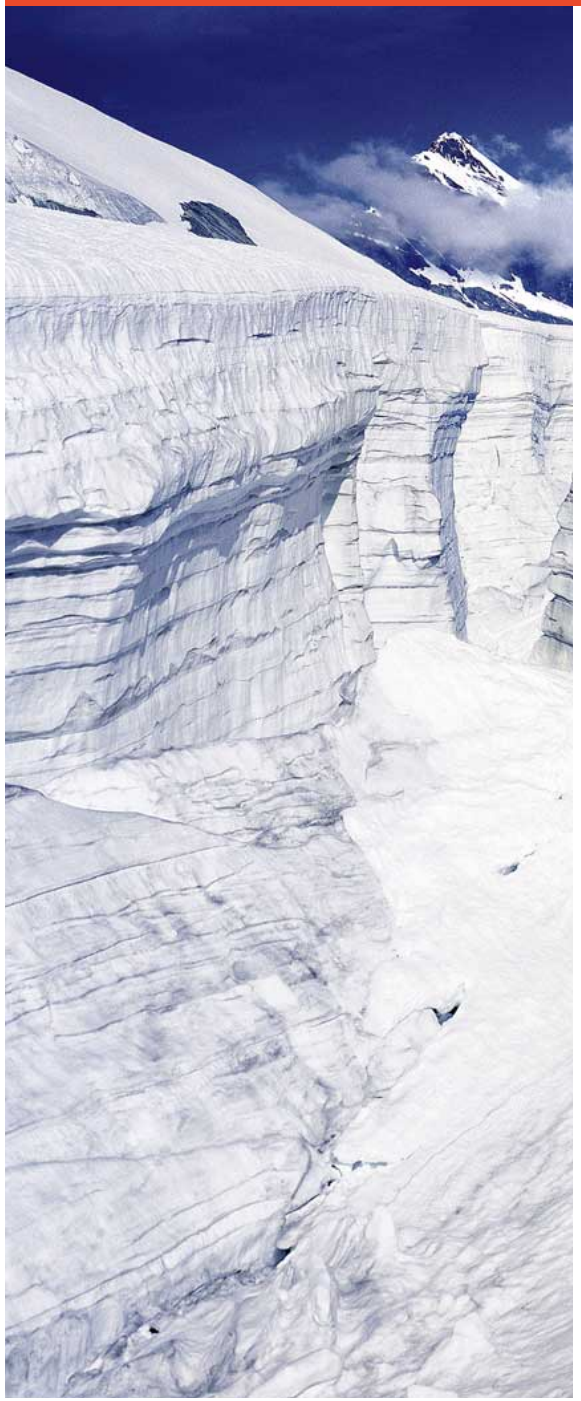
@olliwegner
@stilkov

QCon London 2014

# Please evaluate our talk via the mobile app!



Oliver Wegner • Stefan Tilkov

# From Parts to a Whole: Modular Development of a Large-Scale e-Commerce Site

QCon, London 06/03/2014

innoQ OTTO

# 1. Reviewing architectures

# Generic Architecture Review Results

Building features takes too long

Technical debt is well-known and not addressed

Deployment is way too complicated and slow

Architectural quality has degraded

Scalability has reached its limit

“-ility” problems abound

Replacement would be way too expensive


http://www.flickr.com/photos/krissymayhew/5463349254, Krissy Ma

## Conway's Law

**Organization → Architecture**

"Organizations which design systems are constrained to produce systems which are copies of the communication structures of these organizations." – M.E. Conway

## Reversal 1

## Architecture → Organization

Any particular architecture approach constraints organizational options – i.e. makes some organizational models simple and others hard to implement.

## Reversal 2

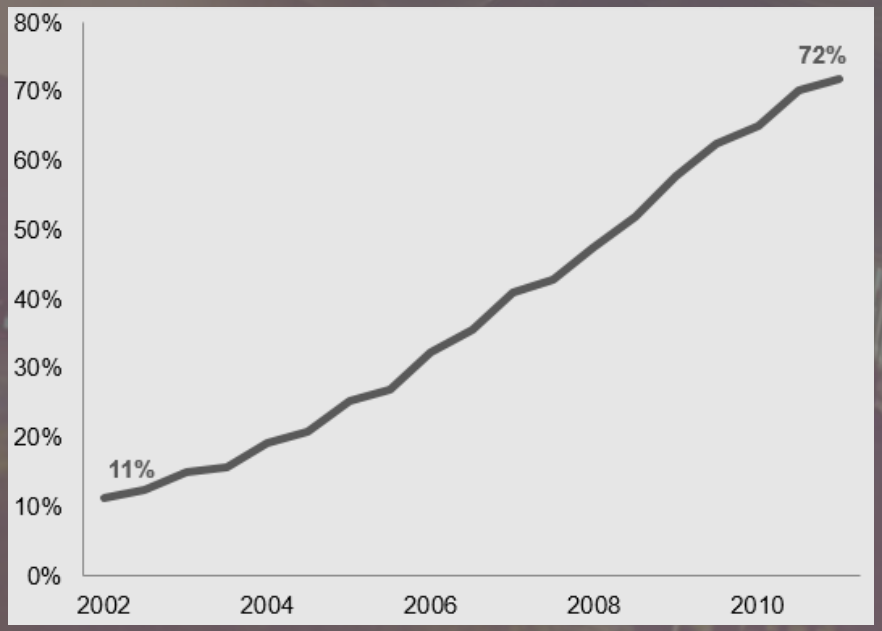
## Architecture → Organization

Choosing a particular architecture can be a means of optimizing for a desired organizational structure.

# 2. Rebuilding Otto.de

For the last **15 years** the E-Commerce business has become more important

Percentage of turnover




**4.200 Employees**

**> 2.1 Billion € turnover**

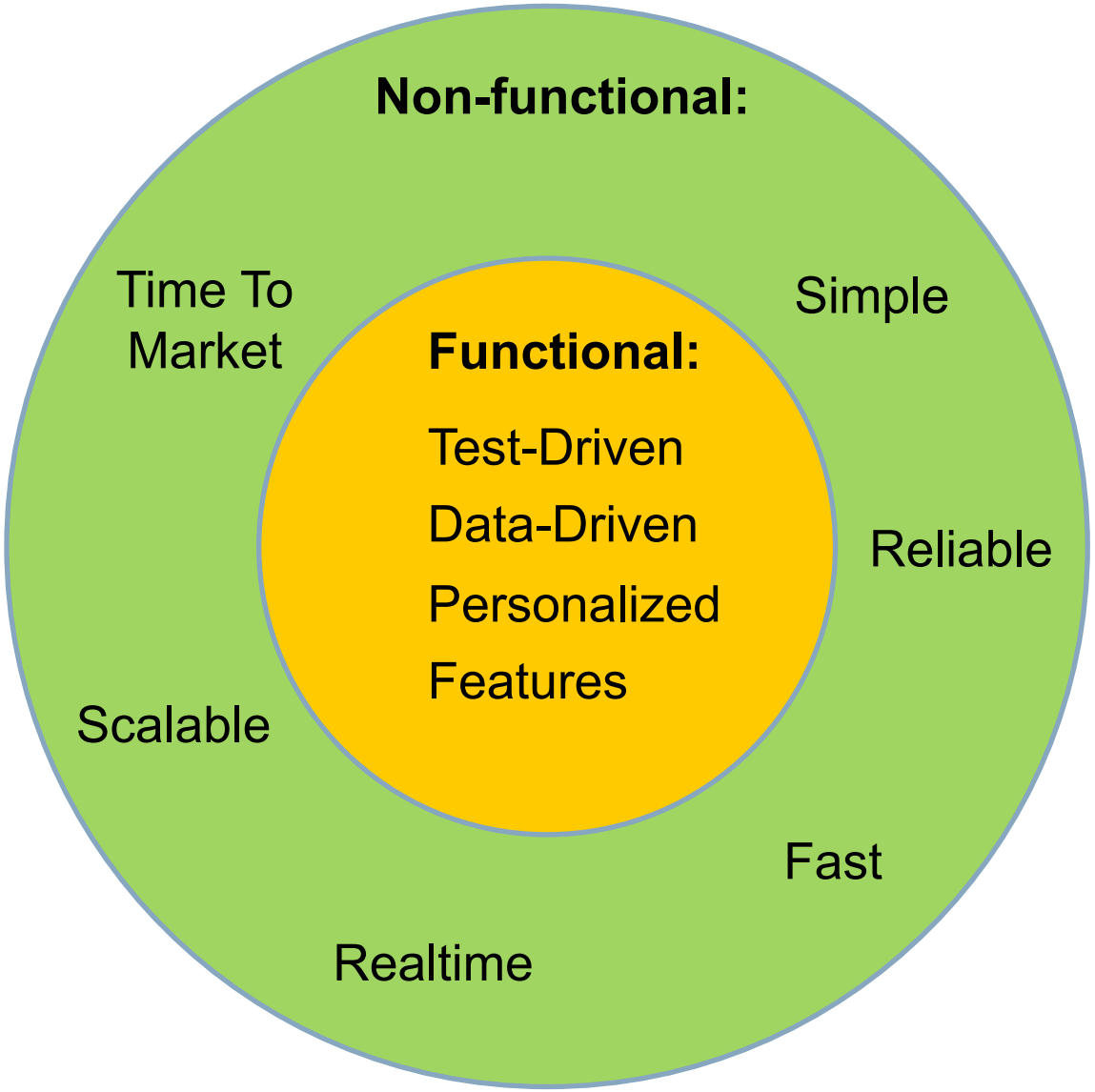
**> 2 Million items on Otto.de**

**80% turnover online**

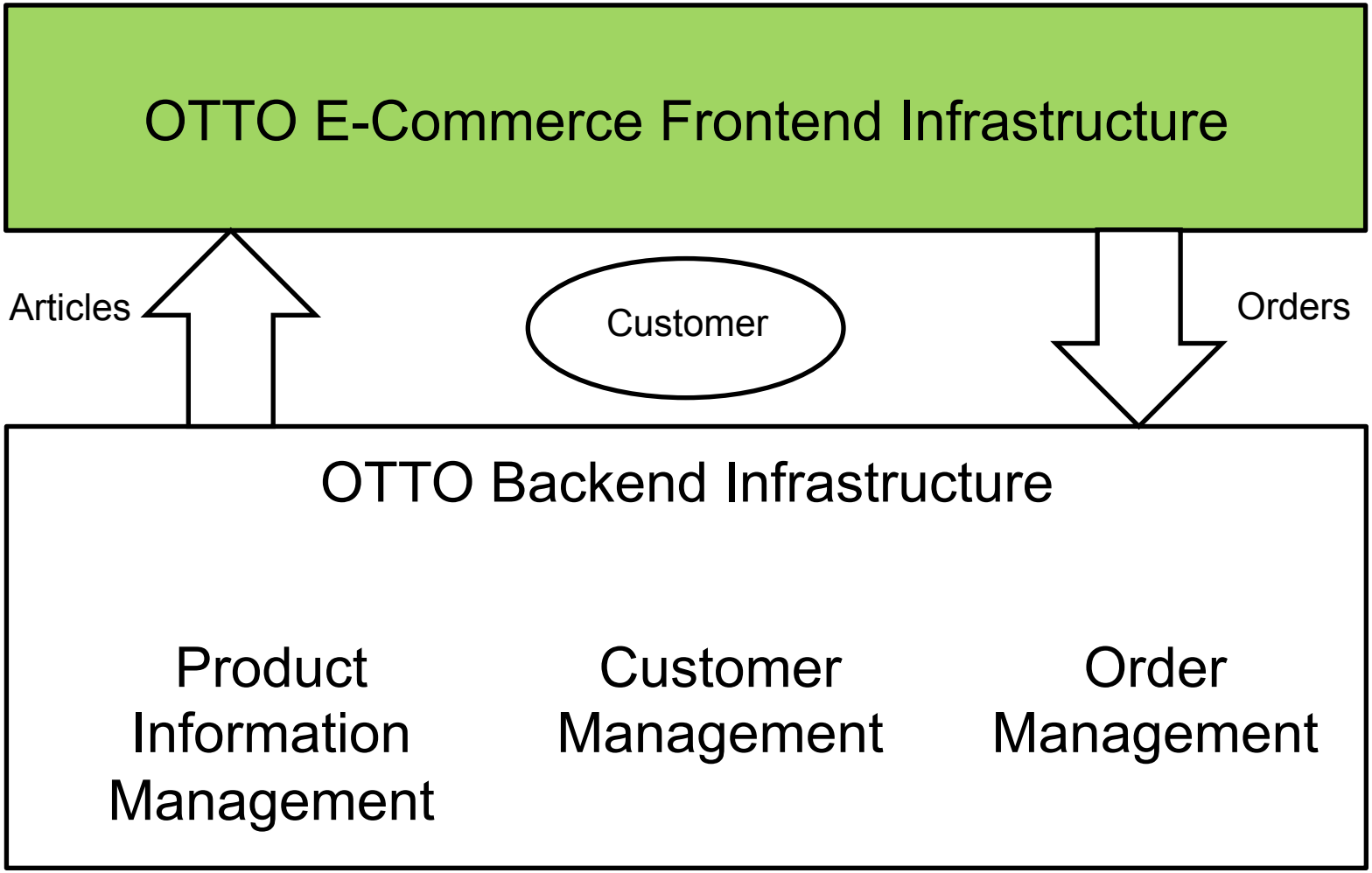

Kleid.
Gefunden auf OTTO .de

But why rebuild Otto.de?

