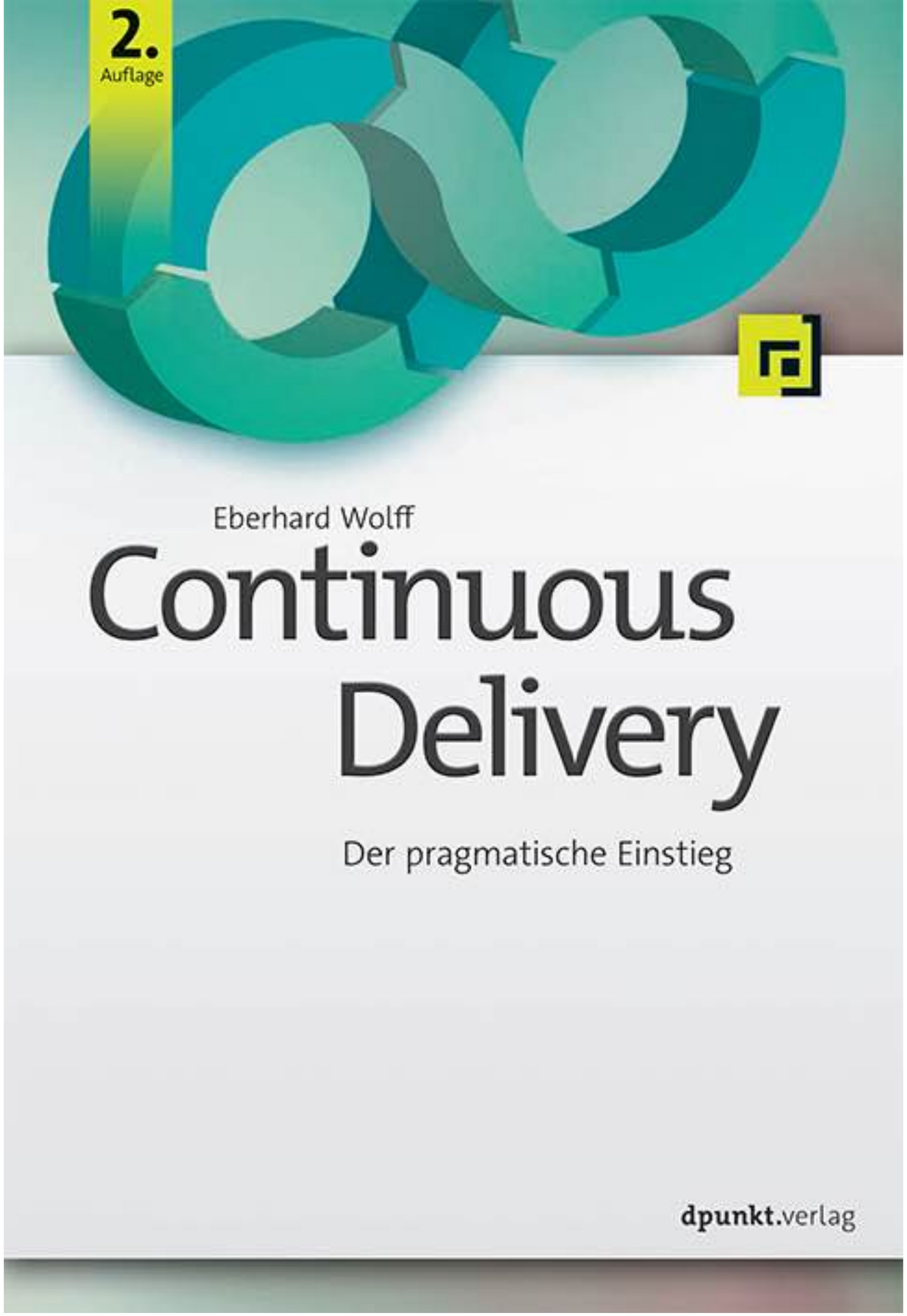
Microservices & Self-Contained Systems to Scale Agile

Eberhard Wolff
Fellow, innoQ
@ewolff



http://continuous-delivery-buch.de/

Eberhard Wolff

Microservices

Grundlagen flexibler Softwarearchitekturen

dpunkt.verlag

http://microservices-buch.de/

Microservices

Flexible Software Architectures

Eberhard Wolff

http://microservices-book.com/

Eberhard Wolff

Microservices Primer

A Short Overview

FREE!!!!

