

# **Generierung und Ordnung von Events in verteilten Systemen mit asynchroner Kommunikation**

BACHLORTHESES  
Studiengang Informatik

vorgelegt von  
**Simon Stockhause**

Mai 2020

Referent der Arbeit: Prof. Dr. Harald Ritz  
Korreferent der Arbeit: M.Sc. Pascal Bormann



## **Zusammenfassung**

Die heutigen Bedürfnisse der Anwender ein stets erreichbaren Service zur Verfügung zu haben, stellt hohe Erwartung an Unternehmen und damit hohe Erwartungen an die Infrastruktur ihrer Mircoserviceanwendungen. Die enorme Skalierbarkeit einzelner Komponenten und die ausgezeichnete Ressourcennutzung der Hardware löst viele Probleme der Vergangenheit. Allerdings schafft diese Umstellung neue Herausforderungen die es zu bewältigen gilt.

- Überwachung
- Logs, Metrics, Traces
- Performance



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Problemstellung . . . . .	1
1.3	Forschungsstand . . . . .	1
1.4	Thesisübersicht . . . . .	1
<b>2</b>	<b>Themenüberblick</b>	<b>2</b>
2.1	Verteilte Systeme . . . . .	2
2.1.1	Überwachung von verteilten Systemen . . . . .	2
2.1.2	Synchronisation . . . . .	2
2.1.3	Ordnung von events . . . . .	2
2.2	Bibliotheksentwicklung . . . . .	2
<b>3</b>	<b>Problembeschreibung</b>	<b>3</b>
3.1	Eventgenerierung . . . . .	3
3.1.1	Eventkorrelation . . . . .	3
3.1.2	Synchronisation von Eventgeneratoren . . . . .	3
3.2	Eventübermittlung . . . . .	3
<b>4</b>	<b>Design</b>	<b>4</b>
4.1	Anforderungsanalyse . . . . .	4
4.1.1	Anforderungen . . . . .	4
4.1.1.1	Funktionale Anforderungen . . . . .	4
4.1.1.2	Nicht-Funktionale Anforderungen . . . . .	4
4.2	Datenmodell . . . . .	4
4.2.1	Eventmodell . . . . .	4
4.2.2	Eventgraph . . . . .	4
4.3	Verarbeitungsmodell . . . . .	4
4.3.1	Agenten . . . . .	4
4.3.2	Collectoren . . . . .	4
<b>5</b>	<b>Implementierung</b>	<b>5</b>
5.1	Bibliothek: Traktor . . . . .	5
5.2	Traktor Agent . . . . .	5

5.3	Traktor Registry . . . . .	5
<b>6</b>	<b>Evaluierung</b>	<b>6</b>
6.1	Genauigkeit der Eventgenerierung . . . . .	6
6.1.1	Uhren und Zeit . . . . .	6
6.2	Darstellung der Events . . . . .	6
6.3	Vergleich mit Jaeger . . . . .	6
6.3.1	Datenmodelle . . . . .	6
6.3.2	Bereitstellung . . . . .	6
6.3.3	Ergebnisse . . . . .	6
<b>Fazit</b>		<b>7</b>
1	Ausblick . . . . .	7

# Abbildungsverzeichnis





# 1 | Einleitung

## 1.1 Motivation

## 1.2 Problemstellung

## 1.3 Forschungsstand

## 1.4 Thesisübersicht

## **2 | Themenüberblick**

### **2.1 Verteilte Systeme**

#### **2.1.1 Überwachung von verteilten Systemen**

#### **2.1.2 Synchronisation**

#### **2.1.3 Ordnung von events**

### **2.2 Bibliotheksentwicklung**

## **3 | Problembeschreibung**

### **3.1 Eventgenerierung**

#### **3.1.1 Eventkorrelation**

#### **3.1.2 Synchronisation von Eventgeneratoren**

### **3.2 Eventübermittlung**

## **4 | Design**

### **4.1 Anforderungsanalyse**

#### **4.1.1 Anforderungen**

##### **4.1.1.1 Funktionale Anforderungen**

##### **4.1.1.2 Nicht-Funktionale Anforderungen**

### **4.2 Datenmodell**

#### **4.2.1 Eventmodell**

#### **4.2.2 Eventgraph**

### **4.3 Verarbeitungsmodell**

#### **4.3.1 Agenten**

#### **4.3.2 Collectoren**

## **5 | Implementierung**

### **5.1 Bilbiothek: Traktor**

### **5.2 Traktor Agent**

### **5.3 Traktor Registry**

## **6 | Evaluierung**

### **6.1 Genauigkeit der Eventgenerierung**

#### **6.1.1 Uhren und Zeit**

### **6.2 Darstellung der Events**

### **6.3 Vergleich mit Jaeger**

#### **6.3.1 Datenmodelle**

#### **6.3.2 Bereitstellung**

#### **6.3.3 Ergebnisse**

# 7 | Fazit

## 7.1 Ausblick