WebMatrix: MillionDollarProject

Frankfurt University of Applied Sciences
Fachbereich 2: Informatik und Ingenieurwissenschaften
Web-basierte Anwendungssysteme
Prof. Dr. Armin Lehmann

Kerem Celal Babacan (1454176)
Sana Meersaed (1452068)
Gülay Solak (955223)

Projektbeginn: 06.12.2024
Projektende: 14.02.2025

Sprecher: Kerem

Gliederung

1. Einleitung
2. Systemarchitektur
3. Technische Umsetzung
    a. User-Service
    b. Advertisement-Service
    c. Post-Service
    d. API-Gateway
4. GitHub & Zusammenarbeit
5. Demonstration
6. Fazit

Sprecher: Sana

Einleitung

- **Projektziel**
  Entwicklung eines Anwendungssystems aus Distributed Web-based Services mit API-Gateway

- **Microservices-Ansatz**
  Trennung von User-, Advertisement- und Post-Verwaltung

- **Flexibilität & Skalierbarkeit**
  Unabhängige Services ermöglichen einfache Erweiterbarkeit

Einleitung

- **<u>Was soll das System tun? Welche Aufgaben soll das System anbieten?</u>**

- Anwendung als Idee eines Markplatzes

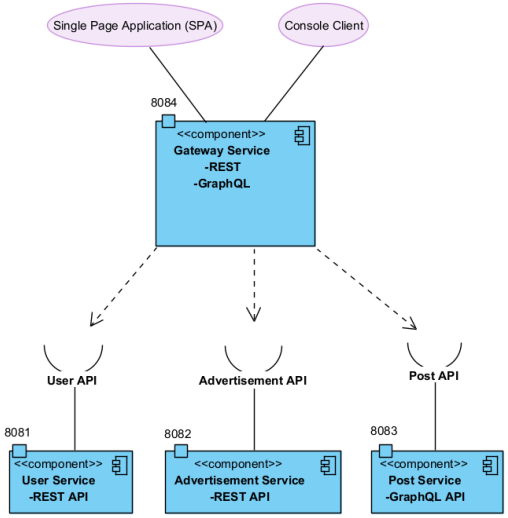
- Benutzerverwaltung (User Service)
- Anzeigenverwaltung (Advertisement Service)
- Postverwaltung (Post Service)

Sprecher: Sana

Systemarchitektur

- **Microservices-Struktur**
  Drei unabhängige Services für User, Anzeigen & Posts
- **API-Gateway als zentrale Schnittstelle**
  Vermittelt und steuert die Service-Kommunikation
- **Kommunikationsprotokolle**
  REST für User & Advertisement, GraphQL für Posts
- **Service-Ports & Skalierbarkeit**
  Jeder Service läuft auf eigenem Port, unabhängig skalierbar

Komponentendiagramm



Sprecher: Sana

User-Service

- **Benutzerverwaltung**: Erstellung, Bearbeitung, Abruf und Löschung von Nutzern
- **REST-API**: Schnittstelle für den Zugriff auf Benutzerdaten
- **Zentrale Funktion im System**: Anzeigen & Posts sind an Benutzer gebunden
- **Validierung & Fehlerbehandlung**: Benutzer kann nur mit gültigen Daten erstellt werden

HTTP-Methode	URI	HTTP-Statuscodes	Consumes	Produces
GET	/users	200 OK, 204 No Content	–	JSON
GET	/users/{id}	200 OK, 404 Not Found	–	JSON
POST	/users	201 Created, 422 Unprocessable Entity	JSON	JSON
PUT	/users/{id}	200 OK, 404 Not Found	JSON	JSON
DELETE	/users/{id}	200 OK, 404 Not Found	–	JSON

Sprecherin: Gülay

Advertisement-Service

- **Anzeigenverwaltung**: Erstellung, Bearbeitung, Abruf und Löschung von Anzeigen
- **REST-API**: Schnittstelle für den Zugriff auf Anzeigen-Daten
- **Verknüpfung mit User-Service**: Anzeigen gehören immer zu einem existierenden Benutzer
- **Validierung & Fehlerbehandlung**: Anzeige kann nur mit gültigen Daten erstellt werden

HTTP-Methode	URI	HTTP-Statuscodes	Consumes	Produces
GET	/users/{userId}/advertisements	200 OK, 404 Not Found	–	JSON
GET	/users/{userId}/advertisements/{id}	200 OK, 404 Not Found	–	JSON
POST	/users/{userId}/advertisements	201 Created, 422 Unprocessable Entity	JSON	JSON
PUT	/users/{userId}/advertisements/{id}	200 OK, 404 Not Found	JSON	JSON
DELETE	/users/{userId}/advertisements/{id}	200 OK, 404 Not Found	–	JSON

Sprecher: Sana

Post-Service

- **Verlinkung von externen Inhalten**: Nutzer können Plattform-Links zu Anzeigen hinzufügen
- **GraphQL-API**: Flexible Abfrage und Erstellung von Posts
- **Verknüpfung mit User- und Advertisement-Service**: Jeder Post gehört zu einer Anzeige und einem Benutzer
- **Validierung & Fehlerbehandlung**: Post kann nur mit gültigen Daten erstellt werden

GraphQL-Operation	Anfrage / Mutation	Beschreibung
Query	getAllPostsByUserId(userId, adId)	Gibt alle Kommentare zu einer Anzeige zurück.
Mutation	createPost(userId, adId, platform, link)	Erstellt einen neuen Kommentar für eine Anzeige.

Sprecher: Kerem

API-Gateway

- **Zentrale Schnittstelle**
  Leitet Client-Anfragen an die passenden Microservices weiter
- **Routing & Weiterleitung**
  Entscheidet anhand der URL, welcher Service angesprochen wird
- **Einheitliche Fehlerbehandlung**
  Gibt konsistente Fehlercodes zurück (z. B. 404, 500)
- **Unterstützung für REST & GraphQL**
  REST für User & Advertisement, GraphQL für Post-Service

Sprecher: Kerem

GitHub & Zusammenarbeit

- Einarbeitung Github mit Verknüpfung zu Visual Studio Code
- Großer Lernerfolg
- Schnelle Einarbeitung und Verständnis
- Zusammenarbeit sehr positiv und vorteilhaft
    - Issues erstellen und kommentieren
    - Abschließen in To-Do-Liste
    - Zusammen am Code arbeiten
    - Commit-Messages

Sprecherin: Gülay

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Demonstration

Fazit

- Erfolgreicher Abschluss des Projektes
- Meilensteine wurden im Laufe des Projektes zeitlich angepasst
- Die Programmstruktur der Anwendung ist einfach und unkompliziert
- Große Lernkurve