# Goals


Non-functional:
Time To Market
Simple
Functional:
Test-Driven
Data-Driven
Personalized
Features
Reliable
Scalable
Fast
Realtime

# How green is the green field?



# Start of the project LHOTSE

Open Source as core technologies

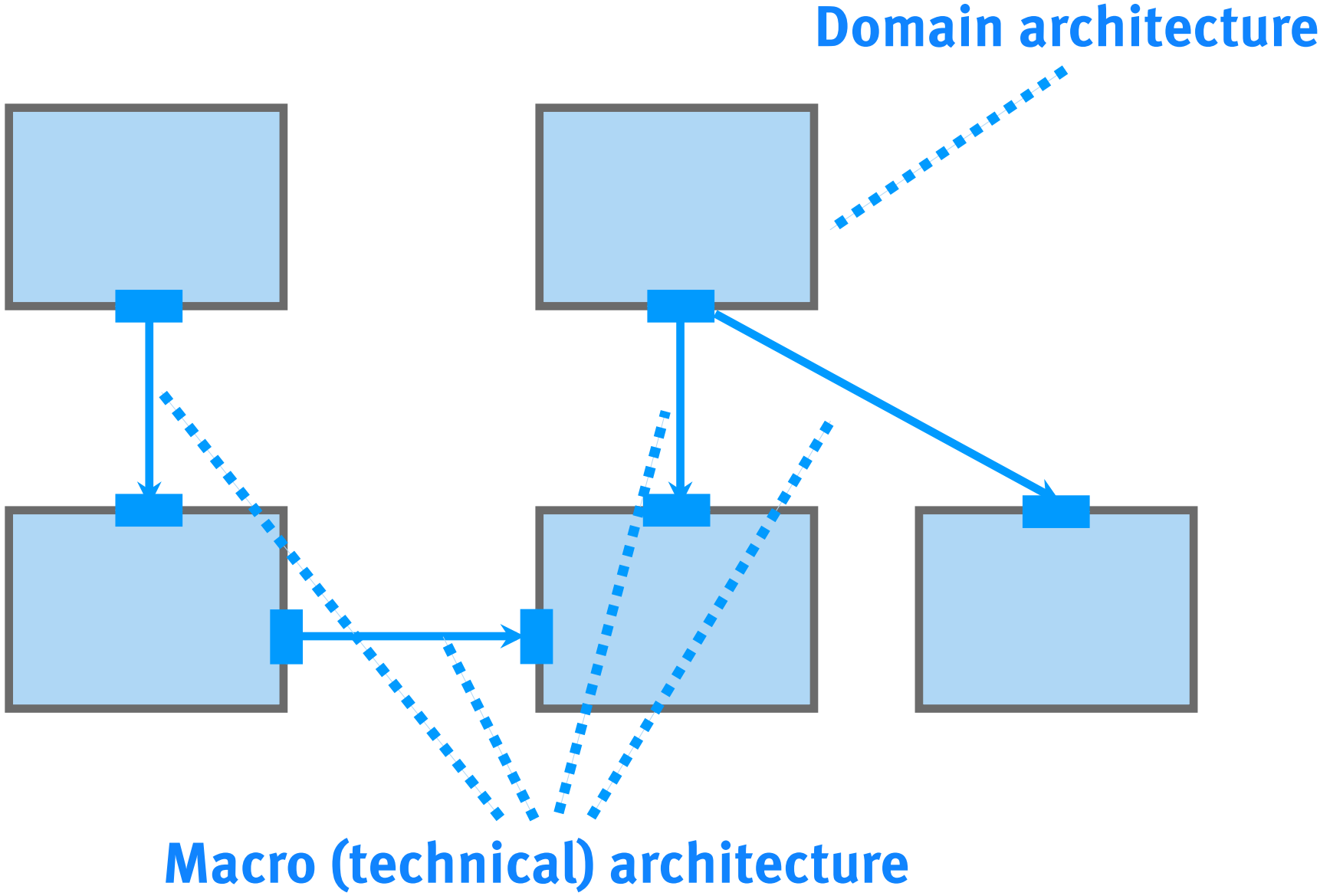
One Prototype to define the technology stack

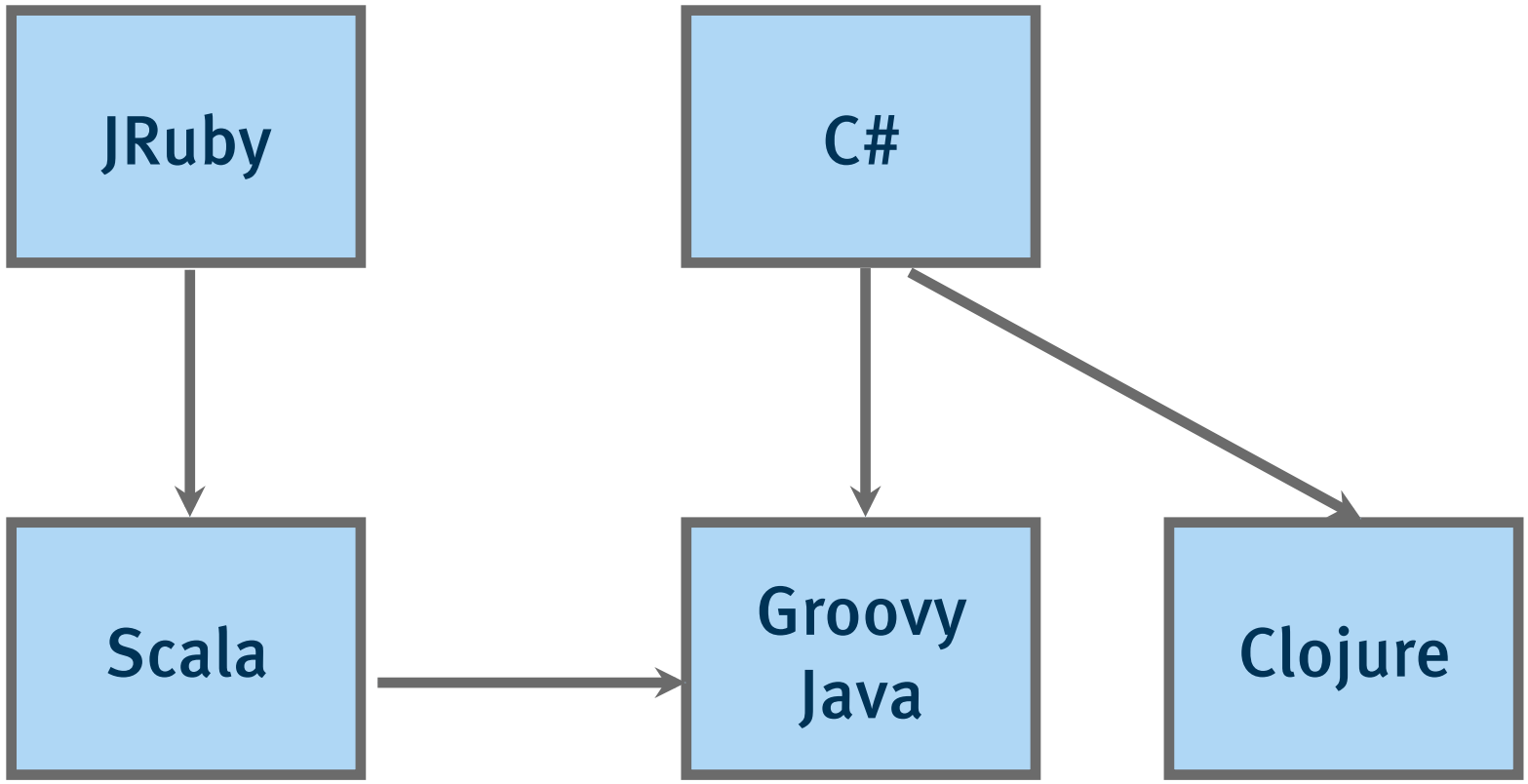
Project organization with autonomous teams

