
codecentric

Microservices and Conversion Hunting:

Build Architectures for Changeability

Bernd Zuther - IT Consultant Big Data
codecentric AG

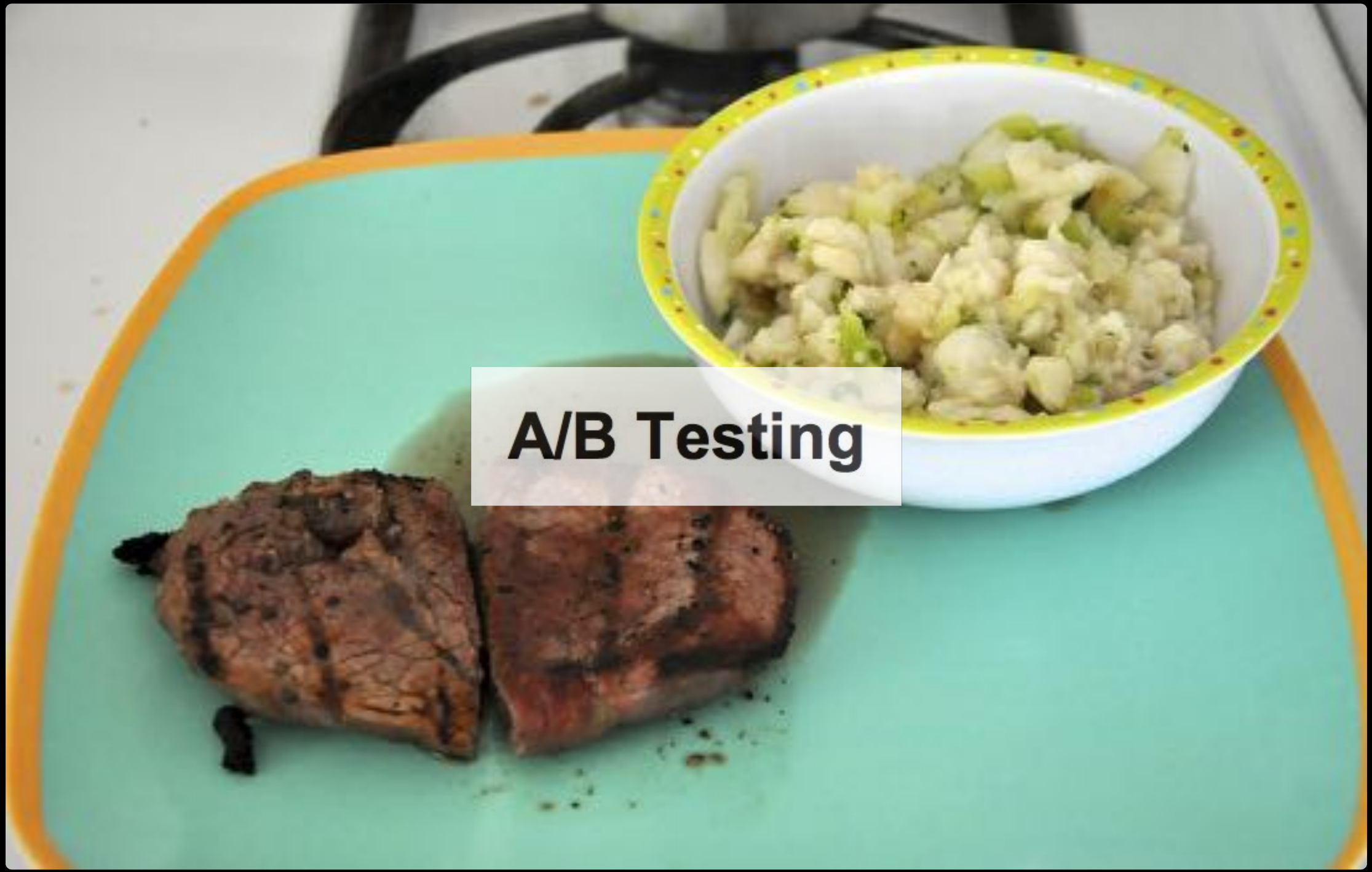

Hello, my name is
Bernd Zuther
@Bernd_Z
github.com/zutherb
bernd.zuther@codecentric.de


Survival of the fittest


Be faster than your competitors


How can we increase sales?


Mike Tyson
Everyone has a plan until they get punched in the face.


A/B Testing

Conversion

- act of converting site visitors into paying customers
- proportion of visits to a website which take action to go beyond a website visit

$$\text{Conversion Rate} = \frac{\text{Number of Goals achieved}}{\text{Visits}}$$

FeatureToggle

JAVA

```
public class TopSellerRecommendationPanel extends AbstractShopBasePanel {

    @SpringBean
    private FeatureToogleBean featureToogleBean;

    ...

    @Override
    protected void onConfigure() {
      super.onConfigure();
      setVisible(featureToogleBean.isTopSellerFeatureEnabled());
    }

    ...

}
```


What have we done?


How does it work?

Disadvantages that we had with the way we develop software

Disadvantages that we had with the way we develop software

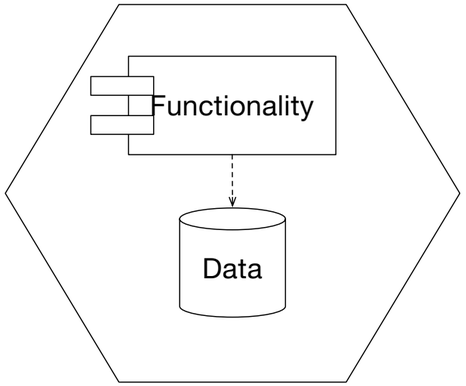
- Development:
  - Long feature development cycles
  - Long feedback times
  - Long build time
  - Strange behavior after changes
  - Application scalability

Disadvantages that we had with the way we develop software

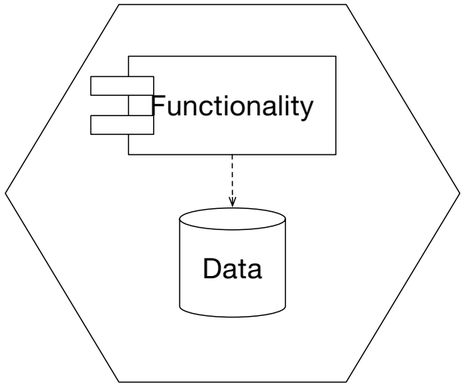
- Development:
  - Long feature development cycles
  - Long feedback times
  - Long build time
  - Strange behavior after changes
  - Application scalability
- Product Management:
  - Long time to market
  - A/B testing difficult to achieve
  - Scalability of workload


How can we build software smarter?

Microservice

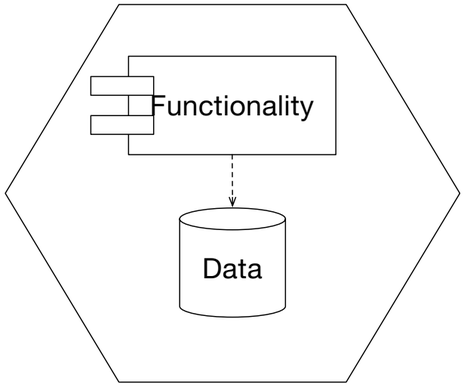


Microservice



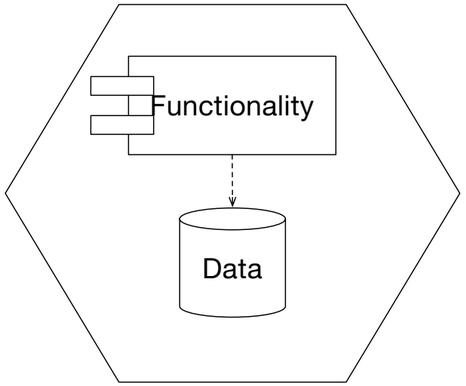
- fits in one brain,

Microservice



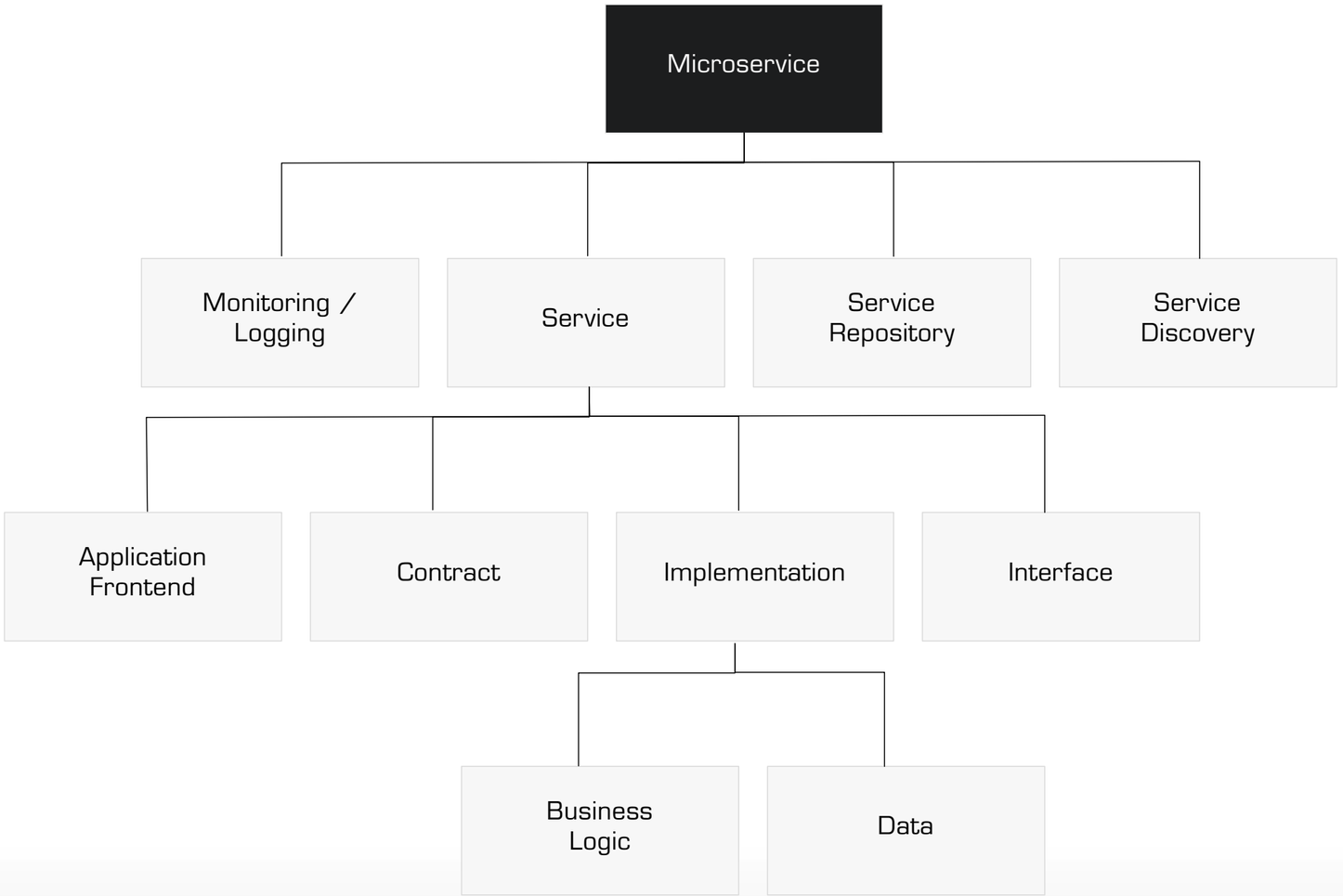
- fits in one brain,
- designed for replaceability,