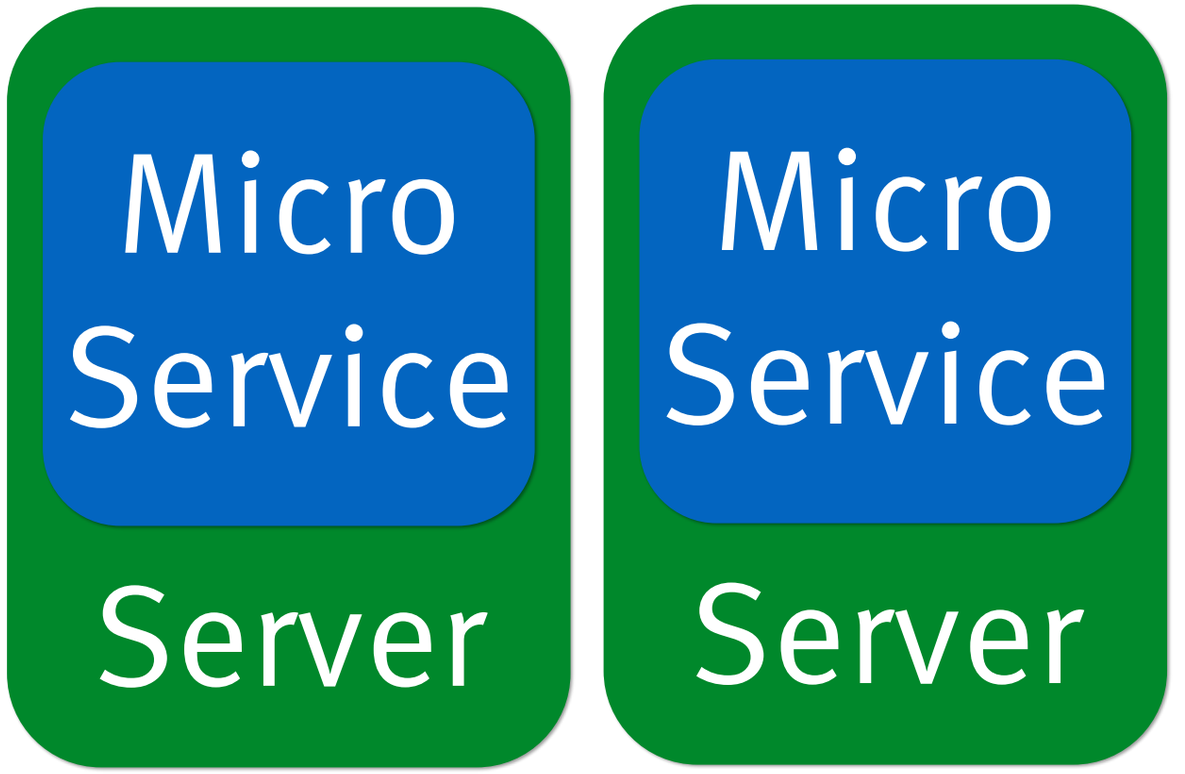
innoQ

http://microservices-book.com/primer.html

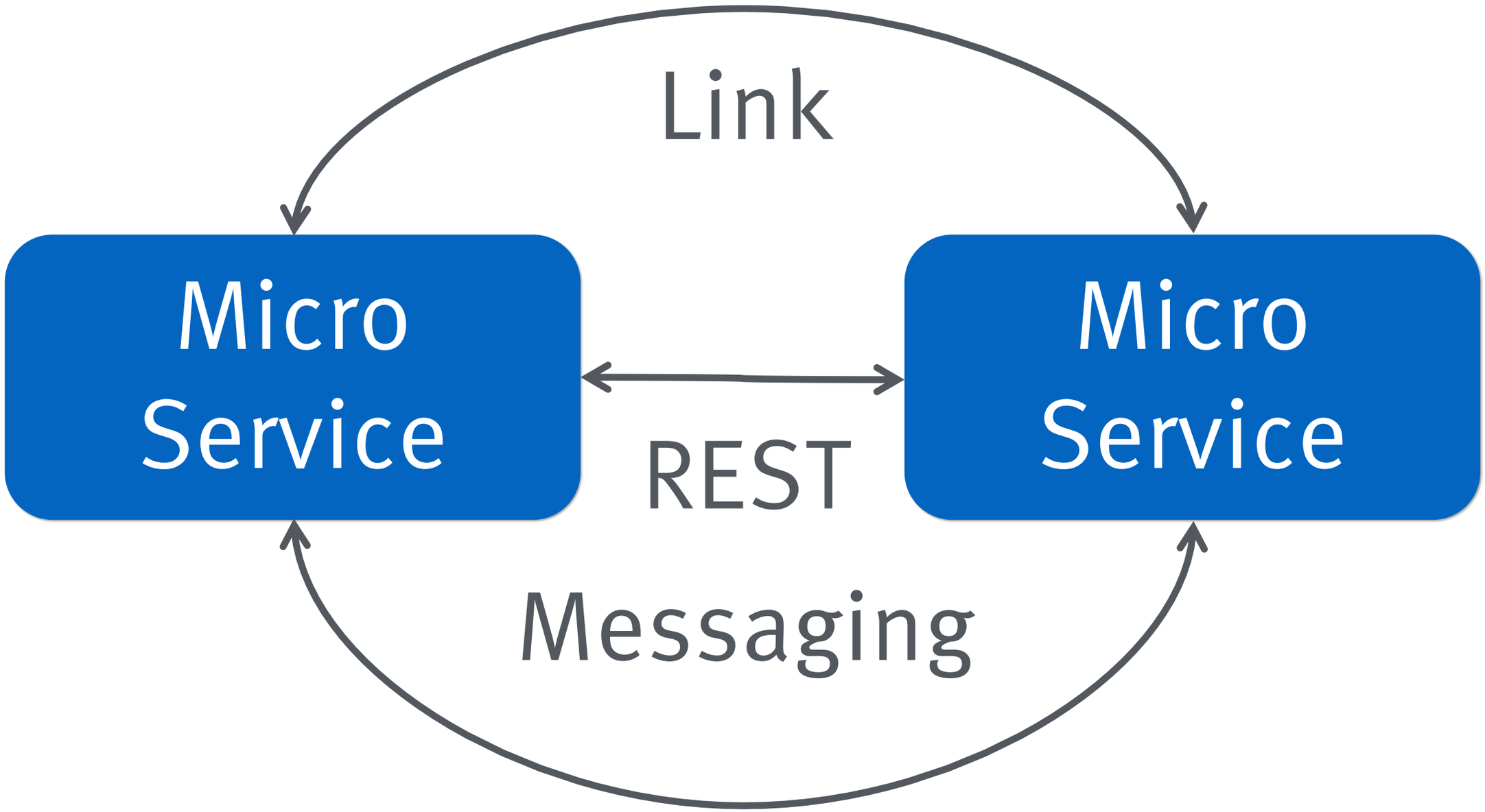
Microservices

Microservices: Definition

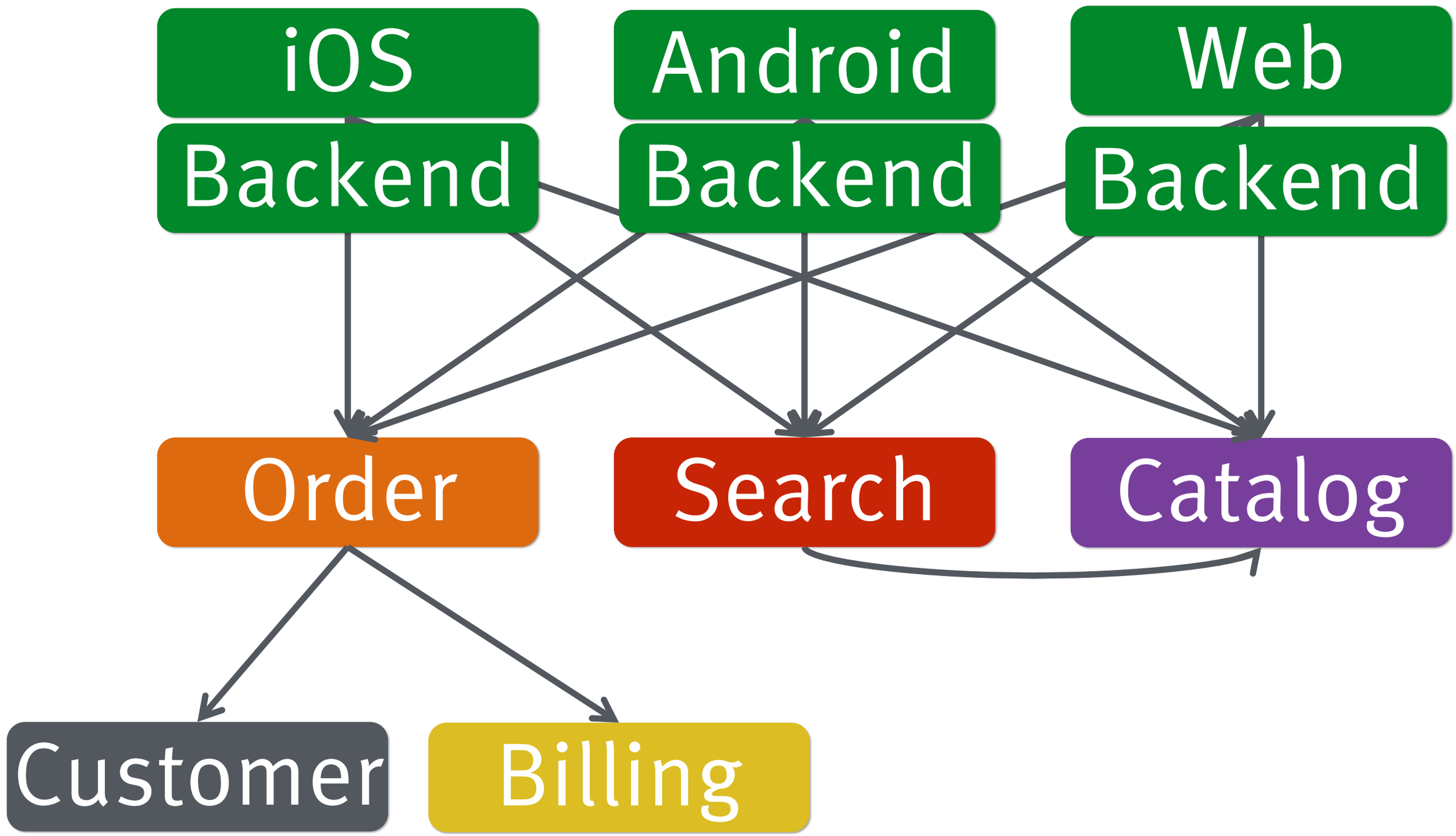
- Independent deployment units
- E.g. process, VMs, Docker containers
- Any technology
- Any infrastructure



Components Collaborate



Layered Microservices



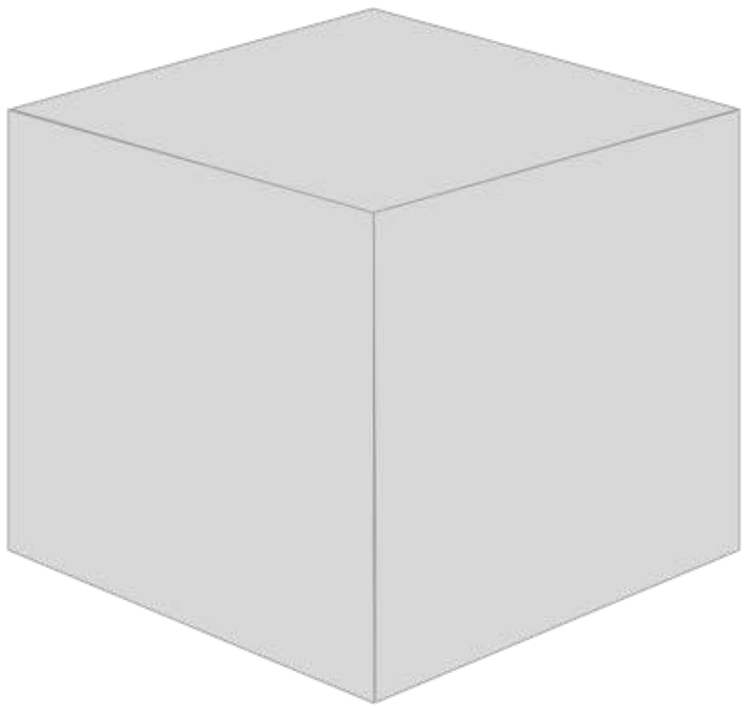
Layered Microservices

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

Layered: Issues

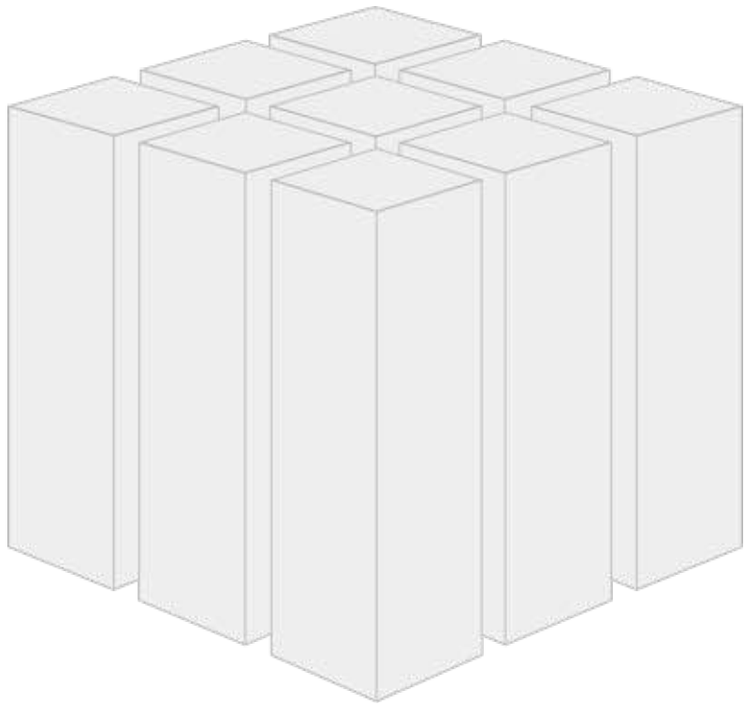
- BFF might contain the same logic – same processes
- BFF might contain most relevant logic
- Change to a business process means changing many services
- Lots of communication

