**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: | 3. 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

### Abkürzungsverzeichnis

**AJAX** Asynchronous JavaScript and XML
**API** Application Programming Interface
**HTML** Hypertext Markup Language
**KI** Künstliche Intelligenz
**UI** User Interface
**BL** Business Logic

# 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

Python ist eine um 1991 von Guido van Rossum entwickelte Programmiersprache. Bei der Entwicklung von Python wurde ein besonderer Fokus auf die Lesbarkeit von Code gesetzt. Dank der simplifizierten Syntax im Vergleich zu anderen höheren Programmiersprachen wie Java oder C#, ist Python auch in Bereichen, wie in der Mathematik oder der Wissenschaft ein häufig genutztes Werkzeug. Python bietet ebenfalls die Möglichkeit, von anderen Entwicklern bereitgestellte Bibliotheken in das eigene Projekt zu integrieren.[1]

Flask ist eine der verfügbaren Bibliotheken, die ein Framework für die Implementierung eines Webbasierten Application Programming Interface (API) bereitstellt. Eine API dient dazu, Funktionen und Routen zu definieren, um die Kommunikation zwischen dem Frontend und dem Backend herzustellen. Das Flask Framework ist im Gegensatz zu anderen Frameworks sehr klein. Dies ermöglicht ein schnelles aufsetzen und entwickeln. Da Flask nur die nötigsten Grundlagen für eine API mitliefert, ist der Code besser lesbar und damit für andere Entwickler besser wartbar.[2]

Die Flask API wird für die Anbindung des Frontends an die Datenbank, sowie die Anbindung an die Kommunikationsschnittstelle von RabbitMQ verwendet. Sie nimmt die Daten oder die Eingaben des Nutzers entgegen und vermittelt sie an den richtigen Dienst, damit sie von einer KI-Schnittstelle ausgewertet werden können. Anschließend kann die API angefragt werden, ob es bereits Antworten von einer Künstliche Intelligenz (KI) zu der vorher geschickten Anfrage gab. Falls die API die Auswertung der KI erhalten hat, wird diese ans Frontend geschickt, um sie dort anzeigen zu können.

## 2.2 Angular Frontend

Eine grundlegende Website wird klassisch mit Hypertext Markup Language (HTML) und JavaScript erstellt. Um eine moderne Website zu entwickeln, die ihren Inhalt nicht beim ersten Aufrufen lädt, sondern erst dann, wenn er benötigt wird, müssen Konzepte wie Asynchronous JavaScript and XML (AJAX) verwendet werden. Angular ist ein von Google gebaut und gepflegtes Open Source Framework, welches das Entwickeln von komplexen webbasierten Anwendungen vereinfachen soll. Angular bietet im Gegensatz zu anderen Webframeworks wie React und Vue.js eine vollumfängliche Bibliothek mit der nahezu alle Aspekte in der Web Entwicklung abgedeckt werden können.[3]

In Angular wird die Programmiersprache TypeScript verwendet. Diese ist eine Erweiterung der Programmiersprache JavaScript und implementiert Konzepte wie feste Typisierung von Variablen. Weitere Konzepte wie Dependency Injection oder die Trennung von Business Logic (BL) und User Interface (UI) ermöglichen eine schnelle Entwicklung von komplexen Systemen.

Das Frontend wird für die Ein- und Ausgabe der Daten verwendet. Der Nutzer kann auf der Webseite seine Suchanfrage in ein Textfeld schreiben und dieses hochladen. An-

[1] Josheph, 2021.
[2] Grinberg, 2018.
[3] Moiseev u. a., 2018.

schließend wird dort die Möglichkeit bereitgestellt, die eingegeben Daten automatisiert zu manipulieren. Im gleichen Zug wird die Eingabe des Nutzers in ein für die KI verständliches Format konvertiert. Im letzten Schritt kann der Nutzer die Anfrage an das Backend schicken, dass mit der Analyse der Eingabe begonnen werden soll. Das Frontend fängt daraufhin an beim Backend in regelmäßigen Abständen nach Antworten der KI zu fragen. Wenn Antworten vorhanden sind, könne diese in einer Liste Visualisiert werden.

### 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

GRINBERG, M., 2018. *Flask web development: developing web applications with python*. O'Reilly Media, Inc.

JOSHEPH, T., 2021. Python. *Python Releases for Windows*. Jg. 24.

MOISEEV, A.; FAIN, Y., 2018. *Angular Development with TypeScript*. Simon und Schuster.