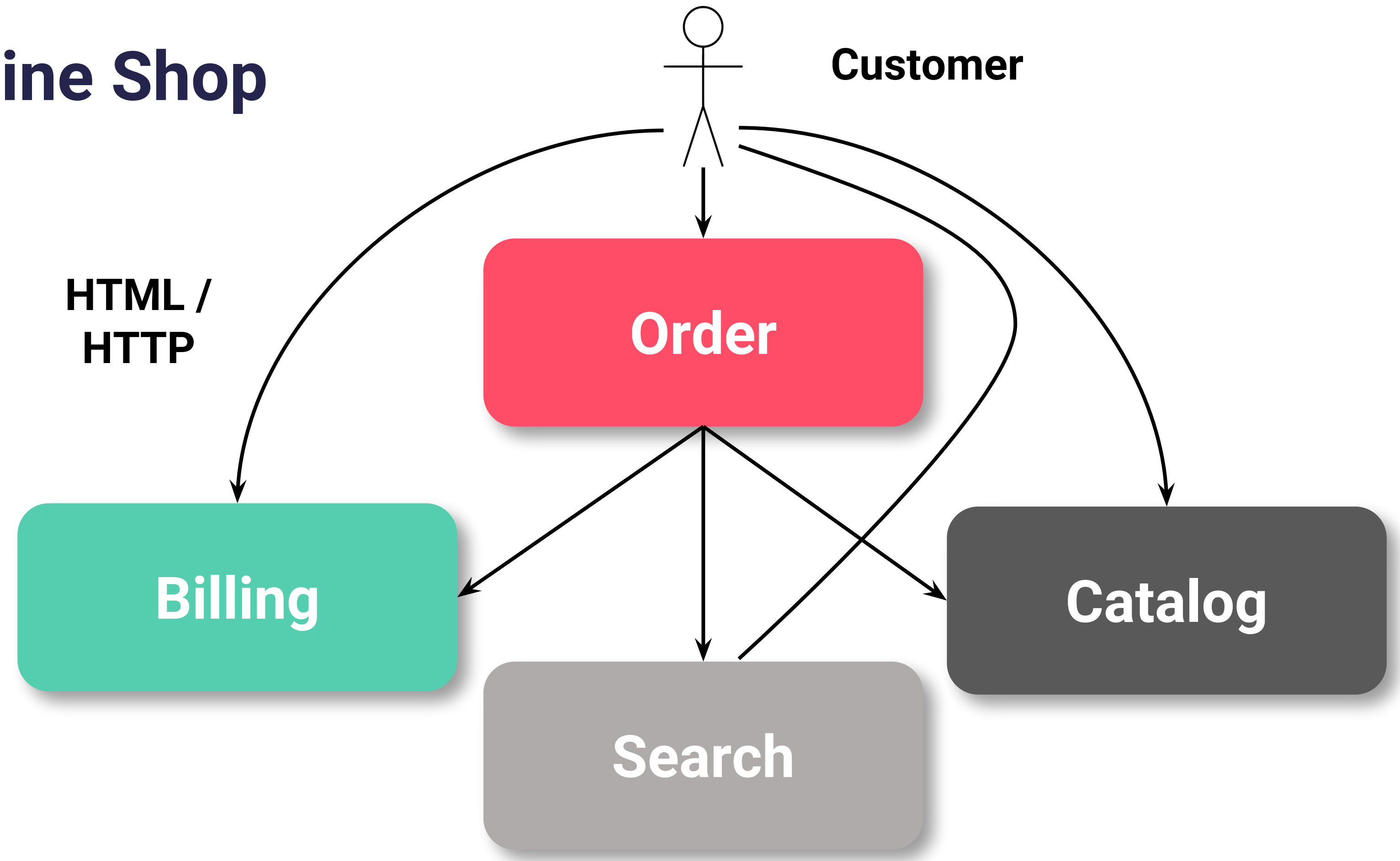
- SCS: Autonomous web application
- Optional service API (e.g. for mobile clients)
- Includes data & logic
- Might contain several microservices
- No shared UI
- No shared business code
- E.g. Otto, Kaufhof, Kühne + Nagel ...