Self-contained System

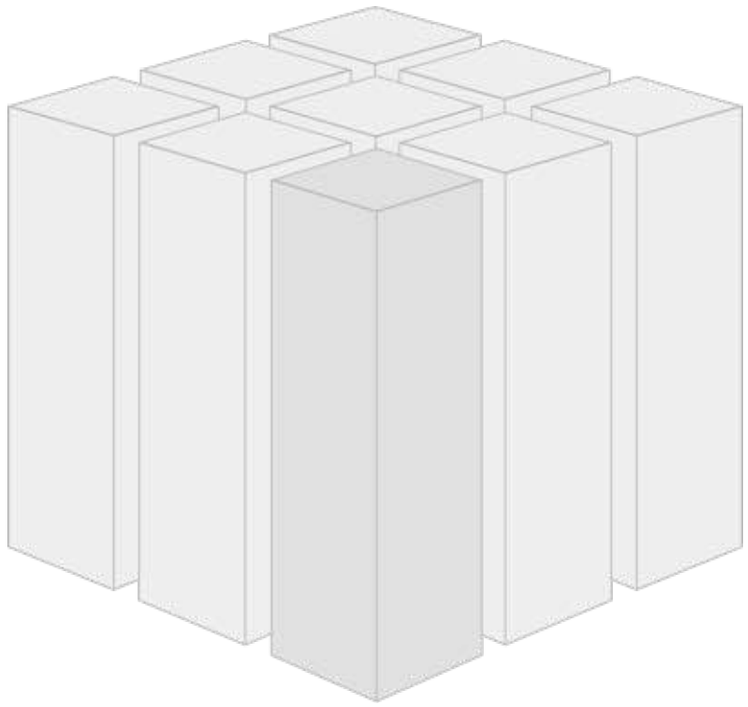


Deployment monolith

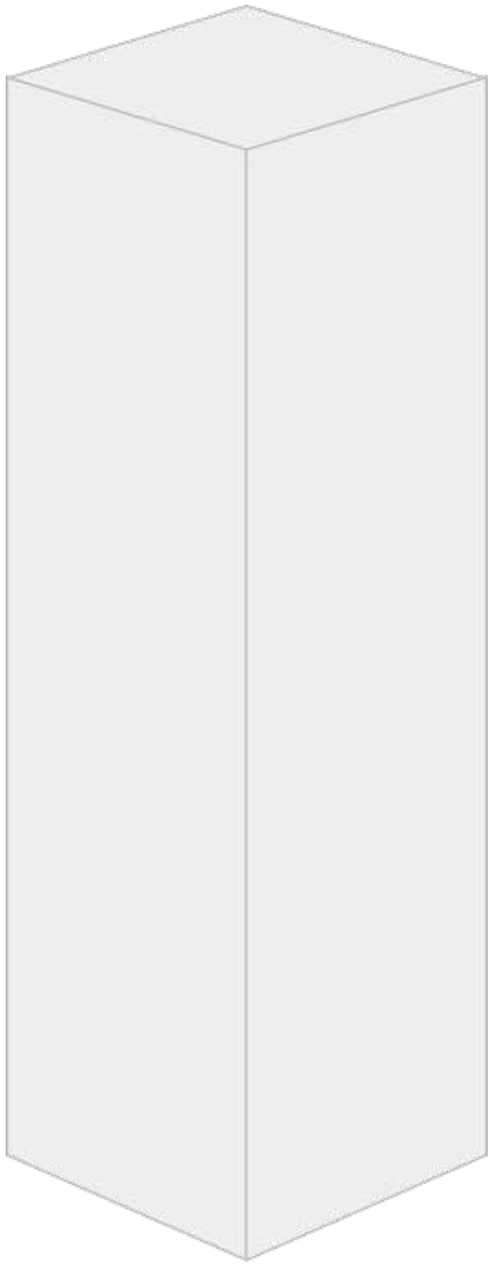
Graphics by Roman Stranghöhner, innoQ
http://scs-architecture.org



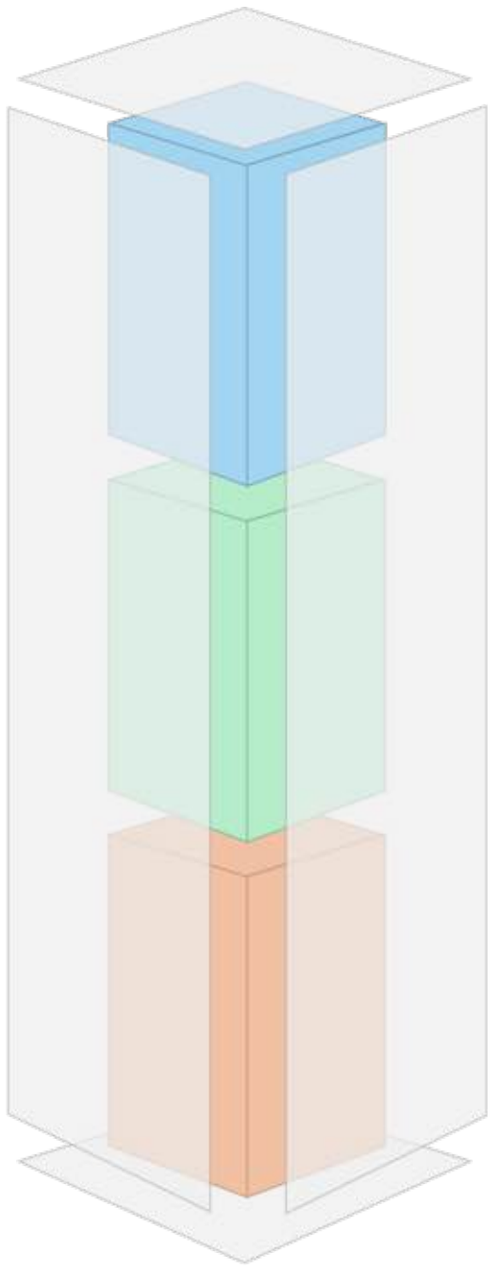
Cut Deployment monolith along domains …



... wrap domain in **separate** web application ...



