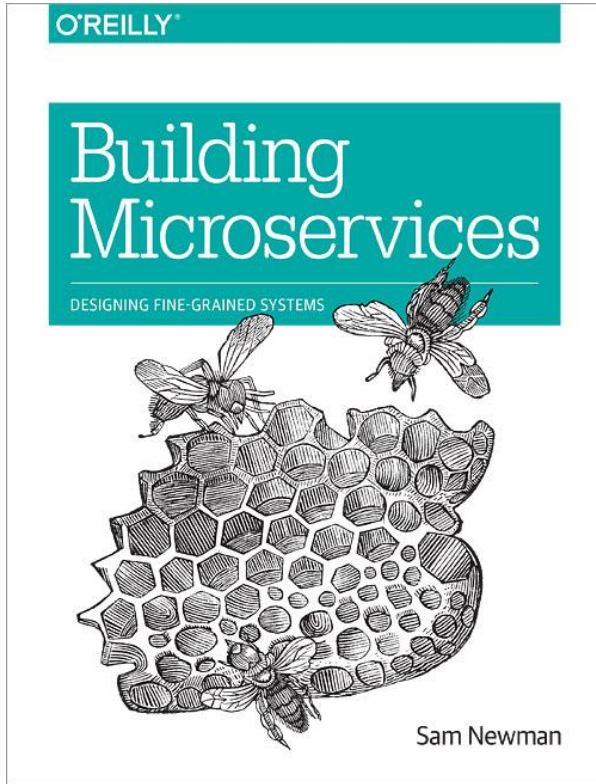


Build und Deployment von Microservices

in Java

Bücher



Agenda

Microservices

Build

Deployment

Microservices

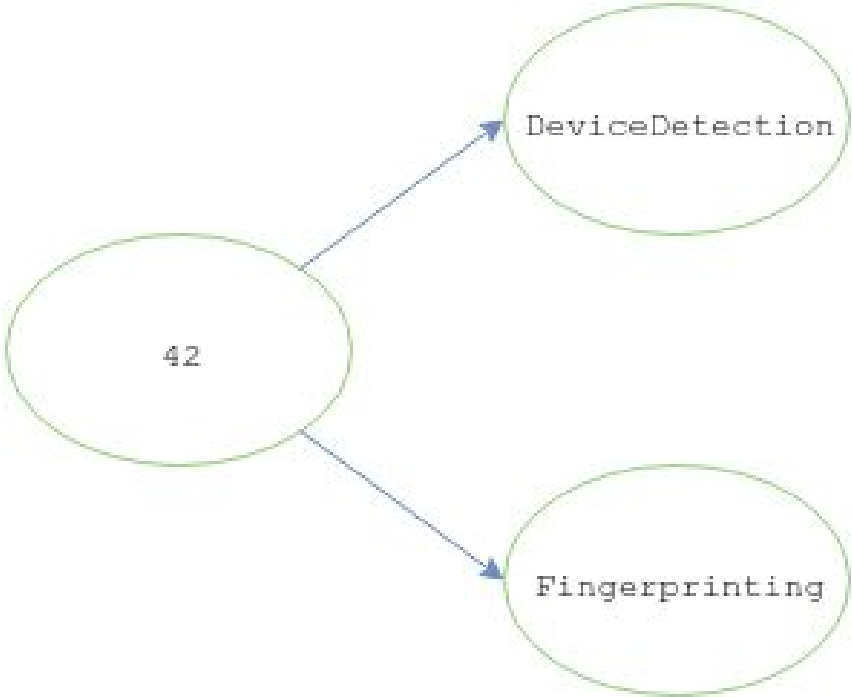
Sam Newman, Building Microservices

“Microservices are small, autonomous services, that work together.”

