

# Wir schlachten einen Monolithen!

JavaLand 2018

PhantasiaLand Brühl

13. März 2018

Dirk Mahler & Stephan Pirnbaum

# Was ist eigentlich ein Monolith?

Mo·no·lith, Monolith

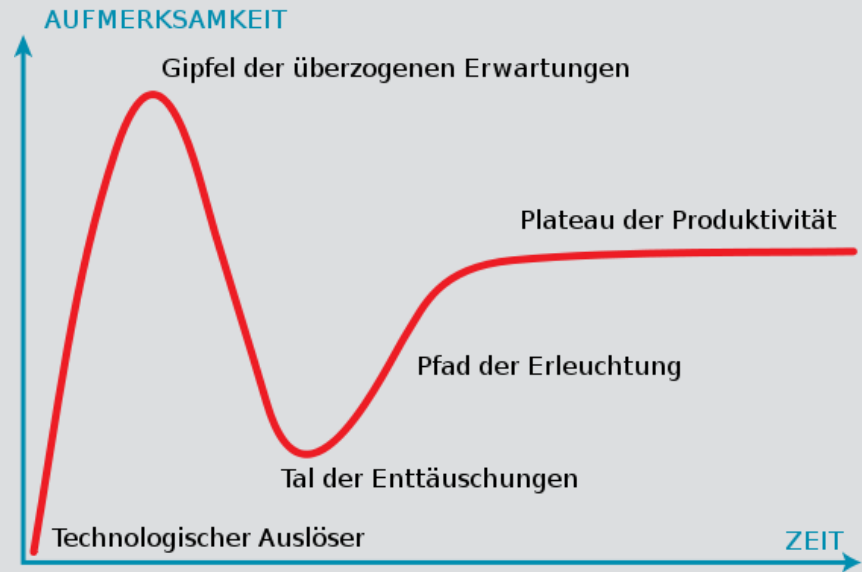
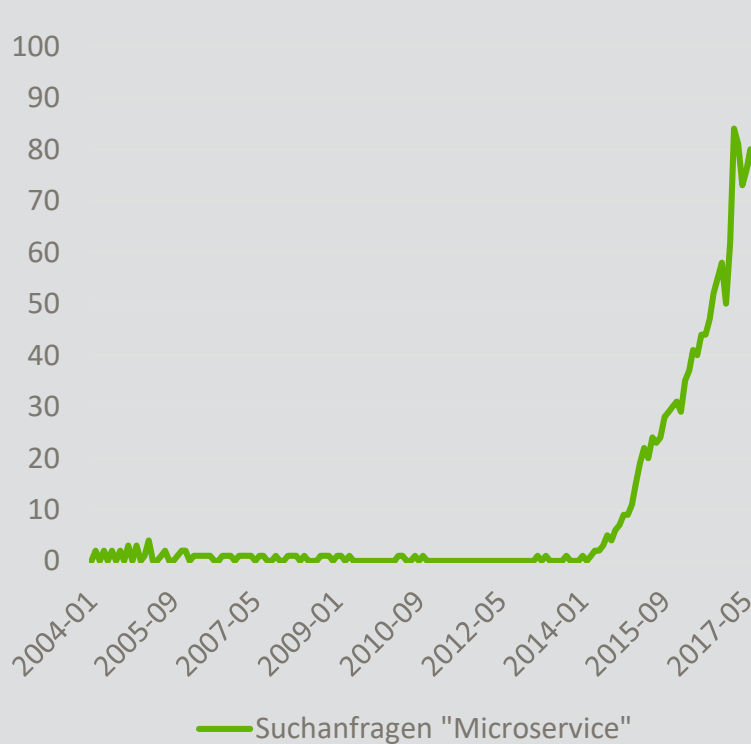
- Substantiv [der]
- Steinsäule, Obelisk o. Ä.  
aus einem einzigen großen Steinblock

# Was ist eigentlich ein Monolith?



[https://upload.wikimedia.org/wikipedia/commons/7/75/India\\_Tamil\\_Nadu\\_Mahendravadi\\_near\\_Kanchipuram.JPG](https://upload.wikimedia.org/wikipedia/commons/7/75/India_Tamil_Nadu_Mahendravadi_near_Kanchipuram.JPG)

# Microservices – Alles nur Hype?



<https://trends.google.de/trends/explore?date=all&q=microservices>

<https://de.wikipedia.org/wiki/Hype-Zyklus>

# Alles besser dank Microservices!?

- Ja!

- Unabhängigkeit
- Fachlich orientiertes Komplexitätsmanagement

- Nein!

- Technische Komplexität durch Verteilung
- Infrastruktur (Deployment, Monitoring, Logging, etc).

# Alles besser dank Microservices!?

*If you can't build a well-structured monolith, what makes you think microservices is the answer?*

Simon Brown

# Alles besser dank Microservices!?

Monolith != Schlecht

Microservices != Gut

# Warum überhaupt zerlegen?

- Schwerfälliger Release-Prozess
- Eingeschränkte Skalierbarkeit einzelner Features
- Kampf um Ressourcen zur Laufzeit
- Beeinträchtigung der Verfügbarkeit



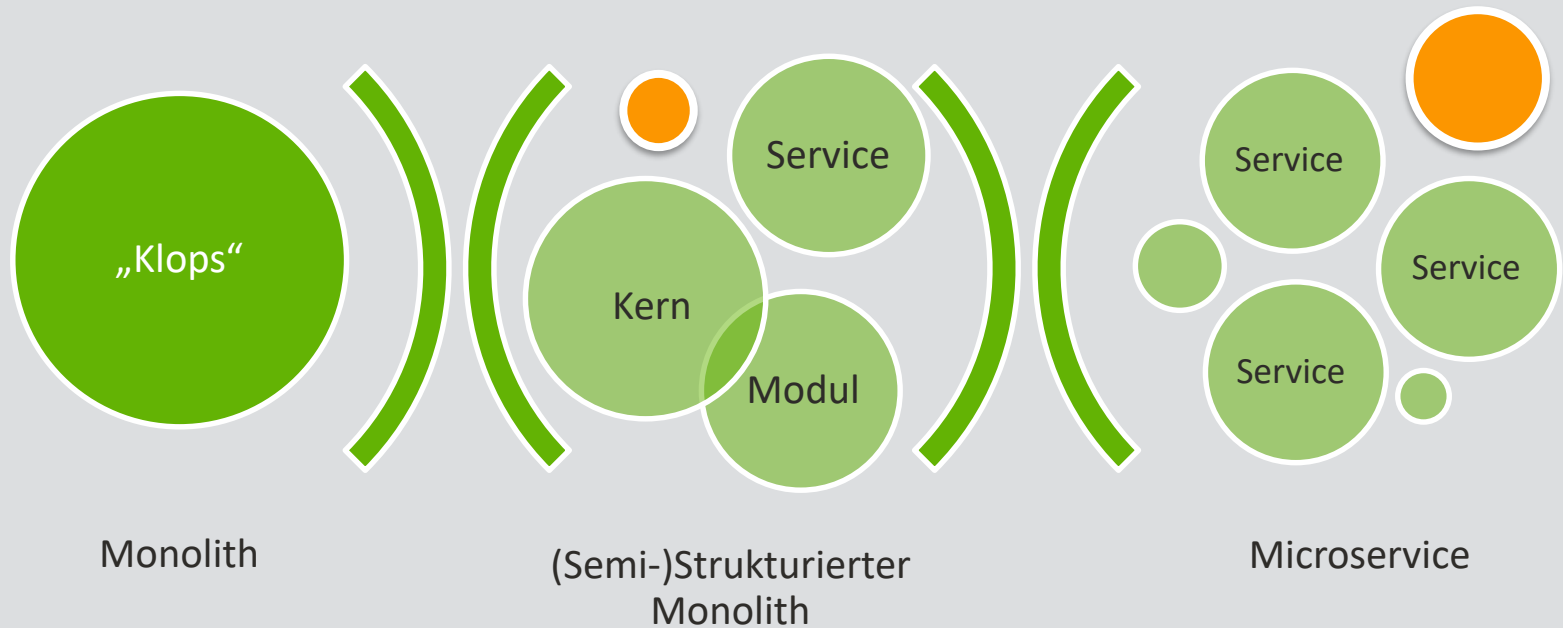
# Vom Monolithen zum Microservice

Kennen wir unseren Monolithen?