Self-contained System (SCS) – individually deployable



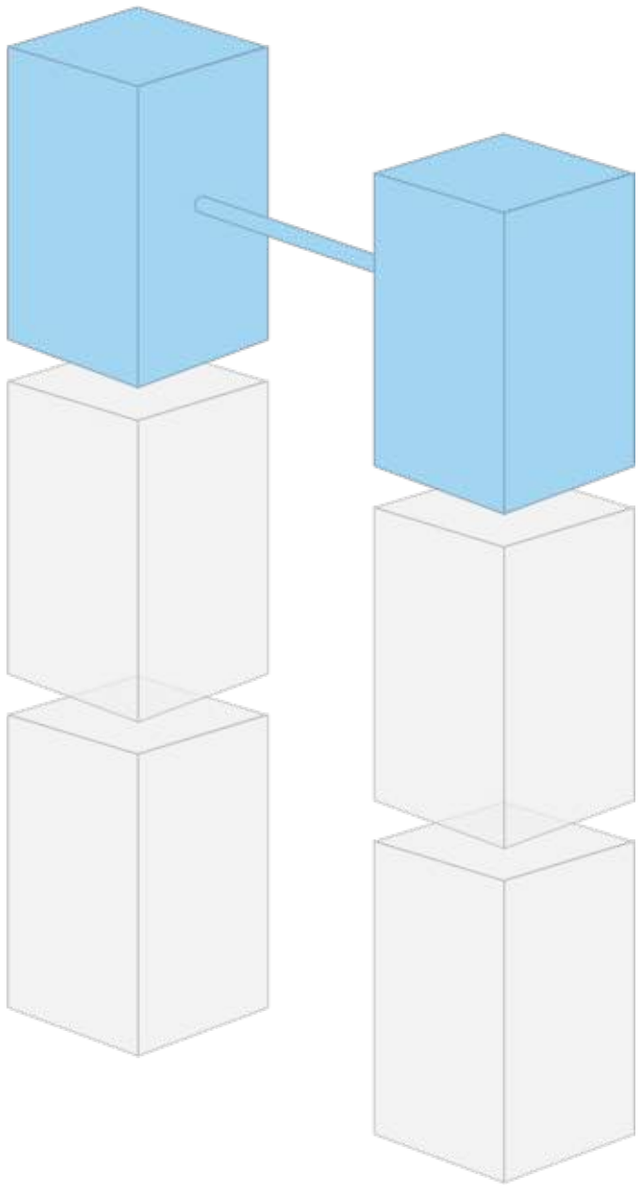
SCS =

user interface+

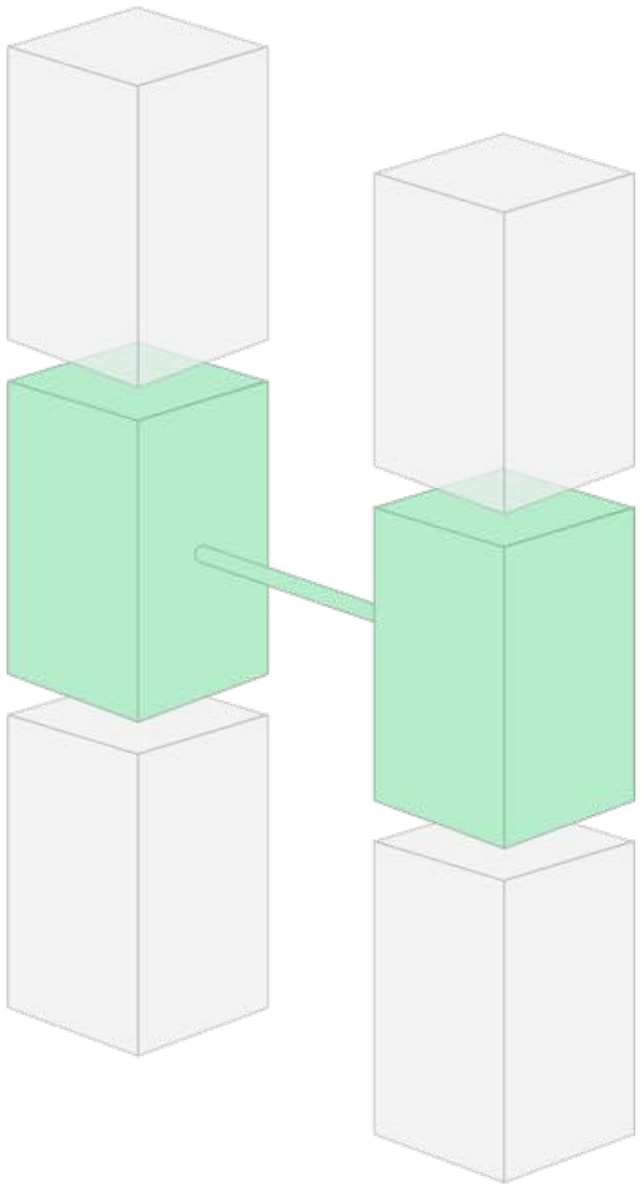
business logic+

data storage

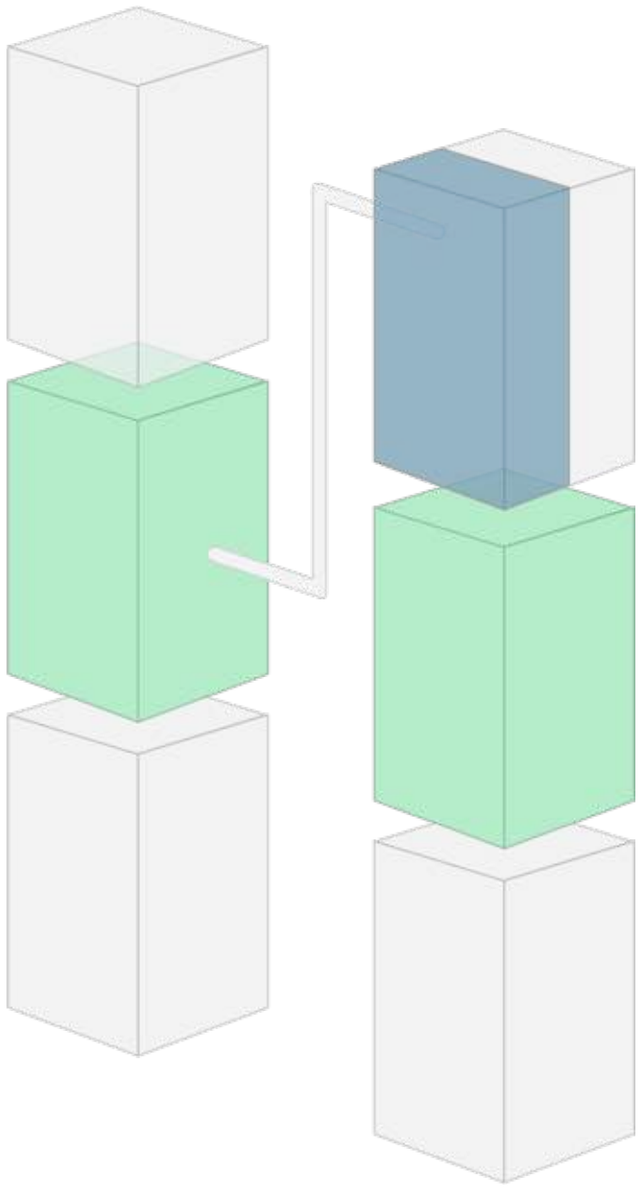
Each SCS contains all layers.



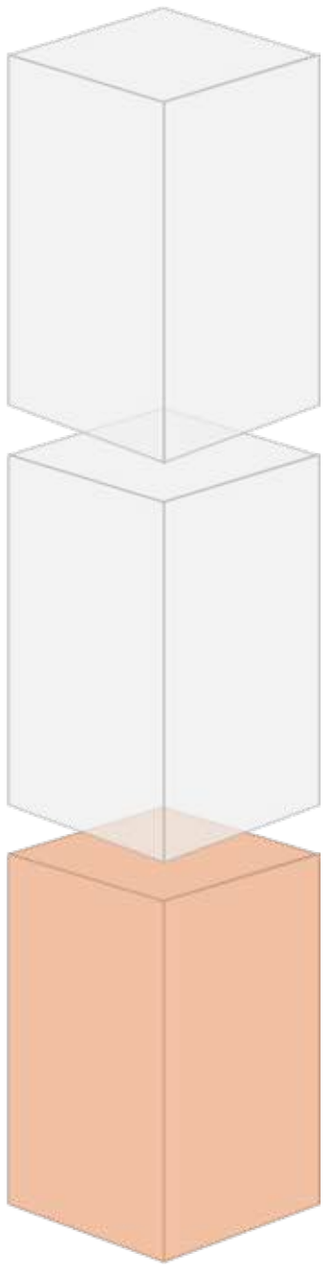
Self-contained Systems should be integrated in the **web interface**



Synchronous remote calls inside the business logic should be **avoided**.



Asynchronous Remote calls reduce dependencies and prevent error cascades.



Every SCS brings its **own data storage** with its own (potentially redundant) data

Domained scoped SCS enables the development, operation and maintenance of an domain by a **single team**.

1 SCS

= 1 Domain

= 1 Web App

= 1 Team

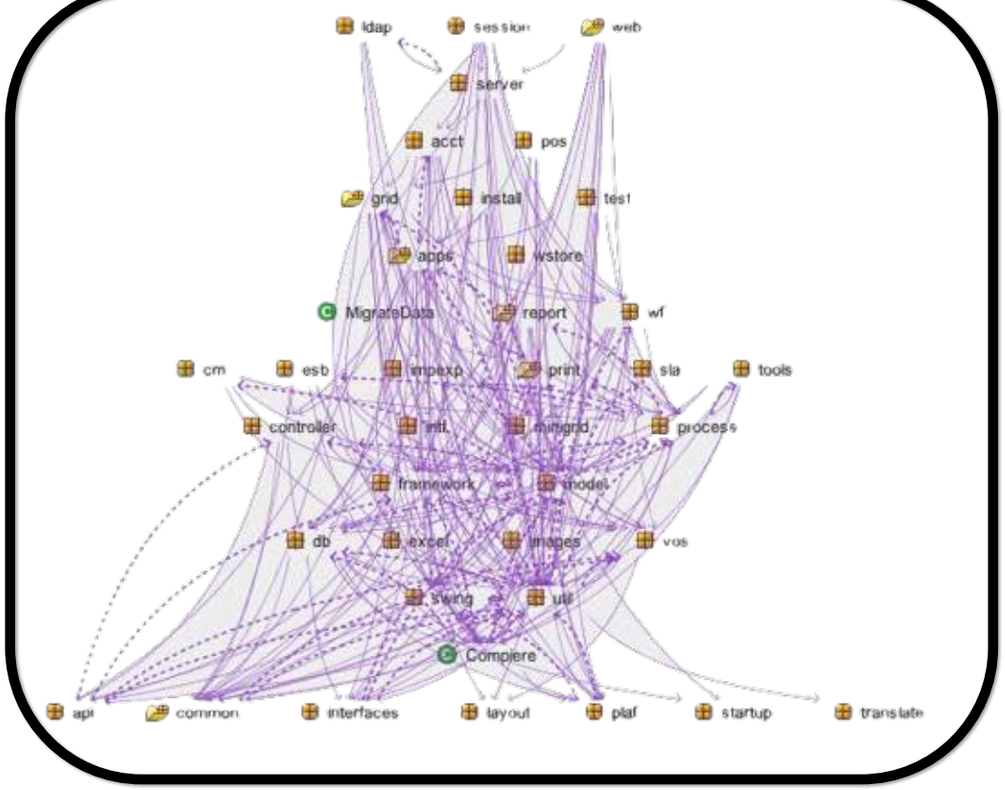
= 1-n Microservices

Why create such a complex system?

Legacy Apps

HTTP

Links



New Stuff

No legacy code

Any technology

Small code base

Sustainable Development

Monoliths

- Architecture rot
- ...not maintainable any more
- ...and can't be rewritten / replaced

Microservices

- Distributed system of small units
- Architecture violations harder
- Architecture won’t rot

- Small units
- Easy to replace
- Service rotten? Replace!

Robust

- One failing Microservice doesn't crash the system
- However, Microservices must deal with failing Microservices

Continuous Delivery

Principles behind the Agile Manifesto

Our highest priority is

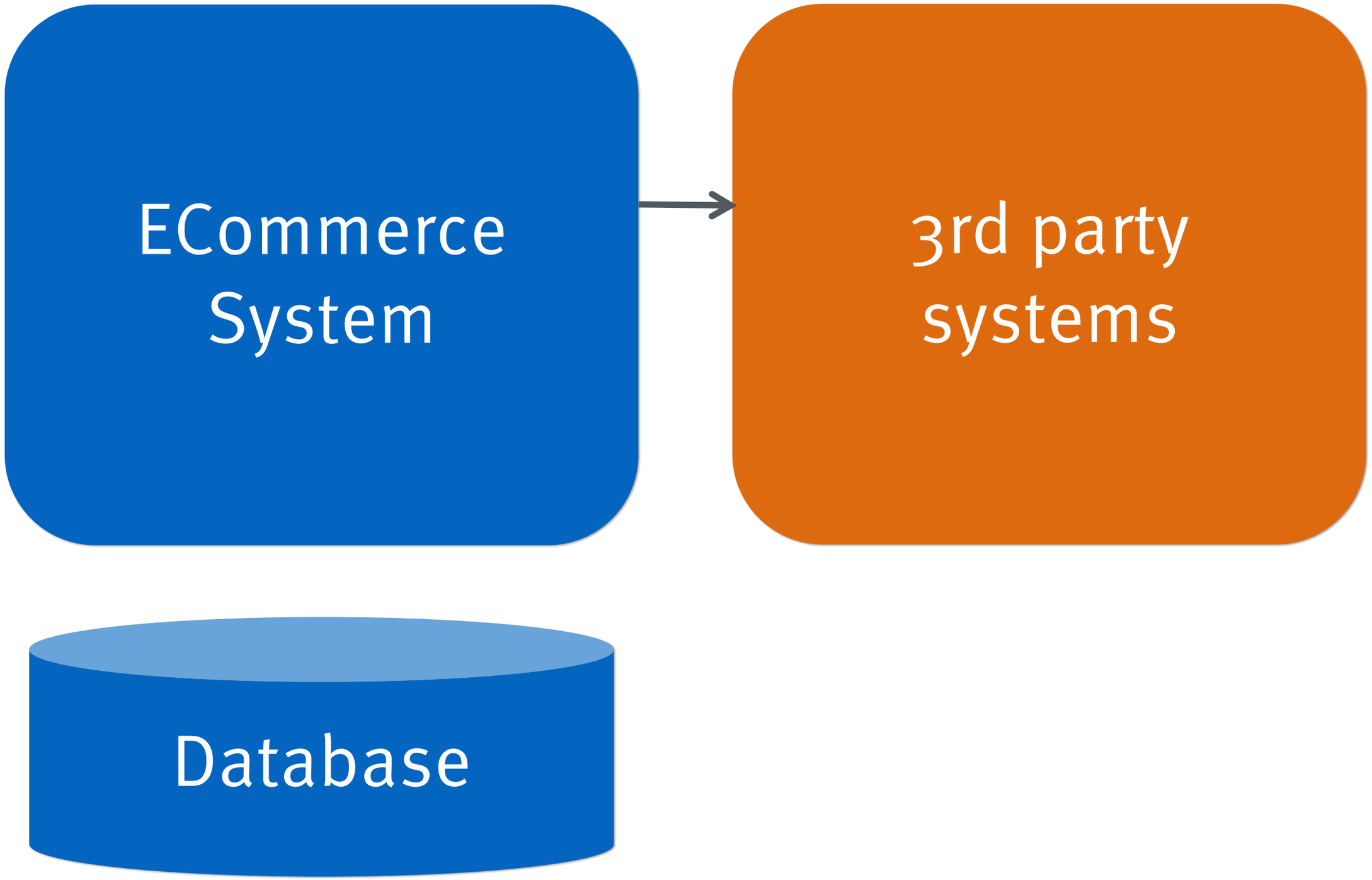
to satisfy the customer

through early and

continuous delivery

of valuable software.