Microservice

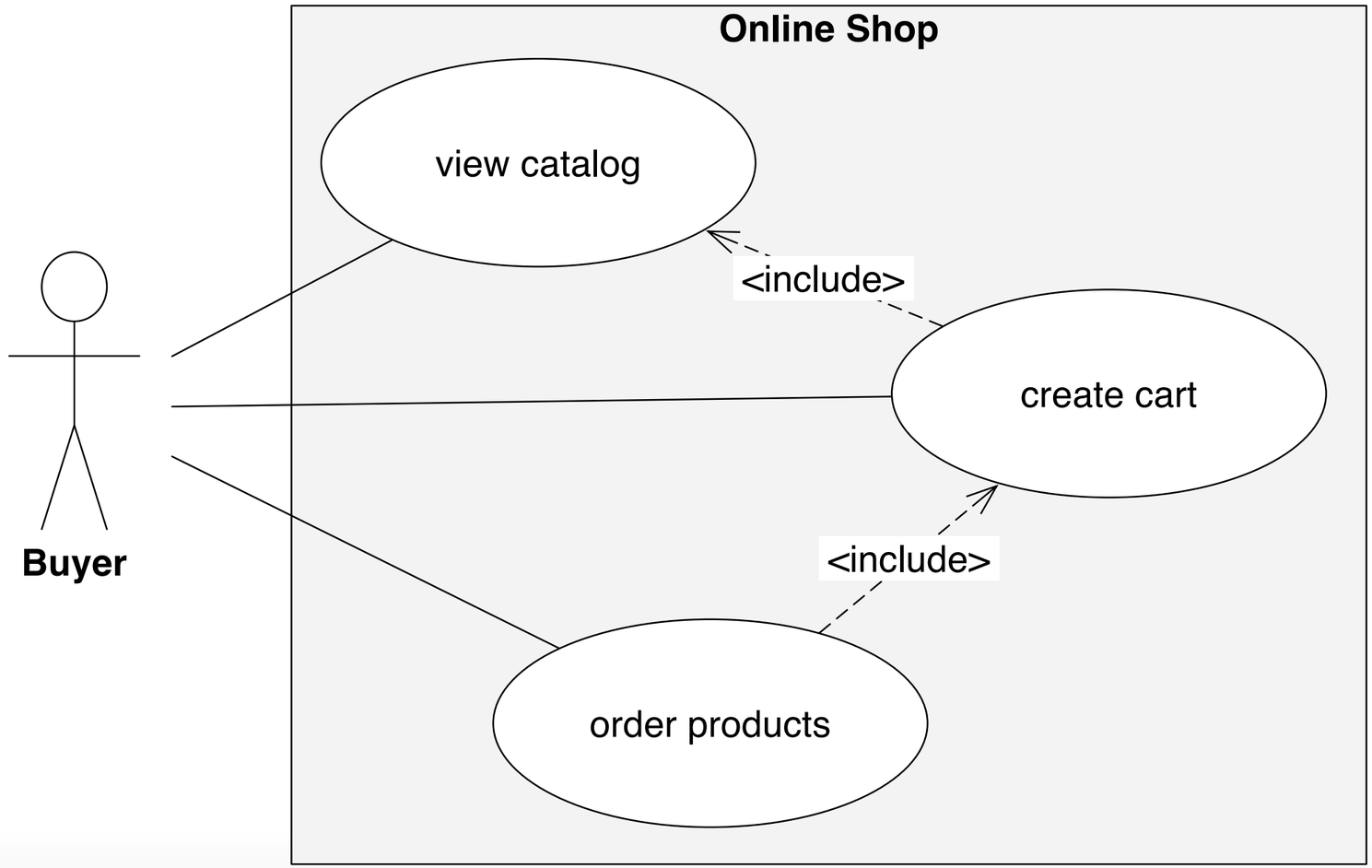


- fits in one brain,
- designed for replaceability,
- autonomy (organisation & technology)