Charakteristika

Charakteristika

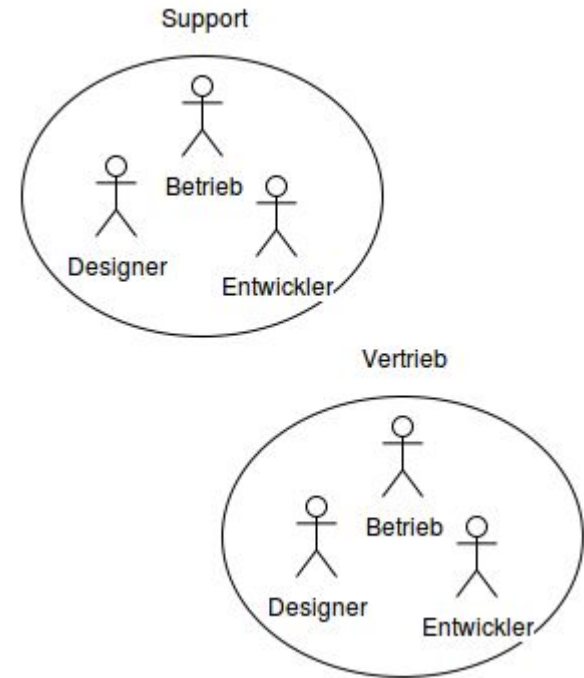
Services als Komponenten



Charakteristika

Services als Komponenten

Organisiert um Geschäftsfähigkeiten



Charakteristika

Services als Komponenten

Organisation von Geschäftsfähigkeiten

Produkte statt Projekte

Charakteristika

Services als Komponenten

Organisation von Geschäftsfähigkeiten

Produkte statt Projekte

Intelligente Endpunkte - dumme Kommunikationswege

Charakteristika

Services als Komponenten

Organisation von Geschäftsfähigkeiten

Produkte statt Projekte

Intelligente Endpunkte - dumme Kommunikationswege

Dezentrale Führung

Charakteristika

Services als Komponenten

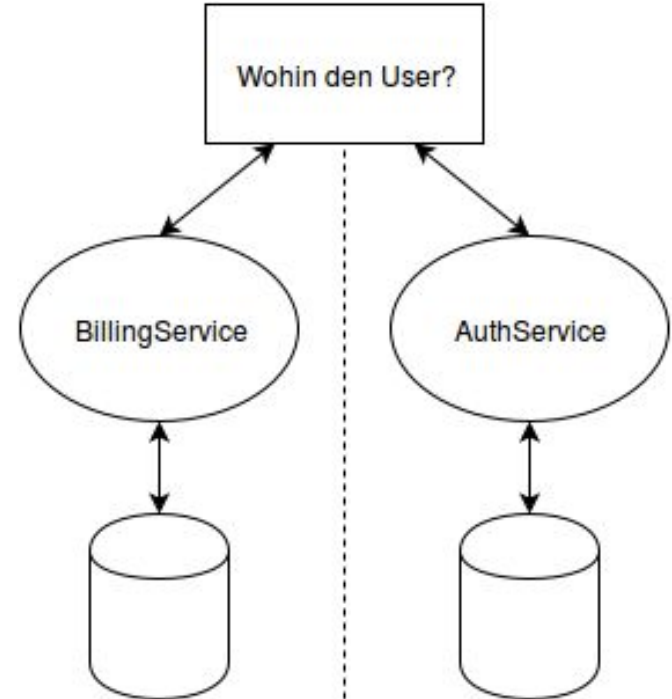
Organisation von Geschäftsfähigkeiten

Produkte statt Projekte

Intelligente Endpunkte - dumme Kommunikationswege

Dezentrale Führung

Dezentrales Datenmanagement



Services als Komponenten

Organisation von Geschäftsfähigkeiten

Produkte statt Projekte

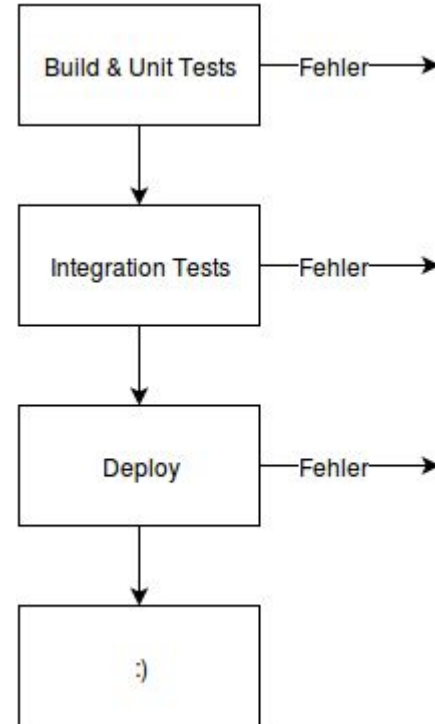
Intelligente Endpunkte - dumme Kommunikationswege