Deployment Monolith



Continuous Delivery: Build Pipeline

ECommerce System

Commit Stage

Automated Acceptance Testing

Automated Capacity Testing

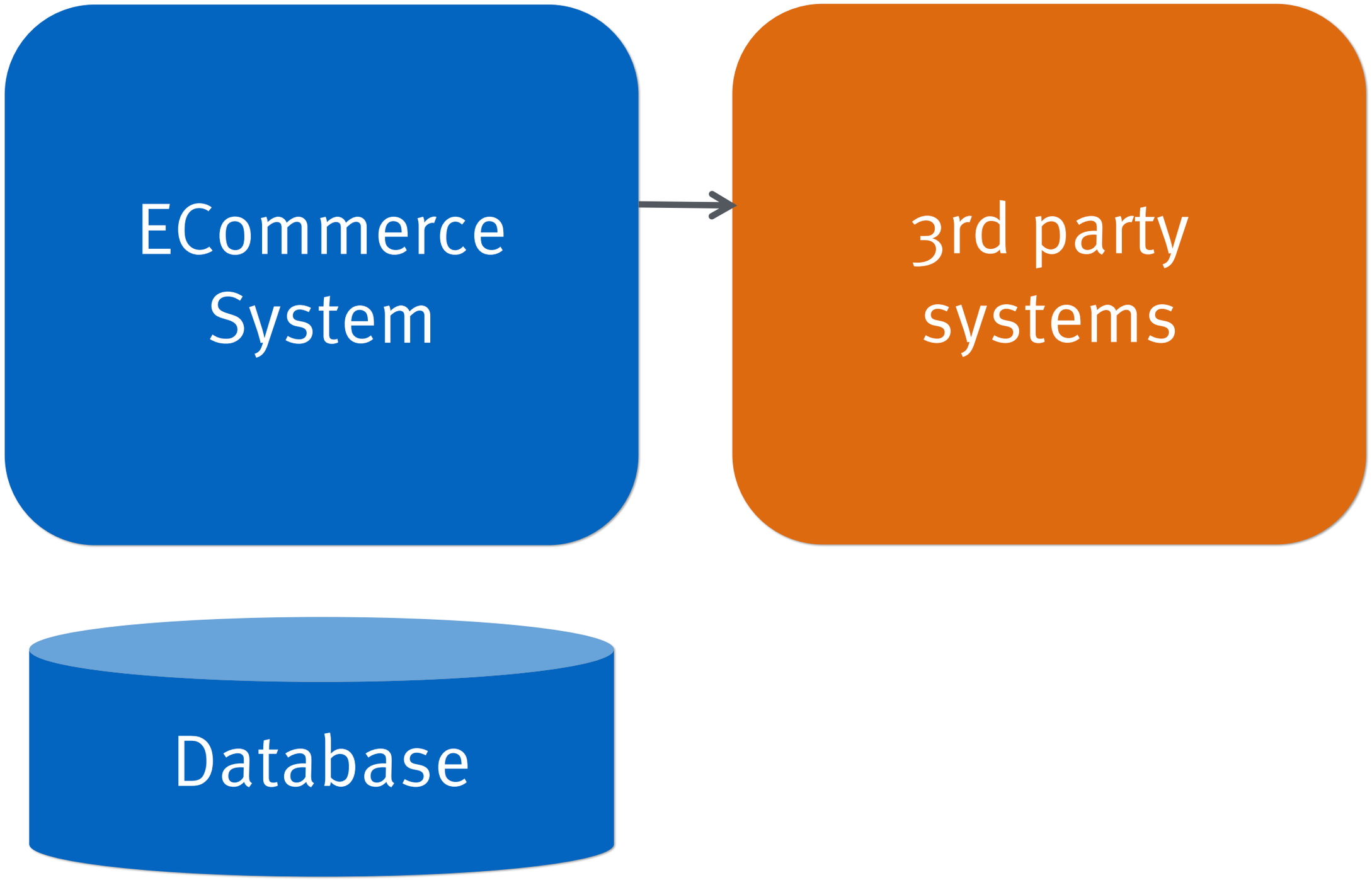
Manual Explorative Testing

Release

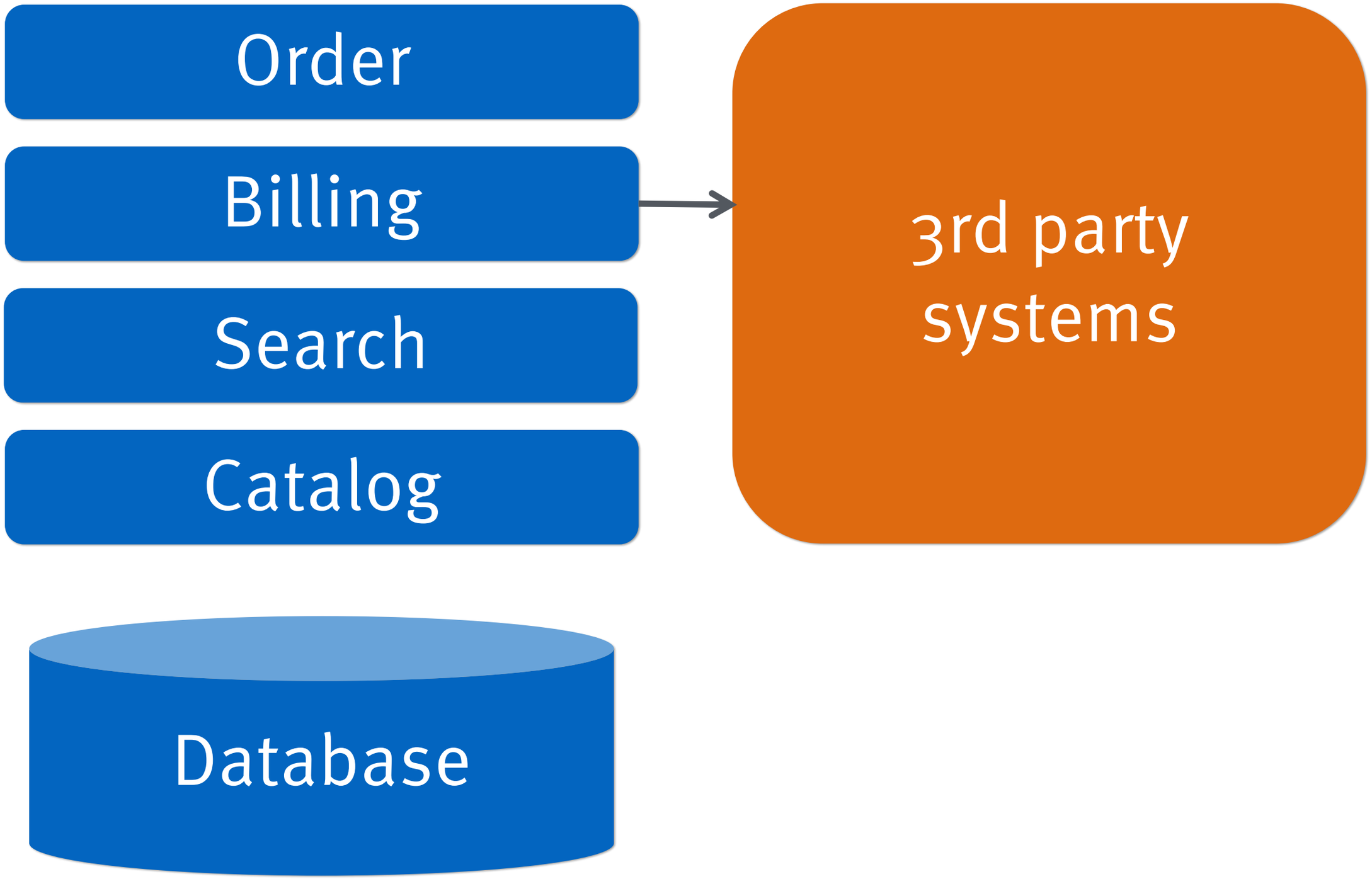
Build Pipeline: Problems

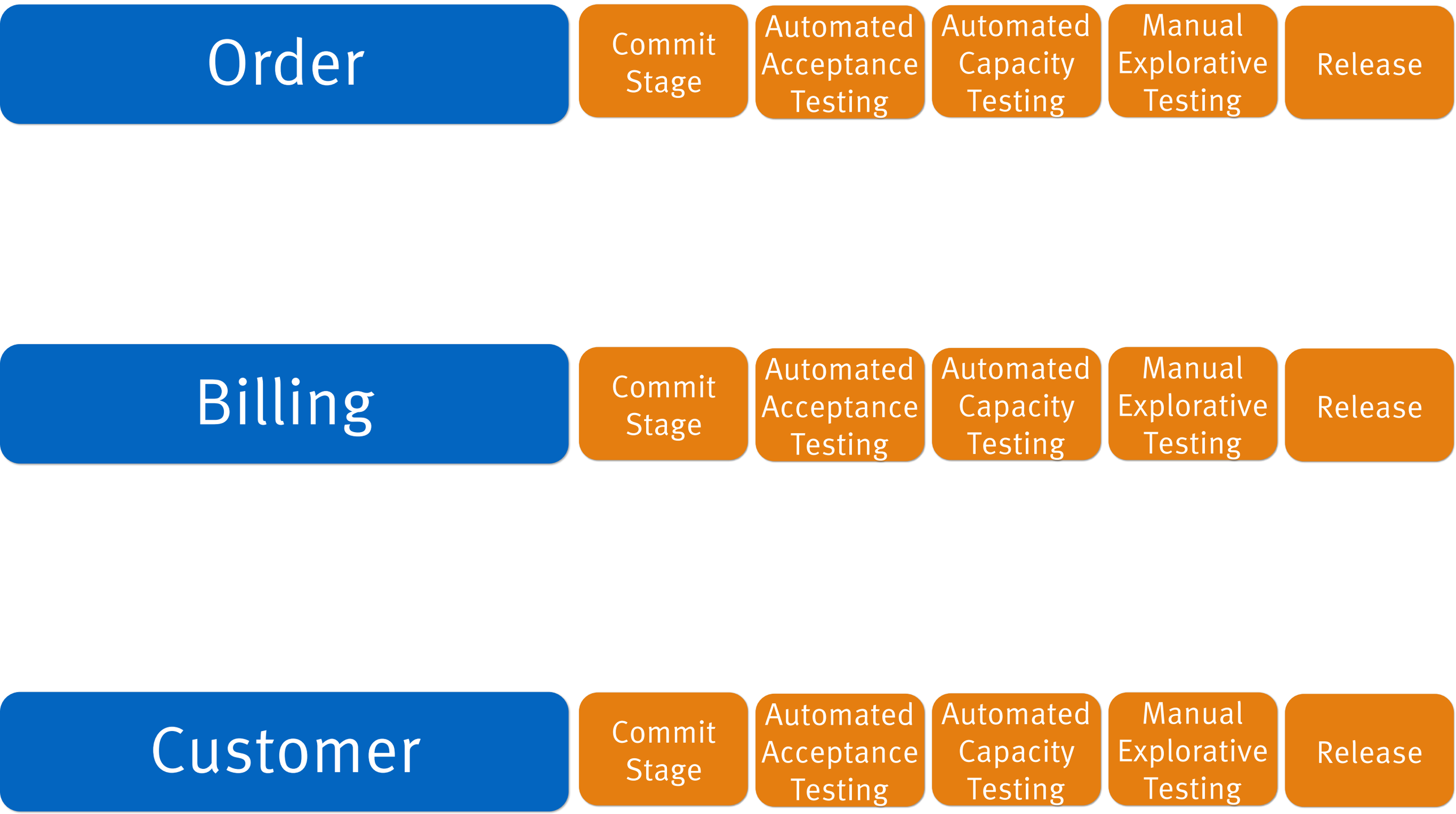
- Complex infrastructure
- Huge database
- 3rd party integration
- Slow feedback
- Test everything for each commit
- Huge deployment unit
- Deployment slow

Microservices



Microservices




Order
Commit Stage
Automated Acceptance Testing
Automated Capacity Testing
Manual Explorative Testing
Release
Billing
Commit Stage
Automated Acceptance Testing
Automated Capacity Testing
Manual Explorative Testing
Release
Customer
Commit Stage
Automated Acceptance Testing
Automated Capacity Testing
Manual Explorative Testing
Release