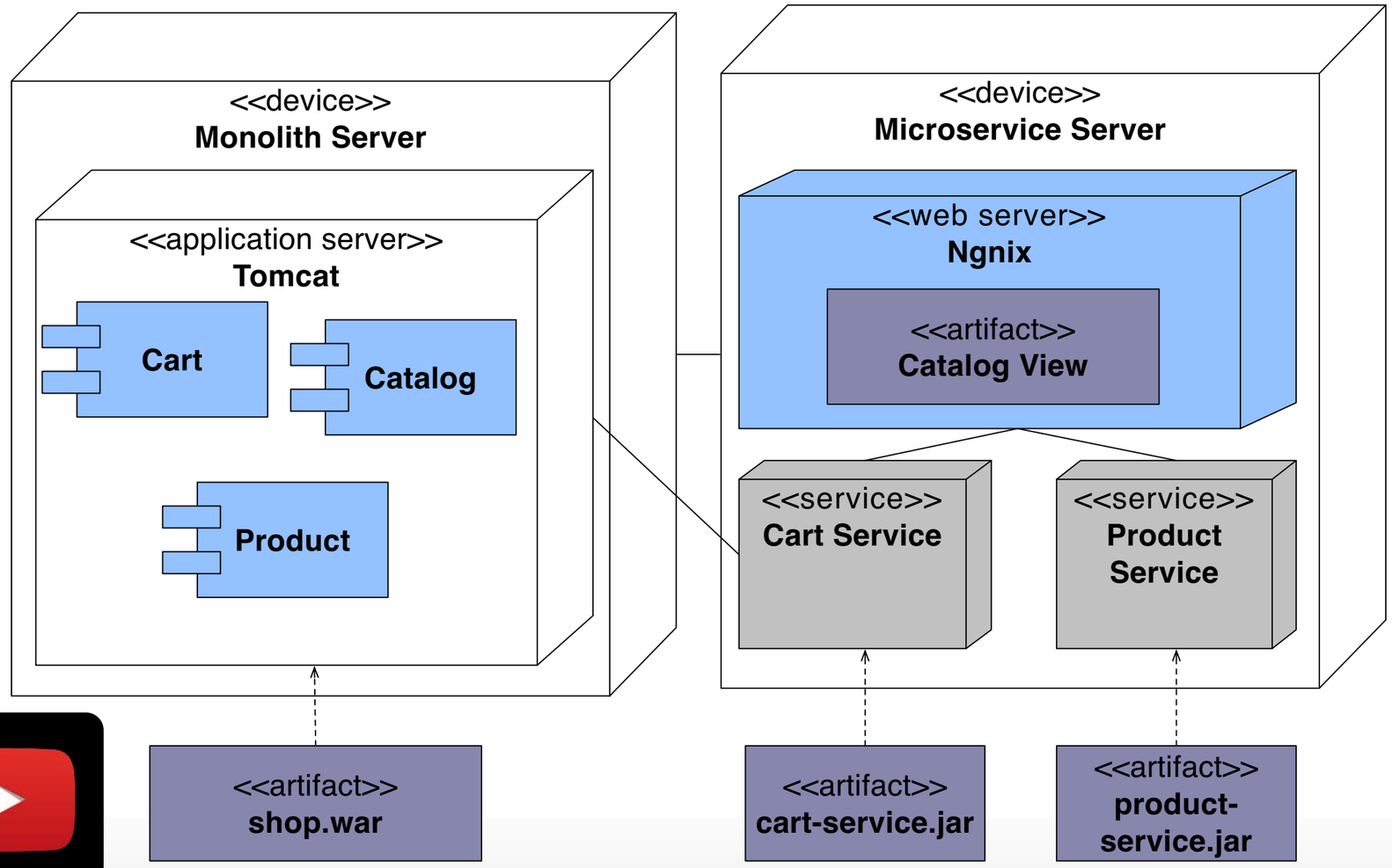
Microservice Taxonomy



Online Shop - Use cases



Online Shop - Migration



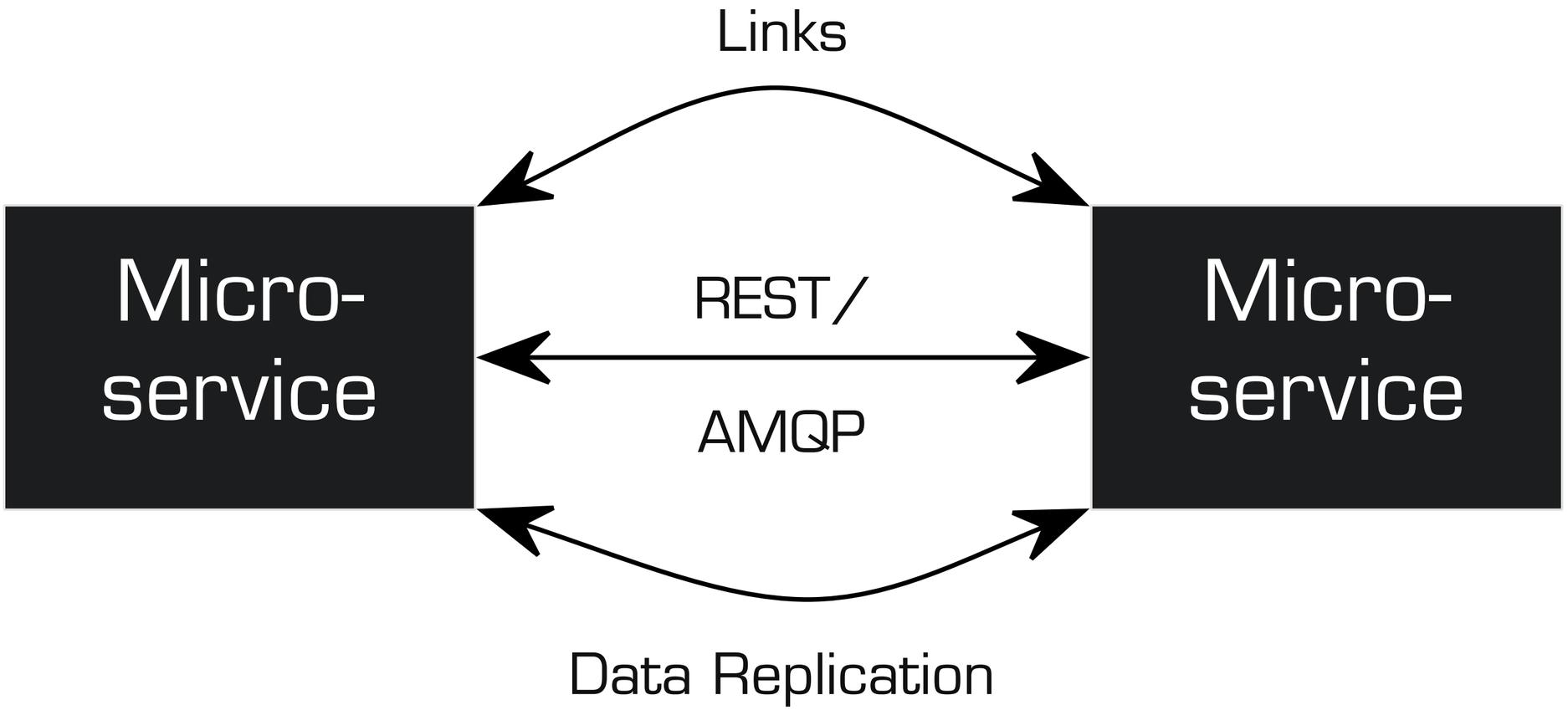

Flexibility

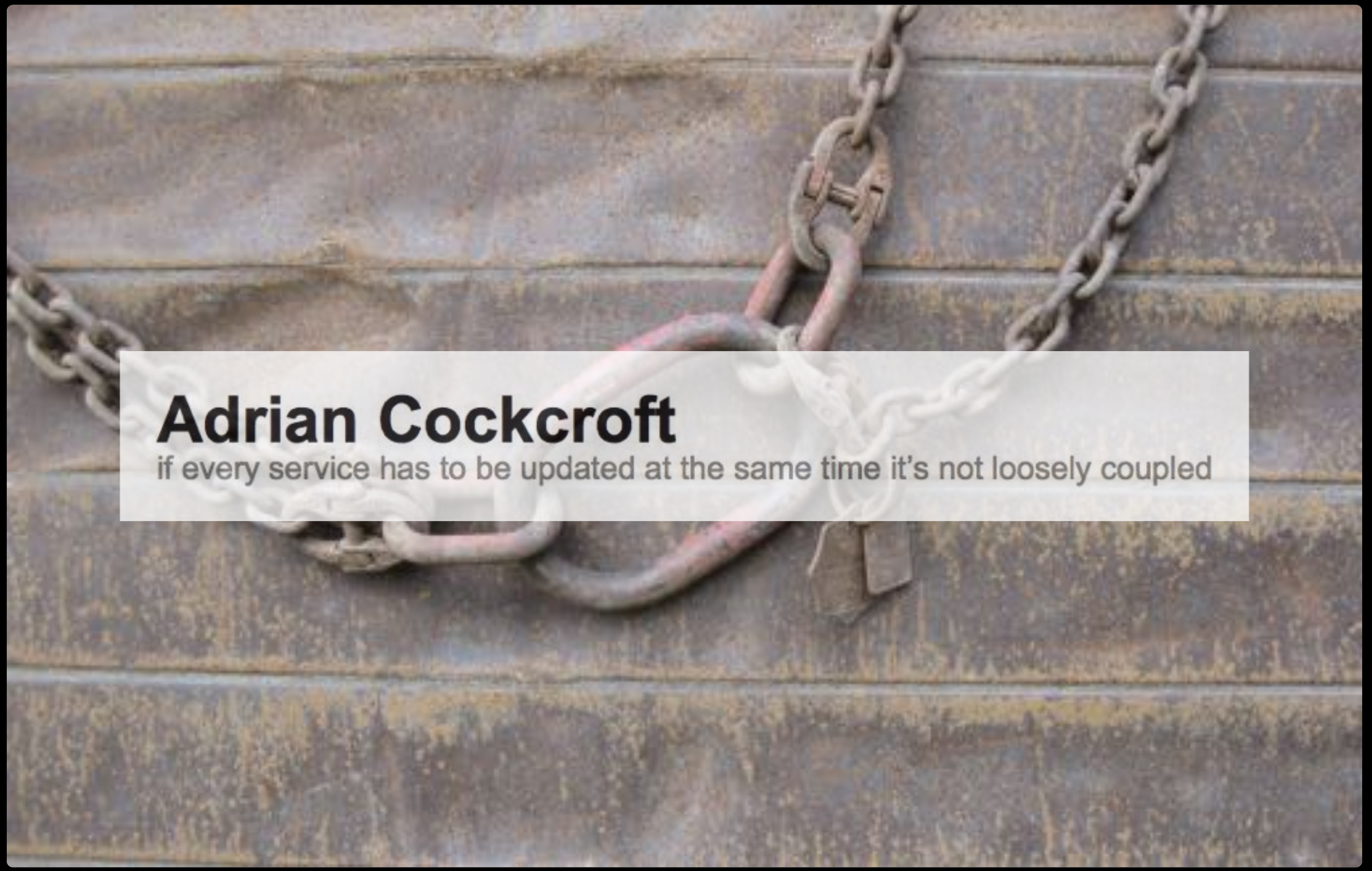

Improve feedback times


Scalability


Challenges

Microservice - Communication




Adrian Cockcroft
if every service has to be updated at the same time it's not loosely coupled

Postel's Law (1980)

RFC 761 (Transmission Control Protocol)

be conservative in what you do, be liberal in what you accept from others

Postel’s Law

JAVASCRIPT

```
{
        firstname: "Bernd",
        lastname: "Zuther",
        street: "Elsenheimerstraße 55a",
        city: "München",
        zip: 80687
}
```

Postel’s Law

JAVASCRIPT

```
{
        firstname: "Bernd",
        lastname: "Zuther",
        birthday: "1983/08/15",
        street: "Elsenheimerstraße 55a",
        city: "München",
        zip: 80687
}
```

Postel's Law

JAVASCRIPT

```
{
        firstname: "Bernd",
        lastname: "Zuther",
        birthday: "1983/08/15",
        addresses : [{
          street: "Elsenheimerstraße 55a",
          city: "München",
          zip: 80687,
          type: "WORK"
        }]
}
```