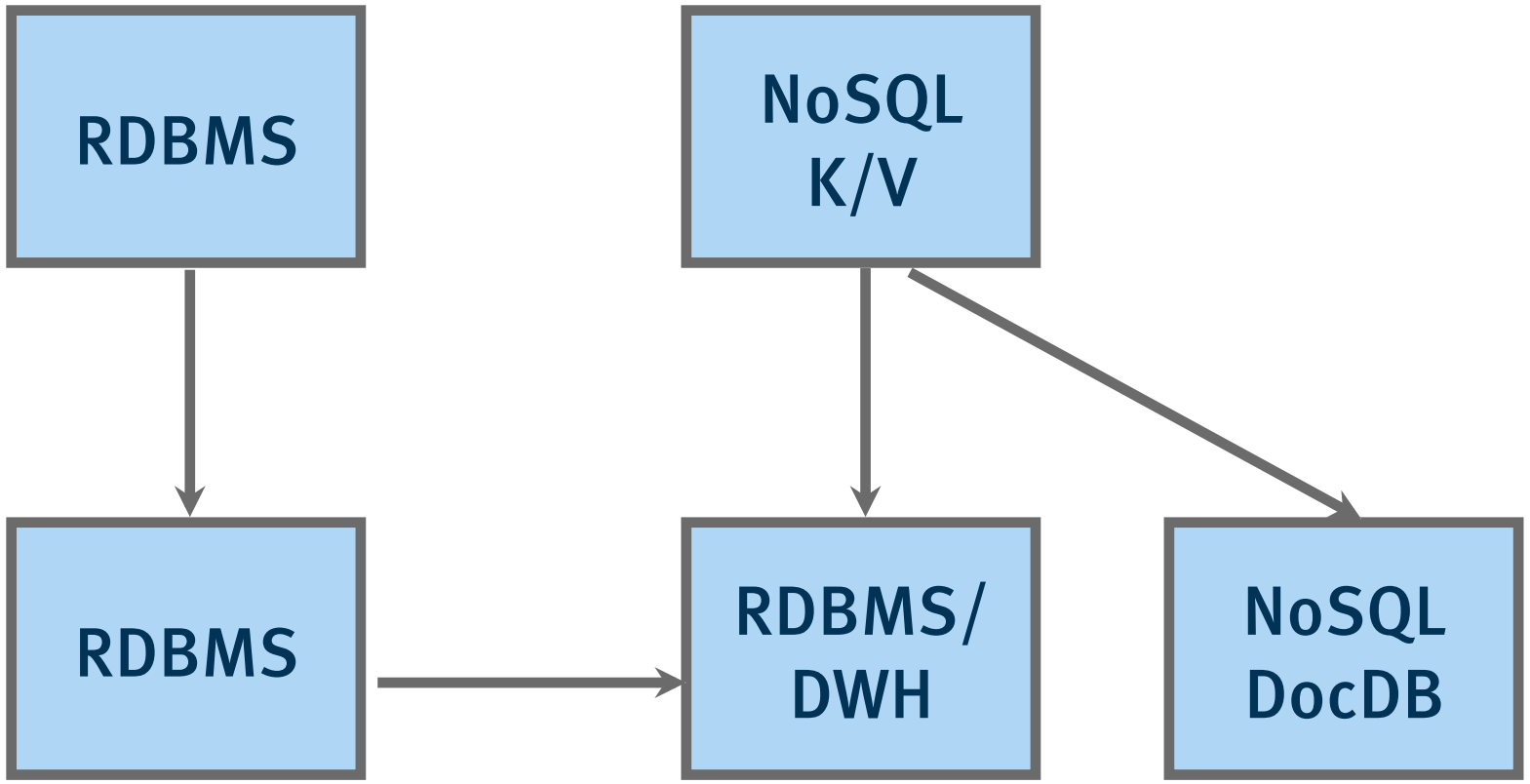
Scrum as an agile development method

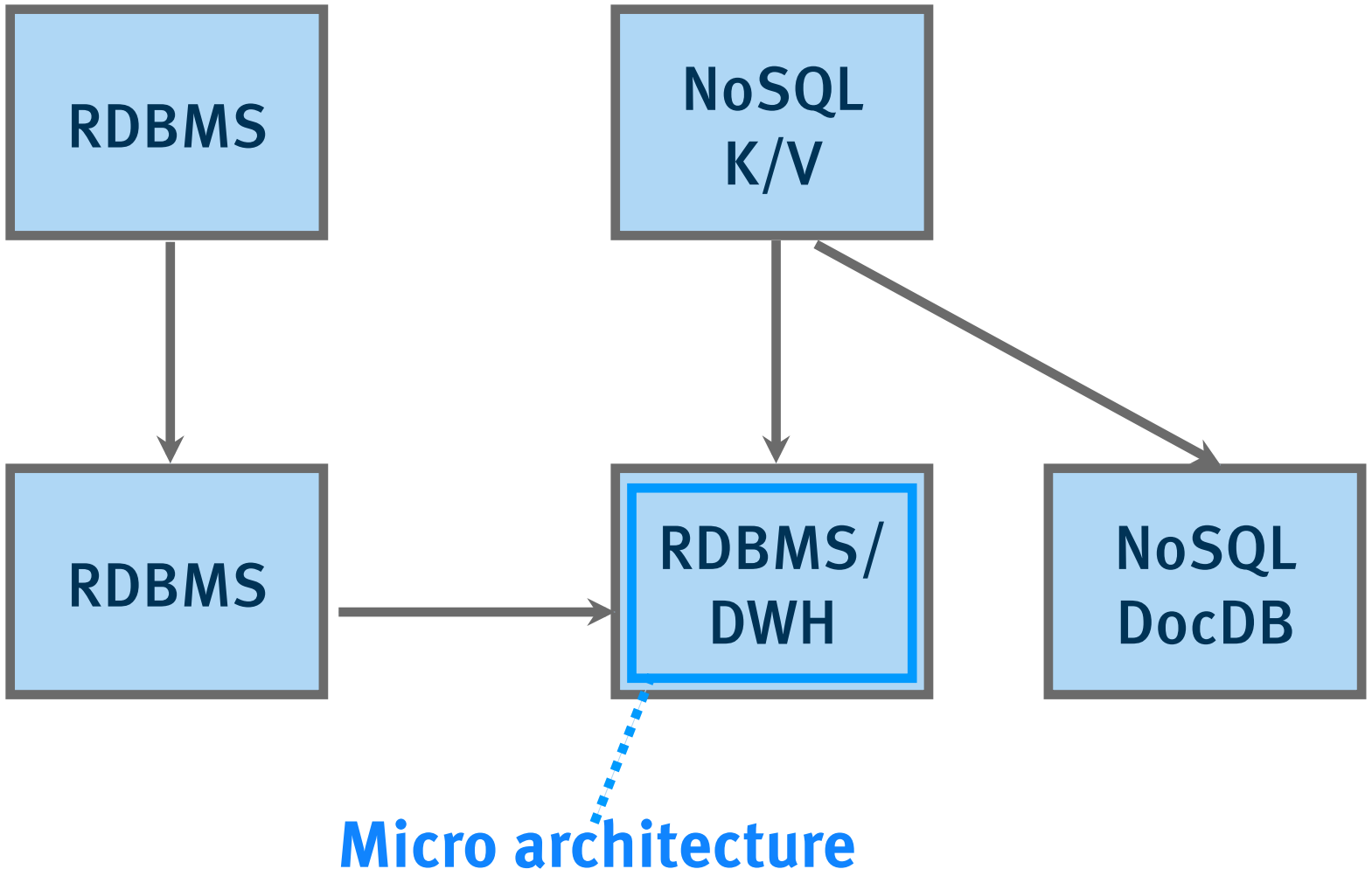
Technical system architecture

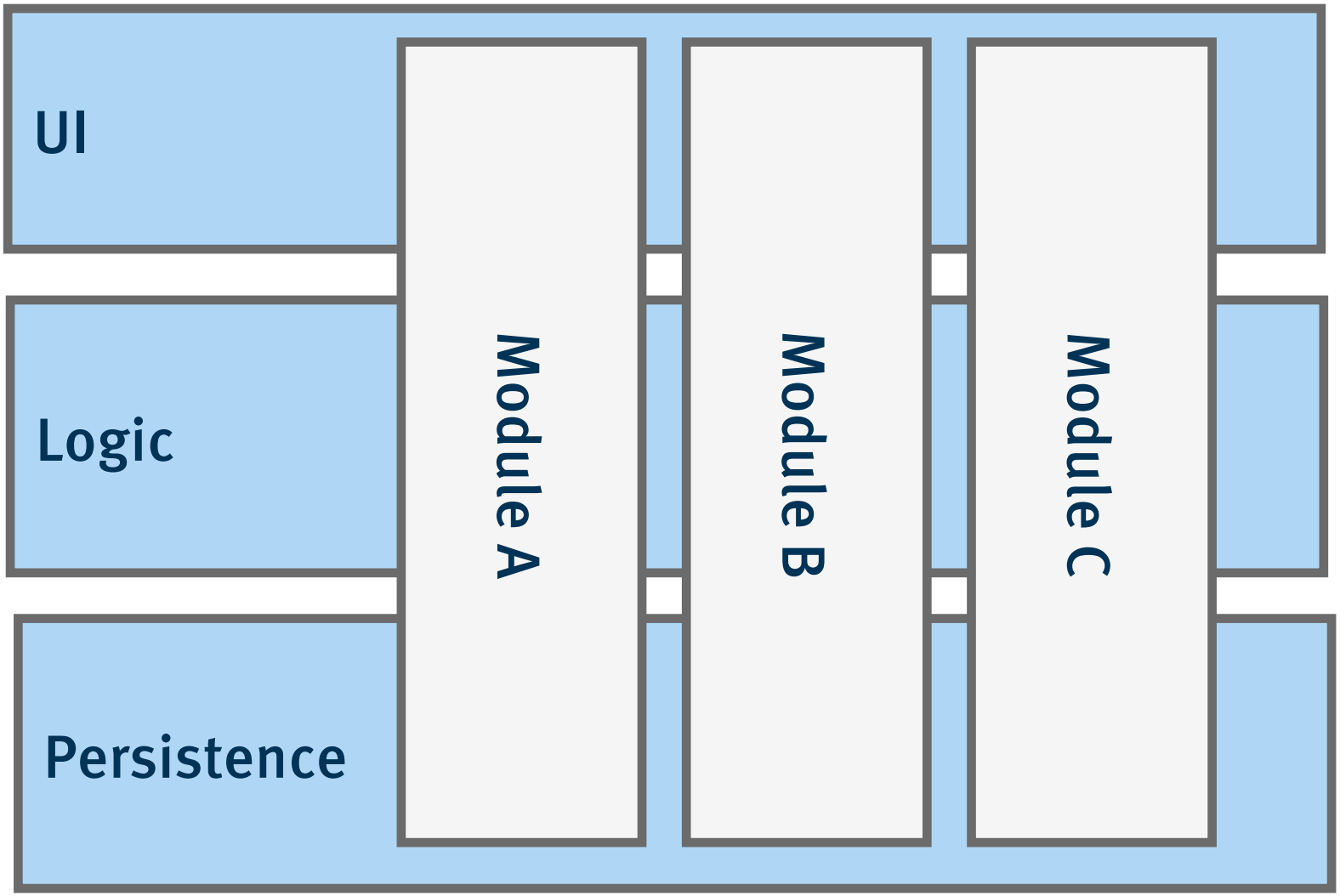
# 3. A system-of-systems approach

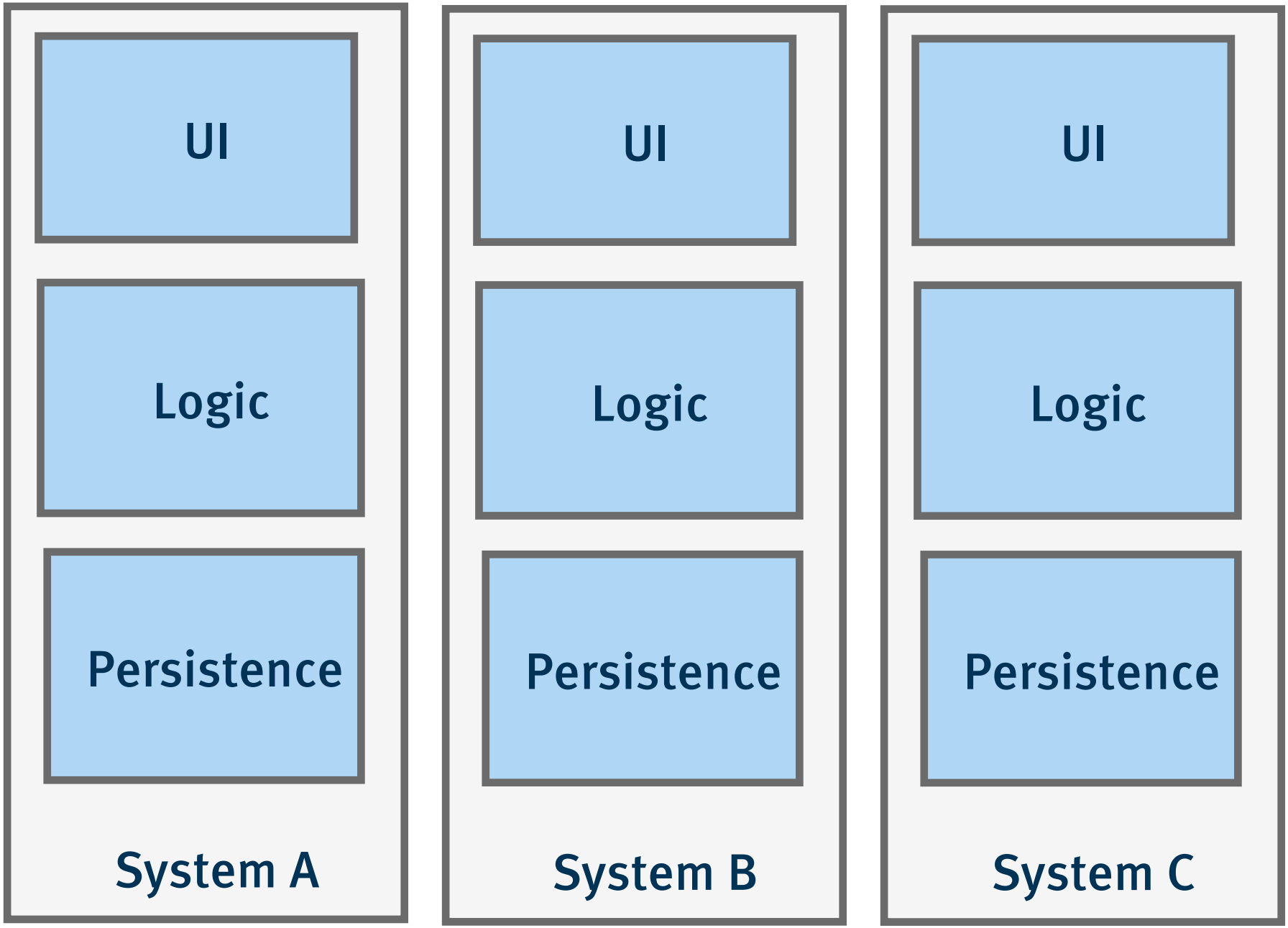

Domain architecture
Macro (technical) architecture


JRuby
C#
Scala
Groovy
Java