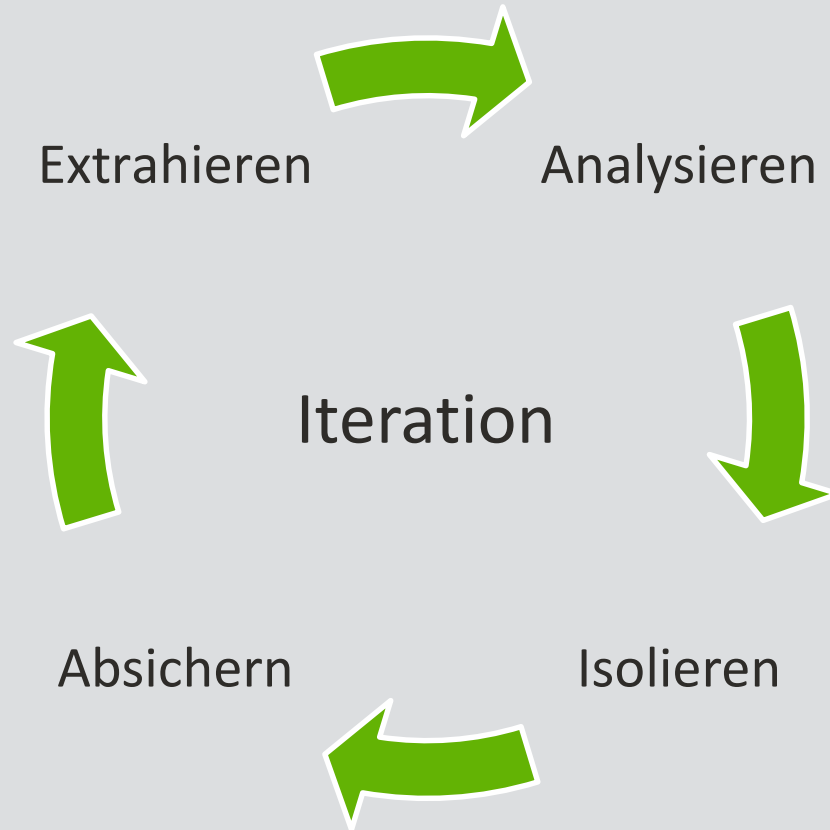
Haben wir Tests?

Was ist mit dem laufenden Geschäft?