25

Postel's Law

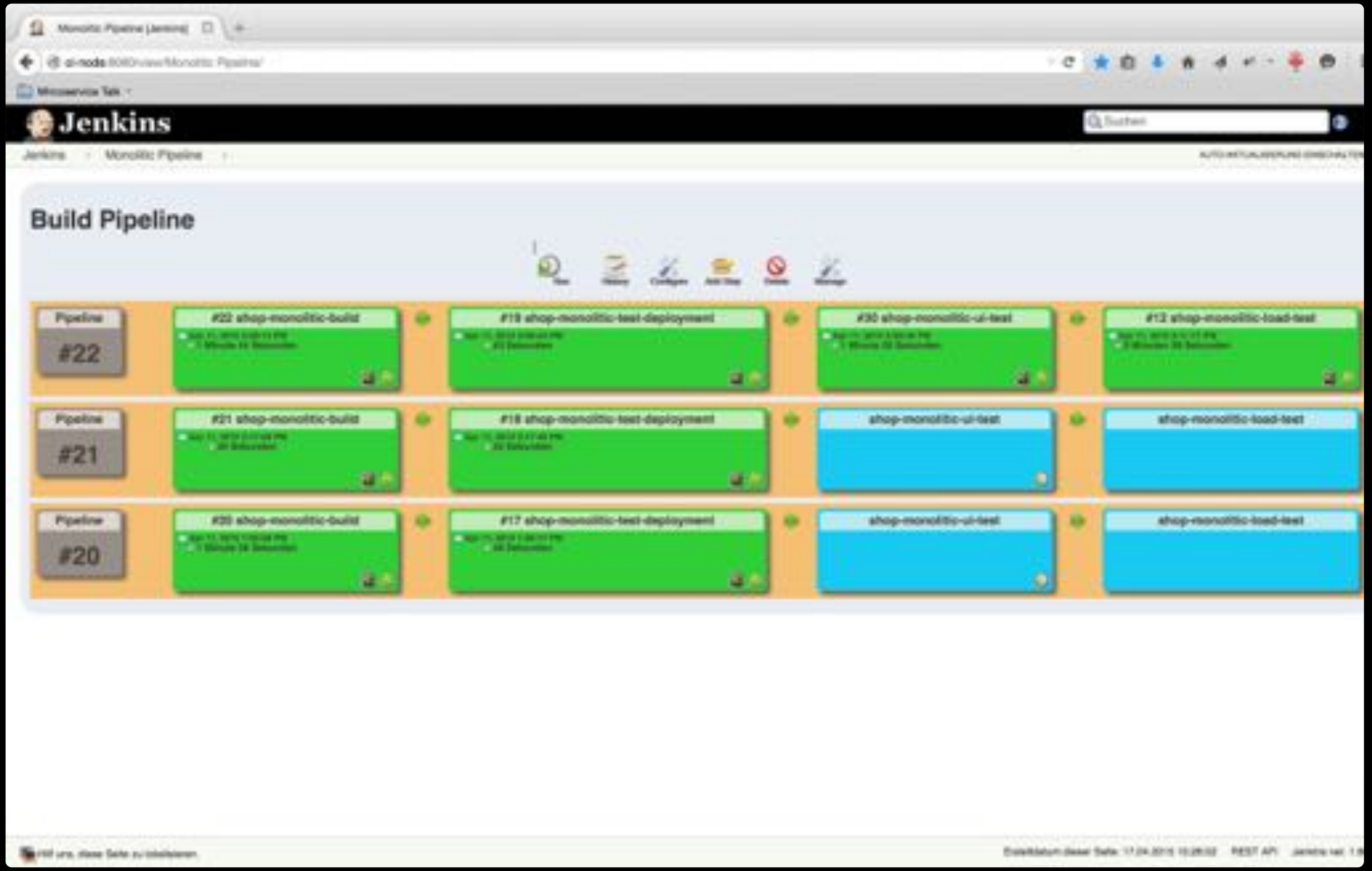
JAVASCRIPT

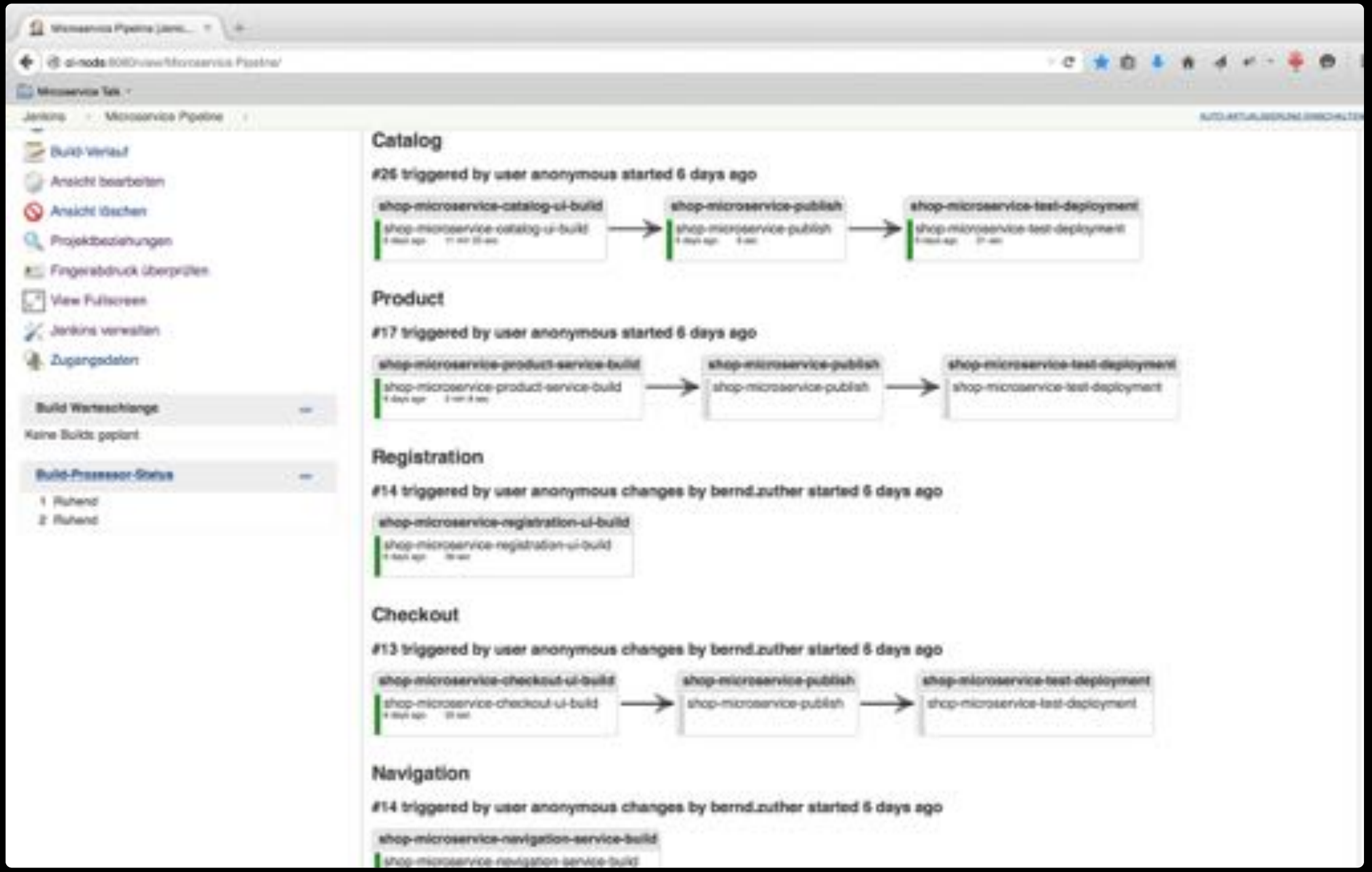
```
{
        firstname: "Bernd",
        lastname: "Zuther",
        birthday: "1983/08/15",
        street: "Elsenheimerstraße 55a",
        city: "München",
        zip: 80687,
        addresses : [{
          street: "Elsenheimerstraße 55a",
          city: "München",
          zip: 80687,
          type: "WORK"
        }]
}
```

Alternatives

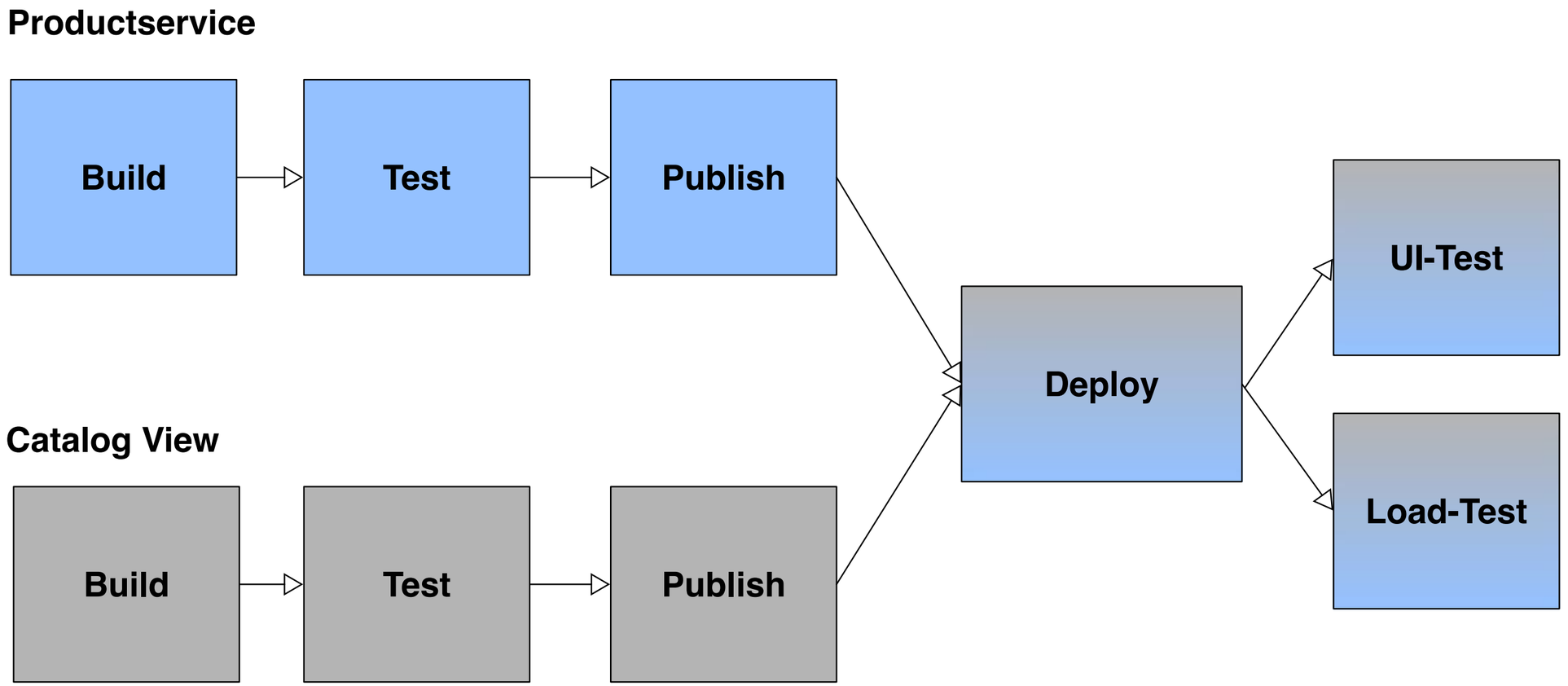
- **URL**:
  - simply whack the API version into the URL, e.g. https://shop/api/v2/cart/all
- **Custom request header**:
  - same URL as before but add a header “api-version: 2”
- **Accept header**:
  - modify accept header to specify the version, e.g. “Accept: application/vnd.cart.v2+json”


What change on your build pipeline?


