



# Microservices and Conversion Hunting:

Build Architectures for Changeability

Bernd Zuther - IT Consultant Big Data  
codecentric AG



Bernd Zuther



@Bernd\_Z

[github.com/zutherb](https://github.com/zutherb)

[bernd.zuther@codecentric.de](mailto:bernd.zuther@codecentric.de)


my name is










A close-up photograph of a lioness feeding on a carcass in a natural, savanna-like environment. The lioness's head and front paws are visible as it consumes the meat. The carcass is partially eaten, showing red flesh and white bone. The background consists of green foliage and a tree trunk.


**Survival of the fittest**





**Be faster than your competitors**



A close-up, slightly blurred photograph of a shopping cart filled with various items. The cart is a wire mesh basket with a red handle. Inside, there are several books, some with orange and yellow covers, and a red box. A hand is visible on the left, holding the handle. In the background, there are decorative items, including a green tinsel branch and a red branch with small flowers. The overall scene suggests a shopping trip, possibly at a bookstore or a holiday market.

**How can we increase sales?**



A black and white photograph of Mike Tyson lying on his back on a boxing ring canvas. He is wearing a white boxing trunks and has his arms spread out. His head is tilted back, and his eyes are closed. The canvas has white circular and linear markings. In the background, the lower legs and feet of several people, likely judges or officials, are visible, standing on the ring's edge. A semi-transparent white rectangular box is overlaid on the image, containing the text.

# 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());
    }

    ...