Build Pipeline for Microservices

- Independent deployment
- Build pipeline per Microservice

- Smaller
- Easier to set up
- Less features (3rd party systems)
- Faster Feedback: Less tests

Independent Scaling

- Can scale each Microservice indenpendently
- Much easier than for a Deployment Monolith

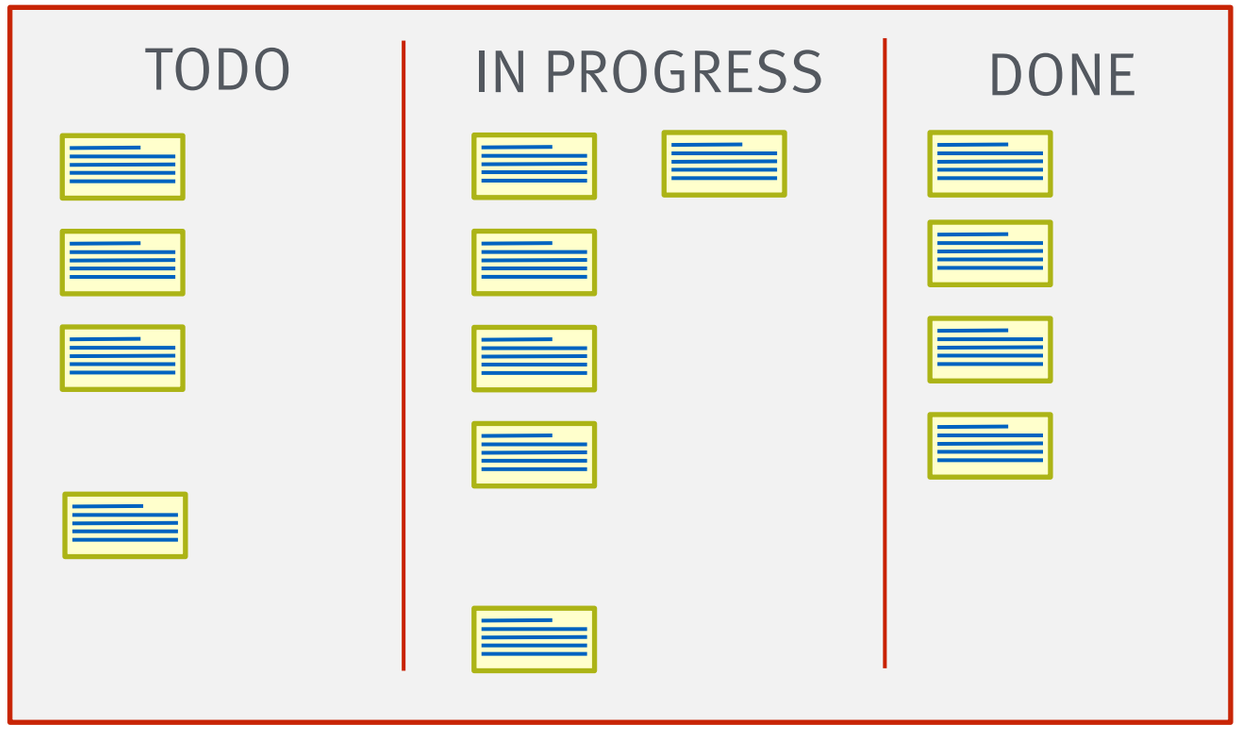
Technology Freedom

- No meetings for introducing a bug fix in a library!
- No big migration away from outdated technology
- Easier experiments
- Higher motivation

Scaling Agility

Scaling Agility

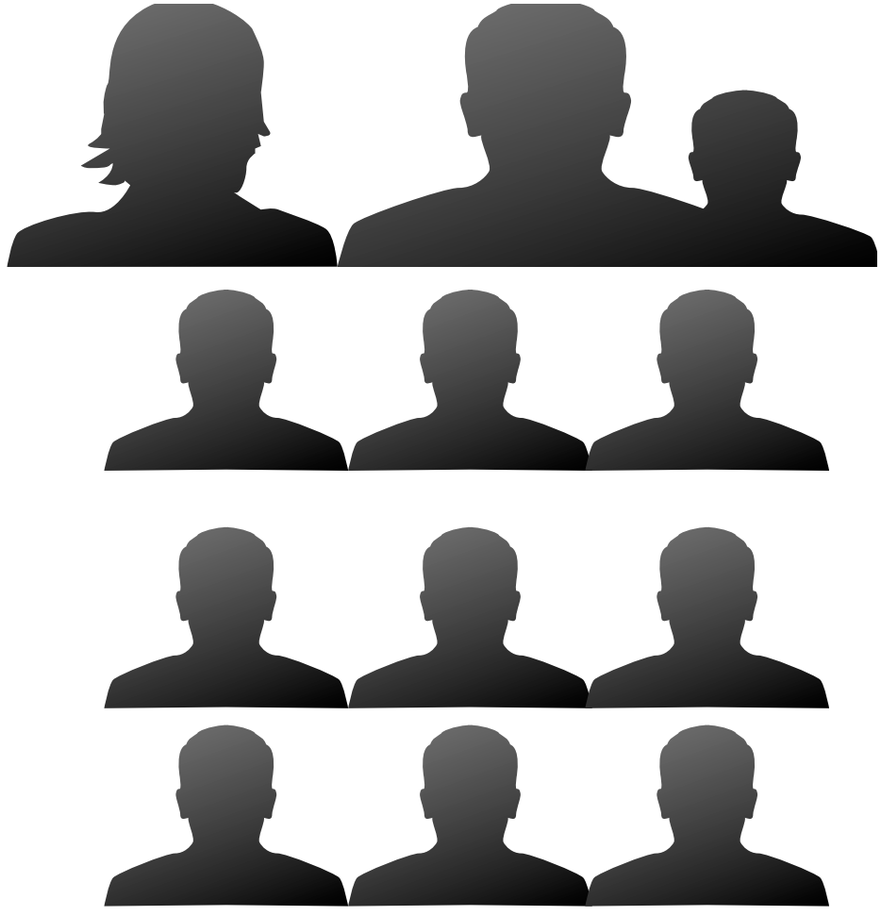
- Do more



- Get more stories implemented
- ...and running in production

Scaling Agility

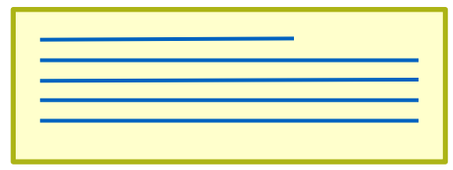
- Add more people
- Let the work in parallel
- Build more teams

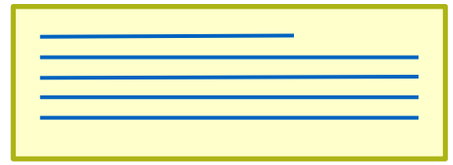


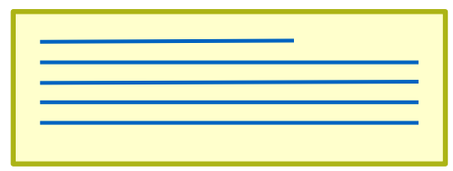
What is the problem?