Jenkins
Jenkins
Monolitic Pipeline
Build Pipeline
Pipeline
#22
#22 shop-monolitic-build
#19 shop-monolitic-test-deployment
#30 shop-monolitic-ui-test
#12 shop-monolitic-load-test
Pipeline
#21
#21 shop-monolitic-build
#18 shop-monolitic-test-deployment
shop-monolitic-ui-test
shop-monolitic-load-test
Pipeline
#20
#20 shop-monolitic-build
#17 shop-monolitic-test-deployment
shop-monolitic-ui-test
shop-monolitic-load-test


Jenkins
Microservice Pipeline
Build Verlauf
Ansicht bearbeiten
Ansicht löschen
Projektbeziehungen
Fingerabdruck überprüfen
View Fullscreen
Jenkins verwalten
Zugangsdaten
Build Warteschlange
Keine Builds geplant
Build-Prozessor-Status
1 Ruhend
2 Ruhend
Catalog
#26 triggered by user anonymous started 6 days ago
shop-microservice-catalog-ui-build
shop-microservice-catalog-ui-build
shop-microservice-publish
shop-microservice-publish
shop-microservice-test-deployment
shop-microservice-test-deployment
Product
#17 triggered by user anonymous started 6 days ago
shop-microservice-product-service-build
shop-microservice-product-service-build
shop-microservice-publish
shop-microservice-publish
shop-microservice-test-deployment
shop-microservice-test-deployment
Registration
#14 triggered by user anonymous changes by bernd.zuther started 6 days ago
shop-microservice-registration-ui-build
shop-microservice-registration-ui-build
Checkout
#13 triggered by user anonymous changes by bernd.zuther started 6 days ago
shop-microservice-checkout-ui-build
shop-microservice-checkout-ui-build
shop-microservice-publish
shop-microservice-publish
shop-microservice-test-deployment
shop-microservice-test-deployment
Navigation
#14 triggered by user anonymous changes by bernd.zuther started 6 days ago
shop-microservice-navigation-service-build

Multi Deployment Pipeline

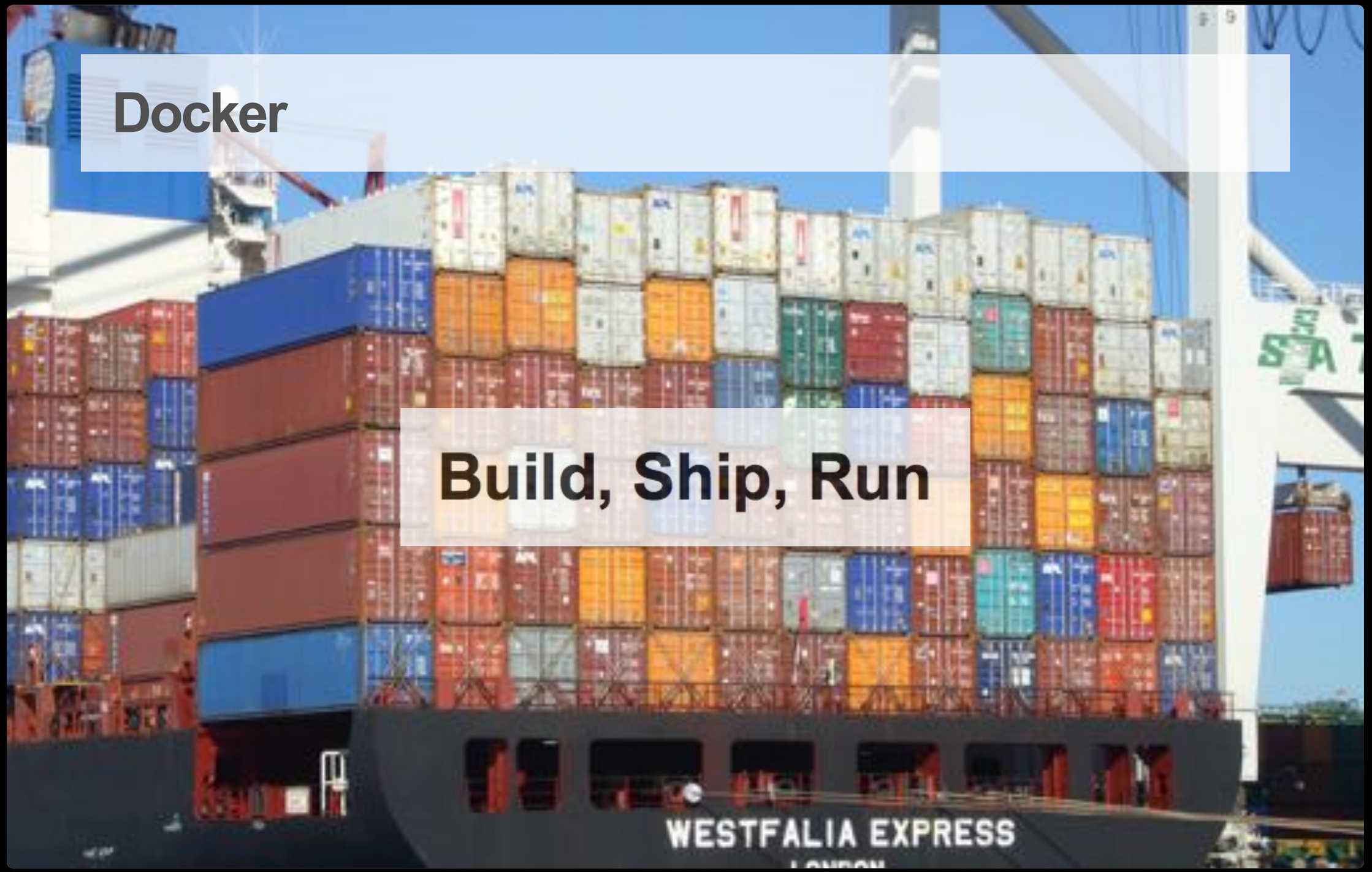



Improve Build Times


Deployments get more complex

How can we package and describe our application?

Docker



Docker-Workflow

BASH

```
bz@cc1 $ docker build -t zutherb/product-service .
bz@cc1 $ docker push zutherb/product-service
bz@cc2 $ docker pull zutherb/product-service
bz@cc2 $ docker run zutherb/product-service
bz@cc2 $ docker ps
CONTAINER ID        IMAGE                                  COMMAND                  CREATED
87bb5524067d         zutherb/product-service:latest       "/product-0.6/bin/pr   14 seconds
```