}
```








A black and white photograph of an industrial facility, likely a refinery or chemical plant. The image shows a complex network of pipes, scaffolding, and structural steel. On the left, there's a multi-story building with windows. In the center, a large, vertical distillation column is prominent, surrounded by a dense web of pipes and walkways. To the right, a tall, slender chimney rises into the sky. Further right, there are several large, cylindrical storage tanks. The foreground is filled with dense foliage, possibly trees or bushes, partially obscuring the lower parts of the industrial structures.

**What have we done?**



A photograph of a blue vintage car with its hood open. A man in a black shirt is leaning over the engine compartment, working on it. Another man in a white cap is standing nearby. In the background, several other people are visible, some standing and some walking. The scene appears to be outdoors, possibly in a parking lot or a street. The text "How does it work?" is overlaid on the image in a white box.

**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

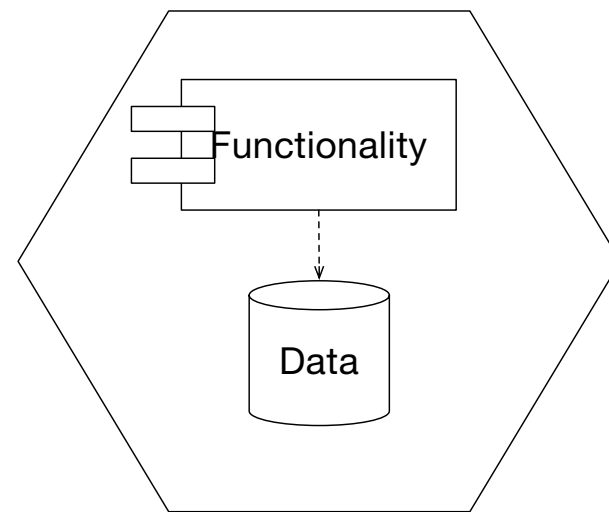




A close-up photograph of a person's hands, wearing a grey and white checkered shirt, carefully connecting a small red LEGO brick to a larger red structure. The background is a light grey wall. In the foreground, a wooden table is covered with a large pile of various colored LEGO bricks (red, blue, yellow, green, white, black) and some partially assembled structures. A white rectangular box with the text "How can we build software smarter?" is overlaid on the center of the image.

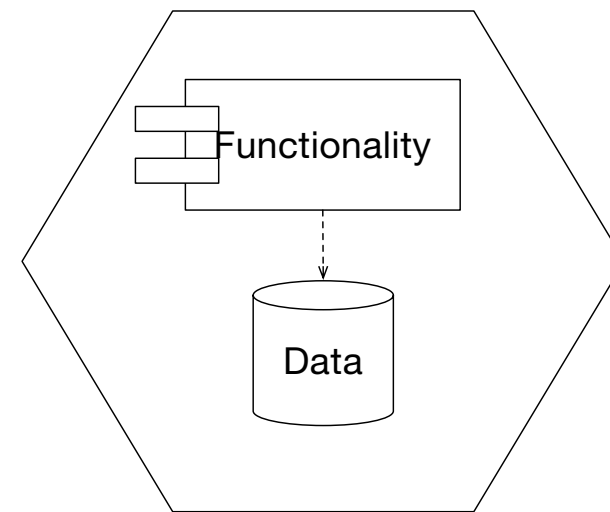
**How can we build software smarter?**

# Microservice



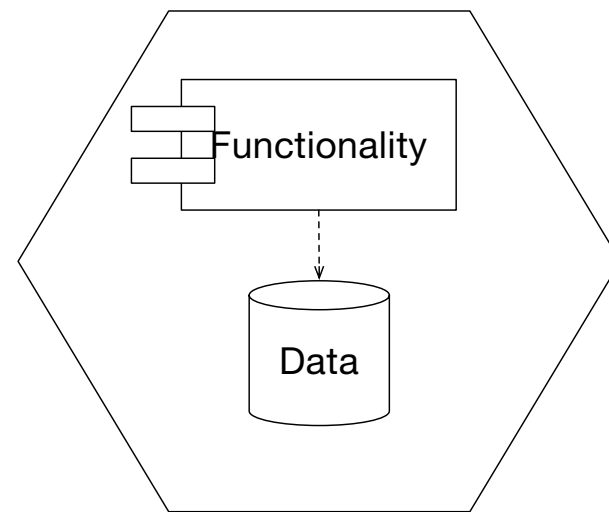


# Microservice



- fits in one brain,