Dezentrale Führung

Dezentrales Datenmanagement

Automatisierung der Infrastruktur

Charakteristika



Charakteristika

Services als Komponenten

Organisation von Geschäftsfähigkeiten

Produkte statt Projekte

Intelligente Endpunkte - dumme Kommunikationswege

Dezentrale Führung

Dezentrales Datenmanagement

Automatisierung der Infrastruktur

Fehlertoleranz

Charakteristika

Services als Komponenten

Organisation von Geschäftsfähigkeiten

Produkte statt Projekte

Intelligente Endpunkte - dumme Kommunikationswege

Dezentrale Führung

Dezentrales Datenmanagement

Automatisierung der Infrastruktur

Fehlertoleranz

Evolutionäres Design

Zusammenfassung

Services als Komponenten

Organisation von Geschäftsfähigkeiten

Produkte statt Projekte

Intelligente Endpunkte - dumme Kommunikationswege

Dezentrale Führung

Dezentrales Datenmanagement

Automatisierung der Infrastruktur

Fehlertoleranz

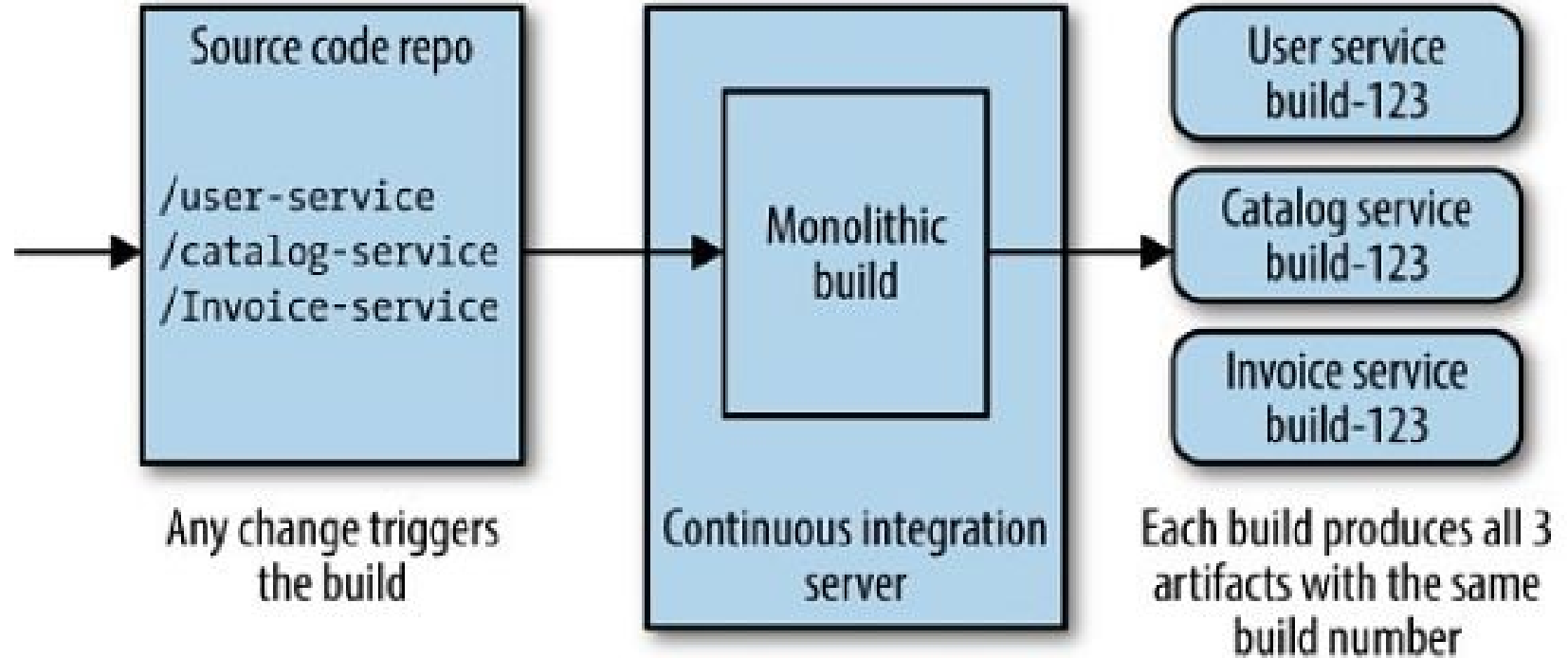
Evolutionäres Design

Build

Continuous Integration/ Continuous Delivery

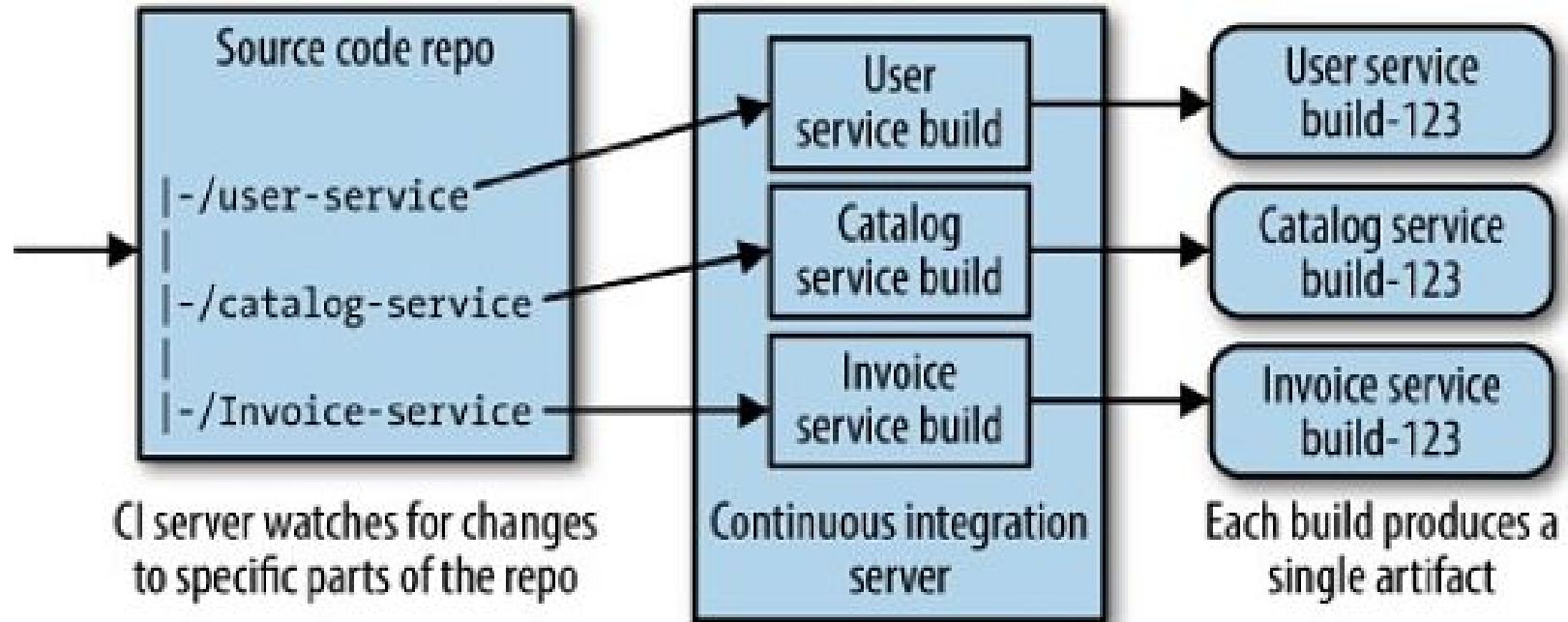
Pipeline?

