

# Middleware – Cloud Computing – Übung

## Web-Services

---

Wintersemester 2024/25

Harald Böhm, Laura Lawniczak, Tobias Distler

Friedrich-Alexander-Universität Erlangen-Nürnberg  
Lehrstuhl Informatik 4 (Systemsoftware)

<https://sys.cs.fau.de>



**Lehrstuhl für Informatik 4**  
Systemsoftware



**Friedrich-Alexander-Universität**  
Technische Fakultät

Web-Services

RESTful Web-Services

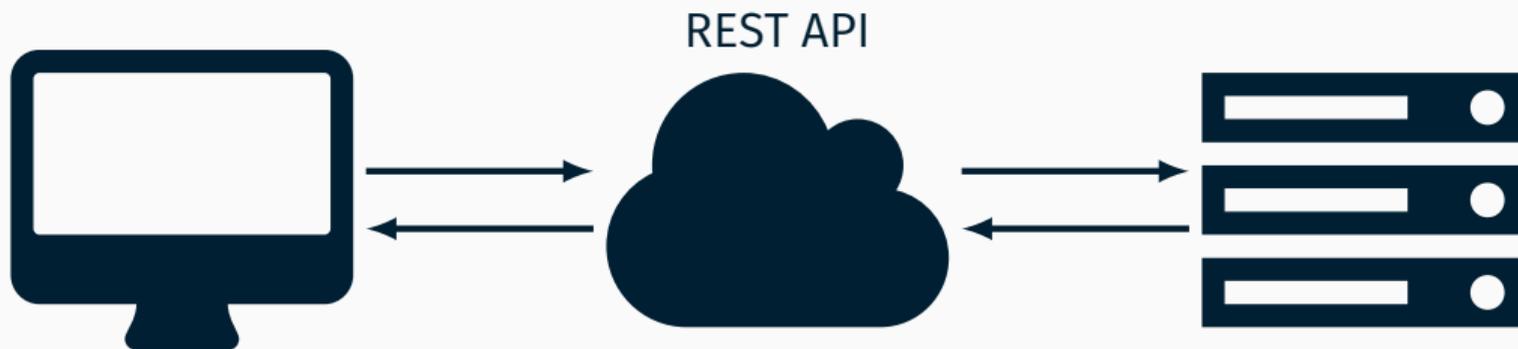
Aufgabe 1

## **Web-Services**

---

### **RESTful Web-Services**

- Web-Service
  - Software-System zur Interaktion zwischen Rechnern über ein Netzwerk
  - Plattformunabhängigkeit durch Einsatz von Web-Standards (z. B. HTTP)
  
- In der Übung: RESTful Web-Service



- Übertragung von Daten in HTTP-Anfragen und -Antworten
  - Header für Metadaten
  - Body für Nutzdaten (optional)
  
- Ausführung von Aktionen mittels HTTP-Operationen
  - GET        Lesezugriff auf eine Ressource
  - PUT        Schreibzugriff auf eine Ressource
  - DELETE    Löschen einer Ressource
  - POST      Übermittlung von Daten an eine Ressource
  - ...
  
- Senden von Statusmeldungen durch HTTP-Status-Codes
  - 200 OK                      Erfolgreiche Bearbeitung
  - 400 Bad Request           Fehlerhafte Anfragenachricht
  - 404 Not Found              Ressource existiert nicht
  - ...