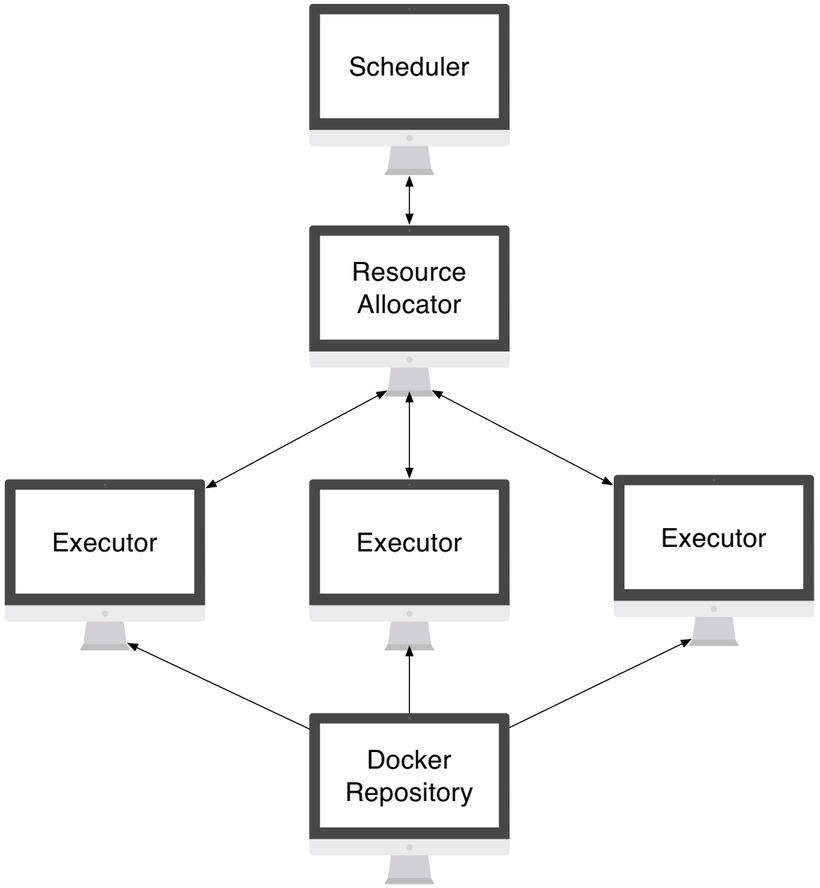
Container-Link

BASH

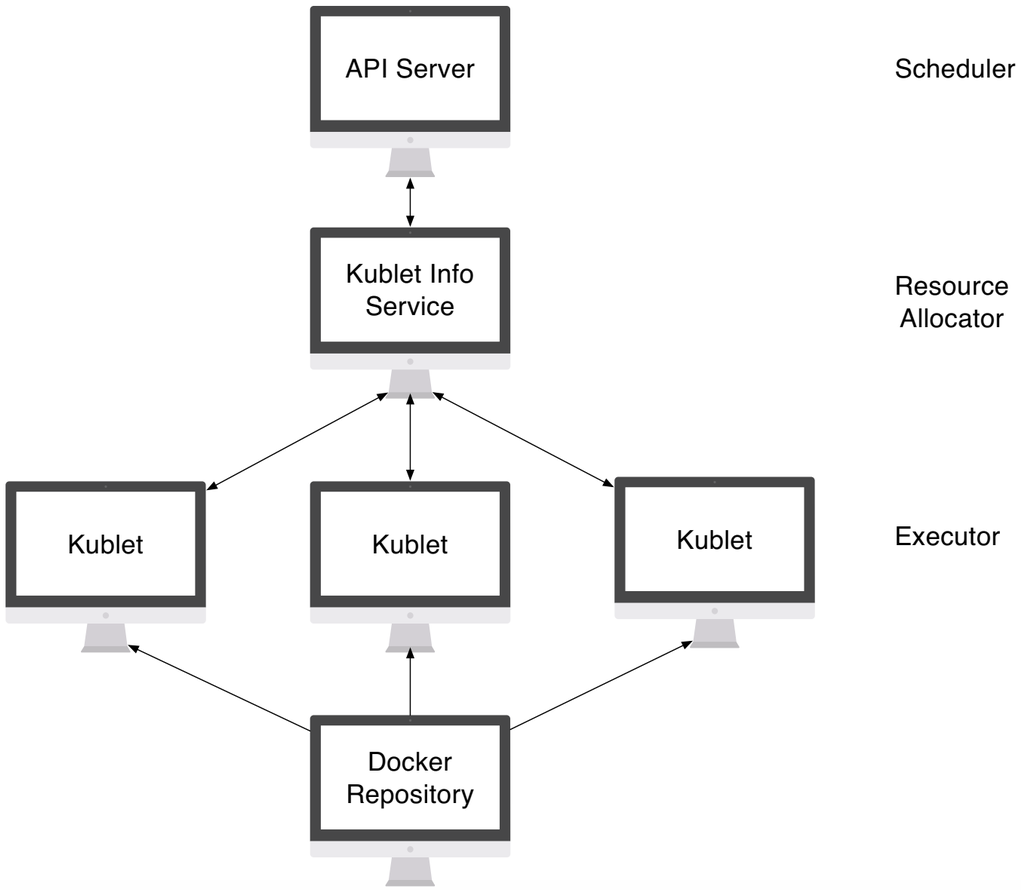
```
bz@cc ~$ docker run -d --name mongodb mongo
705084daa3f852ec796c8d6b13bac882d56d95c261b4a4f8993b43c5fb2f846c
bz@cc ~$ docker run -d --name redis redis
784ebde0e867adb18663e3011b3c1cabe990a0c906396fc306eac669345628cf
bz@cc ~$ docker run -d -P --name cart --link redis:redis zutherb/cart-service
438b2657c7a5c733787fb32b7d28e1a0b84ba9e10d19a8a015c6f24085455011
bz@cc ~$ docker run -d -P -p 8080:8080 --name shop --link cart:cart \
          --link mongodb:mongodb zutherb/monolithic-shop
9926e187faa215ac9044603d51adbd8d679d8076b4a349ebbc9917dade6d560e
bz@cc $ docker exec 9926e187faa215ac9044603d51adbd8d679d8076b4a349ebbc9917dade6d560e env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=87bb5524067d
MONGODB_PORT_27017_TCP=tcp://172.17.0.28:27017
MONGODB_PORT_27017_TCP_ADDR=172.17.0.28
MONGODB_PORT_27017_TCP_PORT=27017
MONGODB_PORT_27017_TCP_PROTO=tcp
```


How can we distributed the containers?

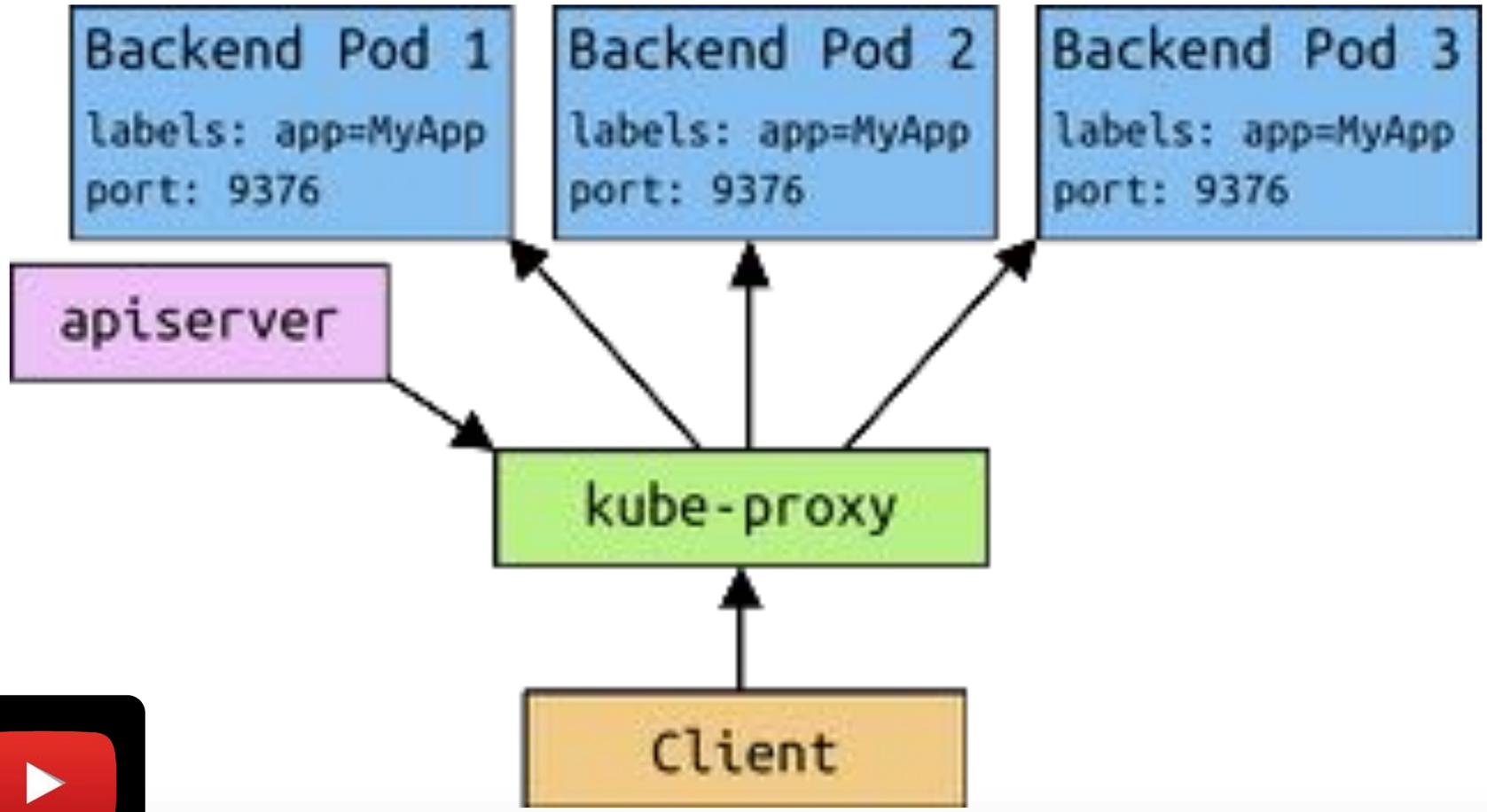
Distributed System



Kubernetes


API Server
Kublet Info Service
Kublet
Kublet
Kublet
Docker Repository
Scheduler
Resource Allocator
Executor