Clojure


RDBMS
NoSQL
K/V
RDBMS
RDBMS/
DWH
NoSQL
DocDB

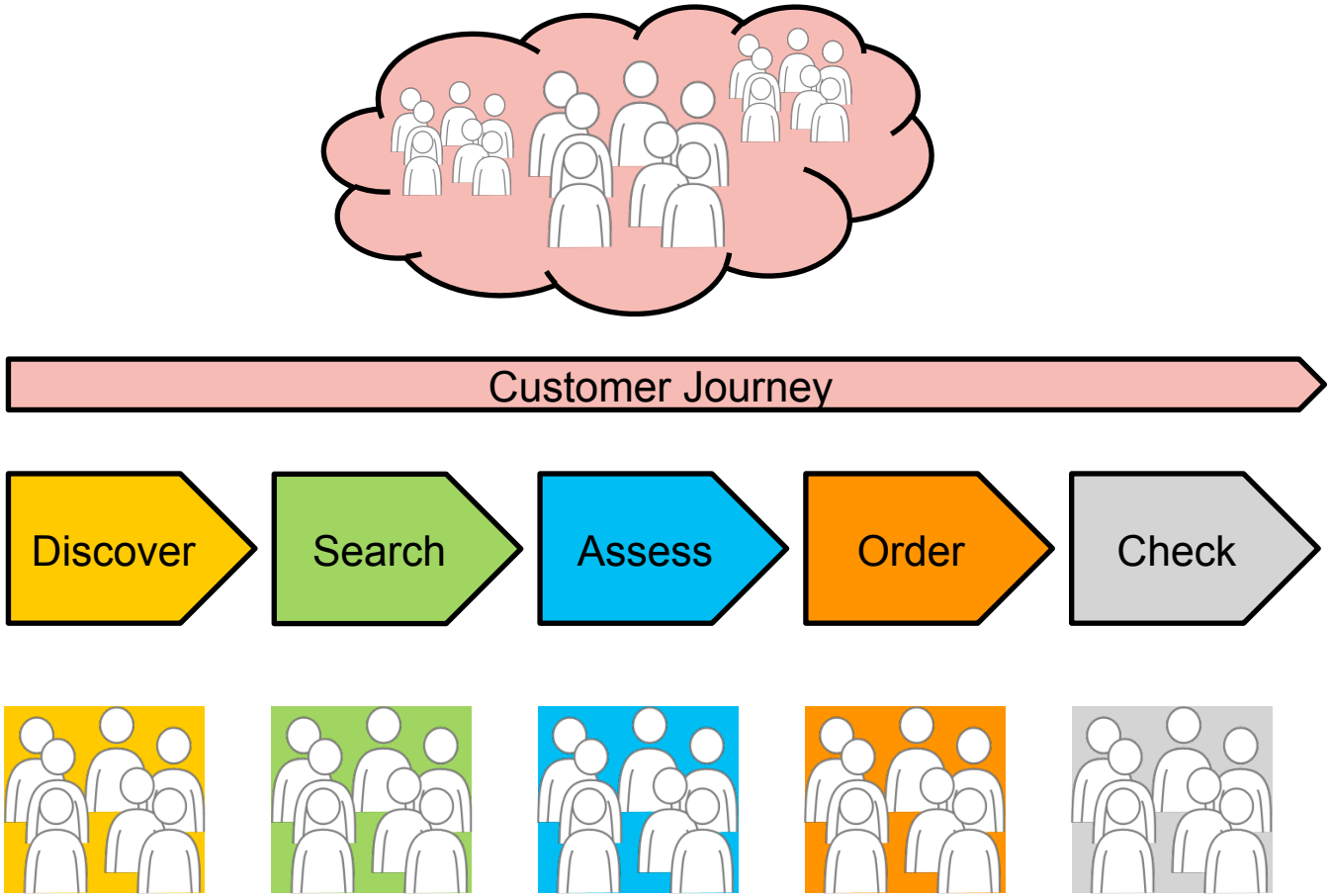

RDBMS
NoSQL
K/V
RDBMS
RDBMS/
DWH
NoSQL
DocDB
Micro architecture


UI
Logic
Persistence
Module A
Module B
Module C

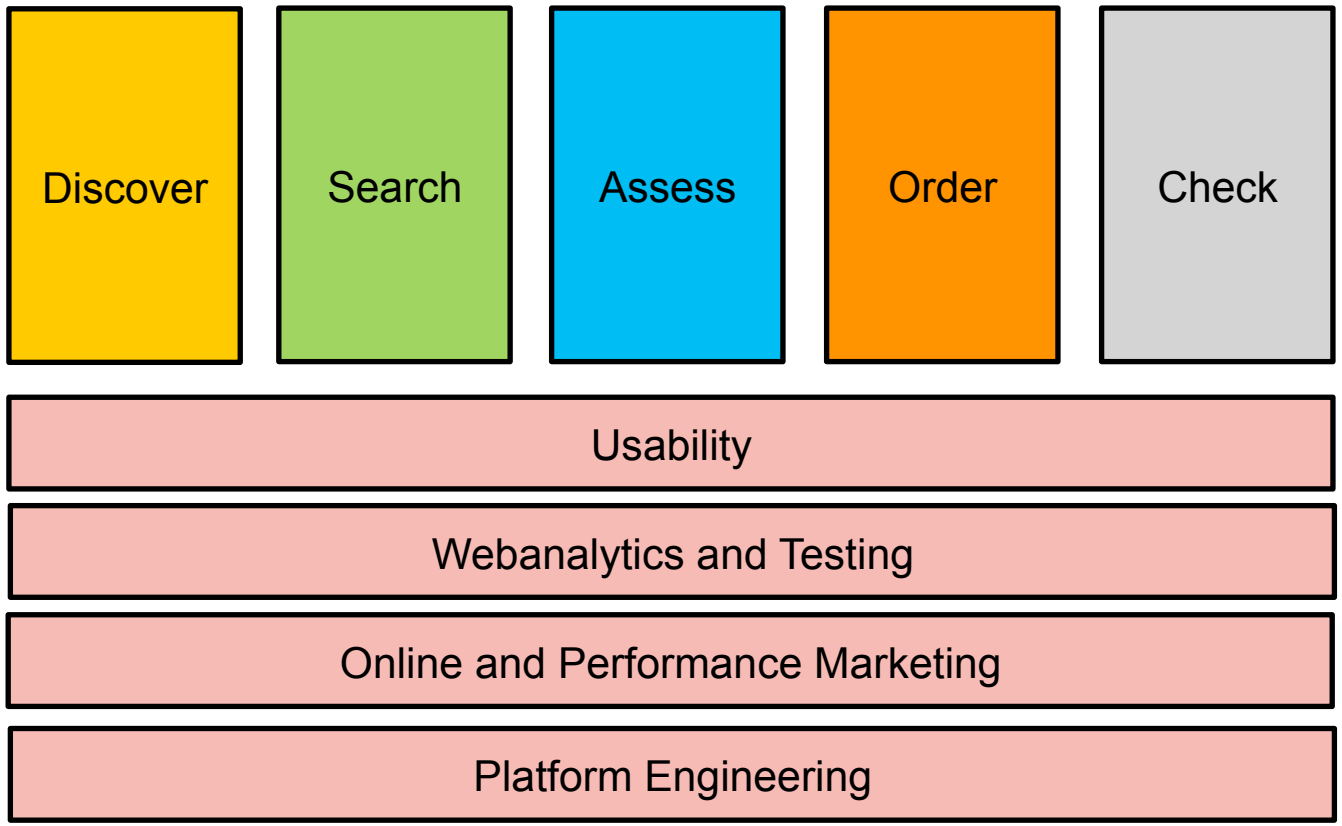

UI
Logic
Persistence
System A
UI
Logic
Persistence
System B
UI
Logic
Persistence
System C