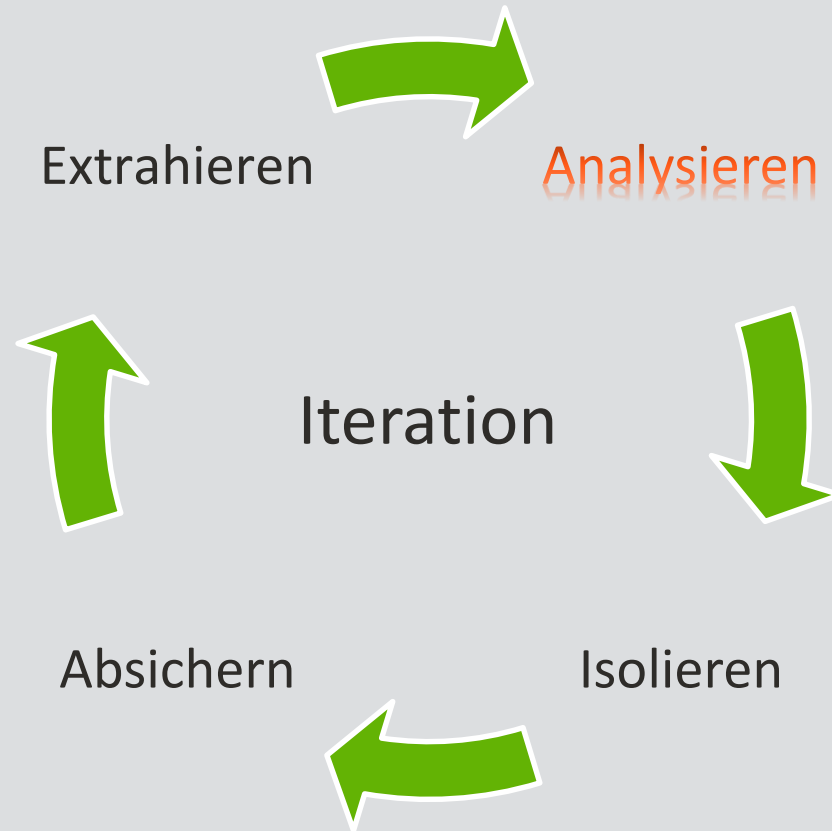
# Vom Monolithen zum Microservice



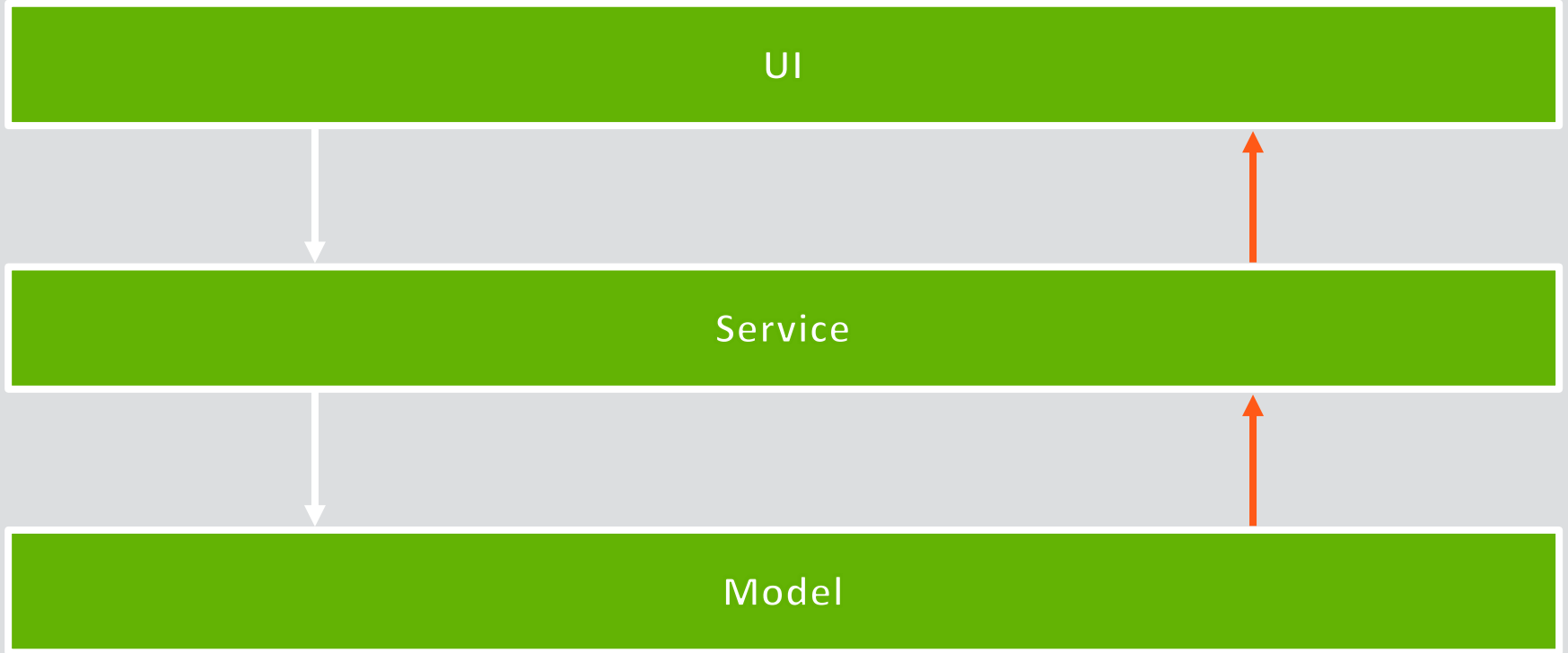
# Iteratives Vorgehen



# Vom Monolithen zum Microservice



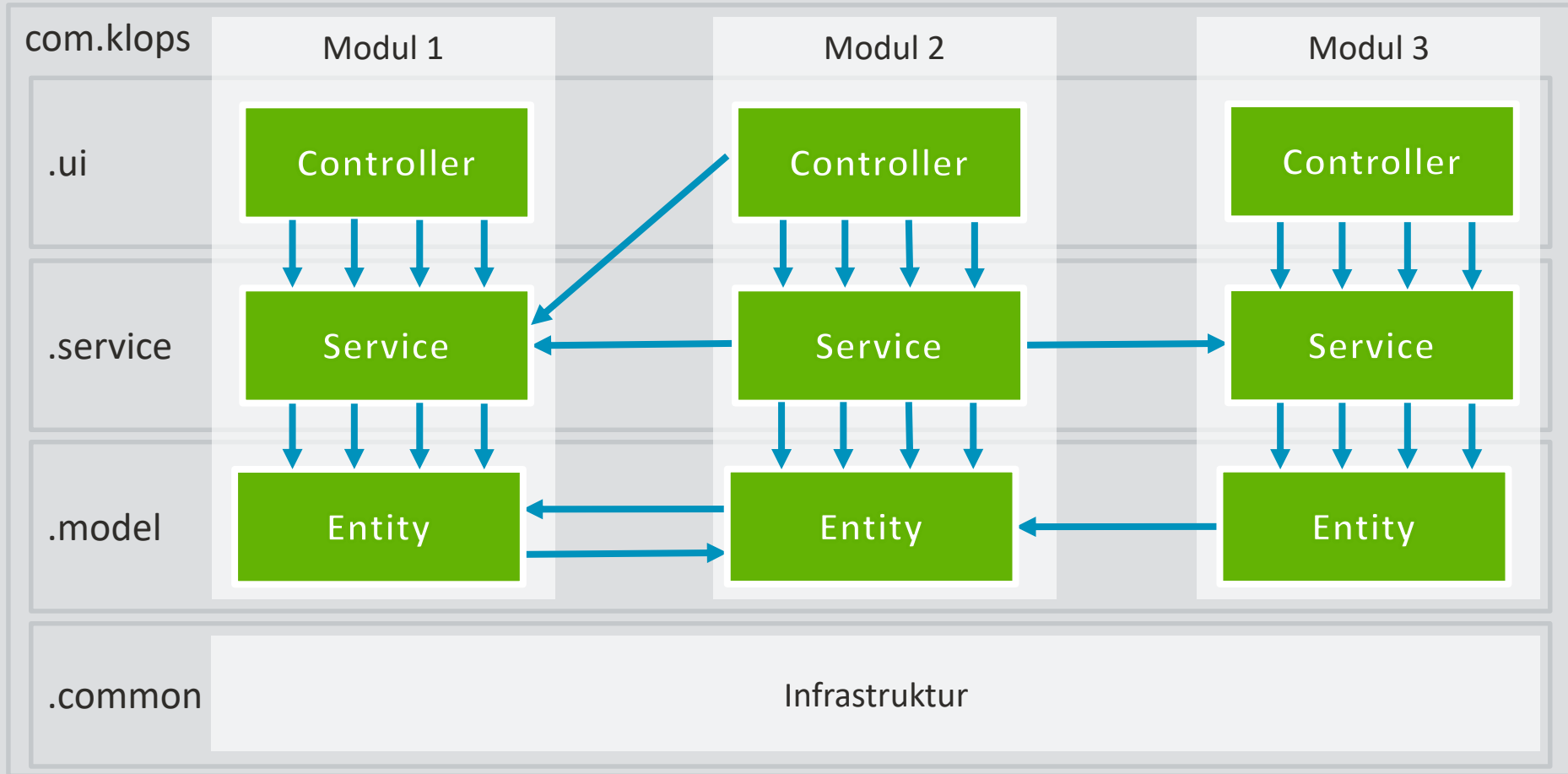
# Analysieren



# Analysieren



# Analysieren

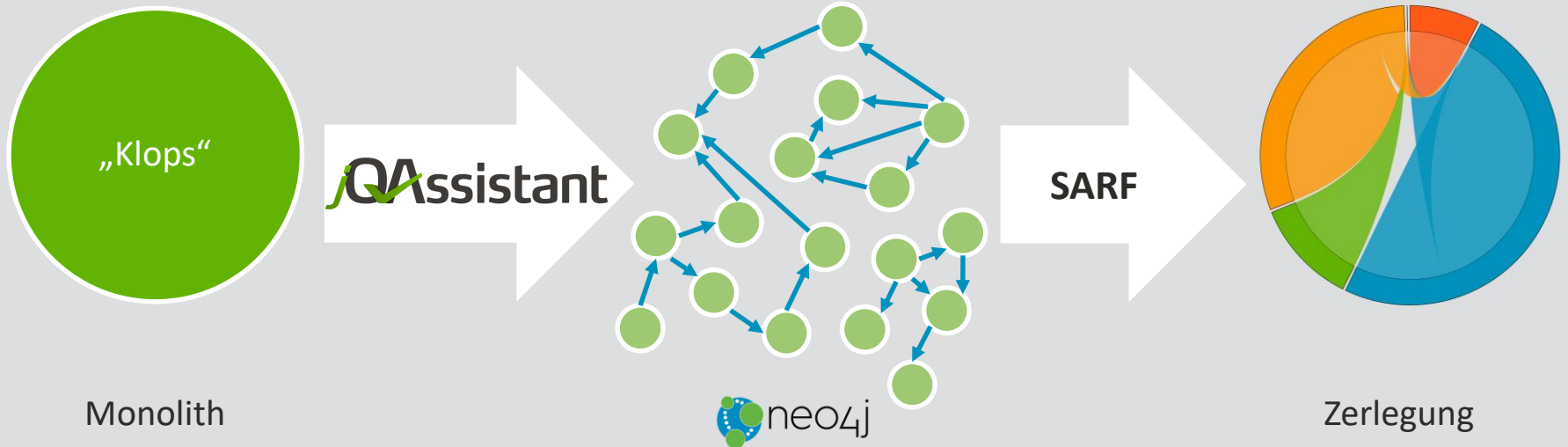


# Analysieren

Beabsichtigte Architektur  
vs.  
Sichtbare Architektur  
vs.  
Reale Architektur