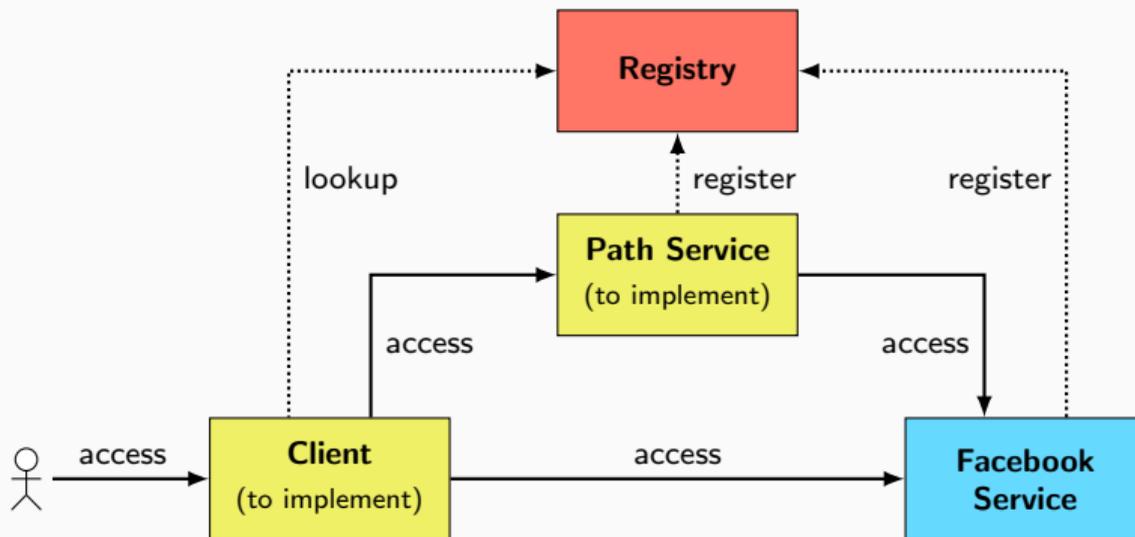
# Web-Services

---

## Aufgabe 1

## Aufgabe 1: Web-Services

- Bereitstellung eines eigenen RESTful Web-Services
- Teilaufgaben
  - Kommandozeilen-Client für Registry-Zugriff
  - Web-Service zur Erweiterung eines bereits bestehenden Web-Services
  - Client zum Zugriff auf beide Web-Services



## ■ Überblick

- Verwaltung von Nutzern und ihrer Freundschaftsbeziehungen zu anderen
- Zusammenhängender Graph aus Freundschaftsbeziehungen
- Bereitstellung auf einem Lehrstuhlrechner

## ■ Verwaltete Informationen für jeden Nutzer

- *ID*            Eindeutige Kennzeichnung des Nutzers
- *Name*        (Klar-)Name des Nutzers
- *Freunde*     Liste mit den IDs von Freunden des Nutzers

## ■ Basisschnittstelle

- GET /facebook/search?string=<s>        Suche nach Nutzern, deren Name die Zeichenkette <s> enthält (maximal 1000 Ergebnisse)
- GET /facebook/names/{id}                Ausgabe des Klarnamens zu einer ID
- GET /facebook/friends/{id}              Ausgabe aller Freunde einer ID

### ■ Gebündelte Abfrage von Klarnamen

- Methode `POST`
- Pfad `/facebook/names`
- Anfrage-Body `String-Array ids` der abzufragenden IDs
- Antwort-Body `String-Array names` der Klarnamen, wobei `names[i]` den Klarnamen von `ids[i]` repräsentiert

### ■ Gebündelte Abfrage von Freundschaftsbeziehungen

- Methode `POST`
- Pfad `/facebook/friends`
- Anfrage-Body `String-Array ids` der abzufragenden IDs
- Antwort-Body `Map<String, HashSet<String>> friends` der angeforderten Freundschaftsbeziehungen
  - Schlüssel in `friends` entsprechen den abgefragten IDs
  - Werte in `friends` enthalten die jeweiligen Freund-IDs

## ■ Überblick

- Im Rahmen von Aufgabe 1 selbst zu implementieren
- Ermittlung der kürzesten Verbindung zwischen zwei Facebook-Nutzern

## ■ Ausgabe der kürzesten Verbindung zwischen {startID} und {endID}

- Methode GET
- Pfad /path/{startID}/{endID}
- Antwort-Body MWPath-Objekt mit Pfad path und Aufrufstatistiken

```
public class MWPath {  
    String[] path;  
    [...] // Statistiken  
}
```

## ■ Implementierung

- Rückgriff auf den Facebook-Dienst
- Bestimmung des kürzesten Pfads
  - Bereitgestellt: Implementierung des Dijkstra-Algorithmus
  - Zu implementieren: Zusammenstellung der Eingabemenge von IDs

# Client für Web-Services

- Zugriff auf Pfad- und Facebook-Dienst per Kommandozeilen-Client
- Zu implementierende Kommandos
  - search Suche nach Nutzern
  - friends Ausgabe der Namen aller Freunde eines Nutzers
  - path Kürzester Pfad (Nutzernamen) zwischen zwei Nutzern