# 4. The Otto architecture

# Buying Process – as you already know it

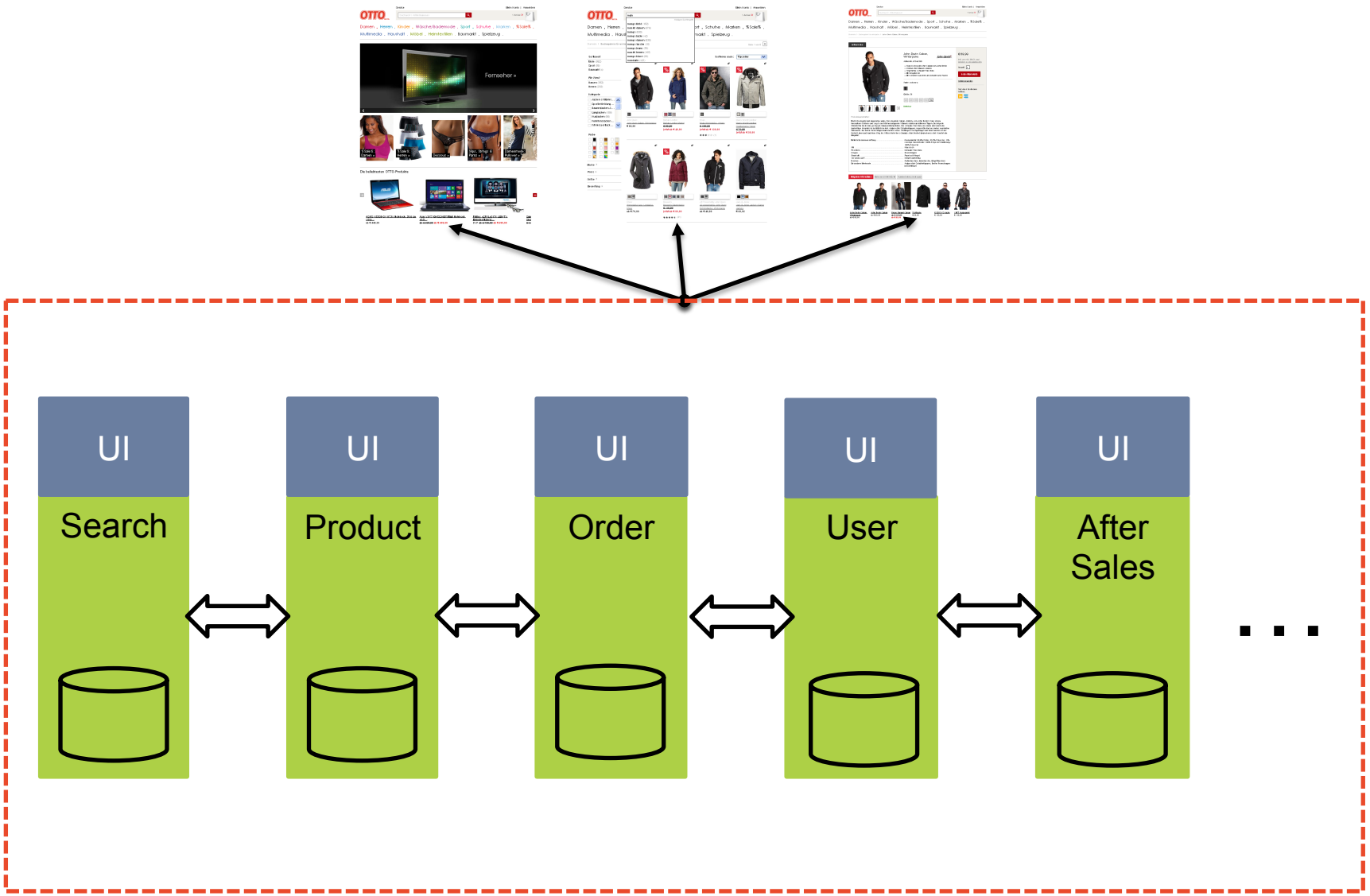


# The E-Commerce Business Architecture – Vertical and Horizontal aspects of the product Otto.de

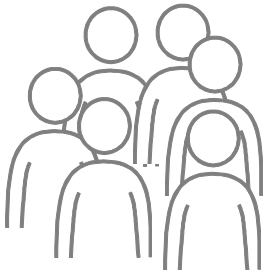


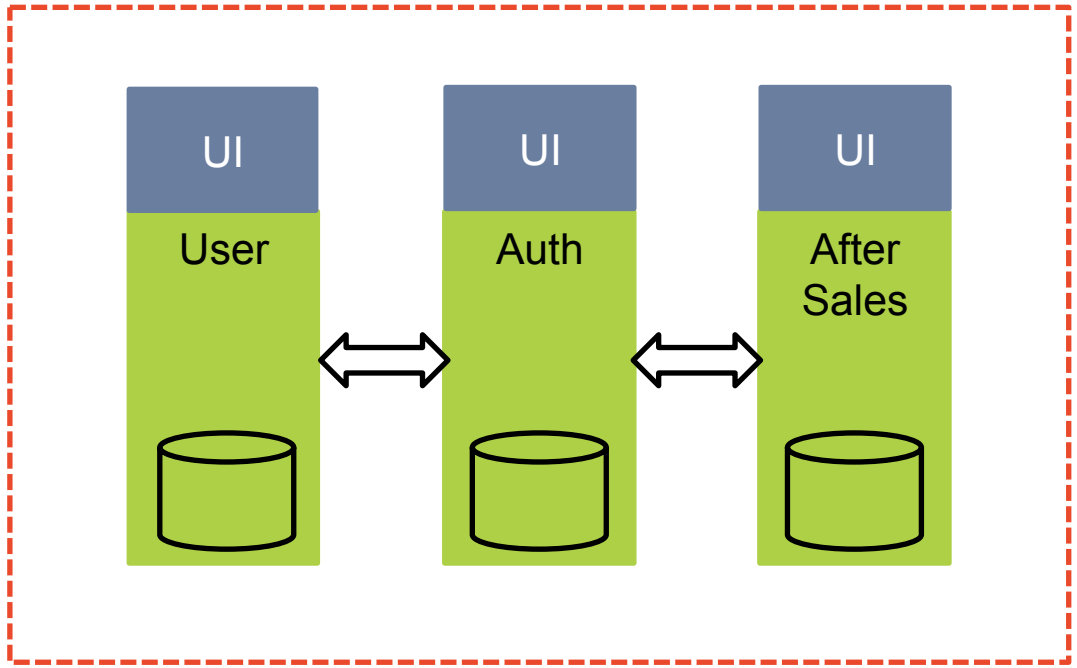

1 Website = 1 Product = 1 System = 1 Engineering Team ?

# System architecture is vertical



# Organizational aspects






1 Team → N Systems

1 System → 1 Team

# Architecture rules

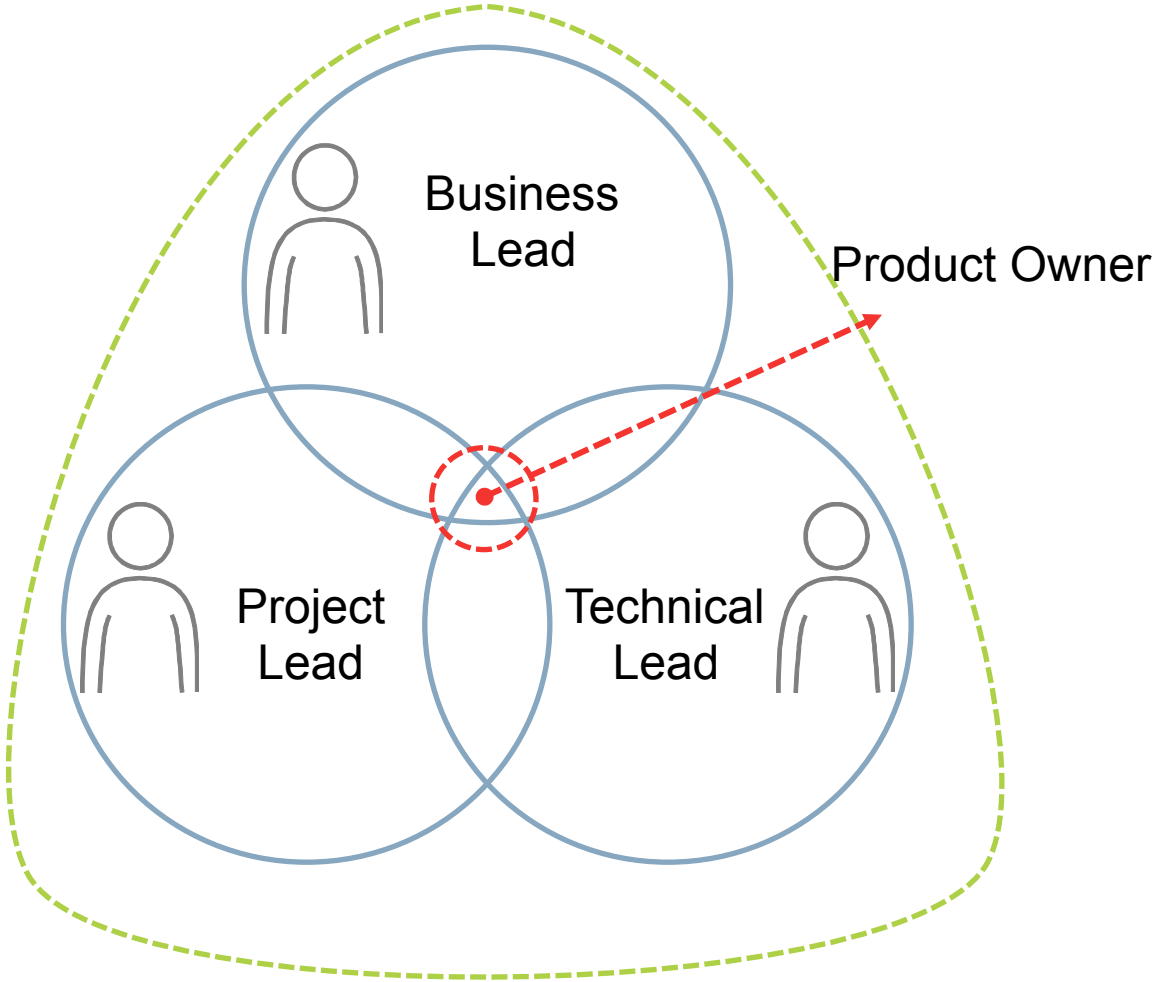
## Macro-Architecture

- RESTful Architecture
- Shared Nothing
- Vertical Design
- Data Governance

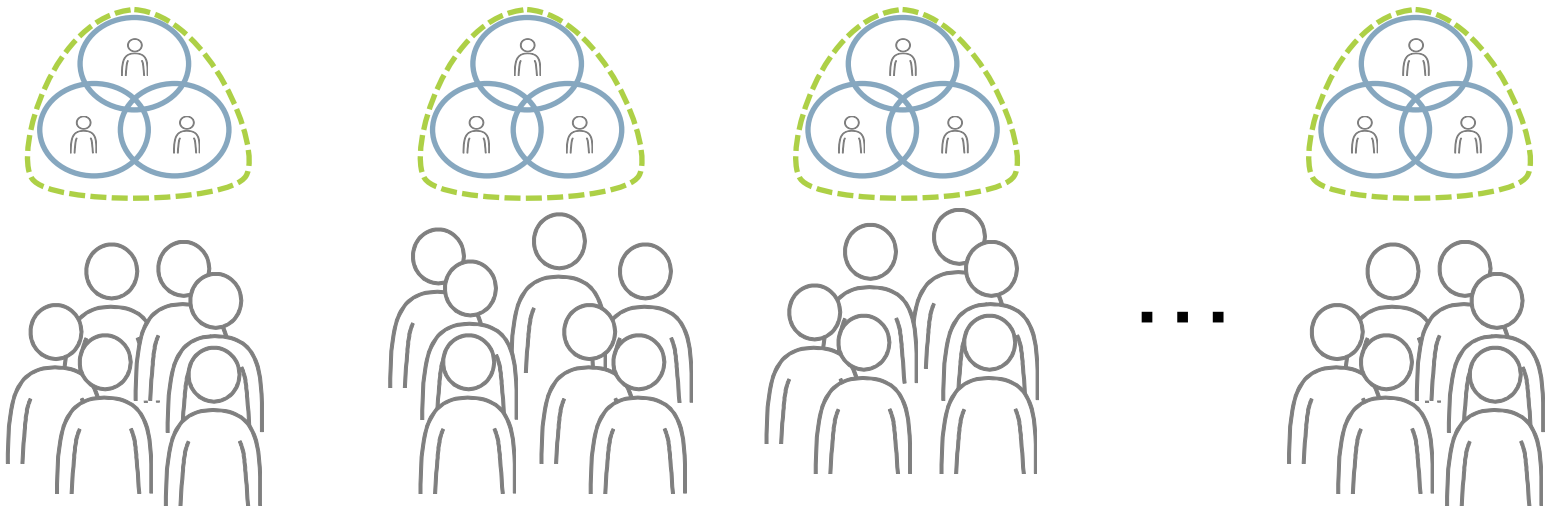
## Micro-Architecture

- Buy when non core
- Common Technologies

# The 3 Faces of Product Ownership



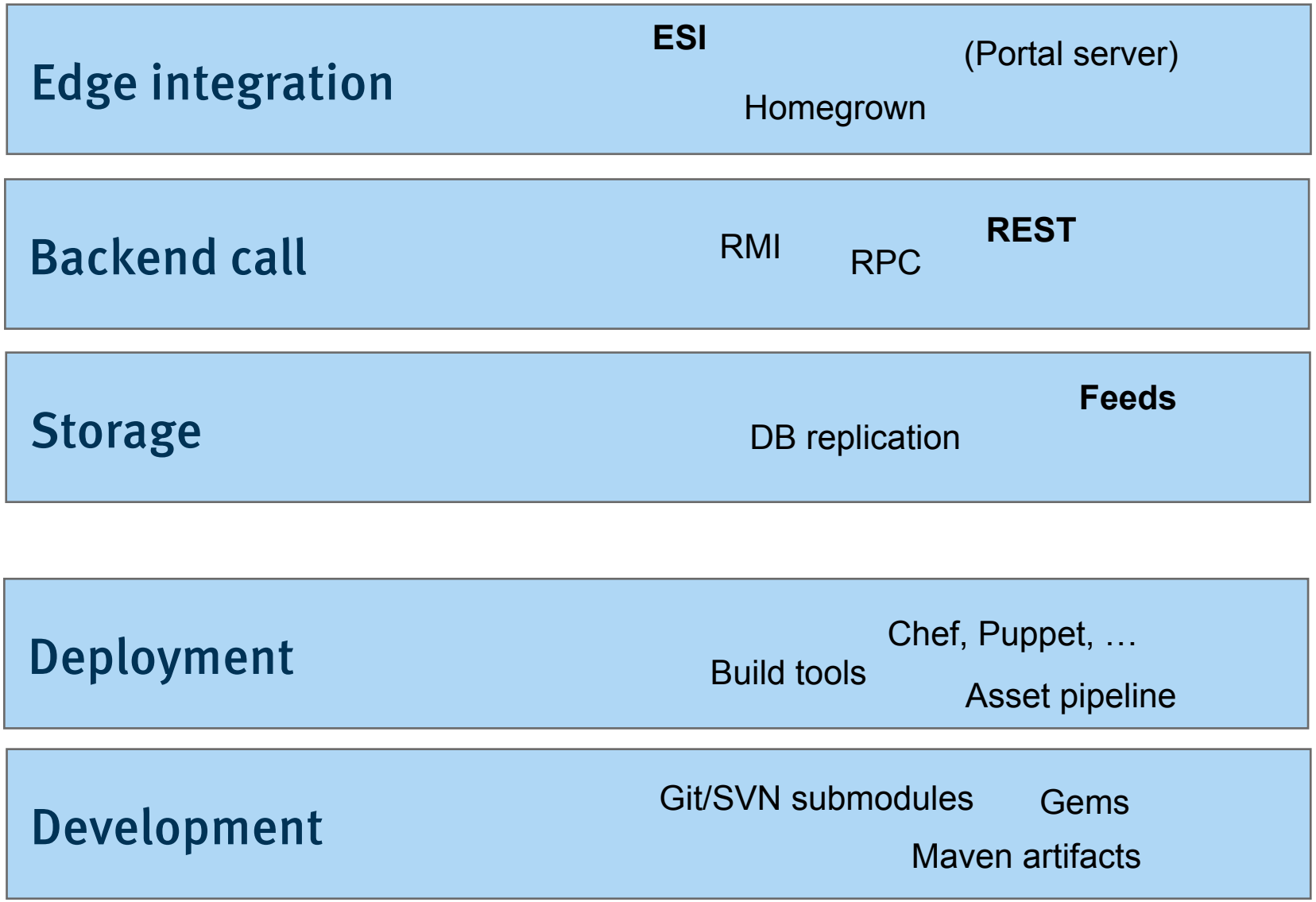
# Project start with distributed teams



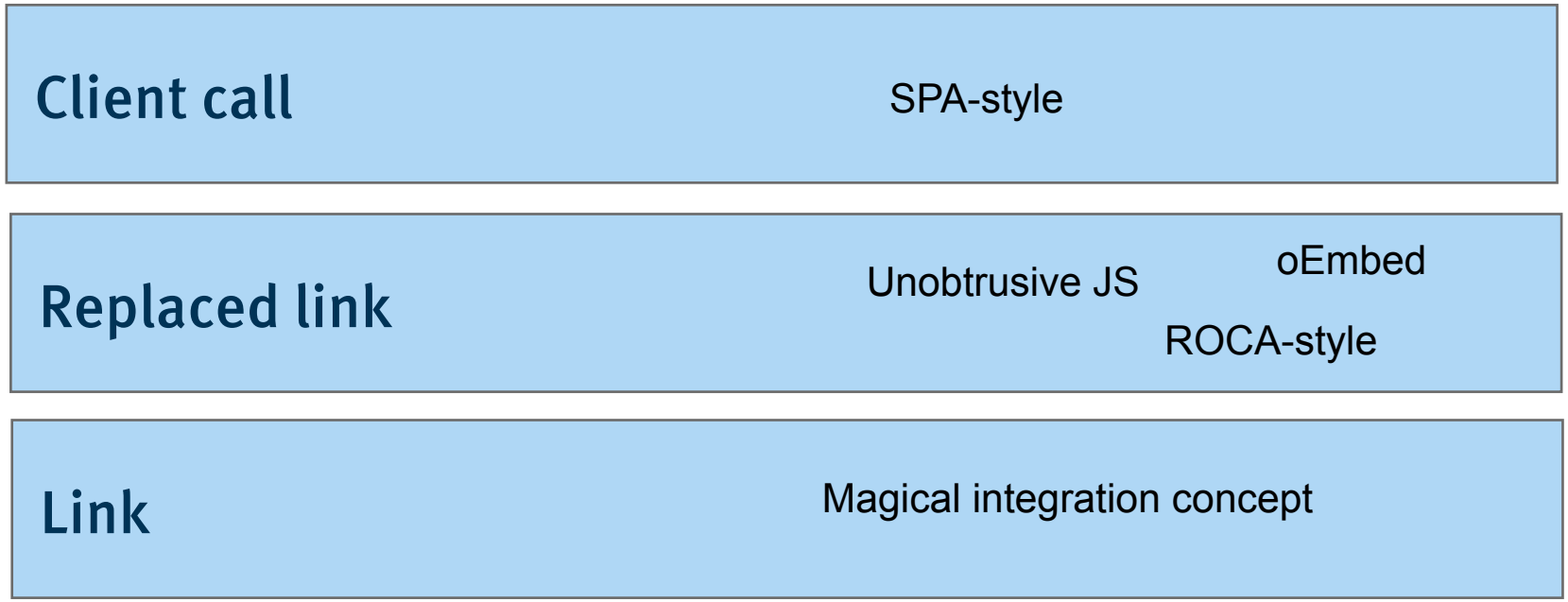

But how to deal with frontend integration?