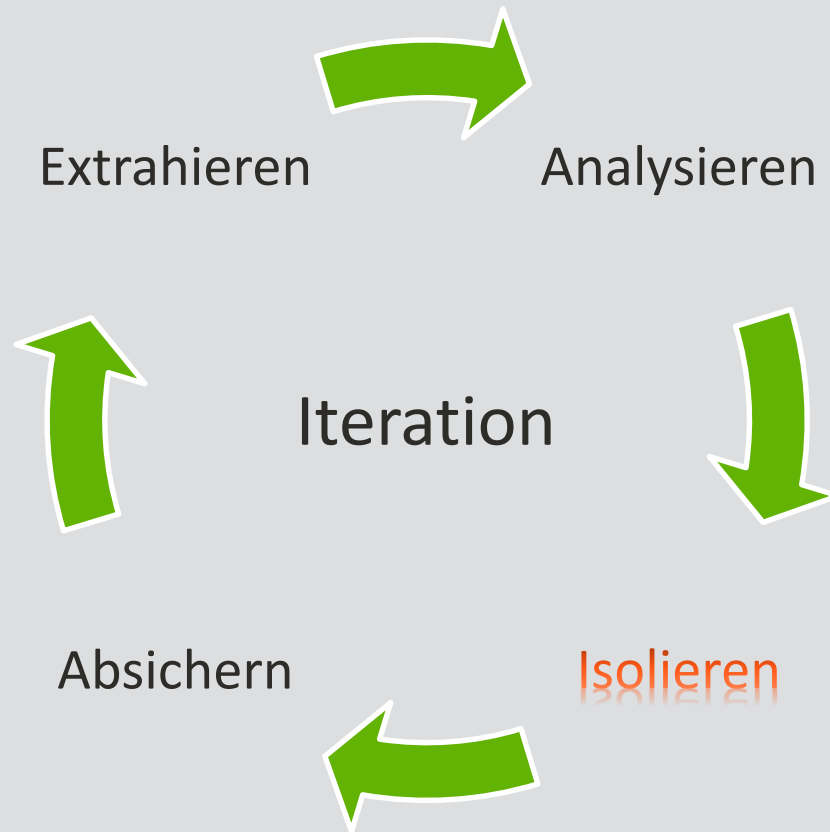


# Analysieren



<http://www.buschmais.de/2018/03/07/dein-system-das-unbekannte-wesen>

# Vom Monolithen zum Microservice



# Isolieren

Monolith

UI

UI

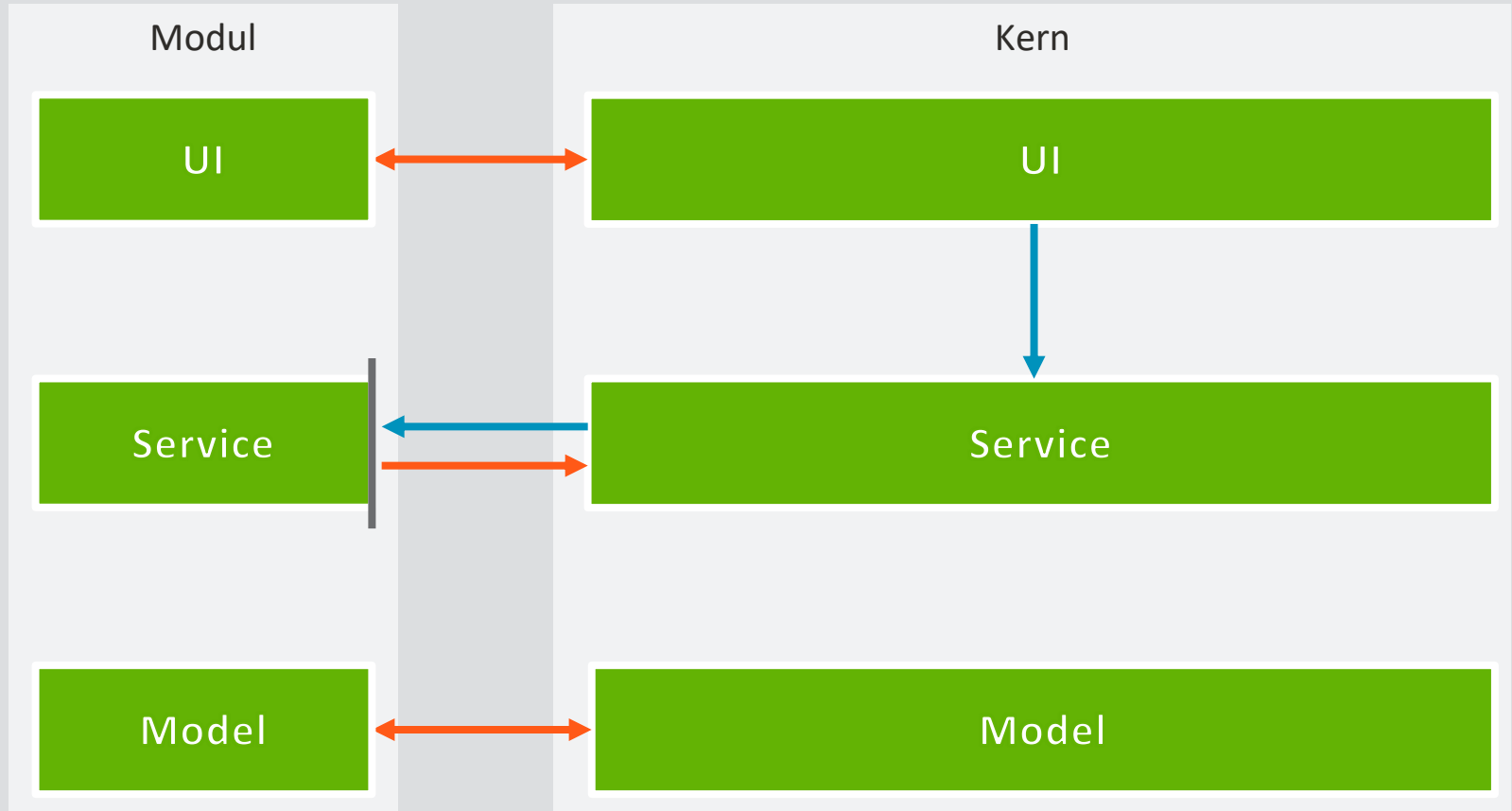
Service

Service

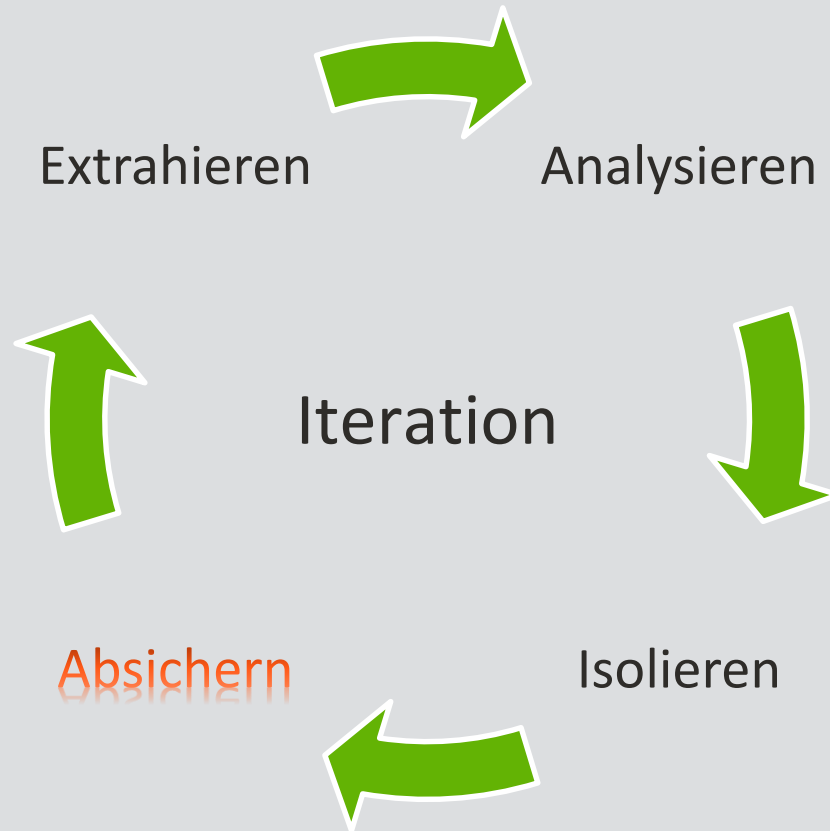
Model

Model

# Isolieren



# Vom Monolithen zum Microservice



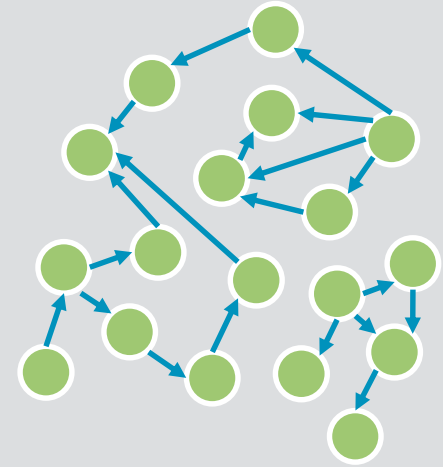
# Absichern



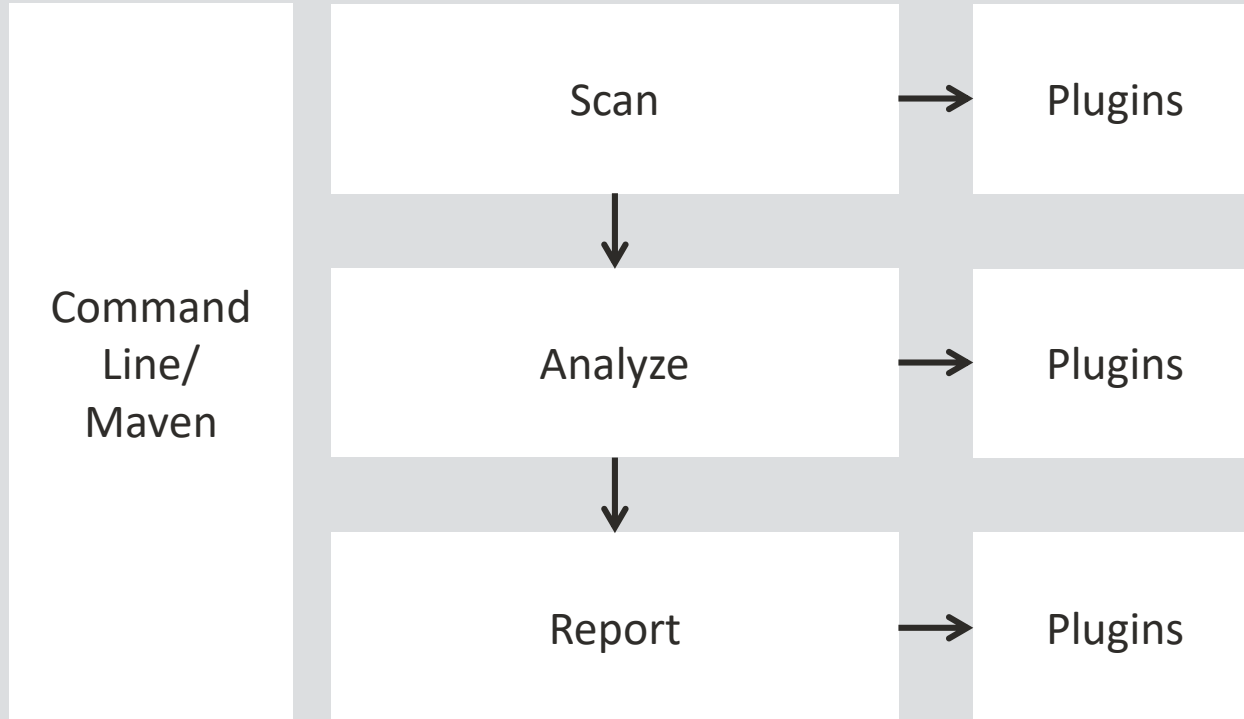
# jQAAssistant

<http://jQAssistant.org>