Ein Repo, ein Build¹

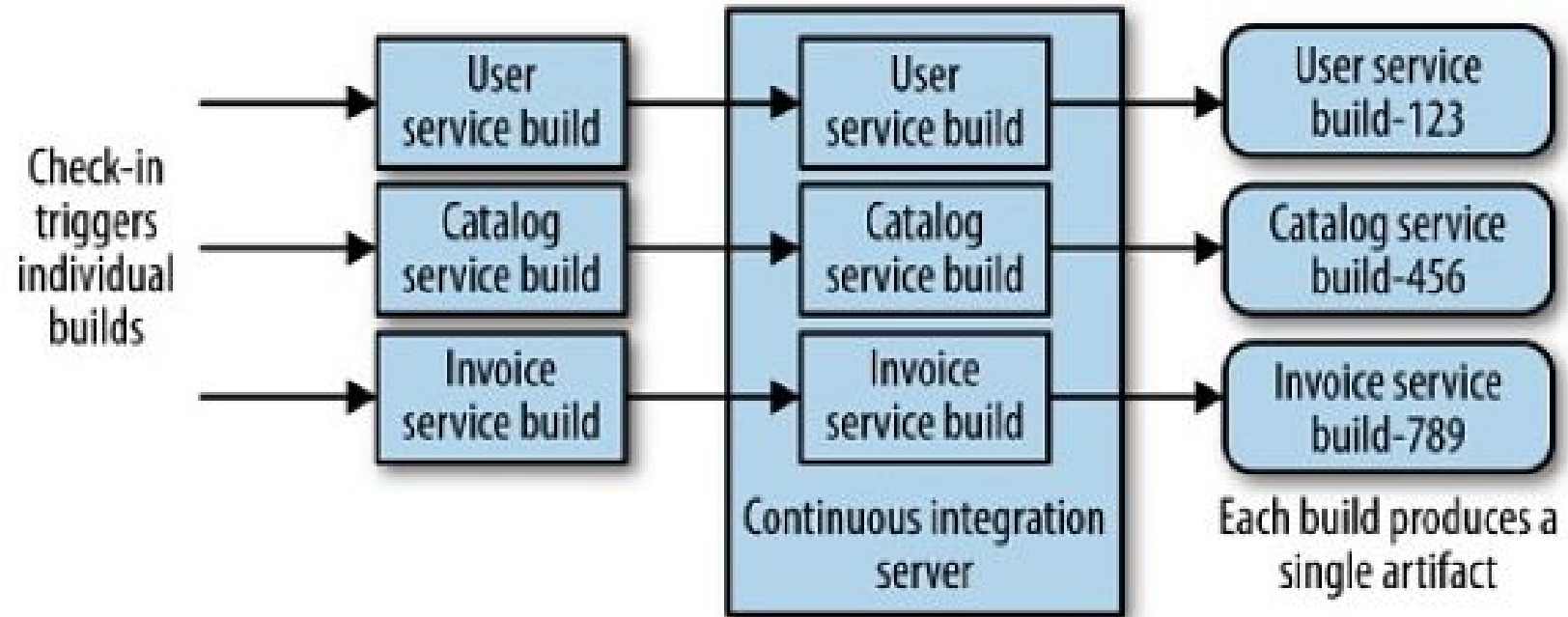


¹ Vgl. S. 105, Newman, Sam, Building Microservices

Ein Repo, mehrere Builds



Mehrere Repos, mehrere Builds







Demo

Just in case

Build

Invoke Gradle script

☐ Invoke Gradle

☒ Use Gradle Wrapper

Make gradlew executable



From Root Build Script Dir



Build step description

Switches

Tasks

clean test shadowJar

Root Build script

Build File

Specify Gradle build file to run. Also, [some environment variables are available to the build script](#)

Force GRADLE_USER_HOME to use workspace ☐

Delete

Build Pipeline

This Build-Pipeline just shows the Build-Pipeline for the HelloWorld Project.