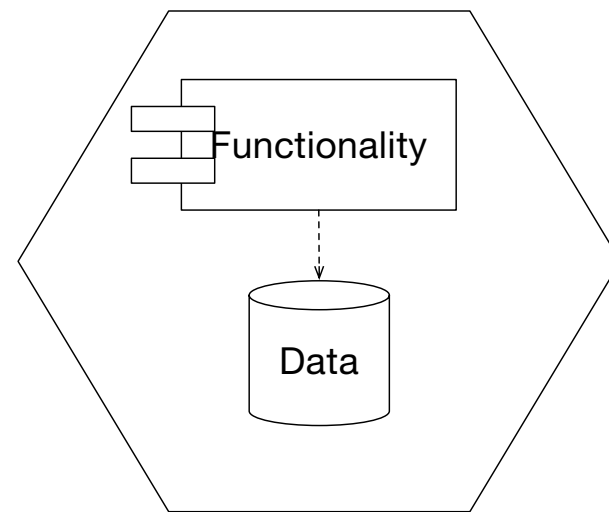
# Microservice



- fits in one brain,
- designed for replaceability,

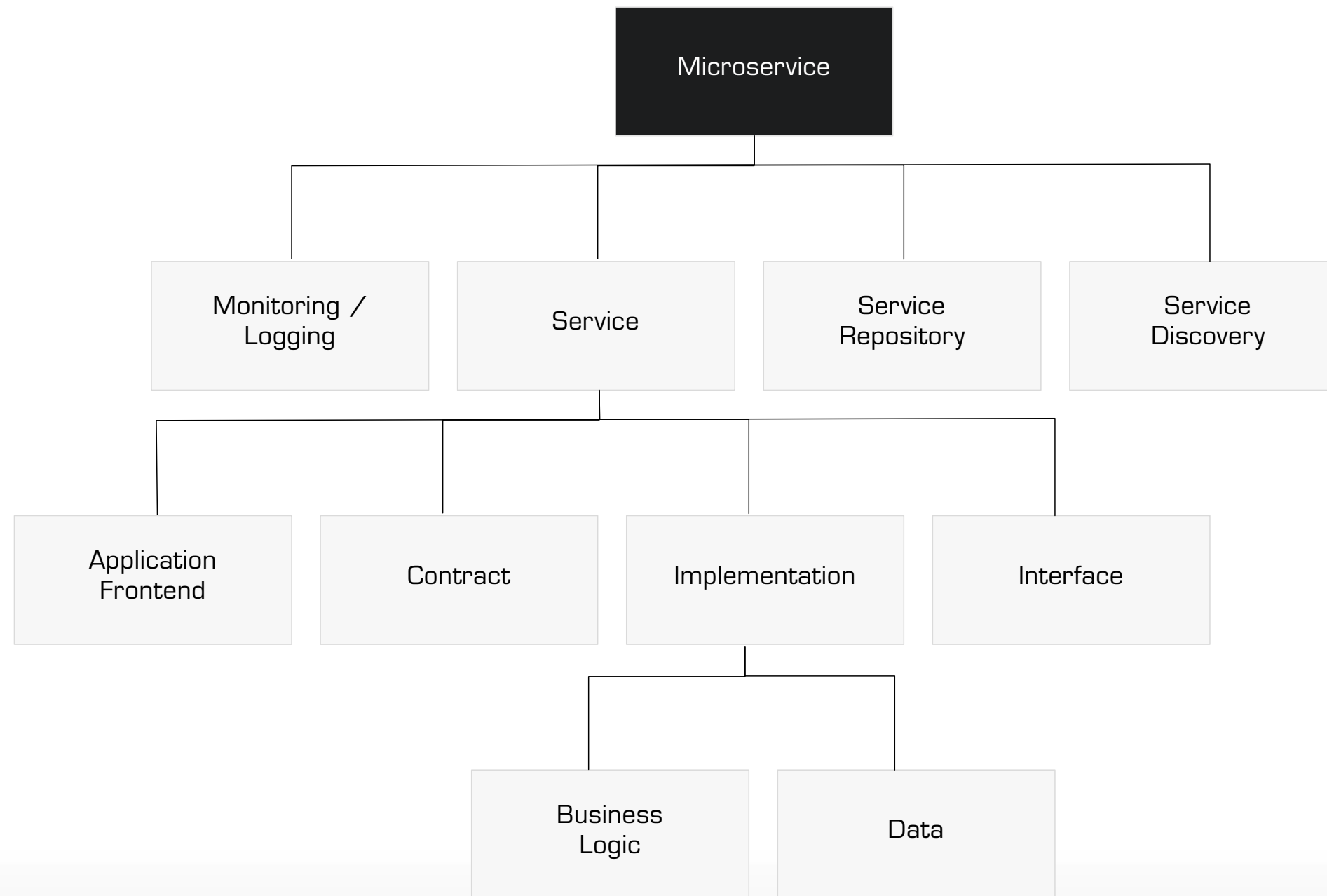


# Microservice



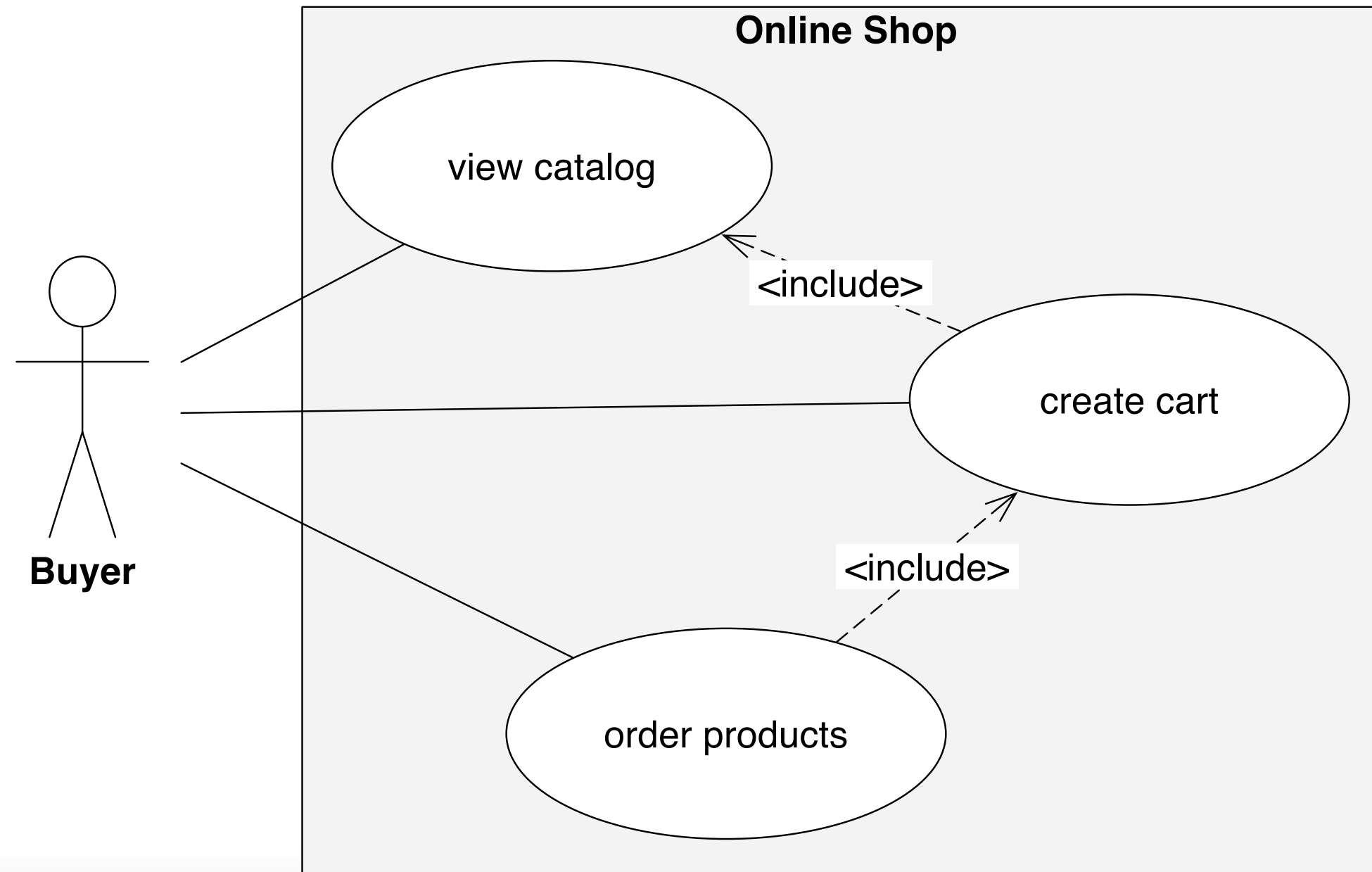
- fits in one brain,
- **designed for replaceability,**
- autonomy (organisation & technology)

# Microservice Taxonomy

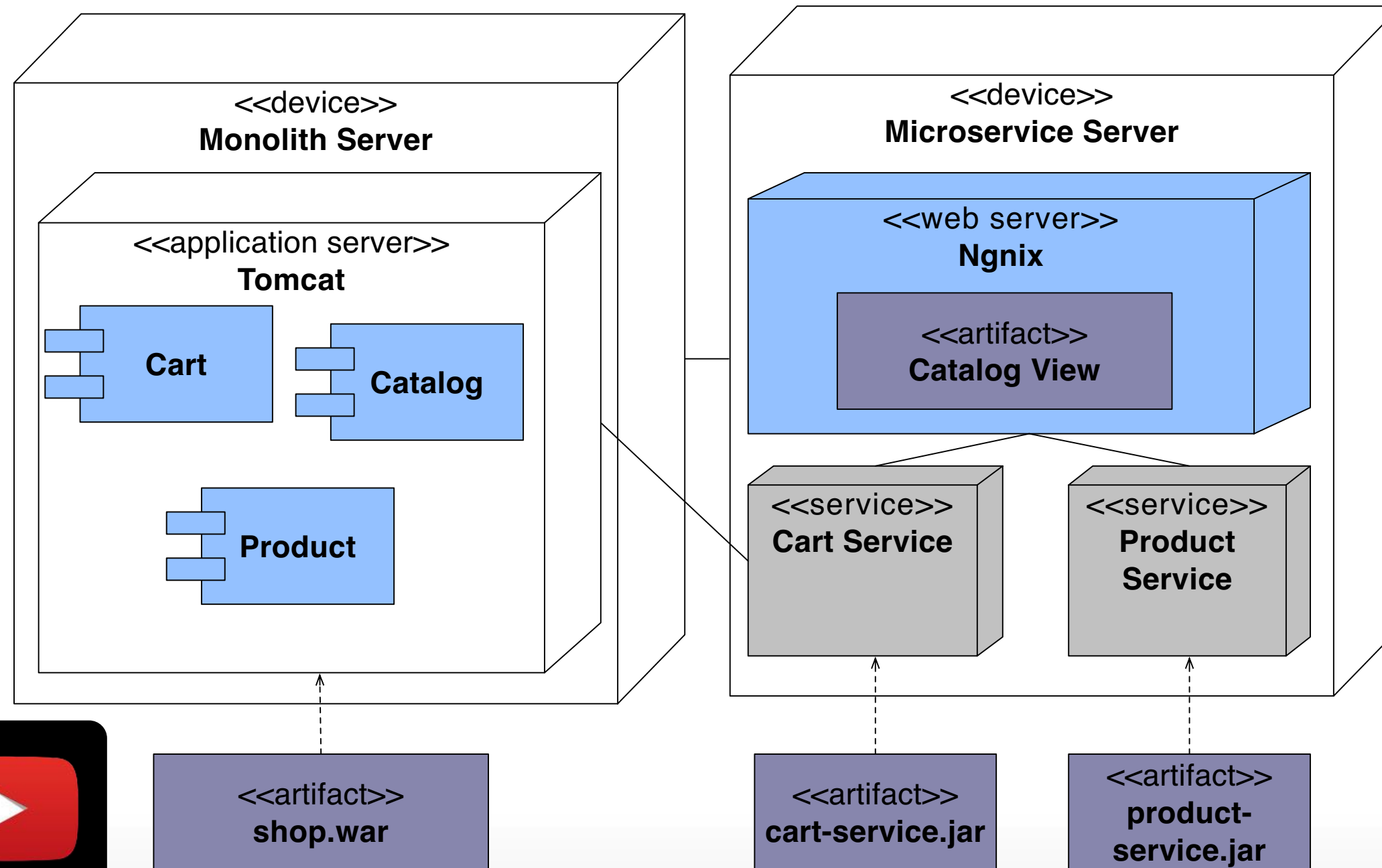




# Online Shop - Use cases



# Online Shop - Migration







**Flexibility**



**Improve feedback times**



A close-up photograph of a hand with the index finger pointing upwards. The skin is a warm, light brown tone. A white rectangular box is superimposed over the middle of the index finger, containing the word "Scalability" in a bold, black, sans-serif font. The background is a soft, out-of-focus light beige color.

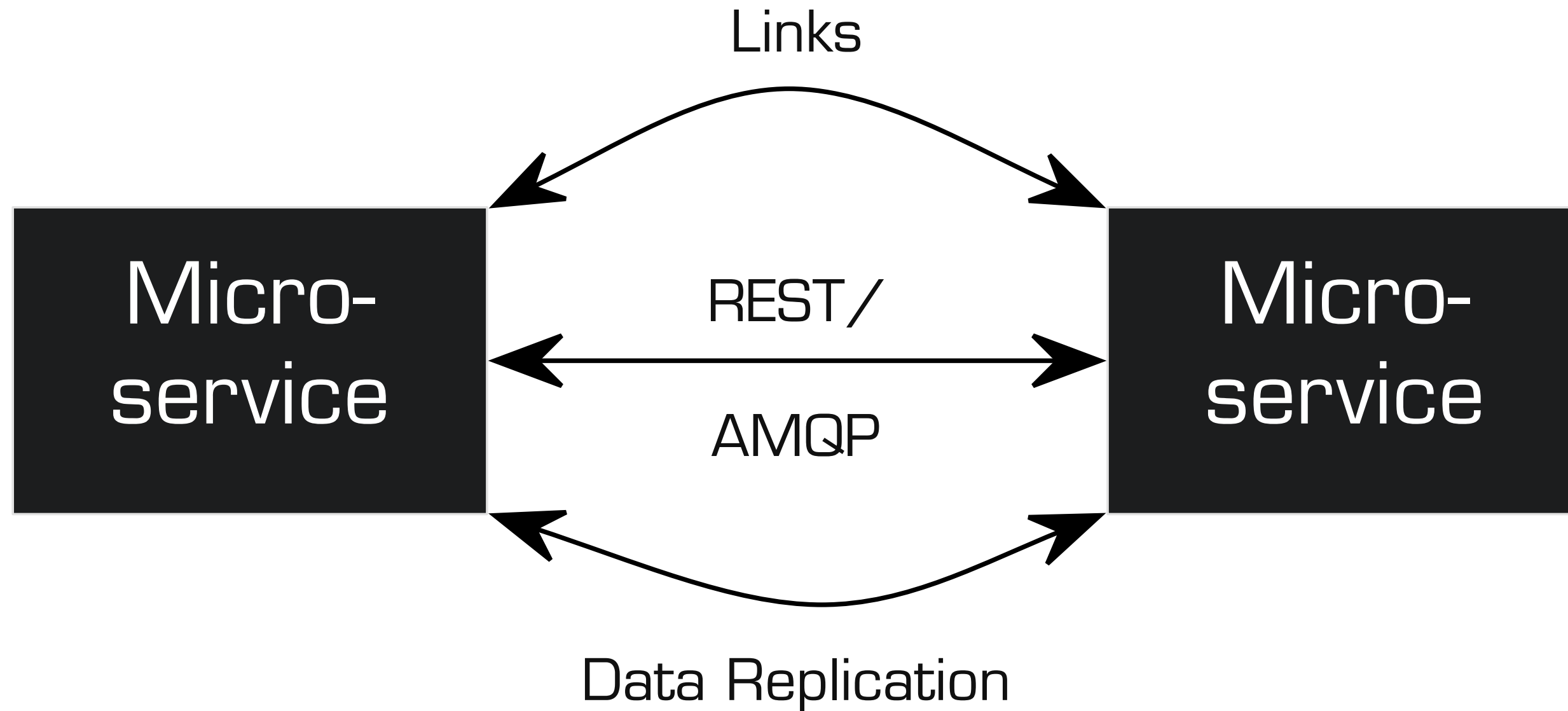
**Scalability**



# Challenges



# Microservice - Communication



The background of the slide is a photograph of a weathered, rusty metal surface. A heavy metal chain is draped across the surface, and a large, rusted padlock is attached to it. The text is overlaid on a semi-transparent white rectangular box in the center of the image.

# **Adrian Cockcroft**

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



The background image shows a collection of vintage electronic equipment, likely from the 1970s or 1980s. In the foreground, there's a green and silver device with two large, prominent knobs. Behind it, a white device with a black control panel and a small display is visible. To the right, a larger white unit with various ports and a coiled cable is present. The equipment is sitting on a dark wooden surface, and the background is slightly blurred, showing some greenery and a white box.

# 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"  
  }]  
}
```