# 5. Frontend Integration

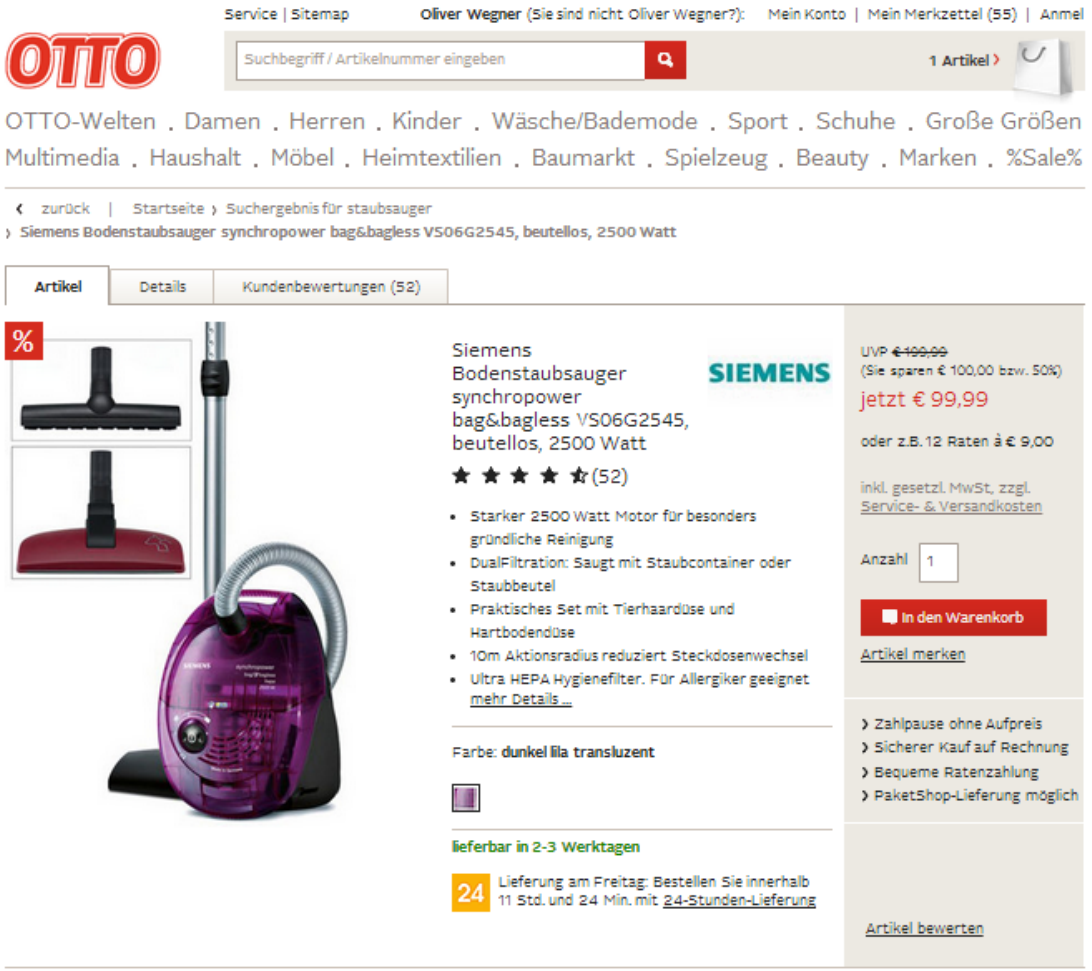
# Server-side integration options



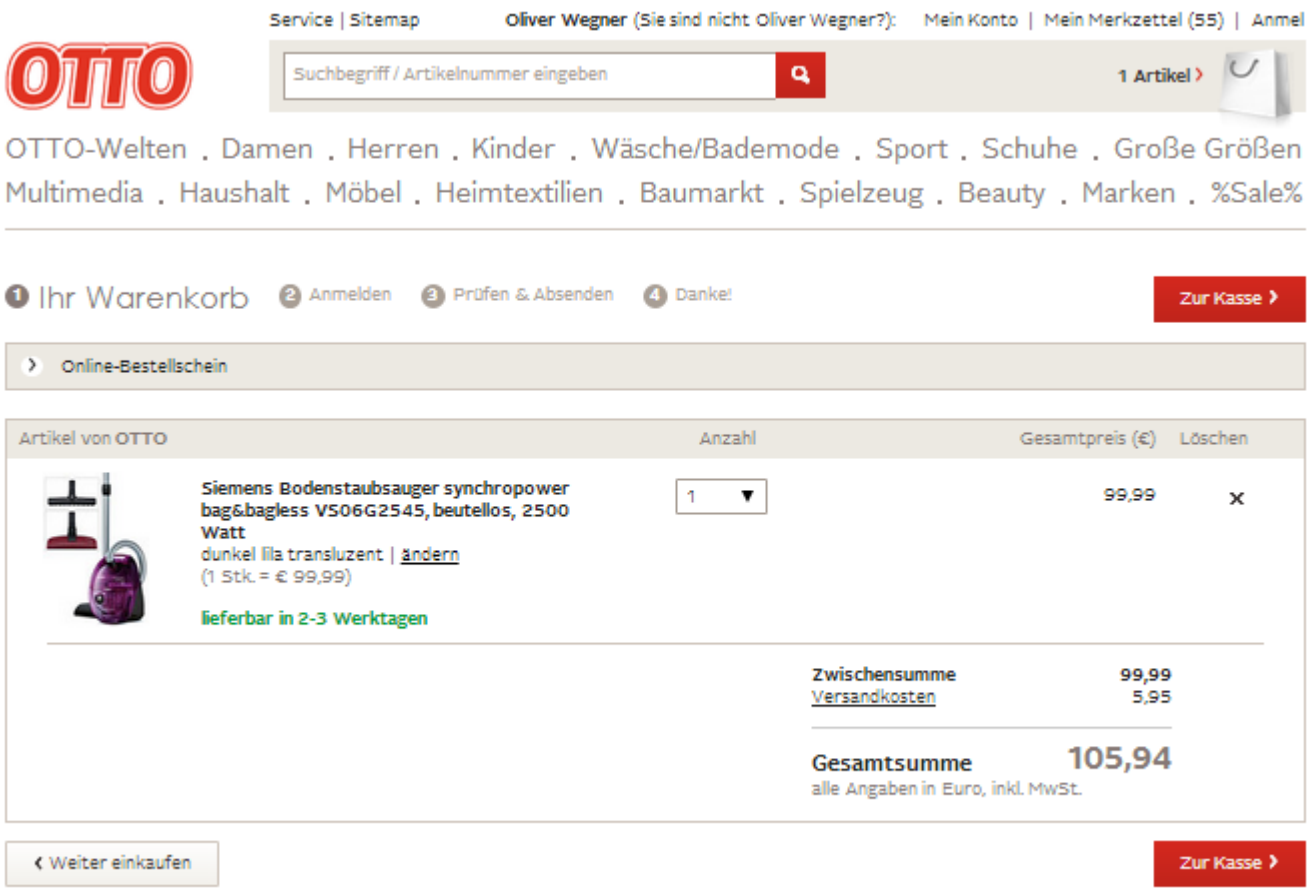
# Client-side integration options



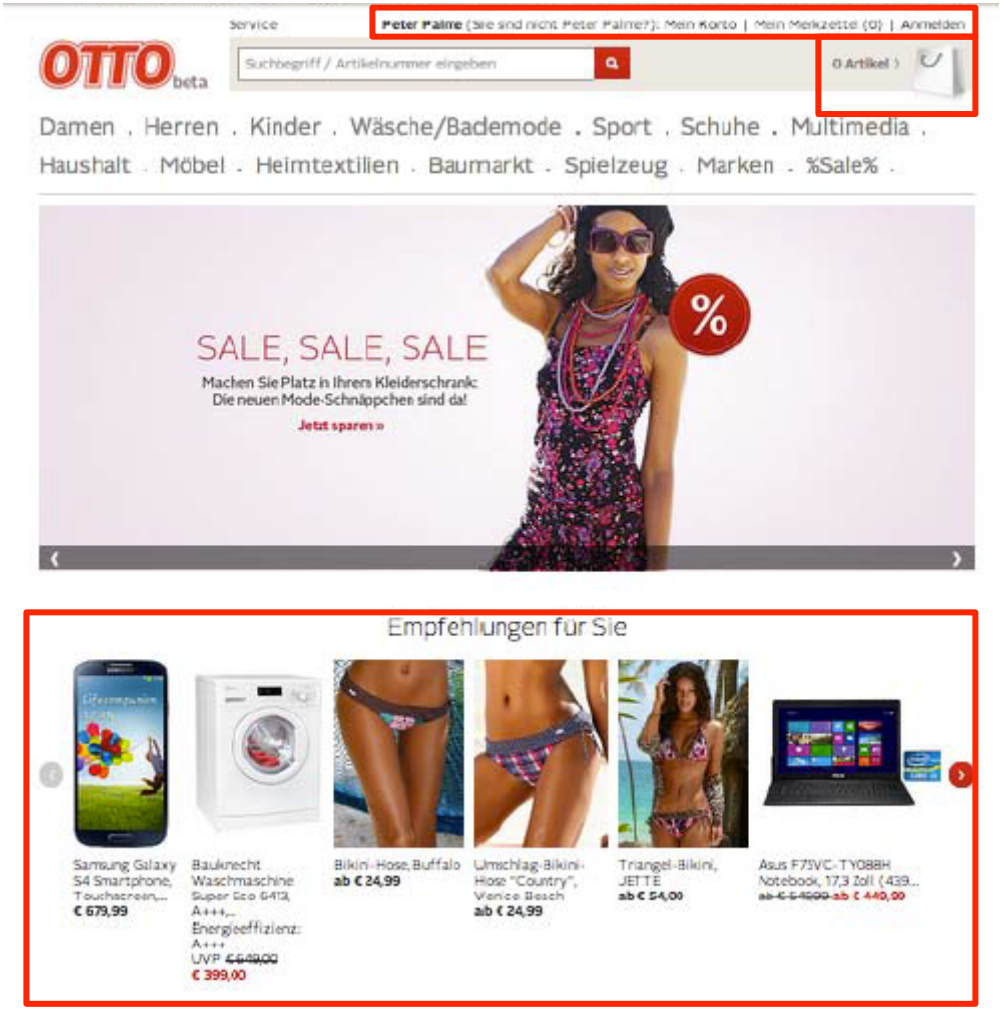
# Otto.de – detail view



# Otto.de - Basket



# Management of JS and CSS from each team



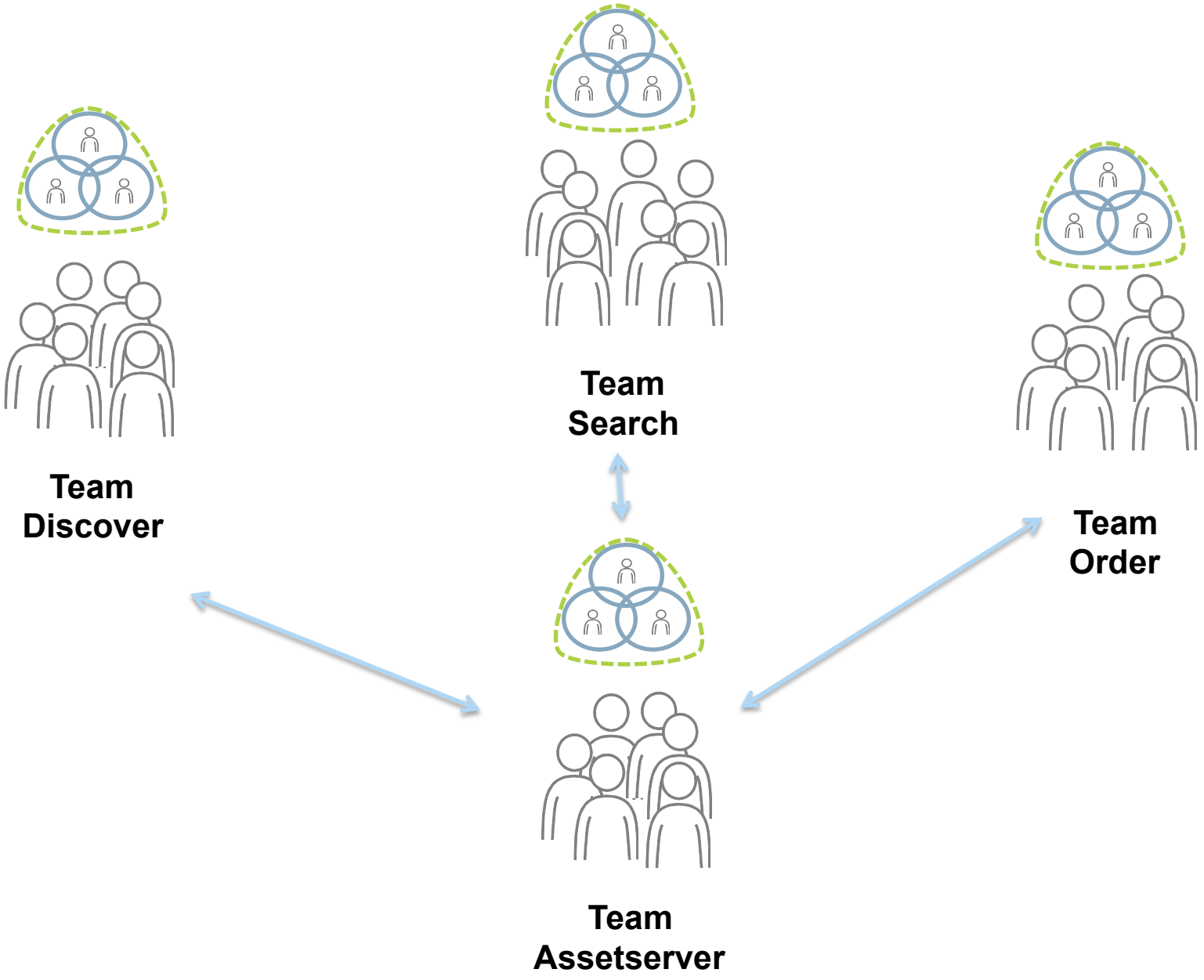

## System View

### Order

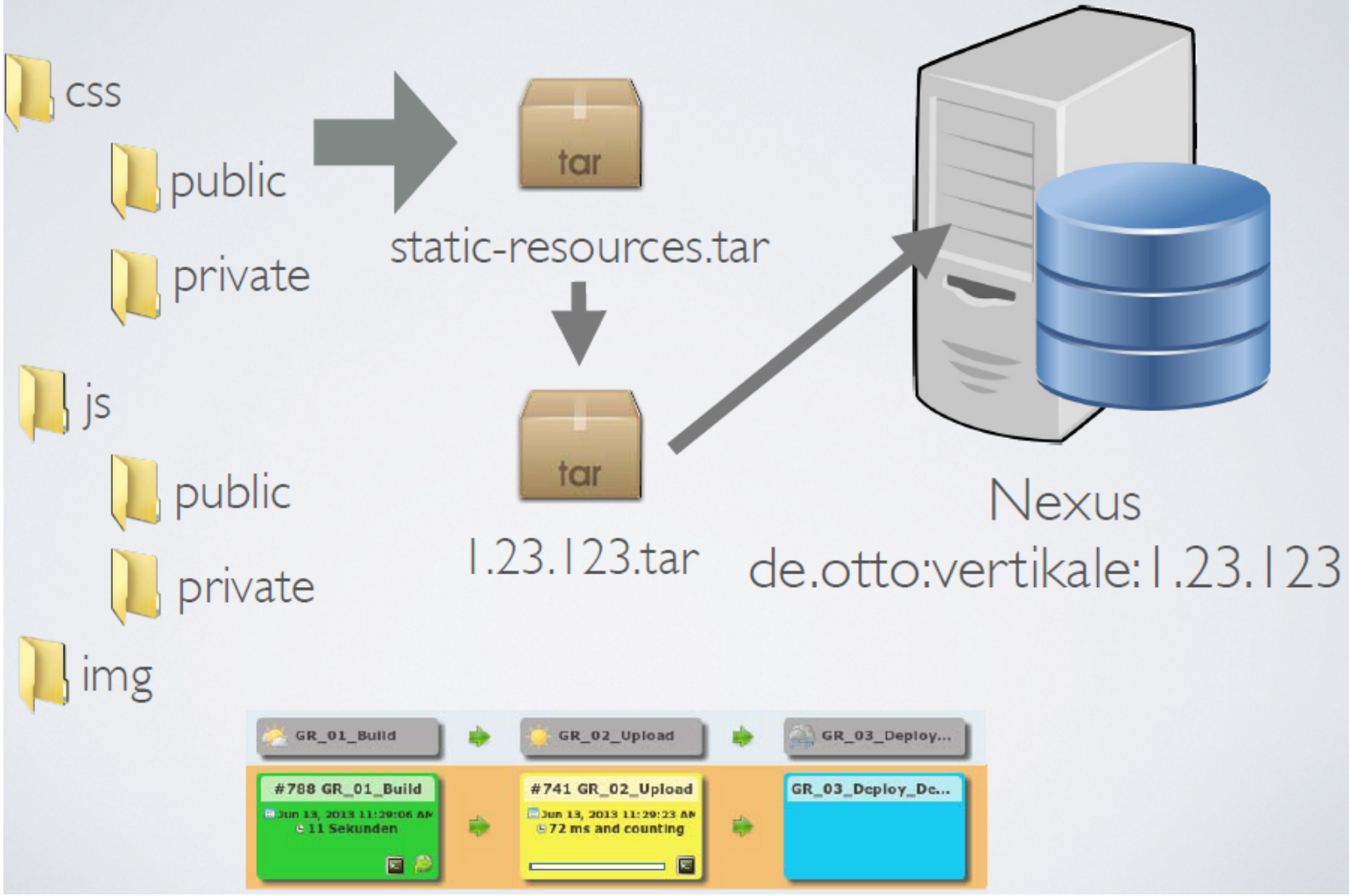