Run

History

Configure

Add Step

Delete

Manage

Pipeline

#1

#1 helloworld-build

Jan 10, 2016 1:46:32 PM
2 min 41 sec

Deployment

Service-to-Host-Mapping

Mehrere Services pro Host

Vorteile

Infrastruktur

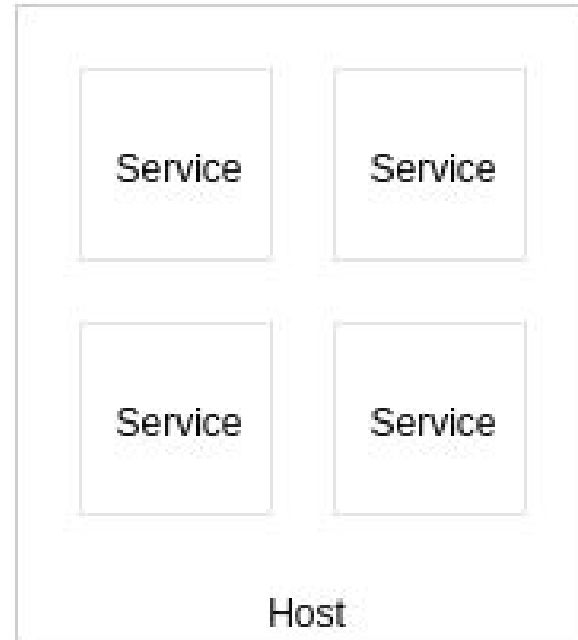
Deployment

Probleme

Konfiguration

Verantwortung

Monitoring



Ein Service pro Host

Vorteile

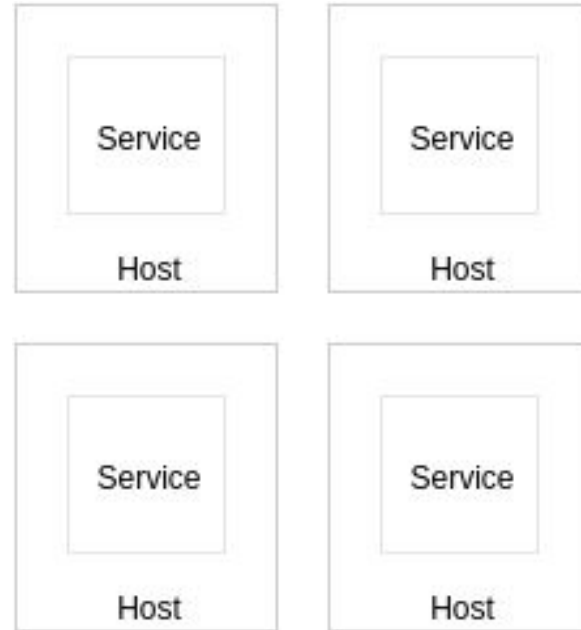
Konfiguration

Verantwortung

Monitoring

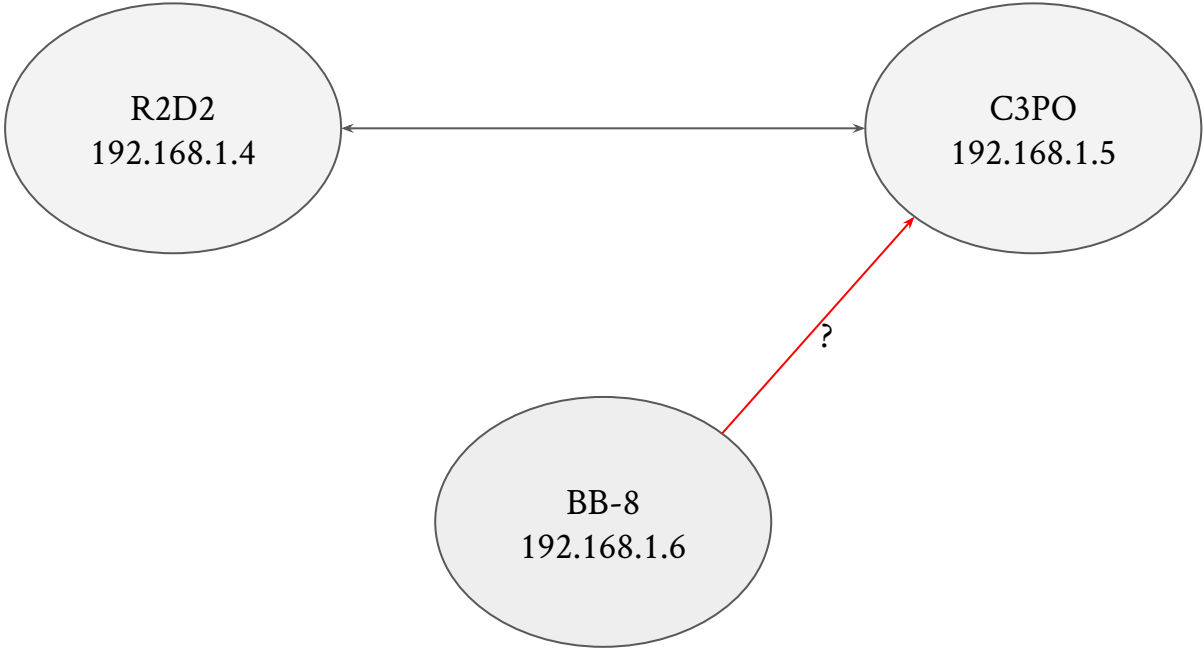
Probleme

Management

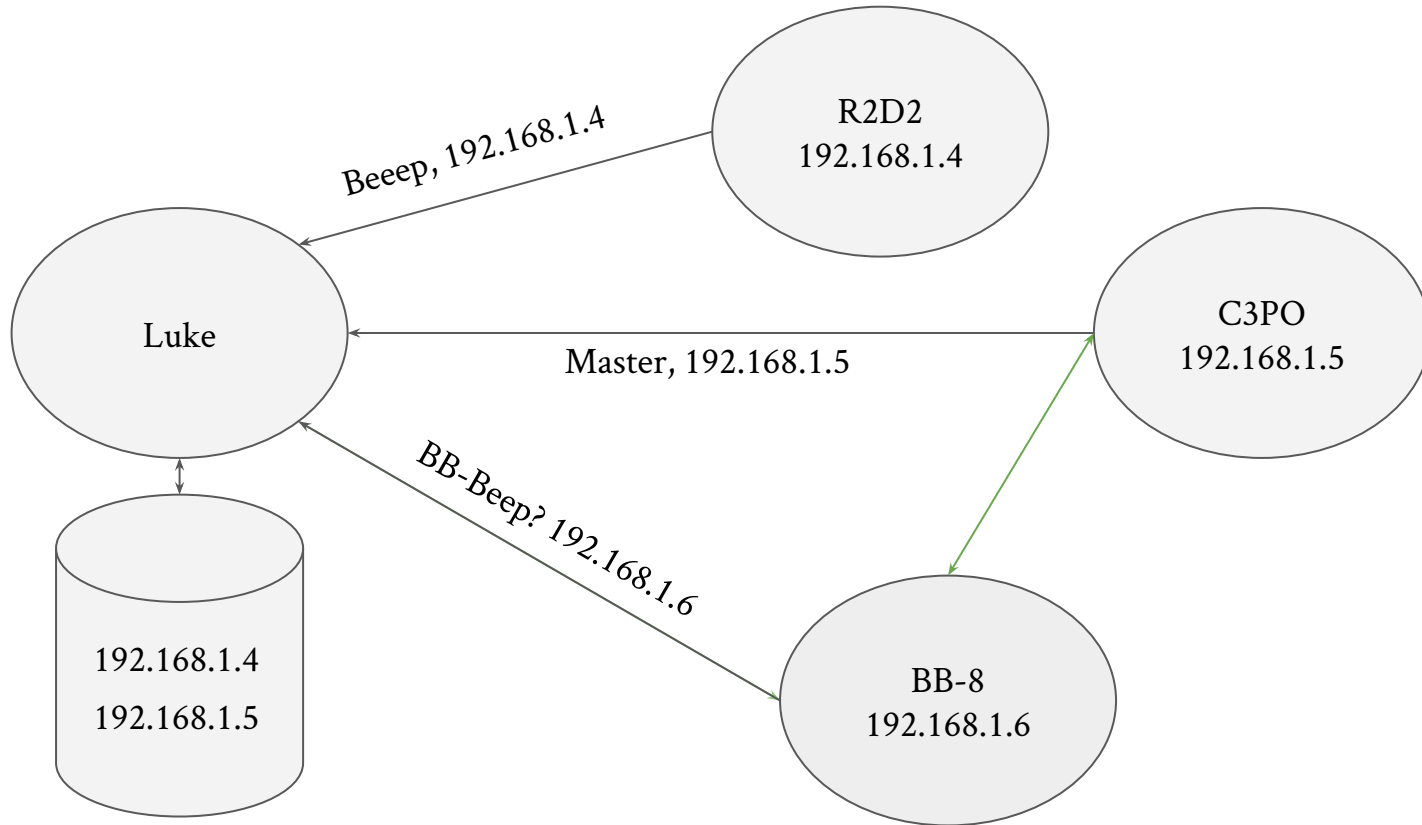


Kommunikation

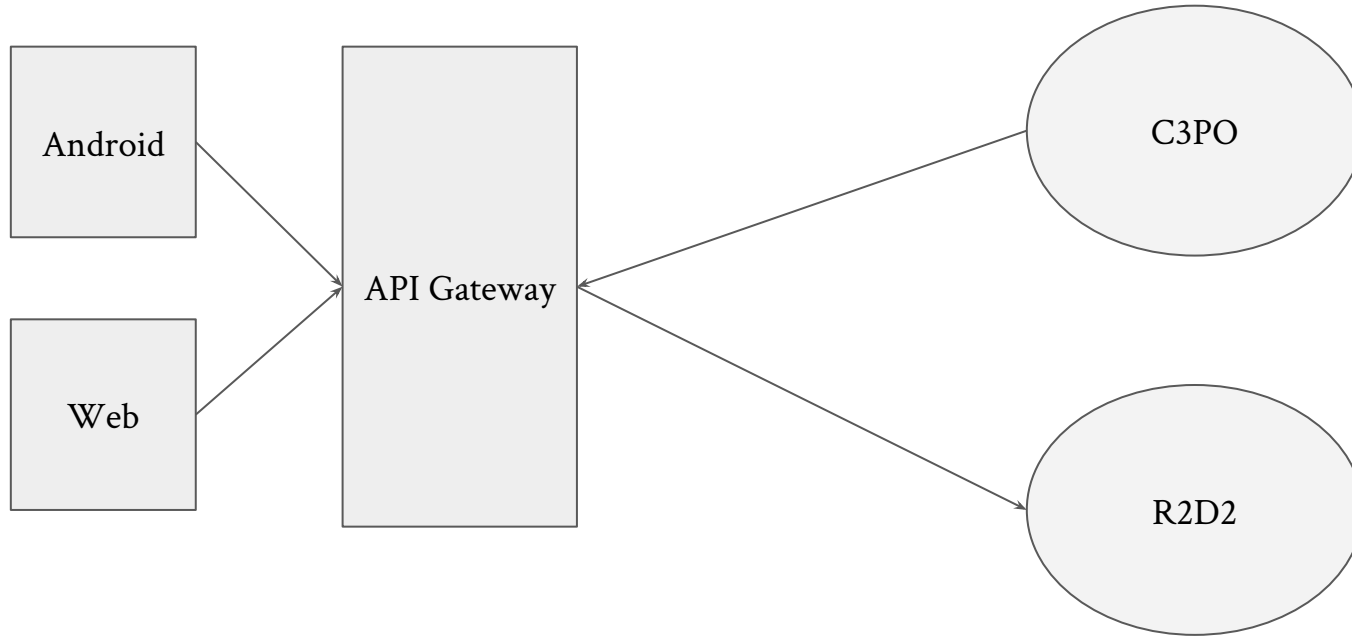
Direkt



Service Registry



API Gateway



Virtualisierung/Container

Docker

```
# Erstelle Container mit derzeitigem Verzeichnis als Kontext  
$ docker build -t helloworld-service[:v2] .