# 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”







The image shows an industrial facility, likely a refinery or chemical plant. In the background, there are several large, white, spherical storage tanks. In the foreground, a complex network of pipes, valves, and structural steel is visible. The scene is illuminated by warm, orange light, possibly from the setting or rising sun, creating a high-contrast environment. The sky is a deep blue, suggesting dusk or dawn. The overall composition emphasizes the scale and complexity of industrial infrastructure.

**What change on your build pipeline?**



# Build Pipeline



Pipeline #22	#22 shop-monolithic-build Sep 15, 2019 12:11 PM 3 Minutes 14 Seconds	#18 shop-monolithic-test-deployment Sep 15, 2019 12:14 PM 20 Seconds	#30 shop-monolithic-ui-test Sep 15, 2019 12:16 PM 1 Minute 02 Seconds	#12 shop-monolithic-load-test Sep 15, 2019 12:17 PM 3 Minutes 30 Seconds
Pipeline #21	#21 shop-monolithic-build Sep 15, 2019 12:04 PM 30 Seconds	#18 shop-monolithic-test-deployment Sep 15, 2019 12:14 PM 20 Seconds	shop-monolithic-ui-test	shop-monolithic-load-test
Pipeline #20	#20 shop-monolithic-build Sep 15, 2019 12:04 PM 3 Minutes 14 Seconds	#17 shop-monolithic-test-deployment Sep 15, 2019 12:01 PM 20 Seconds	shop-monolithic-ui-test	shop-monolithic-load-test