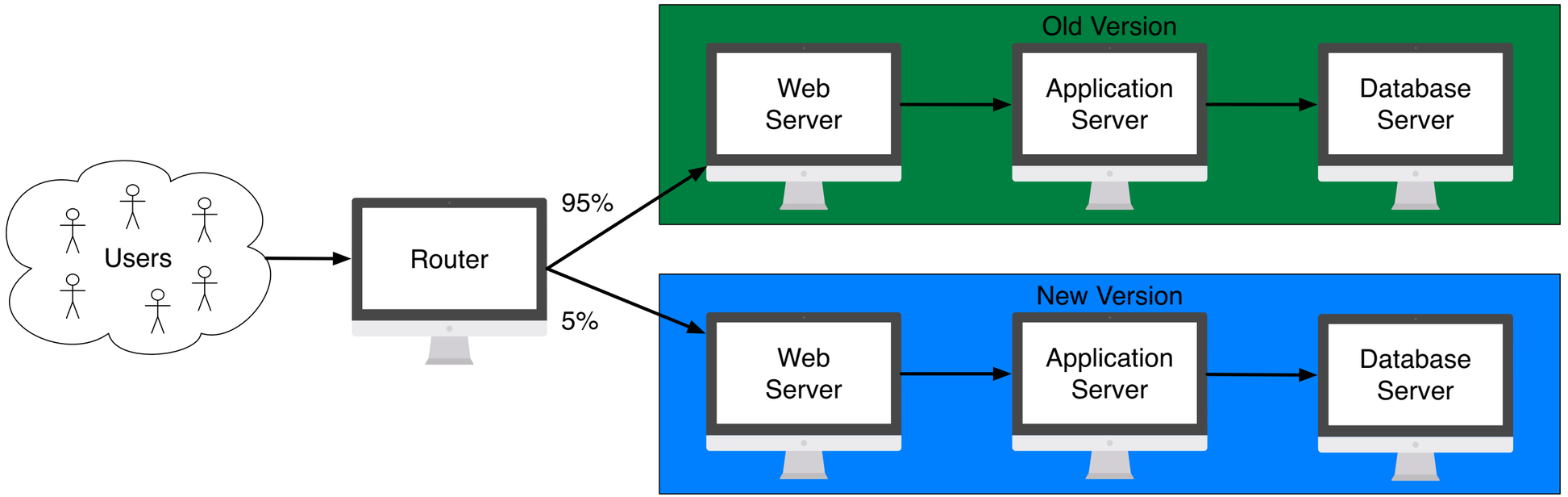
Pods + Replication Controller + Service



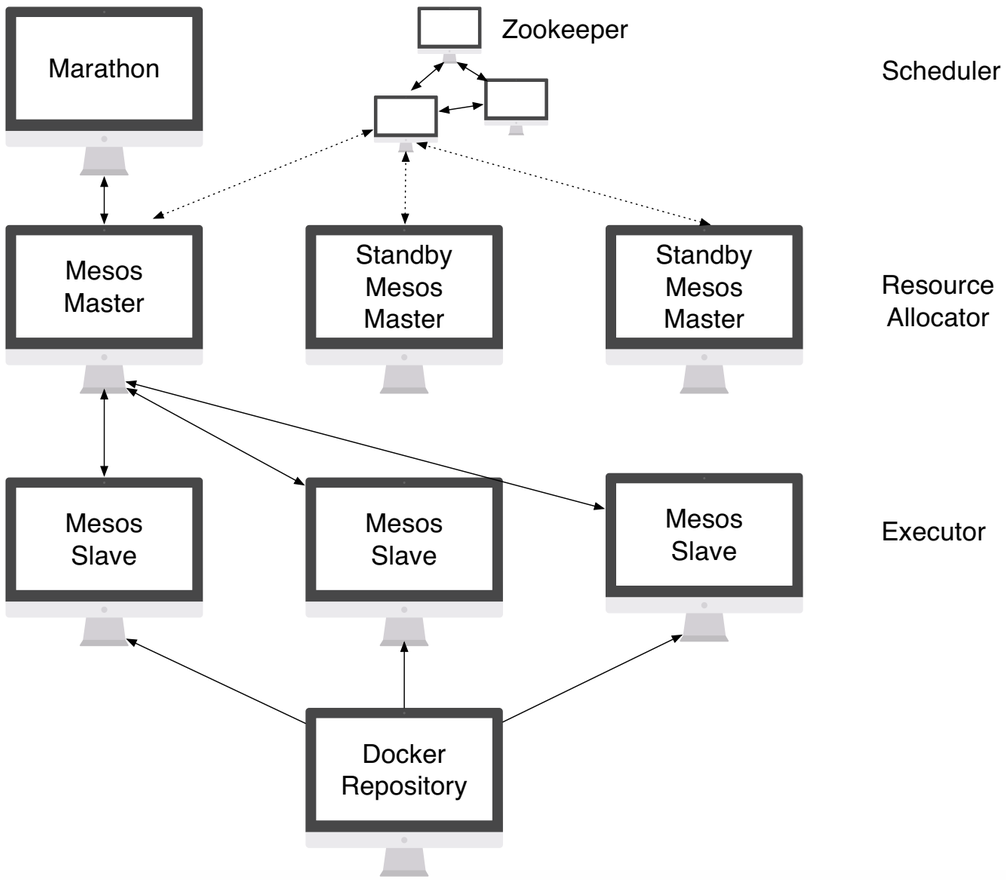
Kubernetes

Pro	Contra
You needn't care where work is executed	No description of the whole application like it is done with docker compose and for deployment scenario
You needn't care about dependencies	Failures analysis will get harder
Service discovery	Master is a single point of failure
Process Supervisor	Few tools and documentation is available yet

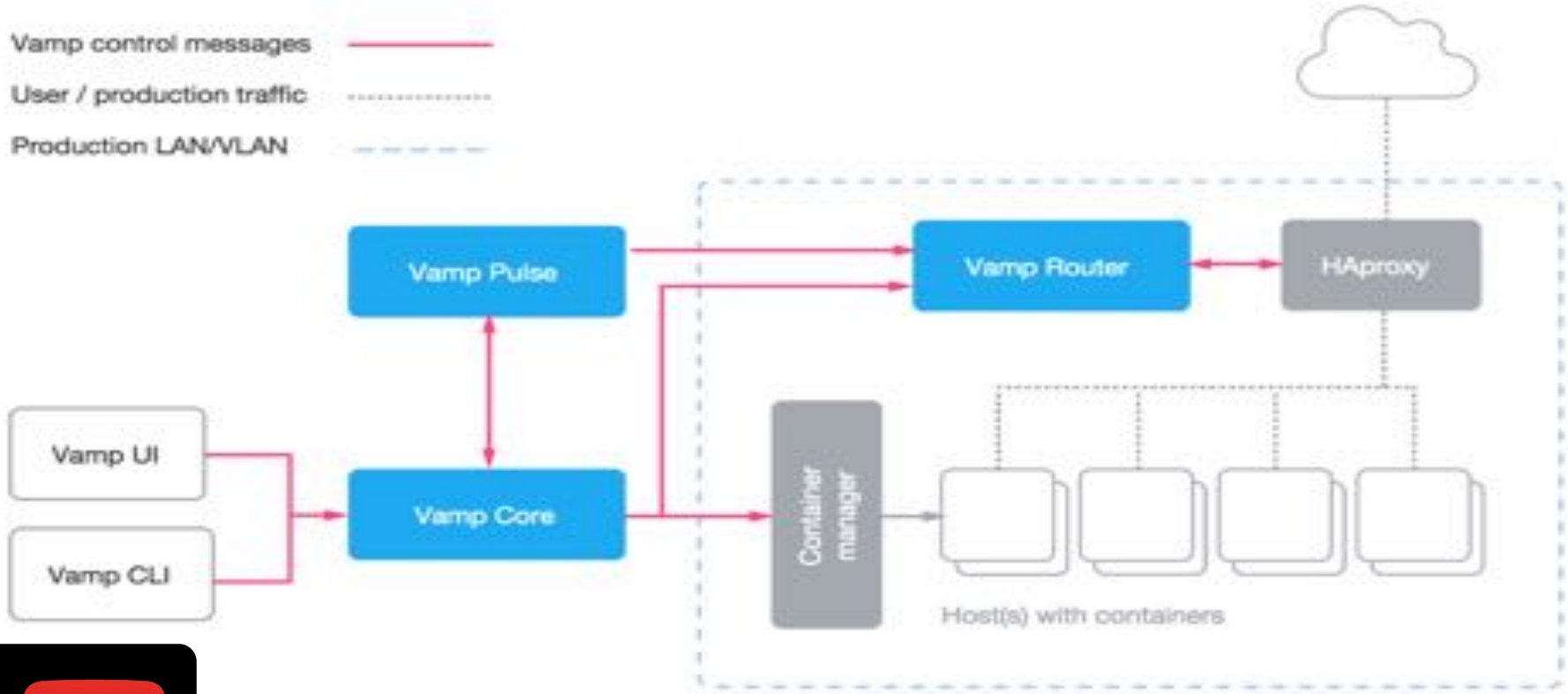
Canary Release



Marathon + Mesos



Vamp (Very Awesome Microservices Platform)

