

# WODSS

Event Planner

**Autoren: Andreas Gassmann, Jonas Frehner,  
Lukas Schönbächler**

**n|w** University of Applied Sciences and Arts  
Northwestern Switzerland

FHNW

Schweiz

März 27, 2017

### **Abstract**

Das Ziel dieser Arbeit ist die Entwicklung eines Tools welches das Managen von IT-Kolloquien der FHNW erlaubt.

# Contents

<b>1</b>	<b>Architektur</b>	<b>3</b>
1.1	Microservices . . . . .	3
1.1.1	Registry . . . . .	3
1.1.2	Eventmanagement . . . . .	3
1.1.3	Frontend . . . . .	3
1.1.4	Mailer . . . . .	3
1.1.5	Scheduler . . . . .	3
	<b>Appendices</b>	<b>4</b>

# 1 Architektur

Unsere Implementation des "Event-Planners" wurde mit Microservices realisiert. Wir verwenden dazu "Eureka", eine Library entwickelt von Netflix <https://github.com/netflix/eureka>. Unsere Applikation verwendet fünf Microservices, welche im nachfolgenden Abschnitt genauer beschrieben sind. Die Kommunikation zwischen den Microservices läuft über REST-APIs.

## 1.1 Microservices

### 1.1.1 Registry

Das "Bindeglied" zwischen den verschiedenen Microservices. Die Microservices verbinden sich auf die Registry um dort die URIs der anderen Microservices zu bekommen. Dadurch wird es möglich, die Applikation auf mehreren Servern verteilt laufen zu lassen.

### 1.1.2 Eventmanagement

Beinhaltet alle Entitäten sowie die Businesslogik unserer Applikation und stellt CRUD Operationen über eine REST-API zur Verfügung. Bestimmte Endpunkte sind dabei nur nach einer Authentifizierung und Autorisierung möglich.

### 1.1.3 Frontend

Hostet die statischen Assets unserer Webseite und stellt sie über eine http/https Schnittstelle zur Verfügung. Die Webseite, welche mit Angular und Ionic erstellt wurde, ist eine Single Page App und lädt alle Daten asynchron vom eventmanagement service.

### 1.1.4 Mailer

Ein sehr simpler Microservice der eine API zum Versenden von Mails bietet. Diese API kann nur mit einem Token angesprochen werden und ist nur für die Verwendung aus anderen Microservices angedacht.

### 1.1.5 Scheduler

Der Scheduler in bestimmten Intervallen oder zu bestimmten Zeiten Tasks aus. Ein Beispiel hierfür wäre, nicht verwendete (keine Referenz mehr) Mediadateien jede Nacht zu löschen. Hier wird auch geprüft, ob das Anmeldefenster eines Events geschlossen wurde und verschickt dann (über den "mailer" microservice" emails an alle Beteiligten.

# Appendices

## **Github Repository**

[https://github.com/lukeisontheroad/simple\\_event\\_planner](https://github.com/lukeisontheroad/simple_event_planner)