Microservice Pipeline Jenkins

ci-nods 6080view/Microservice Pipeline/

Microservice Test

Jenkins - Microservice Pipeline

Build Verlauf

Ansicht bearbeiten

Ansicht löschen

Projektbeziehungen

Fingerabdruck überprüfen

View Fullscreen

Jenkins verwalten

Zugangsdaten

Build Warteschlange

Keine Builds geplant

Build-Processor-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

1 day ago

11:40:20-440

→

shop-microservice-publish

shop-microservice-publish

1 day ago

1:00

→

shop-microservice-test-deployment

shop-microservice-test-deployment

1 day ago

21:40

### Product

#17 triggered by user anonymous started 6 days ago

shop-microservice-product-service-build

shop-microservice-product-service-build

1 day ago

1:00:18-440

→

shop-microservice-publish

shop-microservice-publish

1 day ago

1:00

→

shop-microservice-test-deployment

shop-microservice-test-deployment

1 day ago

21:40

### Registration

#14 triggered by user anonymous changes by bernd.zuther started 6 days ago

shop-microservice-registration-ui-build

shop-microservice-registration-ui-build

1 day ago

10:40

### Checkout

#13 triggered by user anonymous changes by bernd.zuther started 6 days ago

shop-microservice-checkout-ui-build

shop-microservice-checkout-ui-build

1 day ago

21:40

→

shop-microservice-publish

shop-microservice-publish

1 day ago

1:00

→

shop-microservice-test-deployment

shop-microservice-test-deployment

1 day ago

21:40

### Navigation

#14 triggered by user anonymous changes by bernd.zuther started 6 days ago

shop-microservice-navigation-service-build

shop-microservice-navigation-service-build

1 day ago

1:00

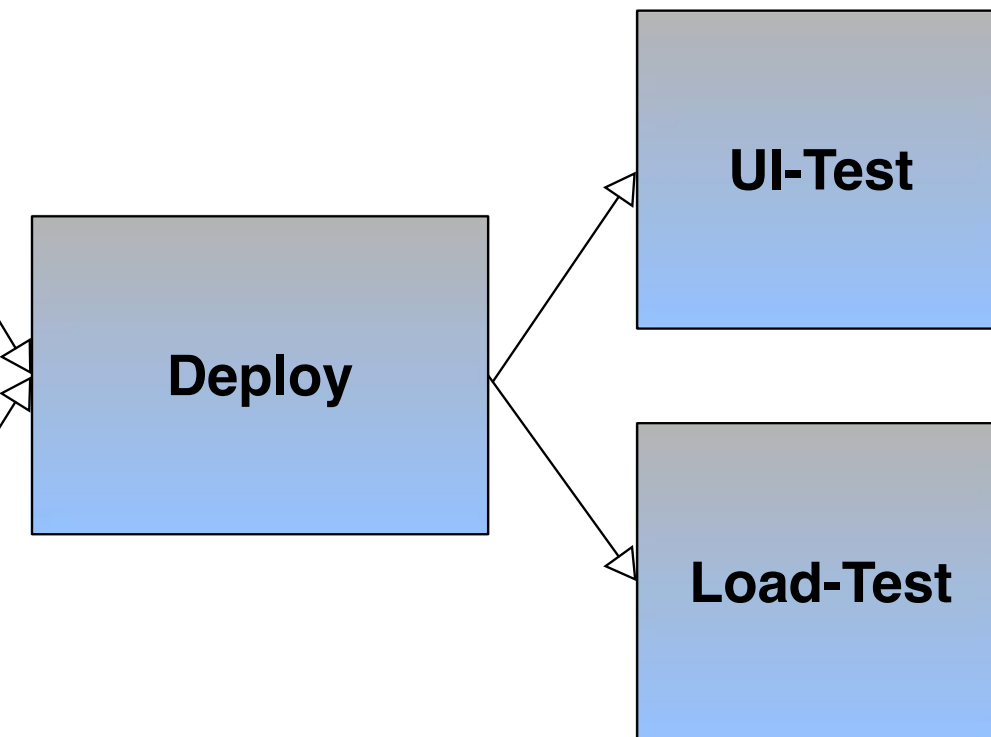
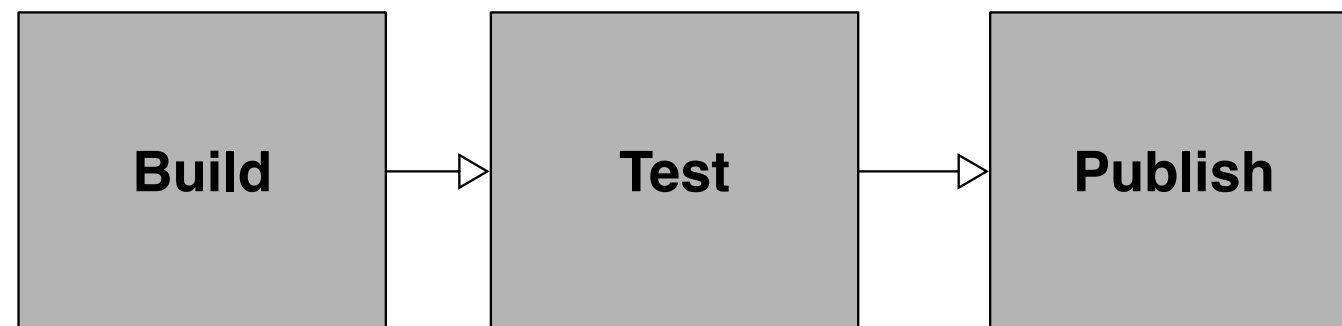


# Multi Deployment Pipeline

## Productservice




## Catalog View




A close-up photograph of a hand, with the index finger pointing upwards. A white rectangular text box is overlaid on the middle of the hand. The background is a plain, light-colored surface.

**Improve Build Times**



**Deployments get more complex**





**How can we package and describe  
our application?**





# Docker

## Build, Ship, Run

# Docker-Workflow

```
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


BASH



# Container-Link

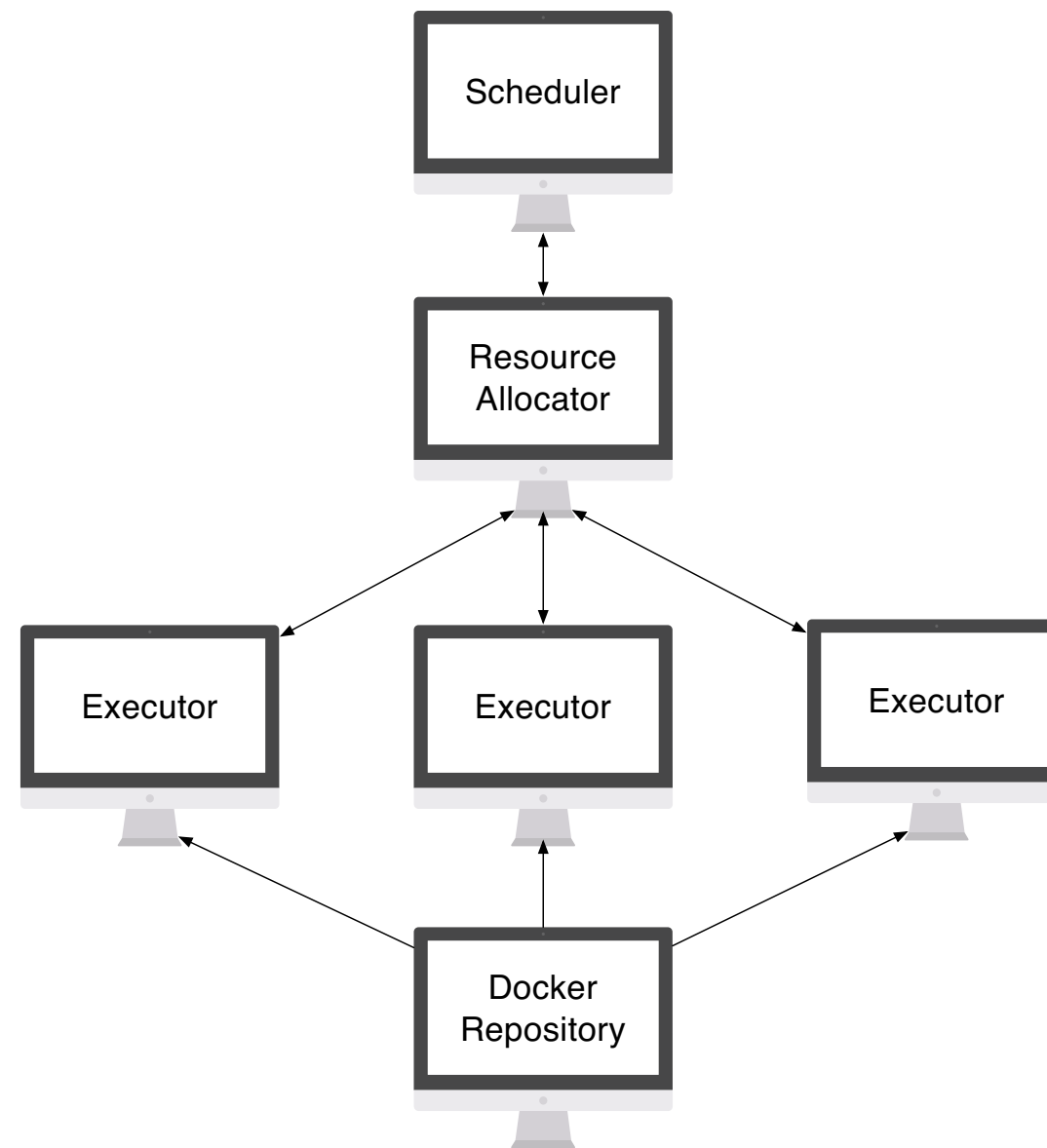
BASH