- Ausführbare Design- und Architekturdokumentation



# Absichern





# Absichern

- Concept

```
MATCH
  (:Maven:Project)-[:CREATES]->(a:Artifact)
WHERE
  a.artifactId IN ["kern","modul"]
SET
  a:Module
RETURN
  a as Module
```

# Absichern

- Constraint

```
MATCH
```

```
  (m1:Module) - [ :CONTAINS ] -> (t1:Type),
```

```
  (m2:Module) - [ :CONTAINS ] -> (t2:Type),
```

```
  (t1) - [ :DEPENDS_ON ] -> (t2)
```

```
WHERE
```

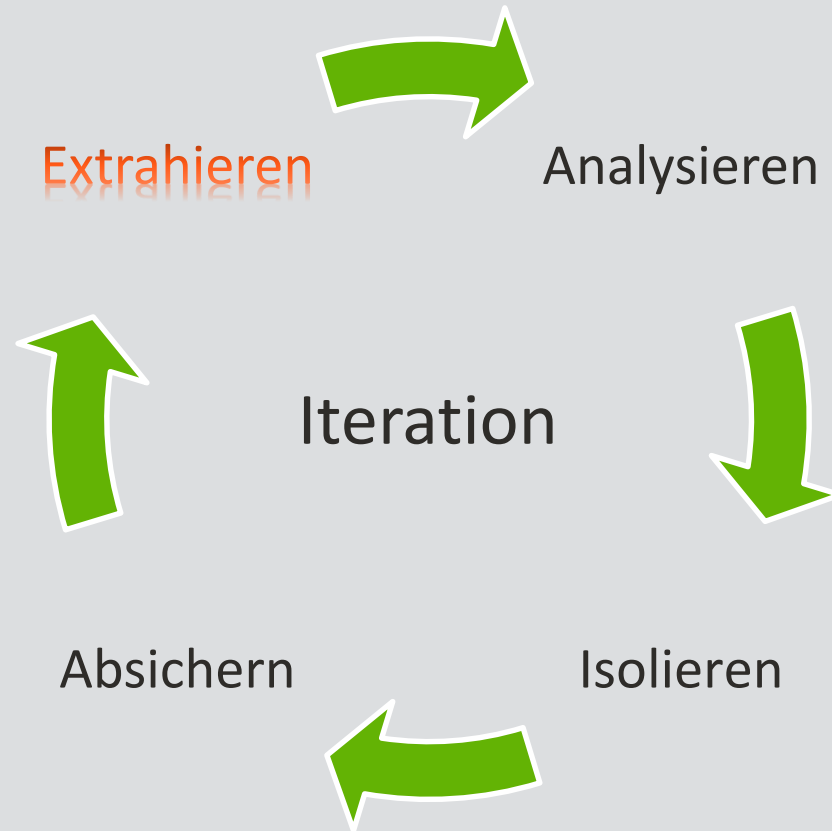
```
  m1 <> m2
```

```
  AND NOT t2:Api
```

