



**Hochschule  
Bonn-Rhein-Sieg**  
*University of Applied Sciences*

**Fachbereich Informatik**  
*Department of Computer Science*

# **Bachelorarbeit**

im Bachelor-Studiengang Wirtschaftsinformatik

**Entwicklung einer Schnittstelle für die Anbindung von  
austauschbaren Datenquellen an KI-Algorithmen**

**von**

**Laurenz Anton Dilba**

Erstprüfer: Prof. Dr. Matthias Bertram  
Zweitprüfer: Prof. Dr. Wolfgang Heiden  
Unternehmen: CONET Solutions GmbH

Eingereicht am: 1. Dezember 2022

### **Erklärung**

Hiermit erkläre ich wahrheitsgemäß, dass ich den vorliegenden Bericht selbst angefertigt habe. Der Bericht gibt die tatsächlich durchgeführten Arbeiten wieder. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Vertrauliche Informationen sind nicht enthalten.

---

Datum

---

Unterschrift Studierender

---

Unterschrift Betreuer

## Inhaltsverzeichnis

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Einleitung</b>  | <b>1</b> |
| 1.1      | Motivation und Hintergrund . . . . .                           | 1        |
| 1.2      | Problemstellung . . . . .                                      | 1        |
| 1.3      | Aufbau . . . . .   | 1        |
| <b>2</b> | <b>Grundlagen</b>  | <b>1</b> |
| 2.1      | Python API mit Flask . . . . .                                 | 1        |
| 2.2      | Angular Frontend . . . . .                                     | 1        |
| 2.3      | Redis API Cache . . . . .                                      | 1        |
| 2.4      | MySQL Datenbank für Services und Logs . . . . .                | 1        |
| 2.5      | Kommunikation mit RabbitMQ . . . . .                           | 1        |
| 2.6      | KI-Service . . . . .   | 1        |
| 2.7      | Logs visualisieren in Grafana . . . . .                        | 1        |
| 2.8      | Deployment mit Docker . . . . .                                | 1        |
| <b>3</b> | <b>Methodik</b>  | <b>1</b> |
| 3.1      | Design Science Research . . . . .                              | 2        |
| 3.2      | Evaluationsmethode . . . . .                                   | 2        |
| <b>4</b> | <b>Projektergebnisse</b>                                       | <b>2</b> |
| 4.1      | Software Architektur . . . . .                                 | 2        |
| 4.2      | REST-API mit Flask . . . . .                                   | 2        |
| 4.2.1    | Aufbau und Implementierung der REST-API . . . . .              | 2        |
| 4.2.2    | Nutzeridentifizierung mit JWT . . . . .                        | 2        |
| 4.2.3    | Caching mit Redis Datenbank . . . . .                          | 2        |
| 4.2.4    | Fehlerbehandlung . . . . .                                     | 2        |
| 4.2.5    | Event Logging . . . . .  | 2        |
| 4.3      | Visualisierung der Logs in Grafana . . . . .                   | 2        |
| 4.4      | Webseite mit Angular . . . . .                                 | 2        |
| 4.4.1    | Aufbau des User Interfaces . . . . .                           | 2        |
| 4.4.2    | Funktionen der Komponenten . . . . .                           | 3        |
| 4.4.3    | Kommunikation zur API . . . . .                                | 3        |
| 4.5      | Kommunikation zwischen API und Services mit RabbitMQ . . . . . | 3        |
| 4.5.1    | RabbitMQ vs. REST-API . . . . .                                | 3        |
| 4.6      | Implementierung des KI-Services . . . . .                      | 3        |
| 4.6.1    | Dynamische Registrierung neuer Services . . . . .              | 3        |
| 4.6.2    | Interpretation der Eingabe mit BERT . . . . .                  | 3        |
| 4.6.3    | Cosinusähnlichkeitssuche in Elastic Search . . . . .           | 3        |
| <b>5</b> | <b>Evaluation</b>  | <b>3</b> |
| 5.1      | Performanceanalyse . . . . .                                   | 3        |
| 5.2      | Skalierbarkeit . . . . .                                       | 3        |
| 5.3      | Ergebnisse des Code-Reviews . . . . .                          | 3        |
| <b>6</b> | <b>Fazit</b>   | <b>4</b> |
| 6.1      | Fazit . . . . .  | 4        |
| 6.2      | Einschränkungen . . . . .                                      | 4        |
| 6.3      | Ausblick . . . . .   | 4        |
| <b>7</b> | <b>Literaturverzeichnis</b>                                    | <b>4</b> |

## **1 Einleitung**

text

### **1.1 Motivation und Hintergrund**

text

### **1.2 Problemstellung**

text

### **1.3 Aufbau**

text

## **2 Grundlagen**

### **2.1 Python API mit Flask**

text

### **2.2 Angular Frontend**

text

### **2.3 Redis API Cache**

text

### **2.4 MySQL Datenbank für Services und Logs**

text

### **2.5 Kommunikation mit RabbitMQ**

text

### **2.6 KI-Service**

text

### **2.7 Logs visualisieren in Grafana**

text

### **2.8 Deployment mit Docker**

text

## **3 Methodik**

text

### **3.1 Design Science Research**

text

### **3.2 Evaluationsmethode**

text

## **4 Projektergebnisse**

text

### **4.1 Software Architektur**

text

### **4.2 REST-API mit Flask**

text

#### **4.2.1 Aufbau und Implementierung der REST-API**

text

#### **4.2.2 Nutzeridentifizierung mit JWT**

text

#### **4.2.3 Caching mit Redis Datenbank**

text

#### **4.2.4 Fehlerbehandlung**

text

#### **4.2.5 Event Logging**

text

### **4.3 Visualisierung der Logs in Grafana**

text

### **4.4 Webseite mit Angular**

text

#### **4.4.1 Aufbau des User Interfaces**

text

#### **4.4.2 Funktionen der Komponenten**

text

#### **4.4.3 Kommunikation zur API**

text

### **4.5 Kommunikation zwischen API und Services mit RabbitMQ**

text

#### **4.5.1 RabbitMQ vs. REST-API**

text

### **4.6 Implementierung des KI-Services**

text

#### **4.6.1 Dynamische Registrierung neuer Services**

text

#### **4.6.2 Interpretation der Eingabe mit BERT**

text

#### **4.6.3 Cosinusähnlichkeitssuche in Elastic Search**

text

## **5 Evaluation**

text

### **5.1 Performanceanalyse**

text

### **5.2 Skalierbarkeit**

text

### **5.3 Ergebnisse des Code-Reviews**

text

## **6 Fazit**

### **6.1 Fazit**

### **6.2 Einschränkungen**

### **6.3 Ausblick**

## **7 Literaturverzeichnis**

ADMINISTRATION, U.S. General Services, [o. D.]. *System Usability Scale (SUS)* [online]. [besucht am 2022-07-25]. Abger. unter: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>.

SNYDER, B., 2000. *Music and memory: An introduction*. (MIT press).

STATISTA, [o. D.]. *Volumen der jährlich generierten digitalen Datenmenge* [online]. [besucht am 2022-07-25]. Abger. unter: <https://de.statista.com/statistik/daten/studie/267974/umfrage/prognose-zum-weltweit-generierten-datenvolumen/>.