### User


### Personalization

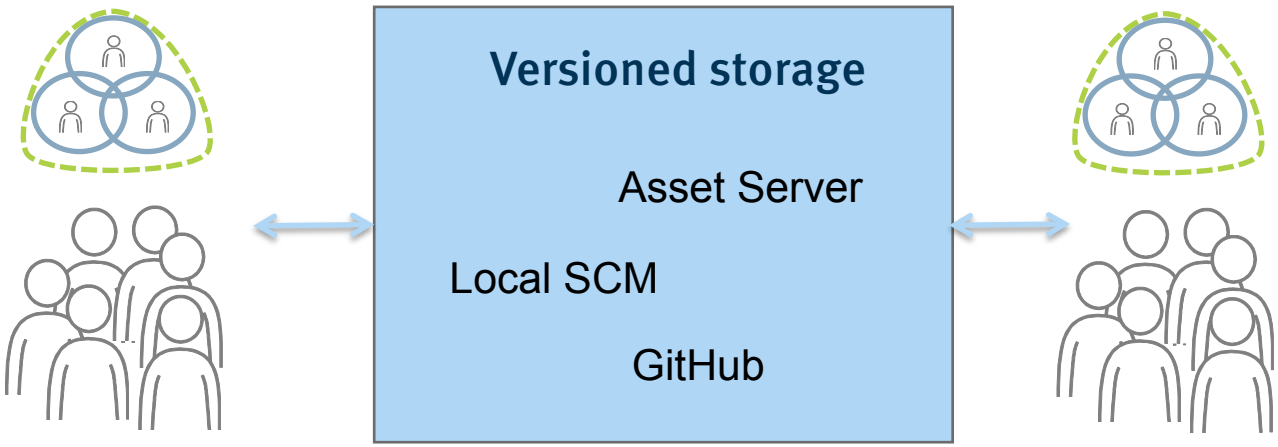
# Separate teams for horizontal aspects



# DEPLOYMENT

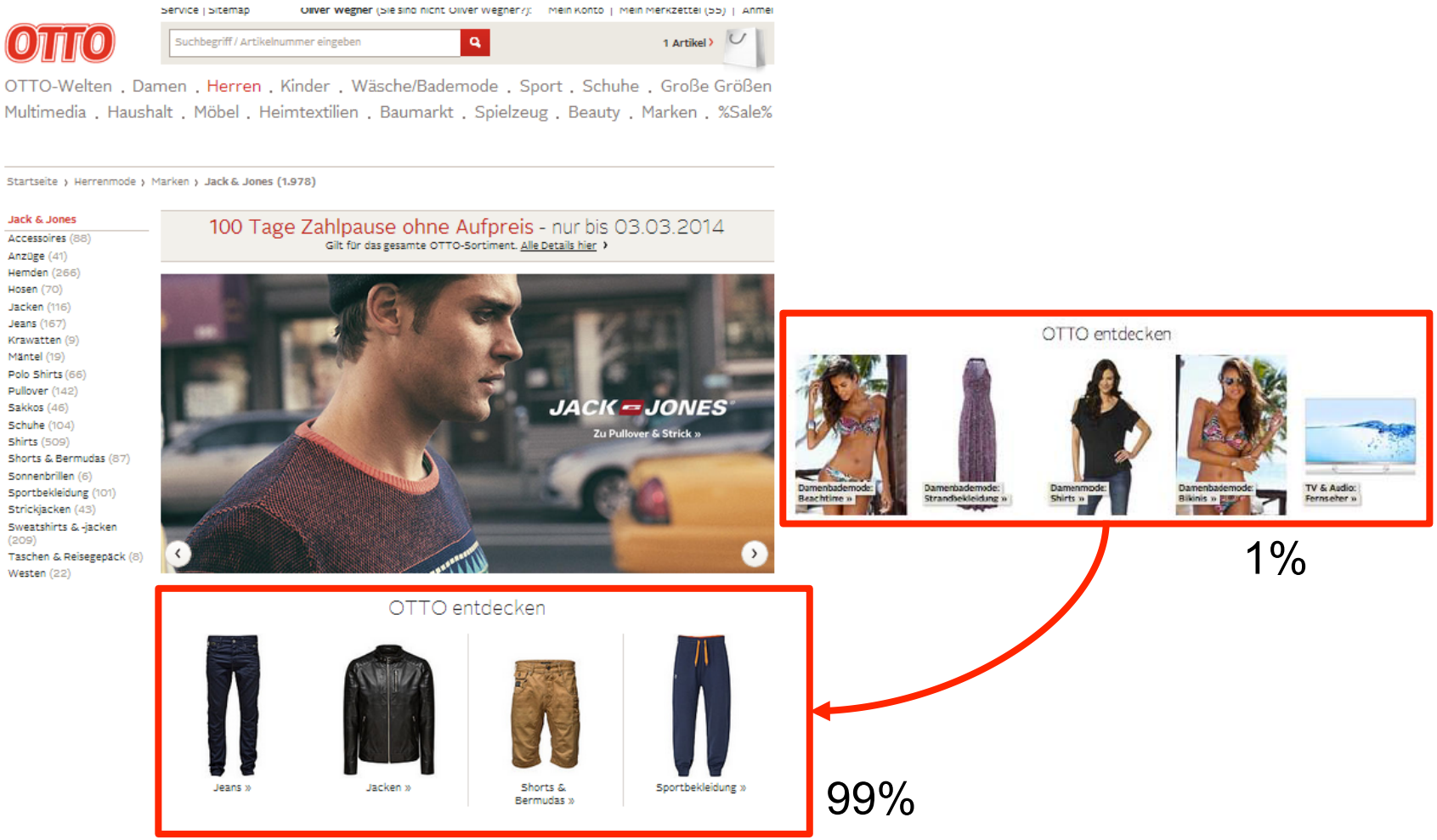

css
public
private
js
public
private
img
tar
static-resources.tar
tar
1.23.123.tar
Nexus
de.otto:vertikale:1.23.123
GR_01_Build
GR_02_Upload
GR_03_Deploy...
#788 GR_01_Build
Jun 13, 2013 11:29:06 AM
11 Sekunden
#741 GR_02_Upload
Jun 13, 2013 11:29:23 AM
72 ms and counting
GR_03_Deploy_De...