Communication

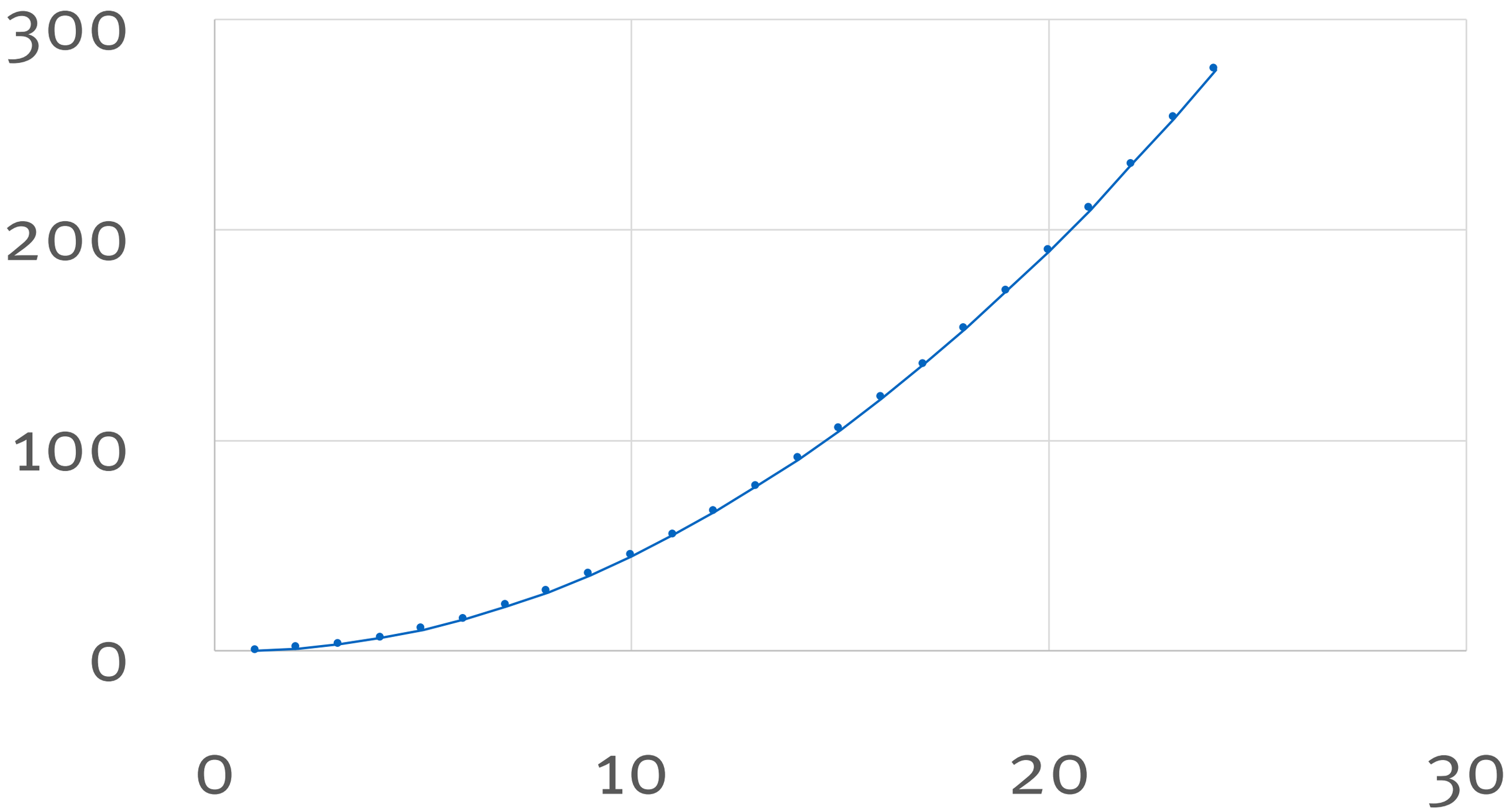
- Agile means communication
- Instead of written requirements
- ...story cards
- + direct communication







Persons vs. Potential Links



Communication is great!

Need more persons

Challenges for Scaling Agile

- Dependencies cause delays
- Too much communication about functionalities,...
- ...releasing software,
- ...and technologies

Challenges for Scaling Agile

- Dependencies cause delays
- Too much communication about functionalities...
- ...releasing software,
- ...and technologies

Conway‘s Law

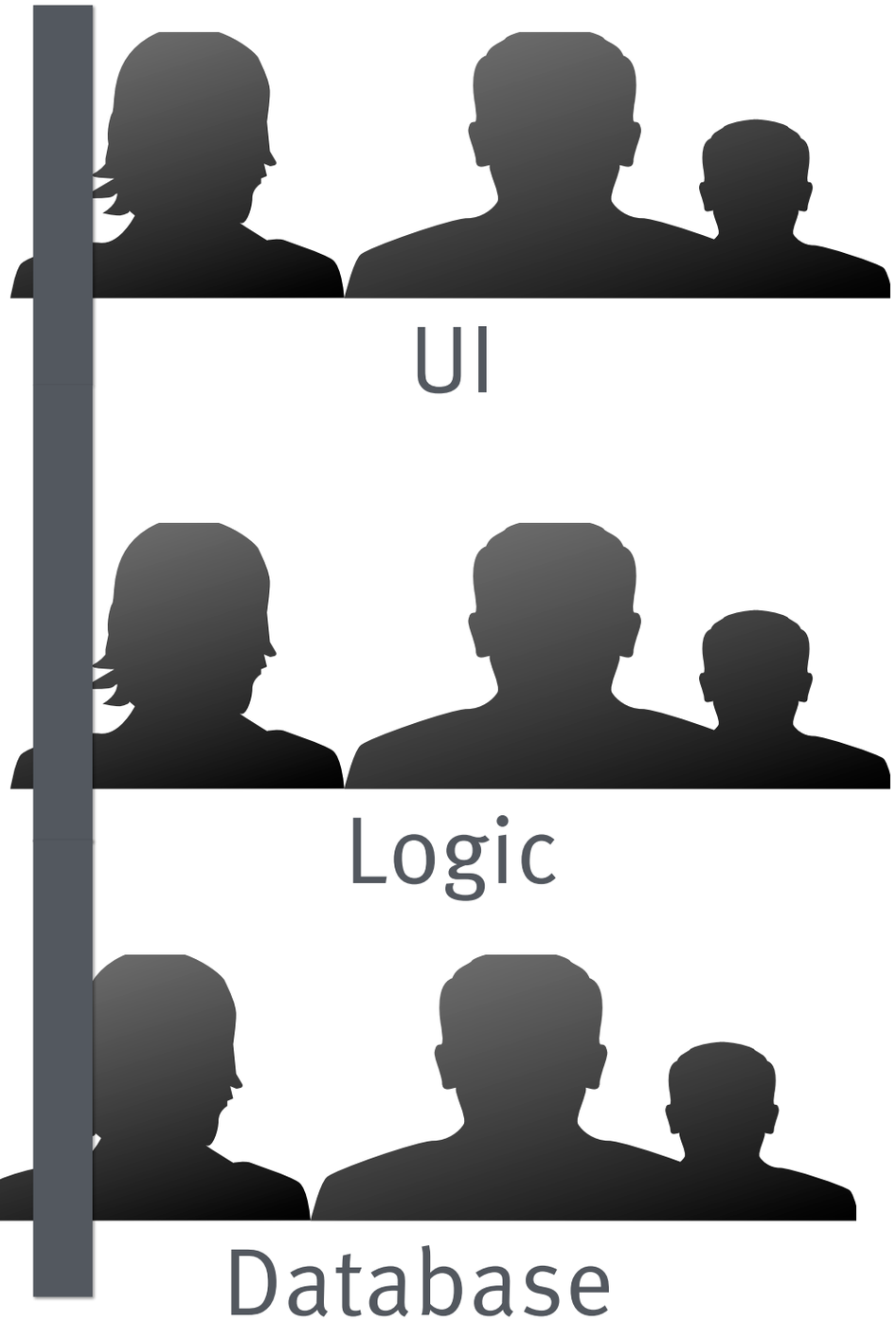
Architecture

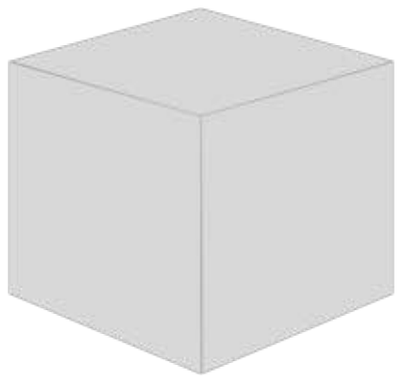
copies

communication structures

of the organization

Change Order Process!





3 sprints

DB
Team Sprint

Logic
Team Sprint

GUI
Team Sprint

time

Order Team =
UI+Logic+Database

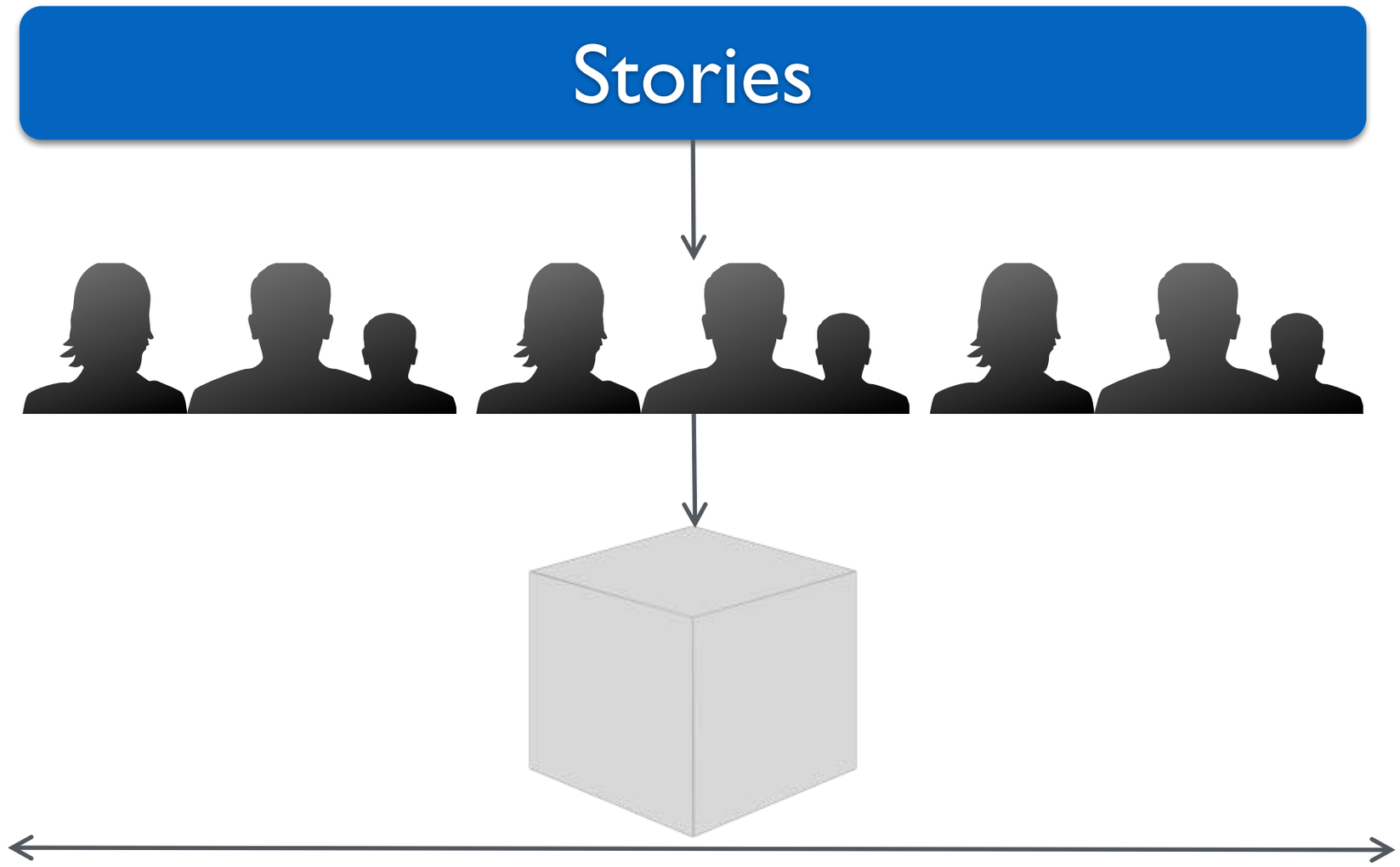
Domained scoped SCS enables the development of a domain by a **single team**