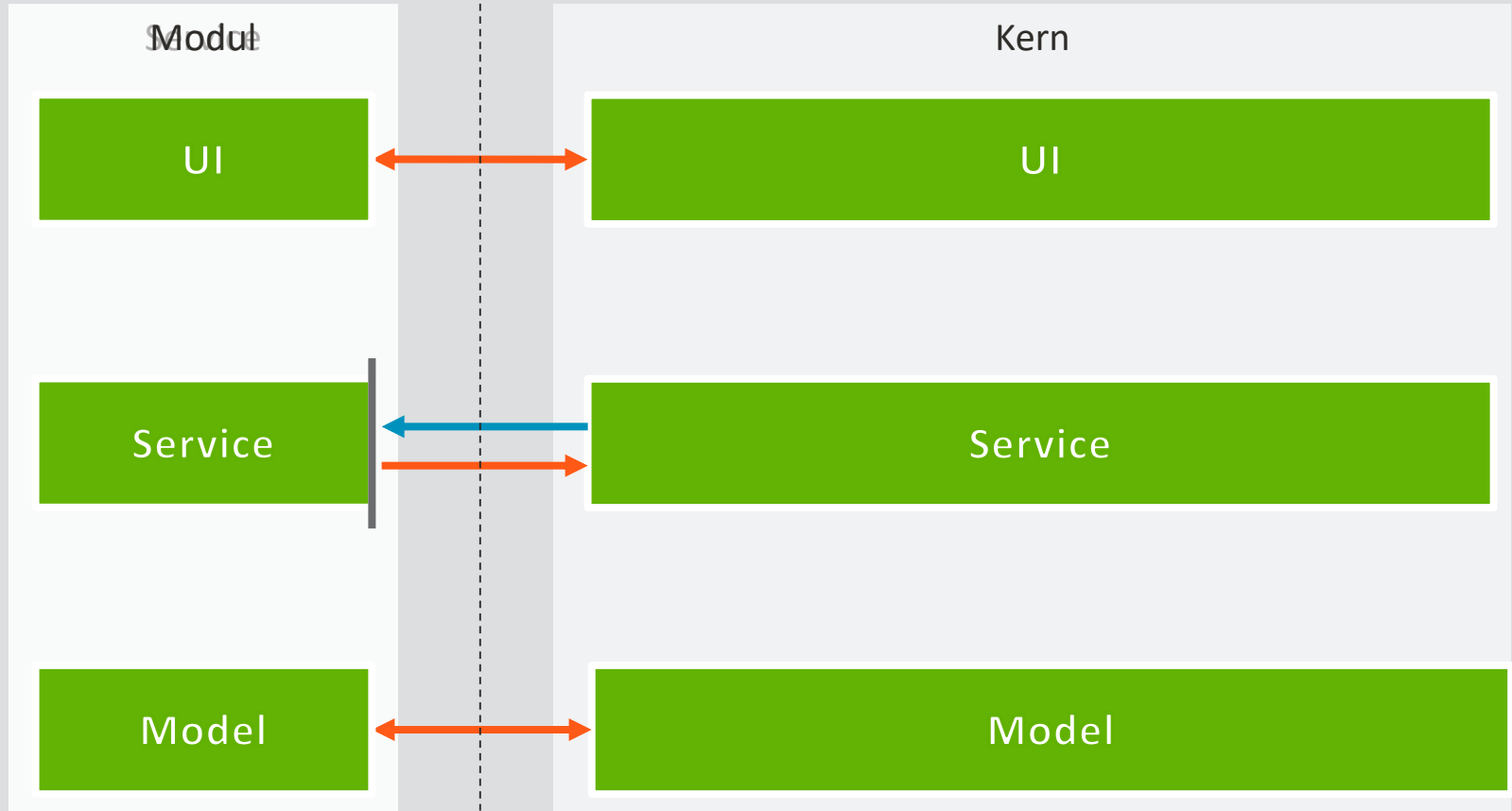
```
RETURN
```

```
  t1 as Dependent, t2 as InternalType
```