```
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
```



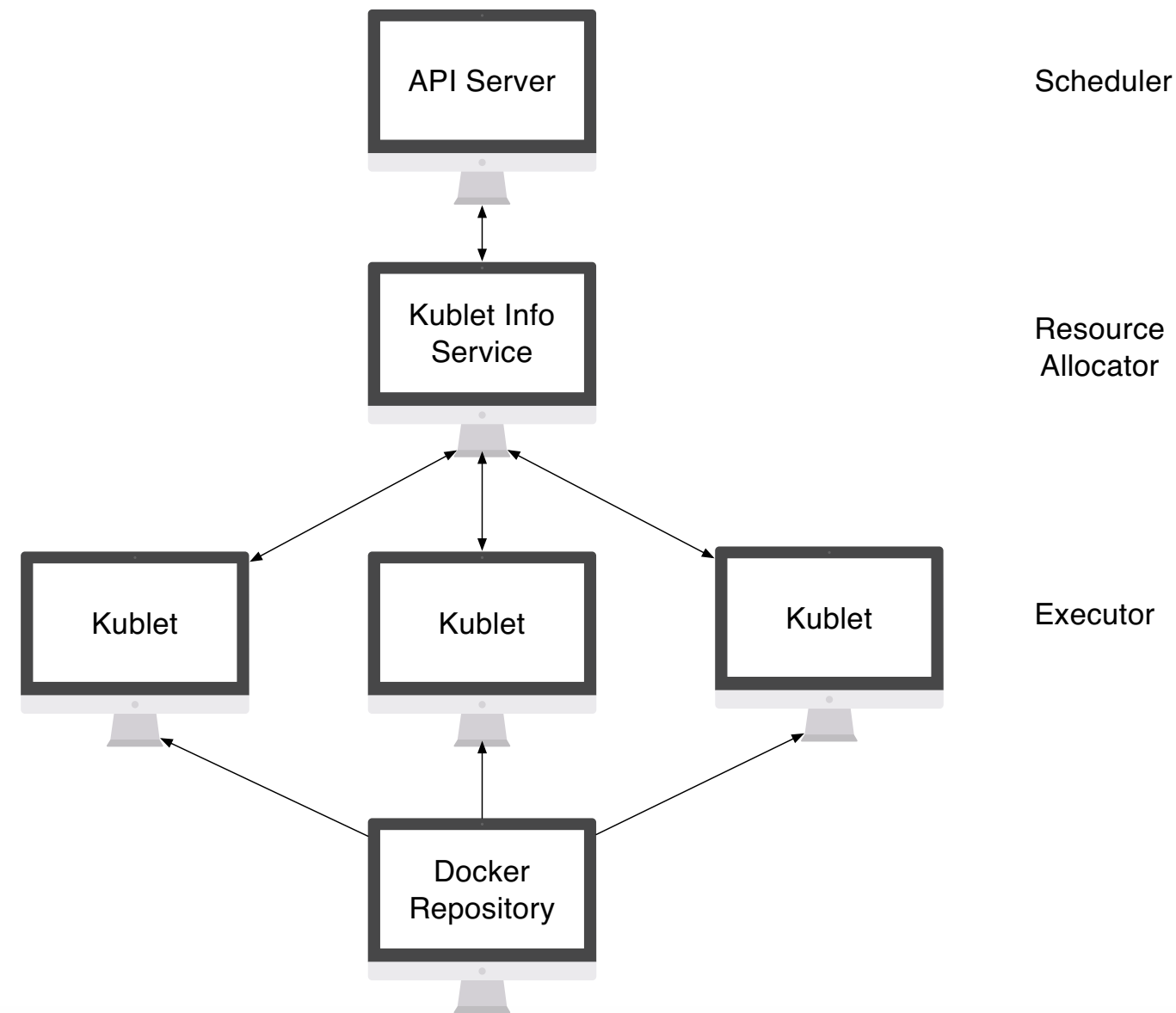
**How can we distributed the containers?**

# Distributed System

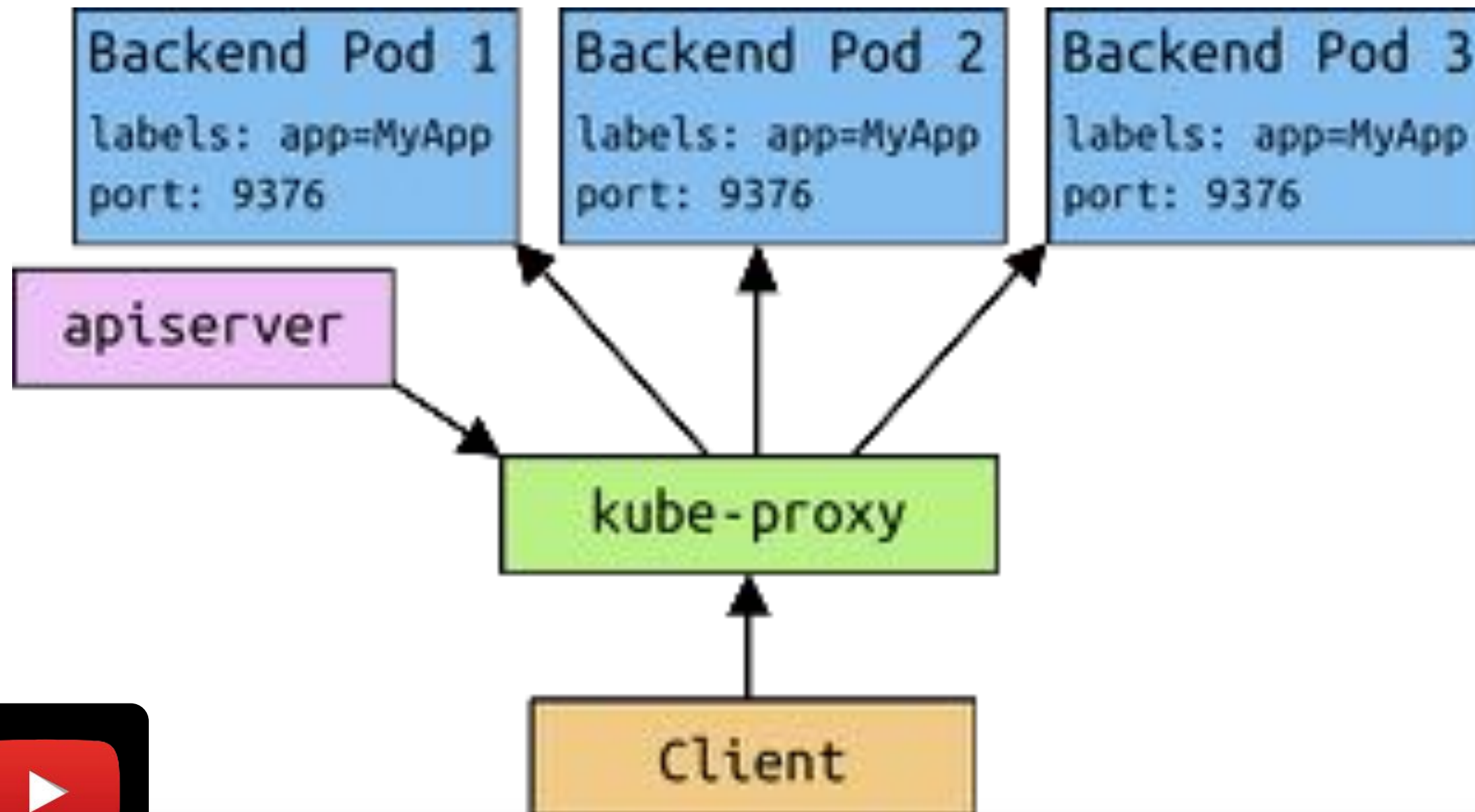




# Kubernetes



# Pods + Replication Controller + Service



# Kubernetes

## Pro

You needn't care where work is executed

You needn't care about dependencies

Service discovery

Process Supervisor

## Contra

No description of the whole application like it is done with docker compose and for deployment scenario

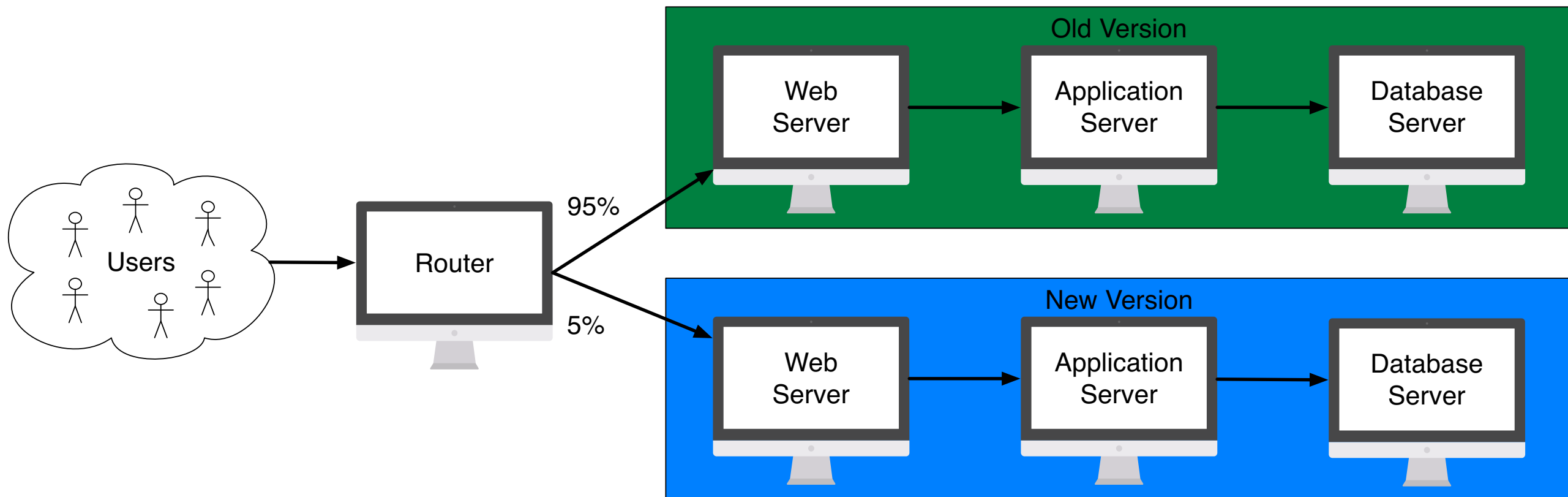
Failures analysis will get harder

Master is a single point of failure

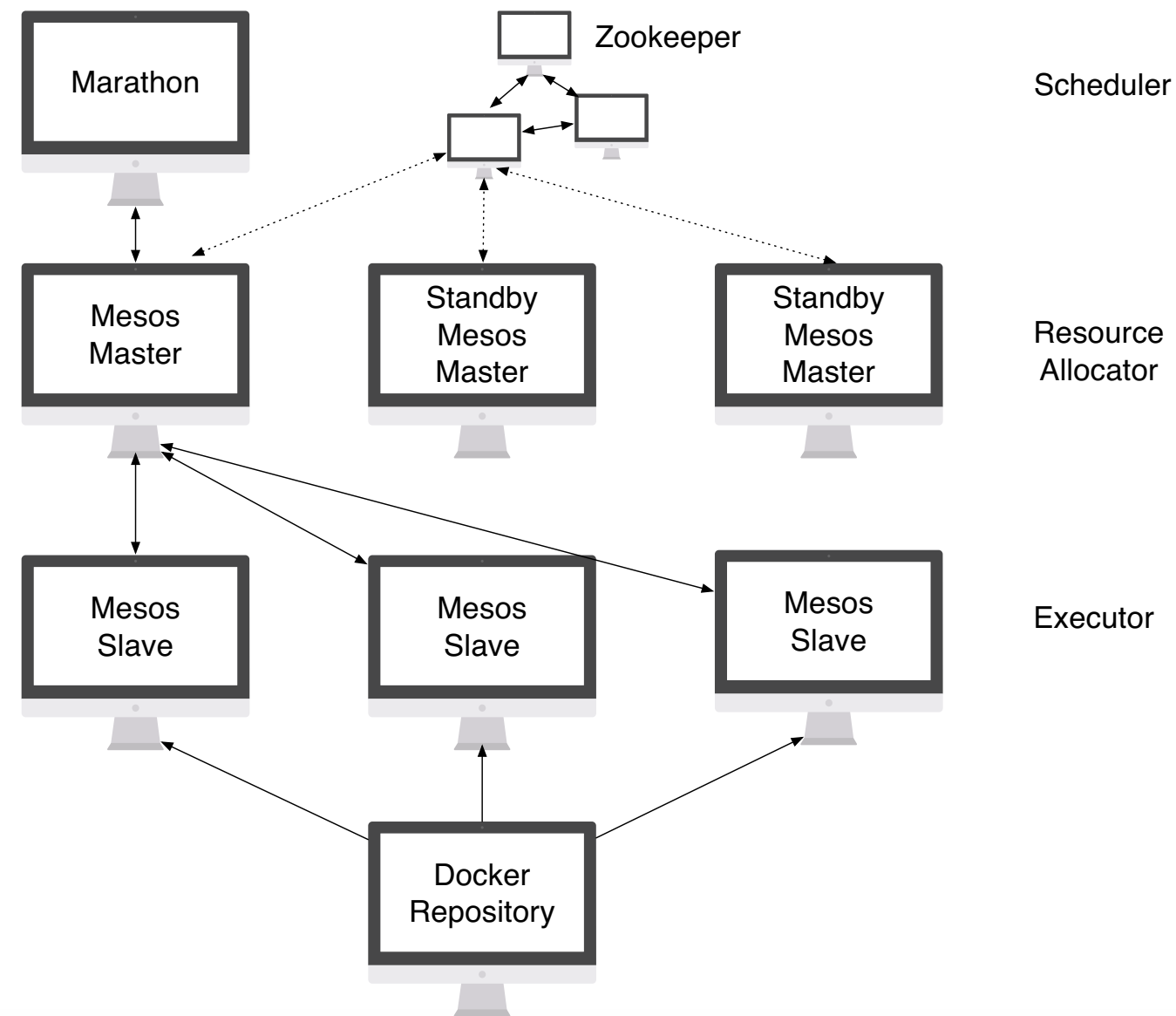
Few tools and documentation is available yet



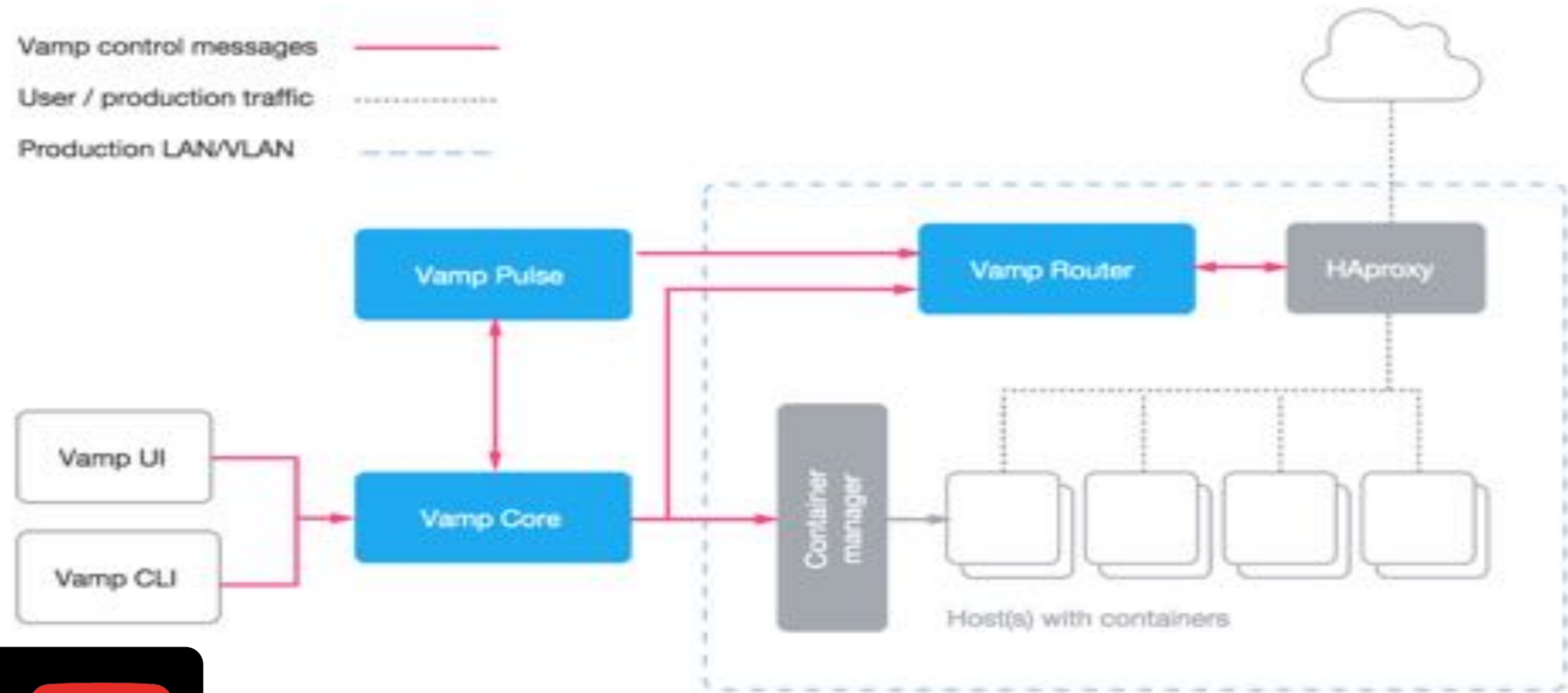
# Canary Release



# Marathon + Mesos



# Vamp (Very Awesome Microservices Platform)





# Vamp + Marathon + Mesos

## Pro

Description of the whole application and for deployment scenario