SCS: Benefits

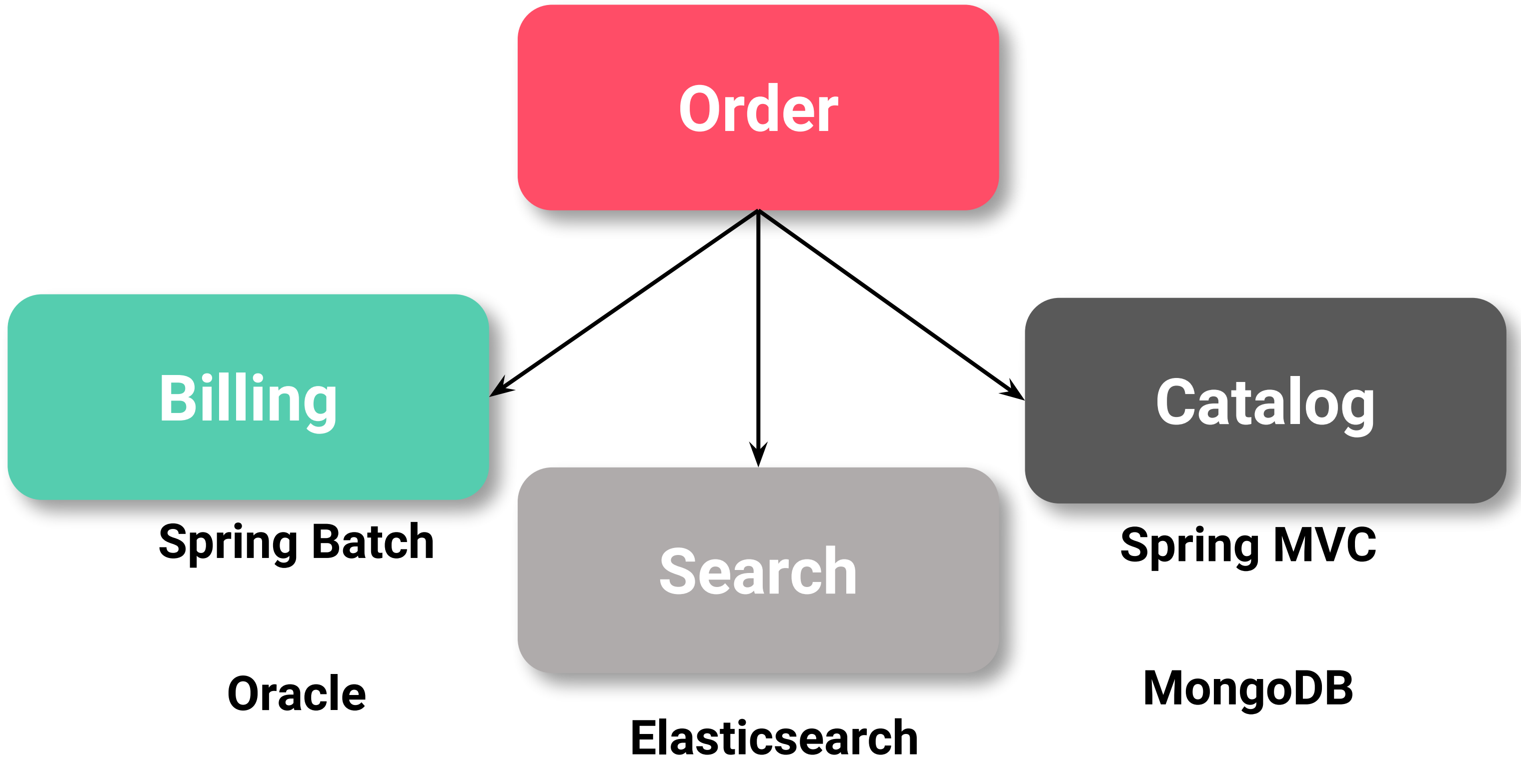
- Business logic for one domain in one SCS
- Change usually local to one SCS
- Less communication between SCS

- I think this should be the goal

Online Shop



Online Shop


Order
Billing
Spring Batch
Oracle
Search
Elasticsearch
Catalog
Spring MVC
MongoDB