# Decoupling

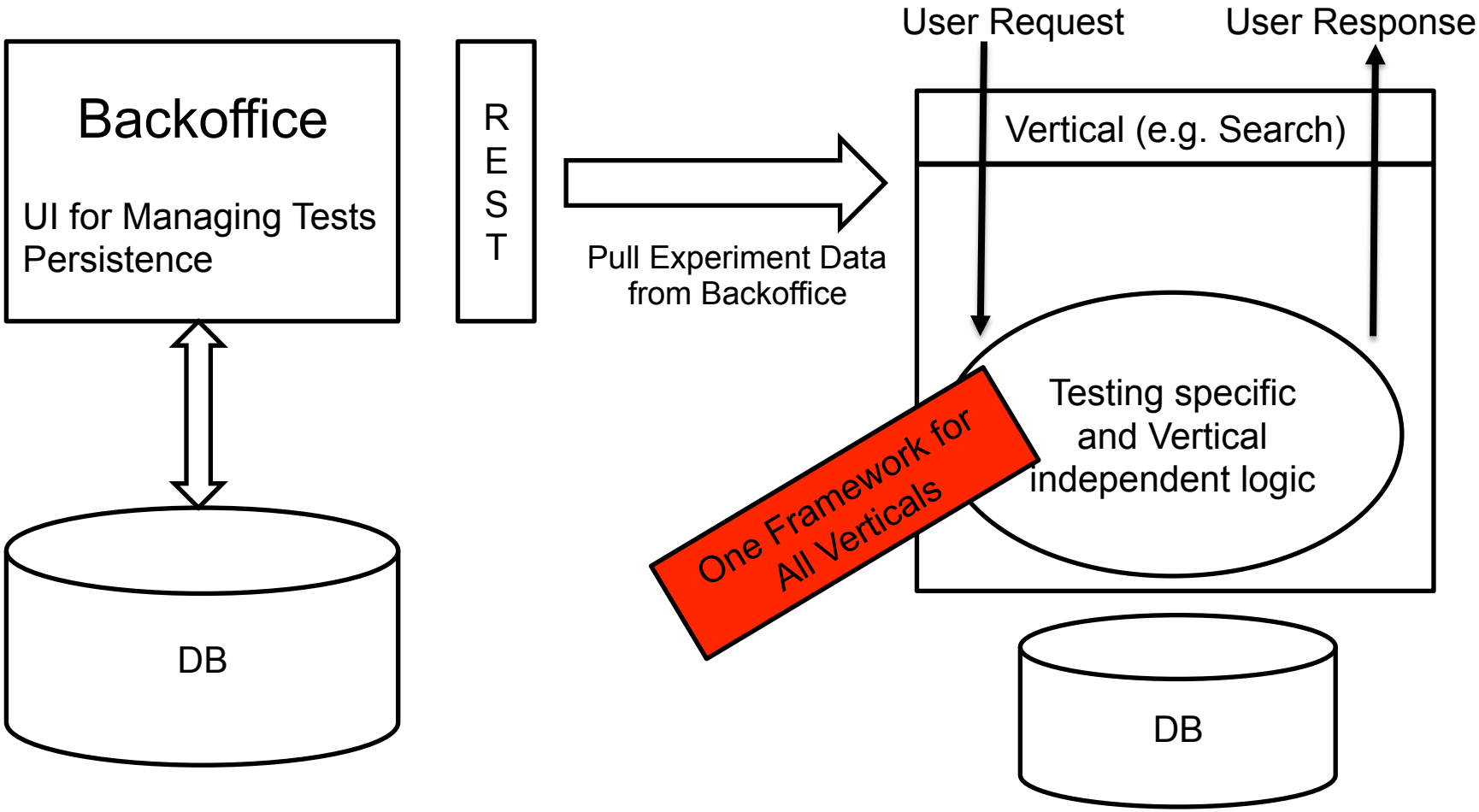


# 6. A/B-Testing

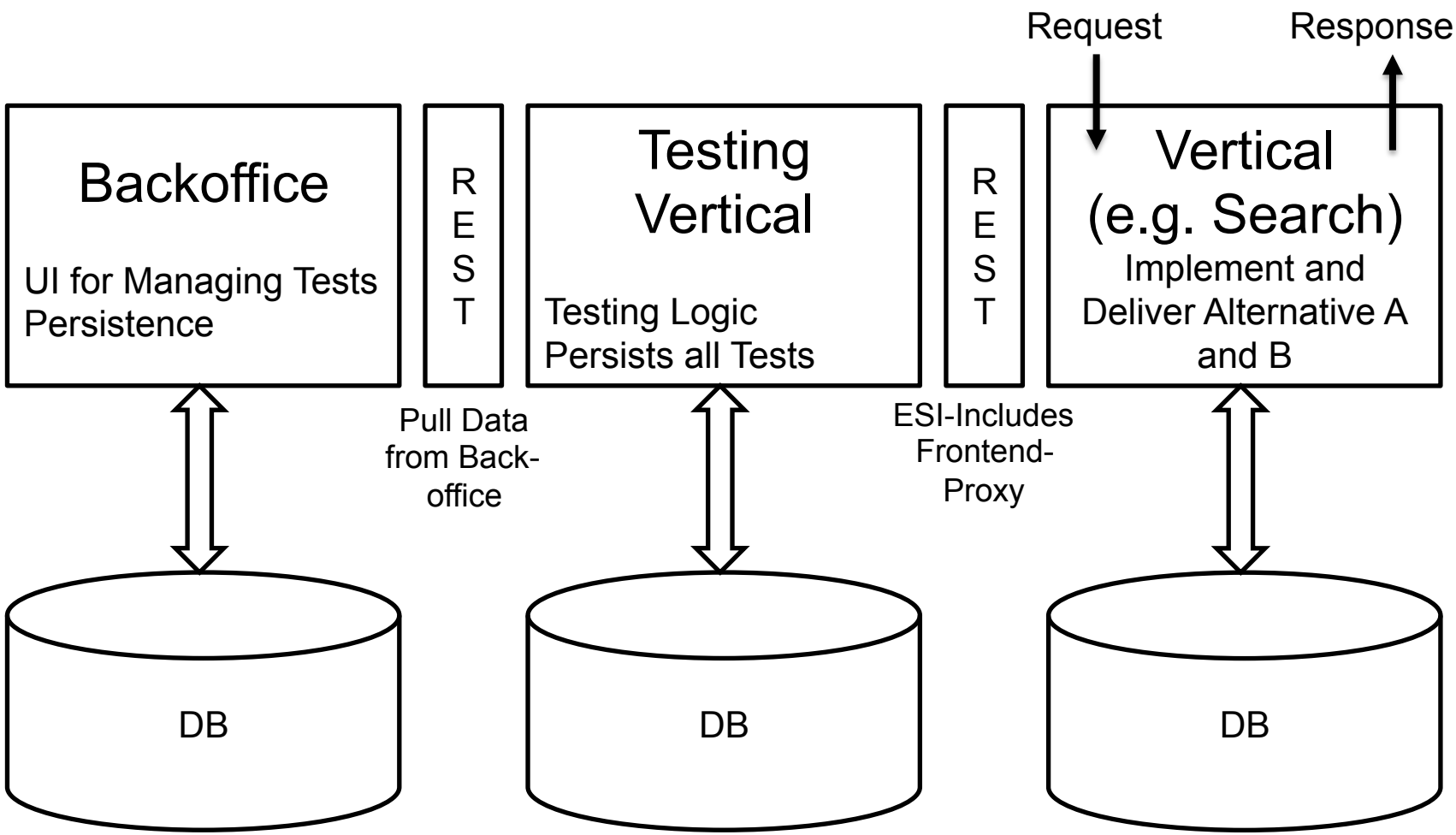
# AB-Testing in a distributed environment – What is an AB-Test



# Solution with a centralized framework every team has to include in their repository



# Solution with a dedicated Vertical and loose coupling



# 7. Conclusion

# Results of the project LHOTSE

2 years in total

Scaled to >100 people

Finished in budget

Finished in quality

Finished before schedule: October, 24th → 4 months earlier

Minimum Viable Product

# Lessons learned in applying a system-of-systems approach

Independent, autonomous systems for maximum decoupling

Prefer “pull” to “push” sharing

Strict macro architecture rules

Minimize cross-functional concerns

Address cross-functional concerns

Minimize need for coordination

Be skeptical of “easy” solutions

Teams with their own decisions

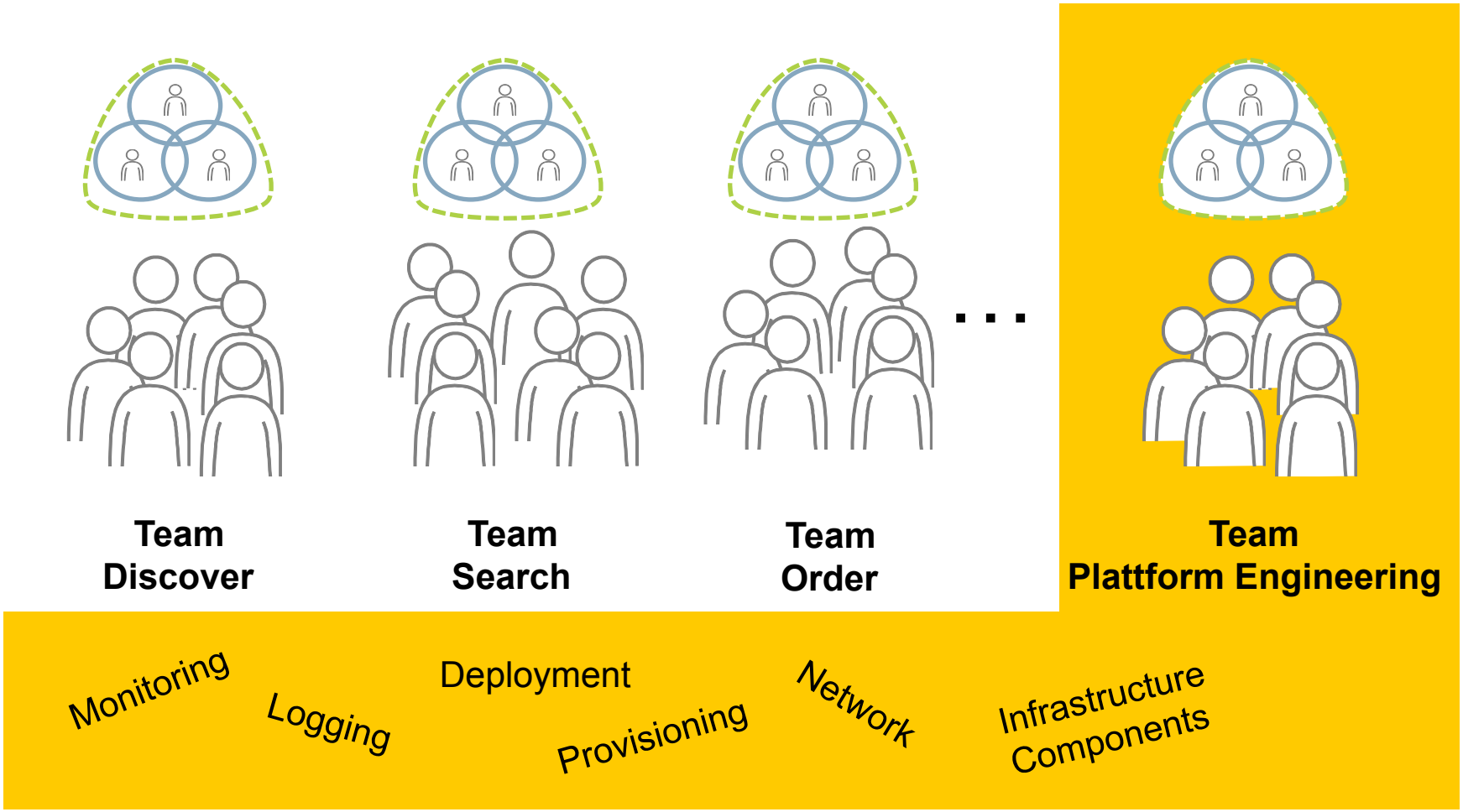
@olliwegner
@stilkov

QCon London 2014

# Please evaluate our talk via the mobile app!

# BACKUP

# Plattform Engineering delivers the basic infrastructure for all verticals



## Microservices?

**Microservices Approach**

- Very small
- 100s
- Does one thing only
- Separate client (?)

**Vertical Systems Approach**

- Medium-sized
- 10s
- Small # of related things
- Includes UI