You needn't care where work is executed

High Availability

Process Supervisor


Service Discovery

## Contra

Many components that has to be understood

Failures analysis will get harder

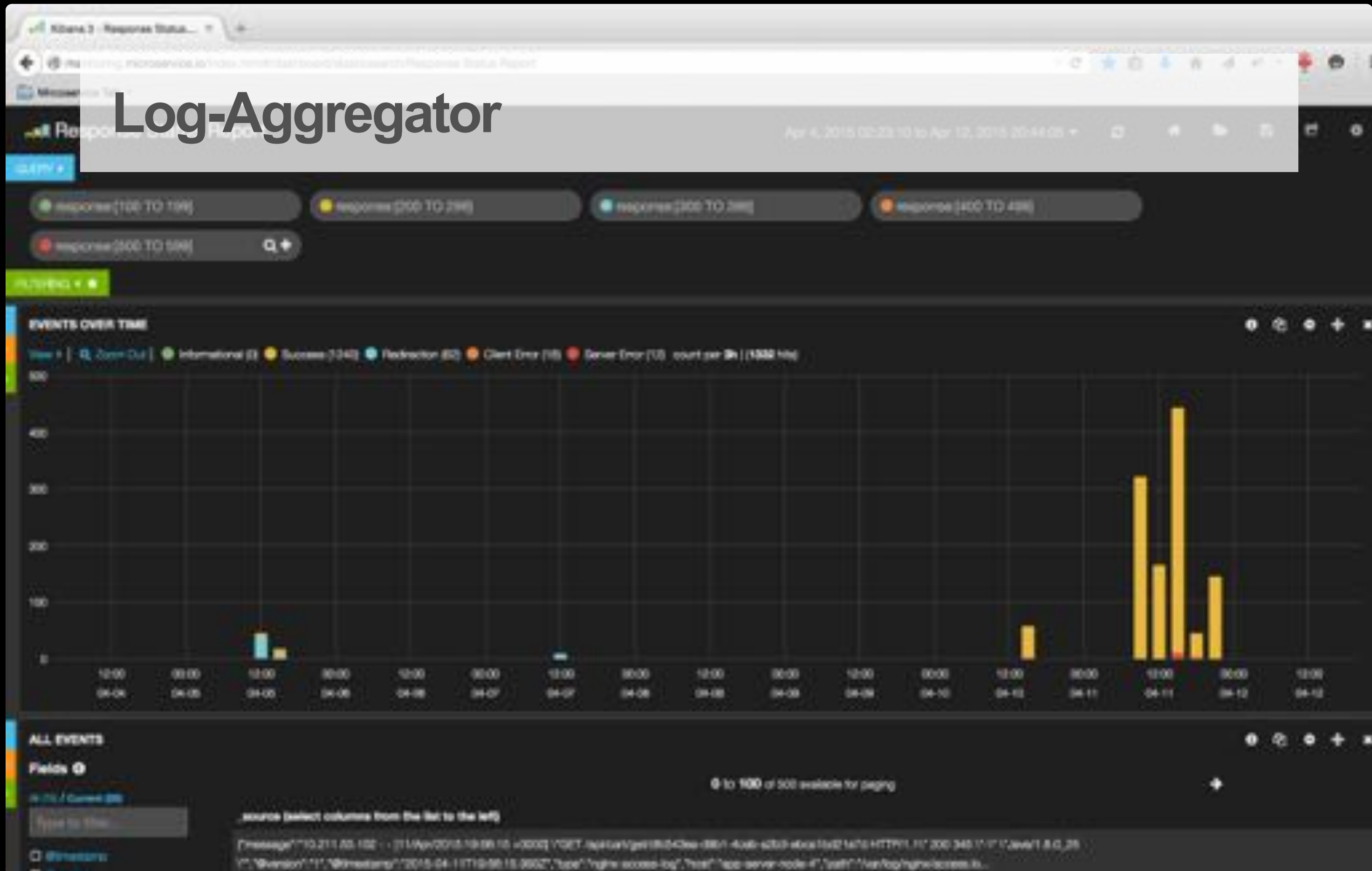


A close-up photograph of a giraffe's head and neck, showing its characteristic brown and white patterned skin. The giraffe is leaning its head down towards a person's hand. The person, a woman with dark hair tied back, is wearing a light-colored polo shirt and a black wristwatch. She is holding a dark, curved object, likely a feeding tool or a piece of food, near the giraffe's mouth. The background is a lush green landscape with trees and a clear sky. A semi-transparent white rectangular box is overlaid on the center of the image, containing the text "How can we keep the zoo clean?".

**How can we keep the zoo clean?**

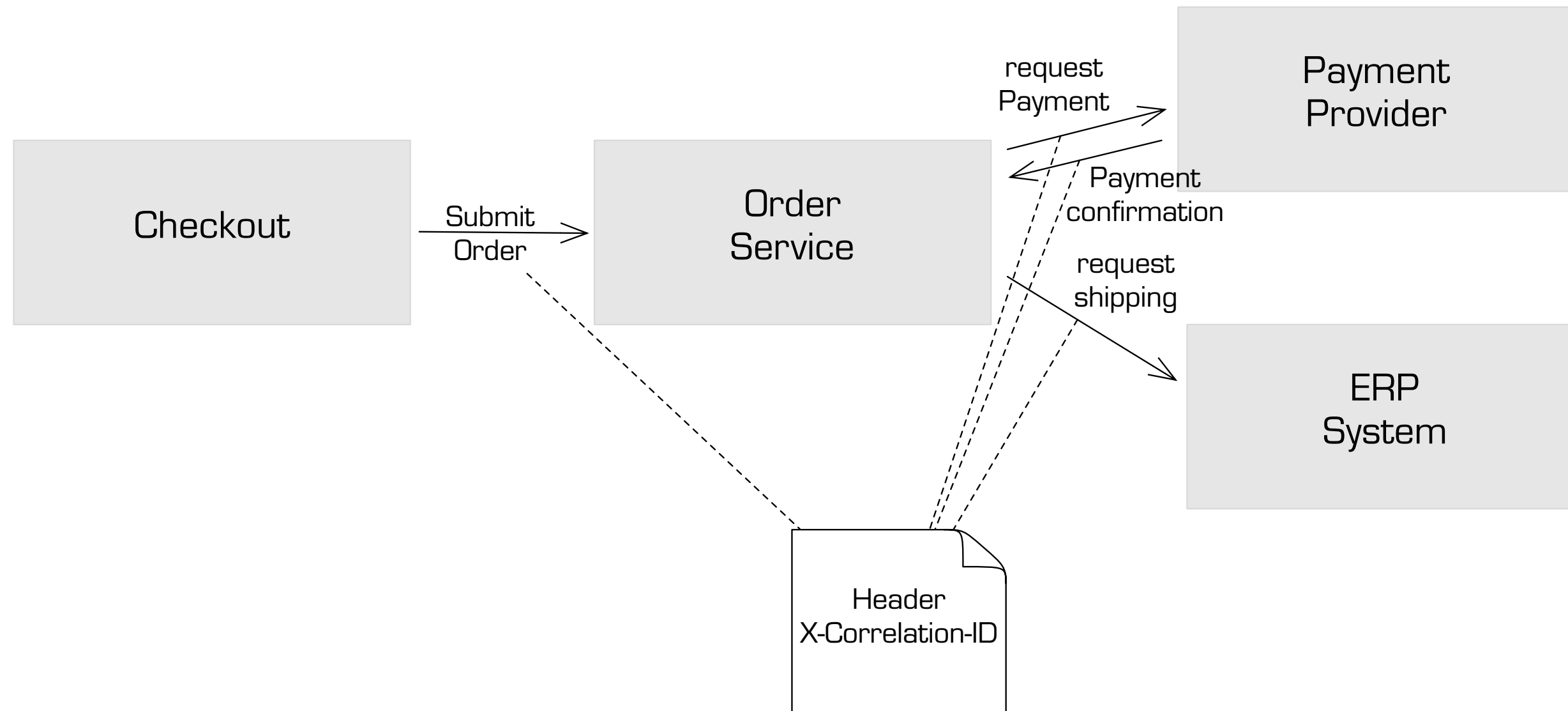