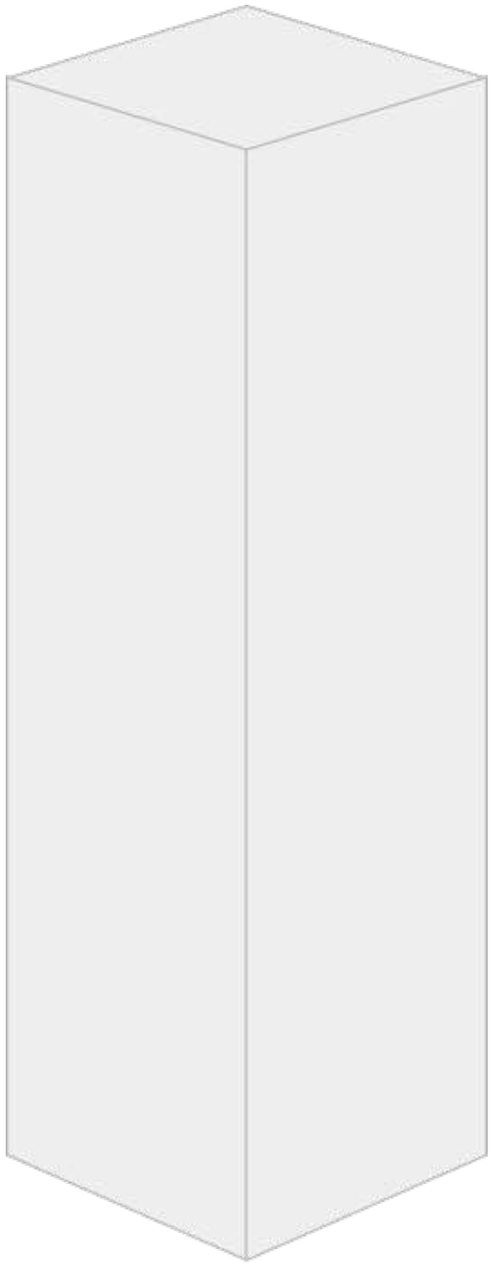
– no coordination

Challenges for Scaling Agile

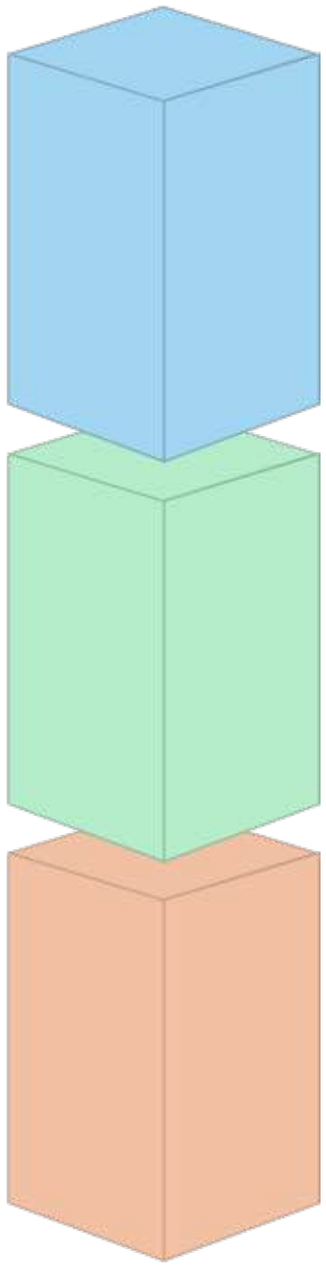
- Dependencies cause delays
- Too much communication about functionalities…
- …releasing software,
- …and technologies

Deployment Monolith



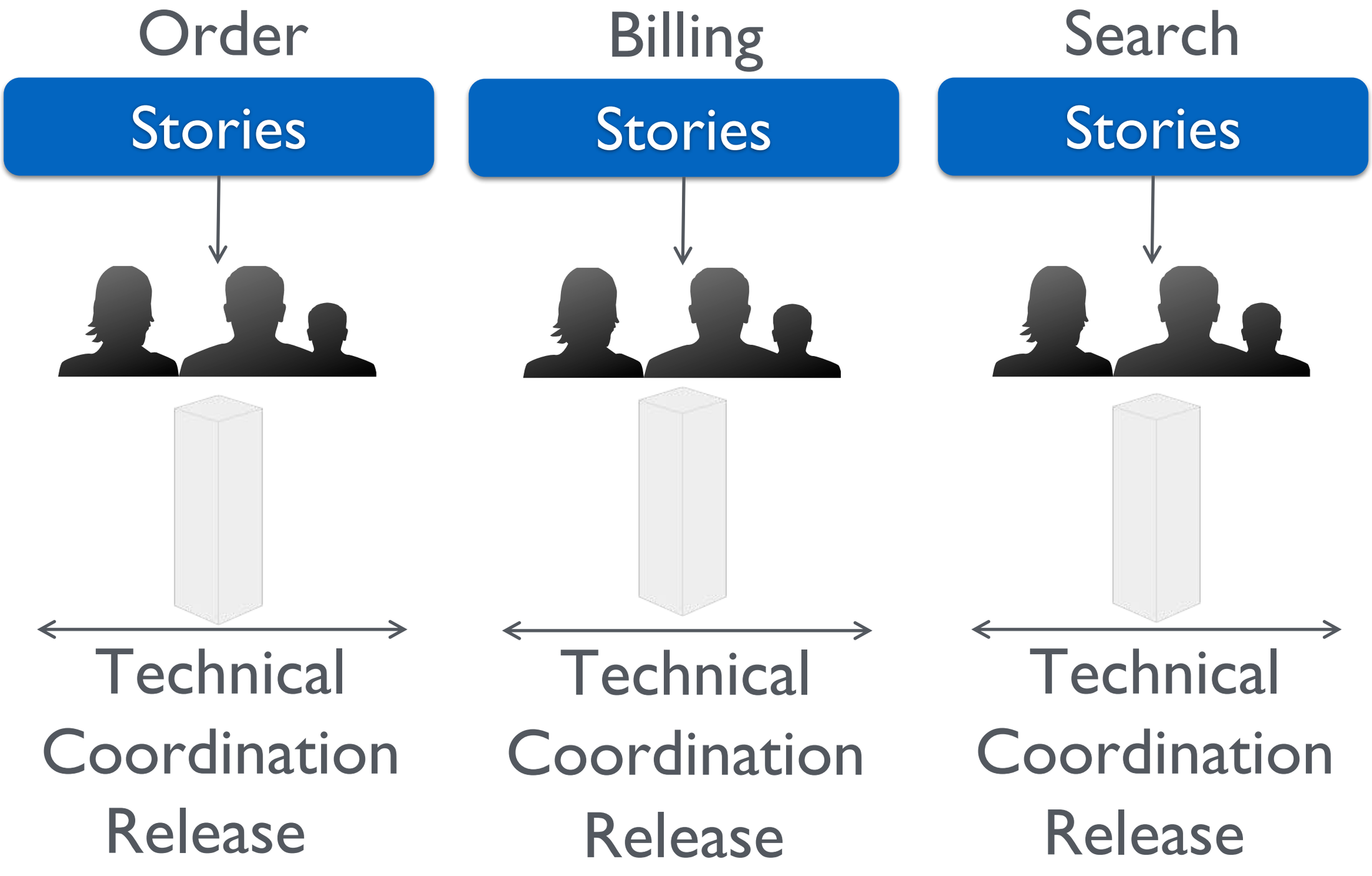


Self-contained System (SCS) – individually deployable



Technical decisions can be made independently from other systems (programming language, frameworks, tooling, platform)

SCS


Order
Billing
Search
Stories
Stories
Stories
Technical Coordination Release
Technical Coordination Release
Technical Coordination Release

Why Microservices/SCS?

Scaling Agile

Handle Legacy more efficient

Sustainable development speed

Robustness

Continuous Delivery

Independent Scaling

Choose best technology for job!

Conclusion

Conclusion: SCS & Microservices

- Microservices have many advantages
- SCS are a way to use Microservices
- ...for large projects
- ...to scale agile

Thank You!

@ewolff

http://microservices-buch.de/

http://microservices-book.com/primer.html

http://scs-architecture.org

EMail slidesjugsaxony@ewolff.com to get:

Slides

\+ Microservices Primer

\+ Sample Microservices Book

\+ Sample of Continuous Delivery Book

Powered by Amazon Lambda & Microservices