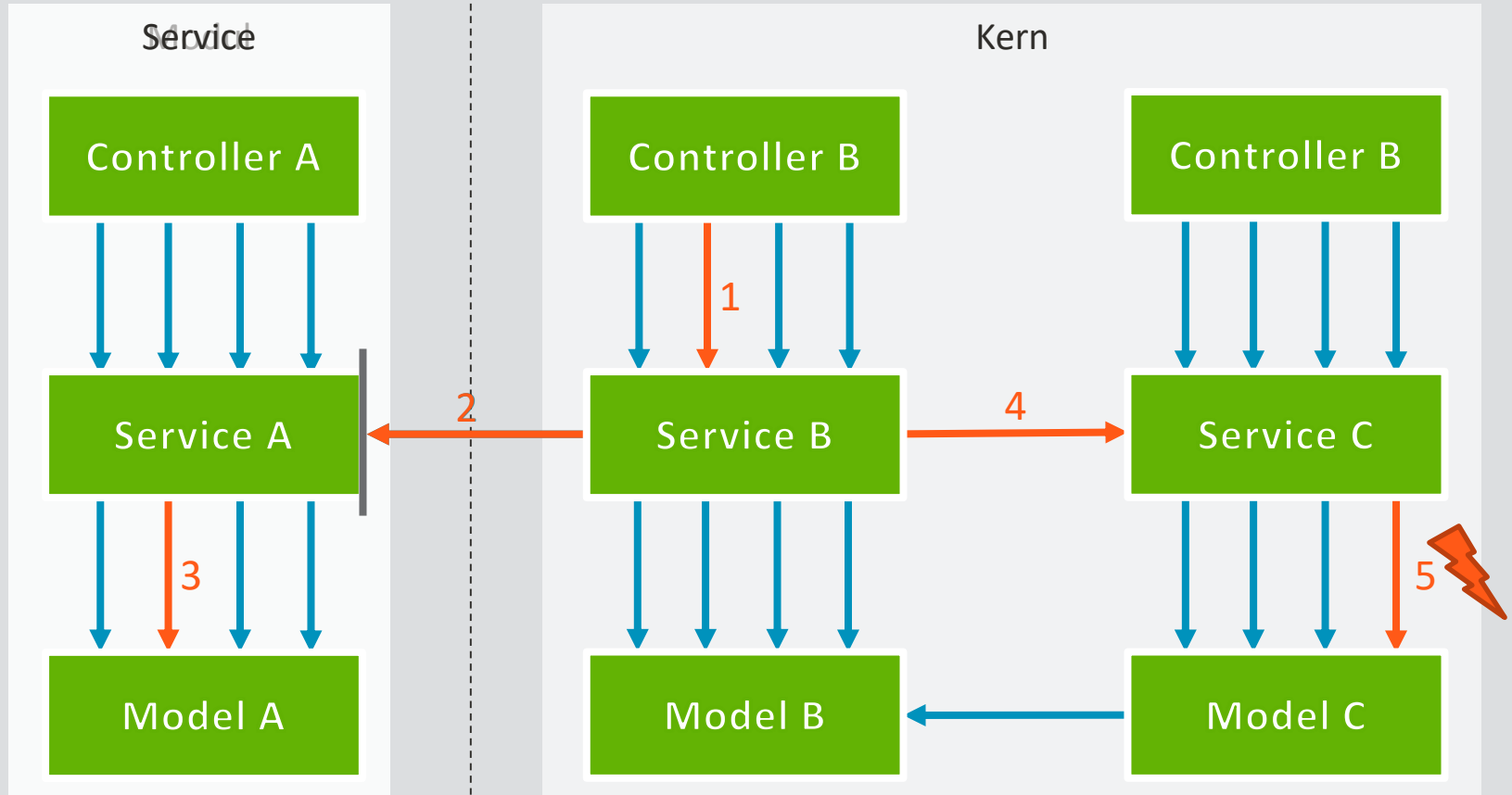
# Vom Monolithen zum Microservice



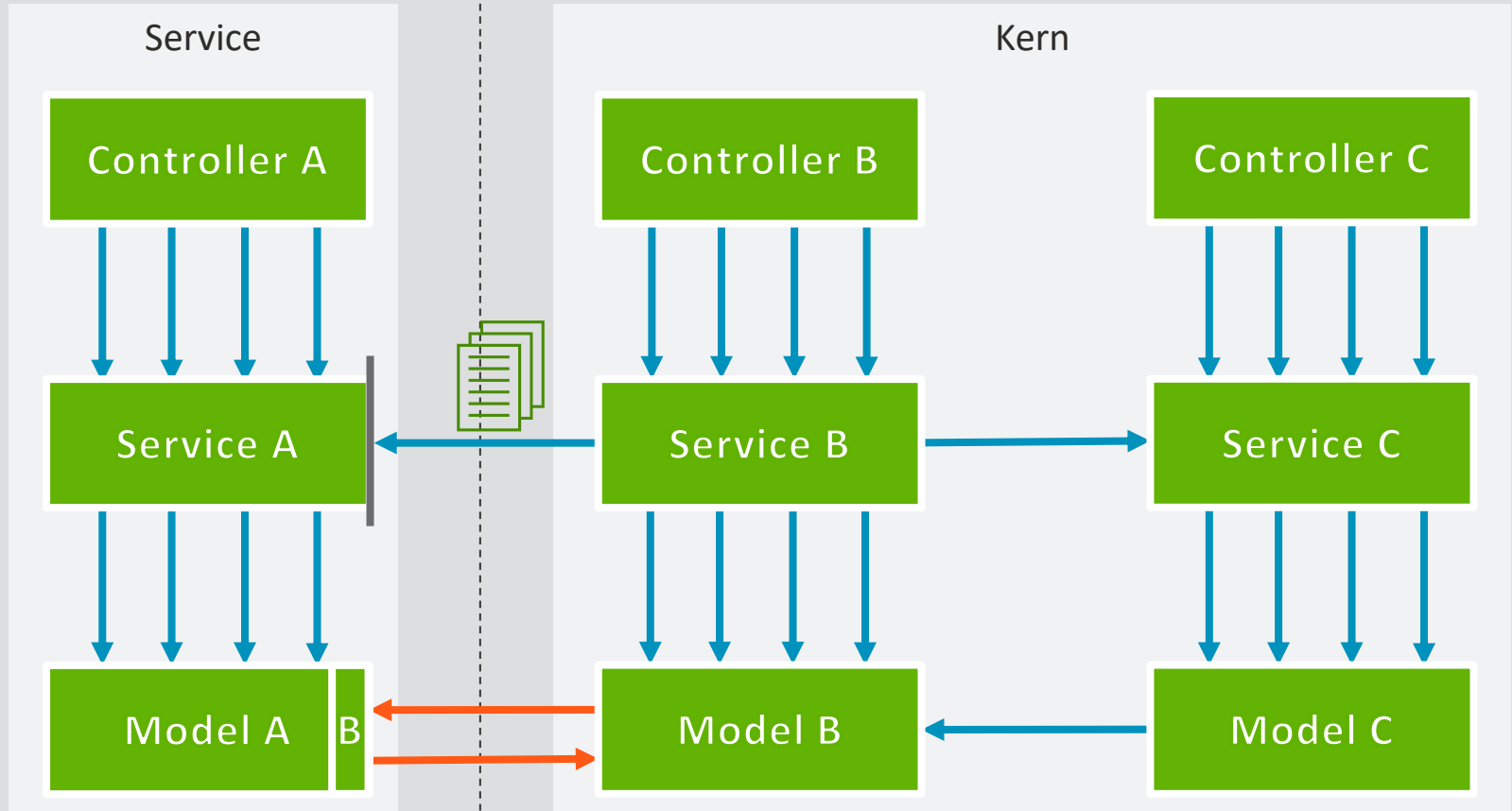
# Extrahieren



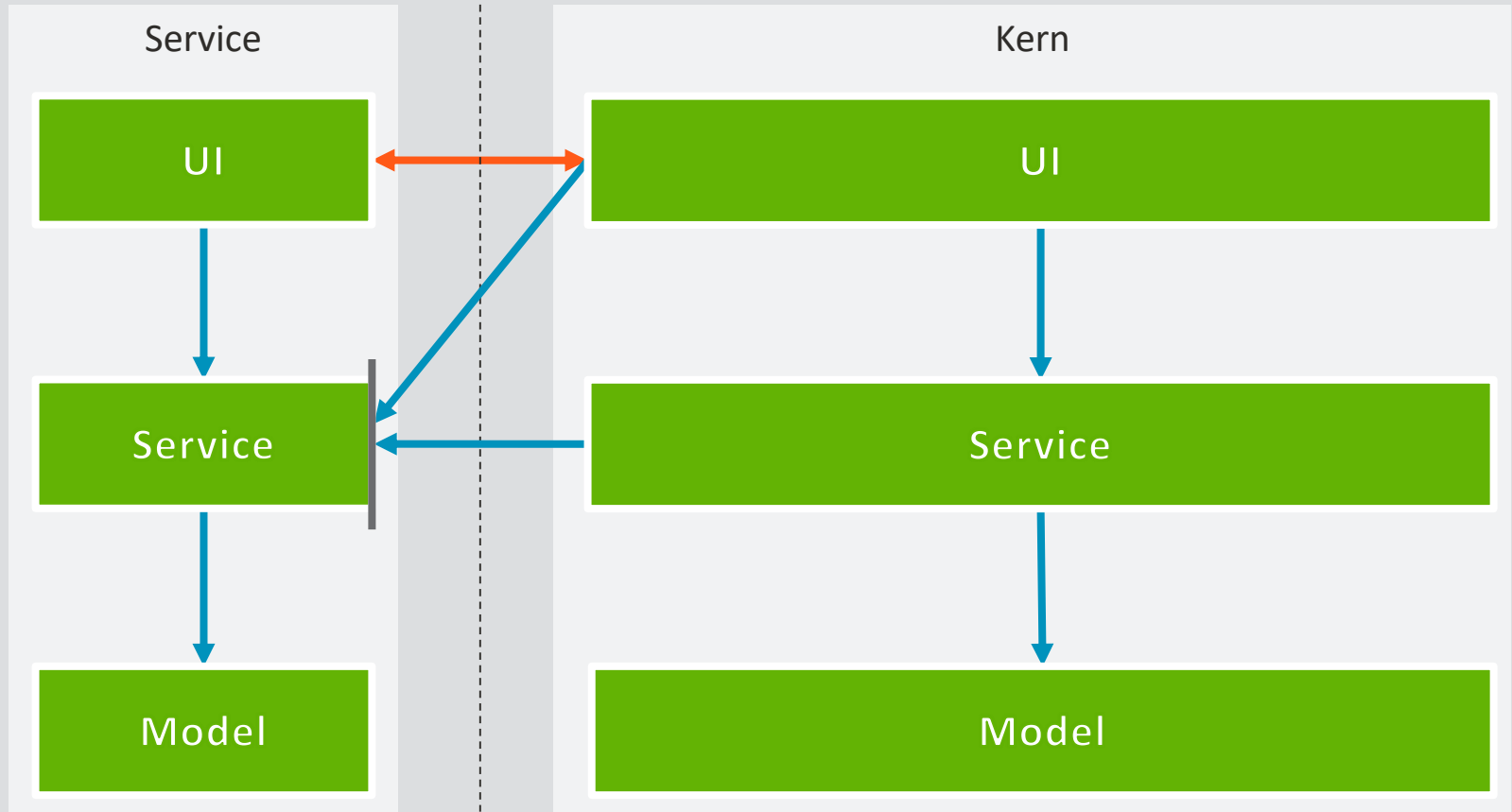
# Extrahieren



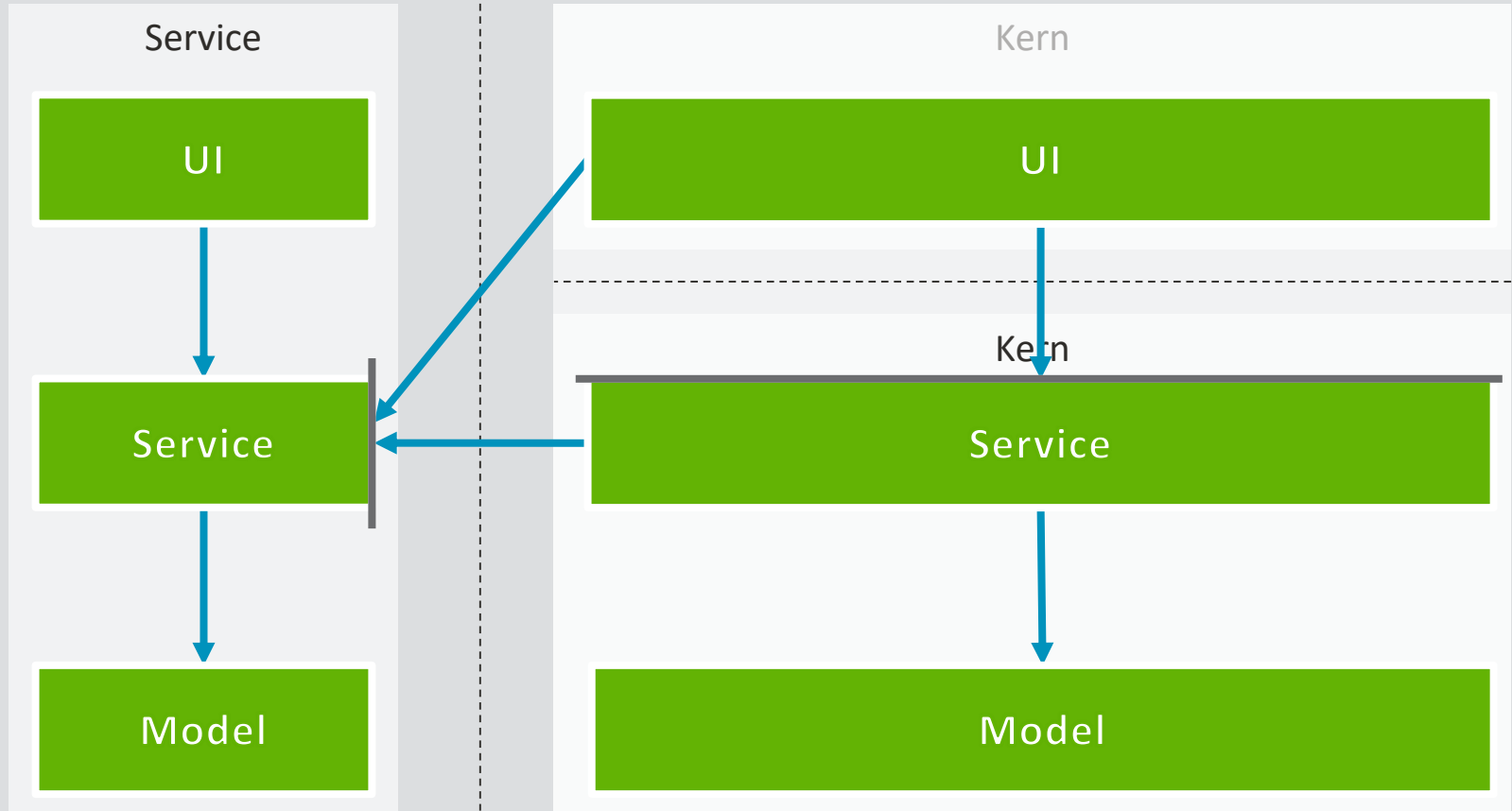
# Extrahieren



# Extrahieren



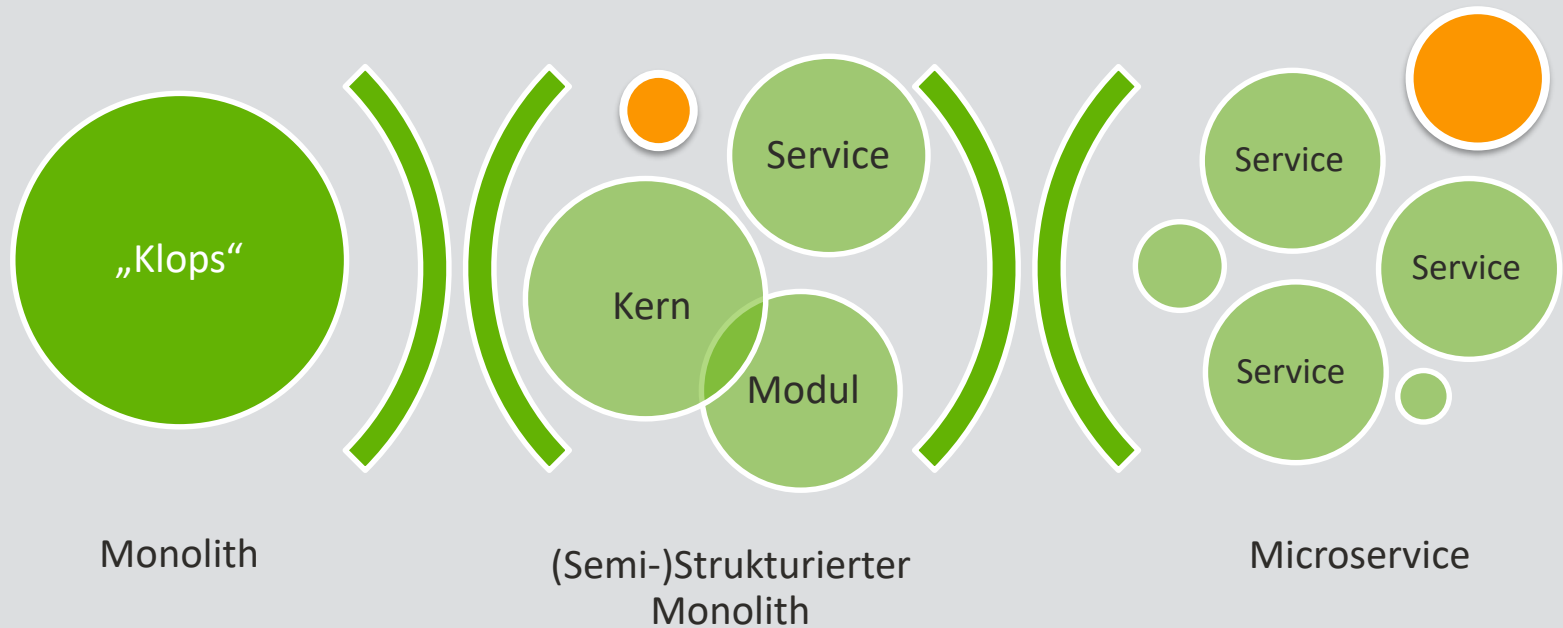
# Extrahieren





- Infrastruktur...
  - Single Sign On
  - Service Discovery & Load Balancer/Proxy
  - Monitoring
  - Logging
  - Delivery & Deployment

# Vom Monolithen zum Microservice



# Fazit

Müssen es immer Microservices sein?

# Fazit

- Monolith vs. Microservices ist Trade-Off-Entscheidung
- Priorisierung nach messbaren Kriterien
- Iteratives Vorgehen in kleinen Schritten
- Restrukturierung des Monolithen als Basis
- Chance zur langfristigen Absicherung

**Vielen Dank!**

**Q&A**