# Log-Aggregator





# Correlation Id



# Mapped Diagnostic Context

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

JAVA

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

XML



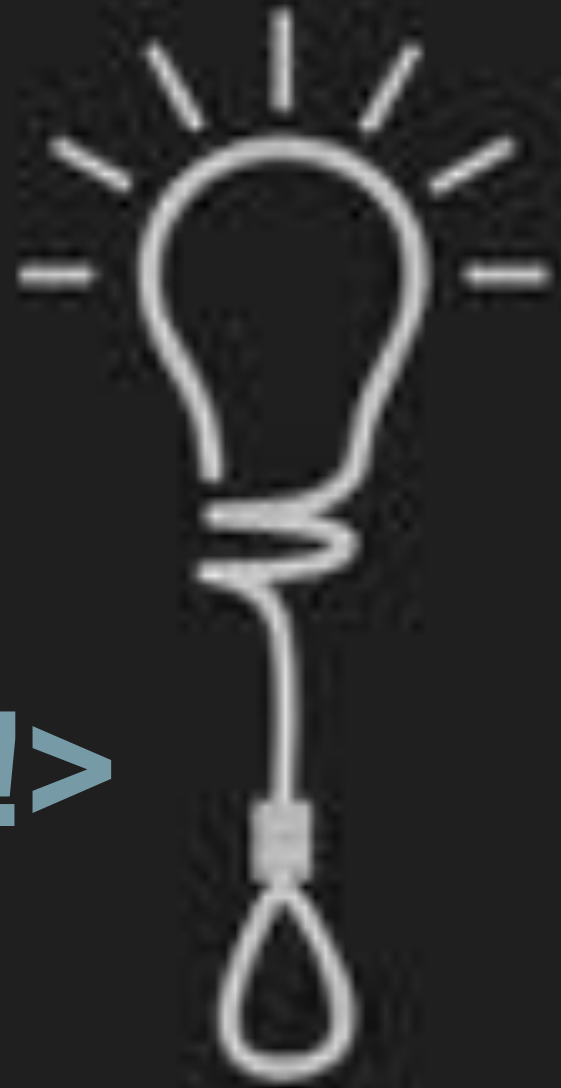
# Take Aways

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+	<a href="https://plus.google.com/+BerndZuther">plus.google.com/+BerndZuther</a>
twitter	<a href="https://twitter.com/Bernd_Z">@Bernd_Z</a>
www	<a href="http://bernd-zuther.de">bernd-zuther.de</a>
github	<a href="https://github.com/zutherb/">github.com/zutherb/</a>

**INNOVATE**

OR DIE

# Get Out of the Building

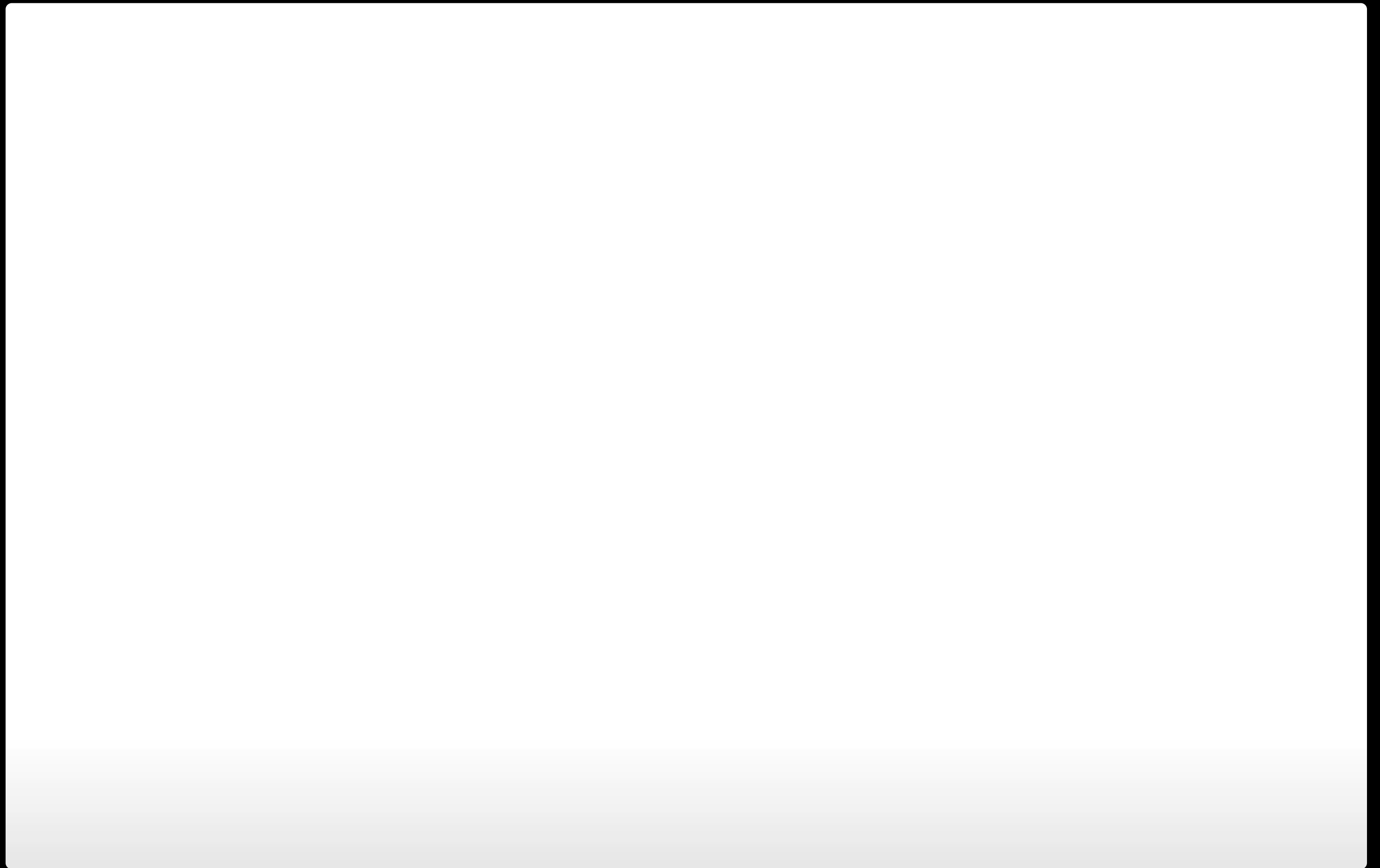
This talk is based on two projects.

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

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



Fork me on GitHub





# 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](#)