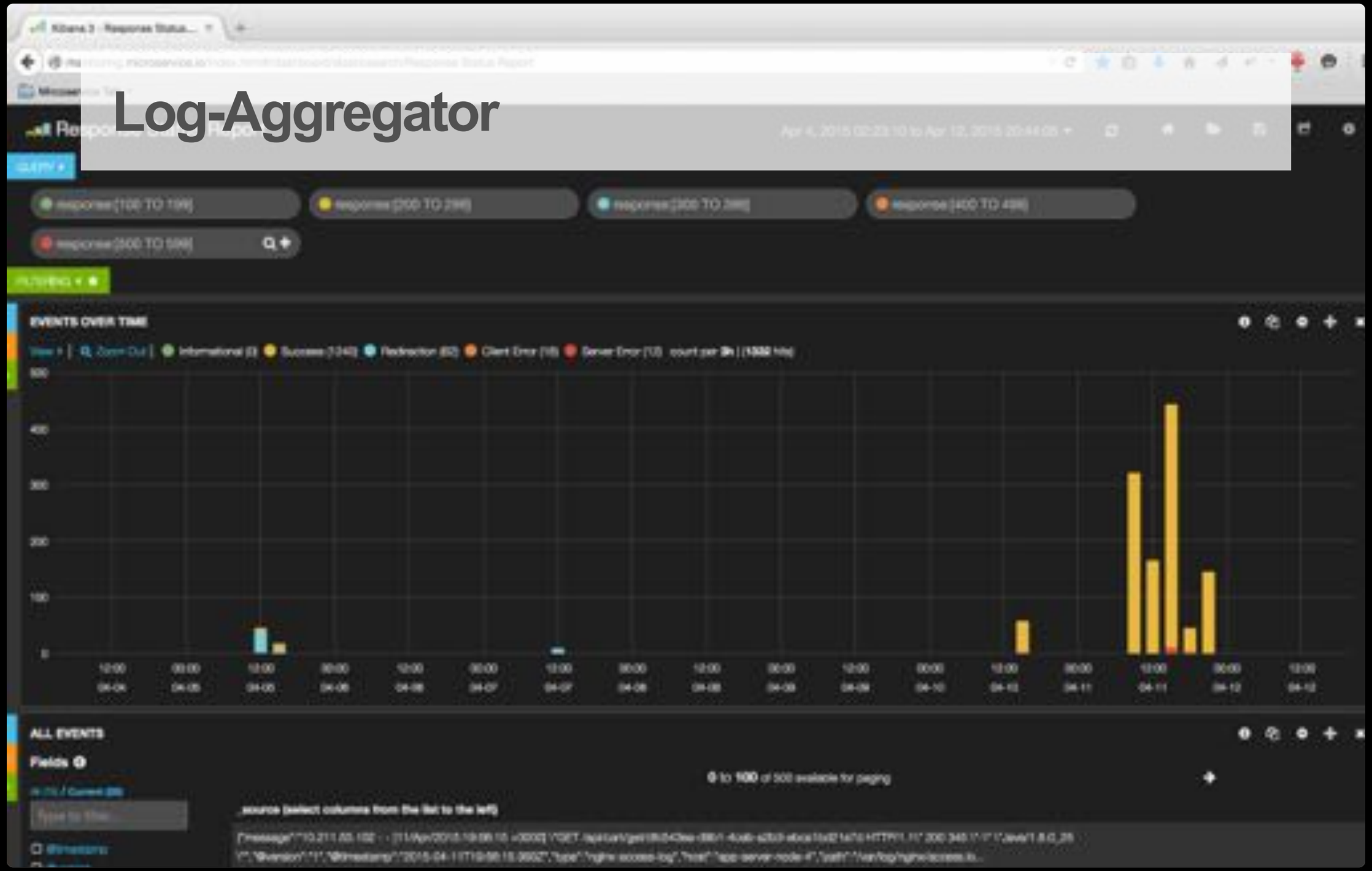


Vamp + Marathon + Mesos

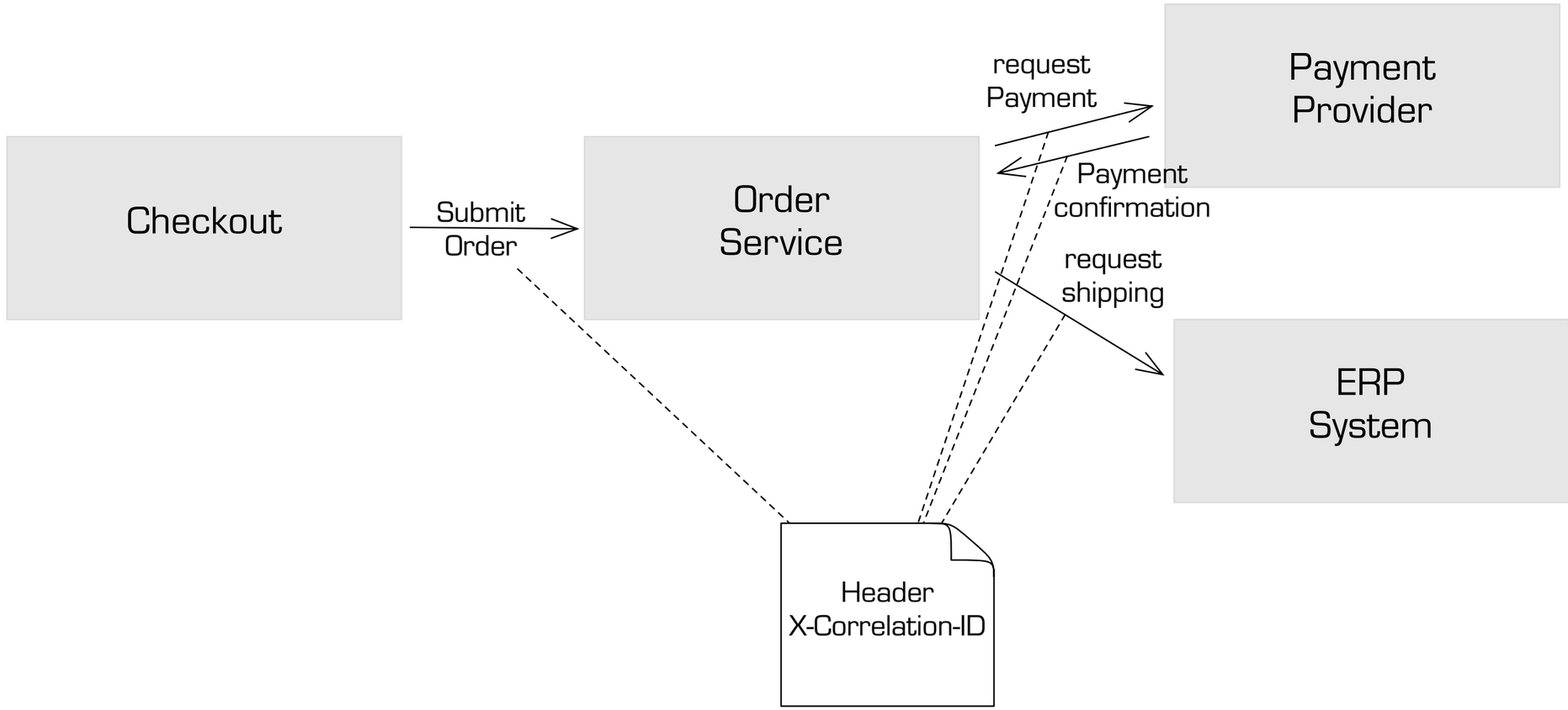
Pro	Contra
Description of the whole application and for deployment scenario	Many components that has to be understood
You needn't care where work is executed	Failures analysis will get harder
High Availability	
Process Supervisor	
Service Discovery	


How can we keep the zoo clean?

Log-Aggregator



Correlation Id



Mapped Diagnostic Context

JAVA

```java
MDC.put("correlationId", "A8FFA290");
Logger logger = LoggerFactory.getLogger(SimpleMDC.class);
logger.info("Create Order");
```

XML

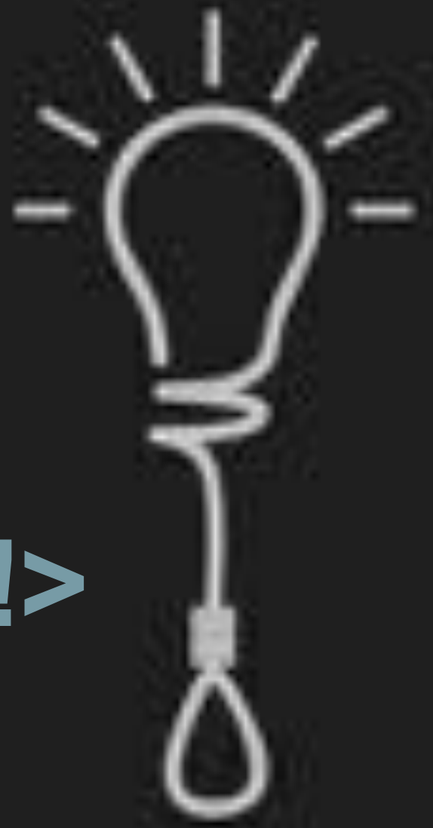
```xml
<appender name="CONSOLE" class="ch.qos.logback.core.ConsoleAppender">
  <layout>
   <Pattern>[%X{correlationId}] - %m%n</Pattern>
  </layout>
</appender>
```

Take Aways

Pro	Needs
supports A/B testing	monitoring
effective technologies	centralised logging
solution orientation	fully automated deployment pipeline
incremental migration	cluster management
	organisation must be ready

Microservices do not guarantee more conversion!
You need to run experiments with your customers!

<Thank You!>



g+	plus.google.com/+BerndZuther
twitter	@Bernd_Z
www	bernd-zuther.de
github	github.com/zutherb/

Get Out of the Building

This talk is based on two projects.

The demo application contains the techology stack and all concepts which were learned in these projects.

https://github.com/zutherb/AppStash/

Links

- Java Aktuell 02/15 - Microservices und die Jagd nach mehr Konversion
- Microservice-Deployment ganz einfach mit Giant Swarm
- Microservice-Deployment ganz einfach mit Kubernetes
- Microservice-Deployment ganz einfach mit Docker Compose
- Microservice-Deployment ganz einfach ohne Docker mit der Linux-Paketverwaltung
- Canary Release mit der Very Awesome Microservices Platform
- Lean Startup
- Test Driven Business Featuring Lean StartUp
- Continuous Delivery mit dem FeatureToggle Pattern
- FeatureToggle
- Do Good Microservices Architectures Spell the Death of the Enterprise Service Bus?
- Microservices

Links

- Pipe
- Service Oriented Architecture : What Is SOA?
- The Twelve Factors
- Three Golden Rules for Continuous Delivery
- Your API versioning is wrong, which is why I decided to do it 3 different wrong ways
- Microservices im Zusammenspiel mit Continuous Delivery, Teil 1 – die Theorie
- Fast, isolated development environments using Docker
- Microservices versus OSGi: Über Sinn und Unsinn der 'neuen Inkarnation des Webservice
- Warnung vor dem Microservice – Versuch einer Definition
- Micro Services in der Praxis: Nie wieder Monolithen!
- Netflix Nebula

Links

- Deployment ganz einfach – Microservice Deployment mit Hilfe der Linux Paketverwaltung
- Splunk – Marke Eigenbau mit Elasticsearch, Logstash und Kibana (ELK Stack)
- Scaling Docker with Kubernetes
- Appstash Project

Images

- Delivery of semlor
- Freebooks
- SOA
- Cuba Car
- Landschaftspark Duisburg-Nord
- Unfall von Herbert Stenger
- Lieferwagen
- Indianapolis Zoo
- Developer At Work
- Icom IC735 + Tuner
- Just a face in the crowd
- [119/366] Thumbs Up
- Containers

Images

- Kaufrausch
- Café con leche - Milchkaffee (CC)
- Cloning Experiments: Jess Payne
- hello, my name is bob ross the bushy-haired painter.
- BLM Nevada
- change machine
- s. Bär
- There's a light at the end of the tunnel...
- 2012 DC Zeitgeist III 11787
- Red Rule
- Schwarzwald Blumenwiese