```

```
# Starte helloworld Container  
$ docker run helloworld
```

```
# Starte Container im Hintergrund und forwarde Port 8000 auf 8000  
$ docker run -d -p 8000:8000 helloworld-service
```

```
# Zeige die laufenden Container an  
$ docker ps
```



Deployment Demo

Microservices

Services als Komponenten, Organisation um Geschäftsfähigkeiten, Produkte statt Projekte, dezentrales Datenmanagement/Führung, Automatisierung, Evolutionäres Design, Fehlertoleranz, dumme Kommunikation

Build

Pipeline Alternativen (Single Repo - Single Build, Single Repo - Multiple Build, Multiple Repo - Multiple Build)

Deployment

Service Registry, API Gateway, Docker, Host pro Service, mehrere Services pro Host

<http://martinfowler.com/articles/microservices.html> (Charakteristiken Microservices)

<http://martinfowler.com/bliki/ContinuousDelivery.html> (Continuous Delivery)

<https://www.youtube.com/watch?v=V0FpbDkKYtA> (Building a continuous delivery pipeline with Jenkins and Gradle)

<http://www.martinfowler.com/articles/continuousIntegration.html>

<http://www.amazon.de/Building-Microservices-Sam-Newman/dp/1491950358>

http://www.amazon.de/Microservices-Grundlagen-Softwarearchitekturen-Eberhard-Wolff/dp/3864903130/ref=sr_1_1?ie=UTF8&qid=1453275042&sr=8-1&keywords=microservices

<https://github.com/noobymatze/helloworld-dropwizard>

<https://www.youtube.com/watch?v=PFQnNFe27kU> (Principles of Microservices by Sam Newman)