

2014

# Parklotse Online

## Grobkonzept

Es soll eine Webanwendung entstehen, die die Anzahl freier Parkplätze auf teilnehmenden Parkplätzen anzeigt. Außerdem sollen die Parkplätze auf einer Karte angezeigt werden und Routen empfohlen werden. Zudem kann man Parkgebühren online abrufen. Man kann einen Parkplatz beobachten/ abonnieren. Wird dieser Parkplatz nun frei, so erhält man vom System eine Nachricht. Die Applikation soll mithilfe von Sensoren in den Parkplätzen ermitteln können, ob bestimmte Parkplätze noch zur Verfügung stehen. Die Zielgruppe sind die Autofahren die in eine bestimmte Stadt freie Parkplätze suchen. Hierbei soll es den Autofahren möglich sein schneller einen freien Parkplatz zu suchen/bekommen.

Web-basierte Anwendungen 2: Verteilte Systeme

Sommersemester 2014

Gruppe 2 (Julian & Robert) | Team 22

Arne Beckmann, Daniel Röger und Patrick Rüschenberg

09.04.2014



## Inhaltsverzeichnis

Systemarchitektur.....	2
Kommunikationsparadigmen.....	3
Ressourcen.....	3
Schemata.....	4
Topics .....	4
Datenstrukturen.....	5
JSON Datenstruktur.....	5
JSON Beispiel .....	5
Kommunikationsmodell.....	6

## Tabellenverzeichnis

Tabelle 1: Ressourcen.....	3
Tabelle 2: Schemata .....	4
Tabelle 3: Topics .....	4

## Abbildungsverzeichnis

Abbildung 1: Kommunikationsmodell .....	6
---	---

## Systemarchitektur

## Kommunikationsparadigmen

### Ressourcen

Um das Parkplatzsystem umzusetzen werden folgende **Ressourcen** benötigt:

Ressourcen	HTTP- Methoden
GPS Standort der Parkplätze	GET, POST
Parkplätze	GET, PUT, POST
Parkpreise	GET, PUT, POST
Parkzeiten	GET, PUT, POST
Platzstatus	GET, PUT
User	
Sensor	GET, PUT

Tabelle 1: Ressourcen

Sie werden von unserem Web- Server eingelesen und dem Client des Users von dort aus zur Verfügung gestellt.

Platzstatus und Sensor können hierbei abonniert und damit zusätzlich asynchron abgerufen werden.

Alle Ressourcen stehen dem User synchron zur Verfügung.

Um dies zu realisieren wird folgendes Schemata verwendet:

## Schemata

Schemata
http

Tabelle 2: Schemata

## Topics

Topics
Land
Bundesland
<b>Stadt</b>
<b>Stadtteil</b>
<b>Einzelner Parkplatz</b>

Tabelle 3: Topics

Nach diesem Schema richtet sich die Aufteilung der Topics. Hier kann der User entscheiden, in welchem Umfang er einen Parkplatz suchen möchte.

Er kann hier in der Größenordnung: Stadt, Stadtteil oder Einzelner Parkplatz suchen.

Land und Bundesland werden voraussehend hinzugefügt, falls sich das System irgendwann erweitert.

Bei diesem Projekt benötigt man einen Topic- Server zwischen dem Web- Server und dem Client. Wenn der User einen Parkplatz abonniert hat, published der Server eine Nachricht, sobald sich der Status des abonnierten Parkplatzes ändert. Diese Nachricht wird als Pushbenachrichtigung an den Client gesendet, solange er auf der Suche nach einem Parkplatz ist.

## Datenstrukturen

### JSON Datenstruktur

Unsere JSON Datenstruktur besteht aus einem Array in dem die einzelnen Parkplätze als JSON Objekt gespeichert werden. In diesen Objekten wird die Adresse sowie die weiteren Informationen zu den Parkplätzen wie z.B. die Maximale Parkdauer gespeichert. Hier wird ebenfalls der Status, ob ein Parkplatz frei oder belegt ist hinterlegt.

### JSON Beispiel

```
var Parkplatz = [  
  # erster Parkplatz  
  {  
    "Parkplatz_ID": 0,  
    "Land": "Deutschland",  
    "Bundesland": "NRW",  
    "Stadt": "Gummersbach",  
    "Bezirk": "Zentrum",  
    "Straße": "Steinmüllerallee",  
    "Nr.": 0,  
    "Gebuehren_in_Euro_pro_Stunde": 0.5,  
    "Maximal_Parkdauer_in_Stunden": 2,  
    "Status_Frei": true,  
    "Freuenparkplatz": false,  
    "Behindertenparkplatz": false,  
    "Eltern-Kind_Parkplatz": false  
  },  
  # zweiter Parkplatz  
  {  
    ...  
  }  
]
```

## Kommunikationsmodell

Unser Kommunikationsmodell zeigt die synchrone und asynchrone Kommunikation zwischen dem Parkplatz suchendem (Autofahrer) und unserem Parkplatz System.

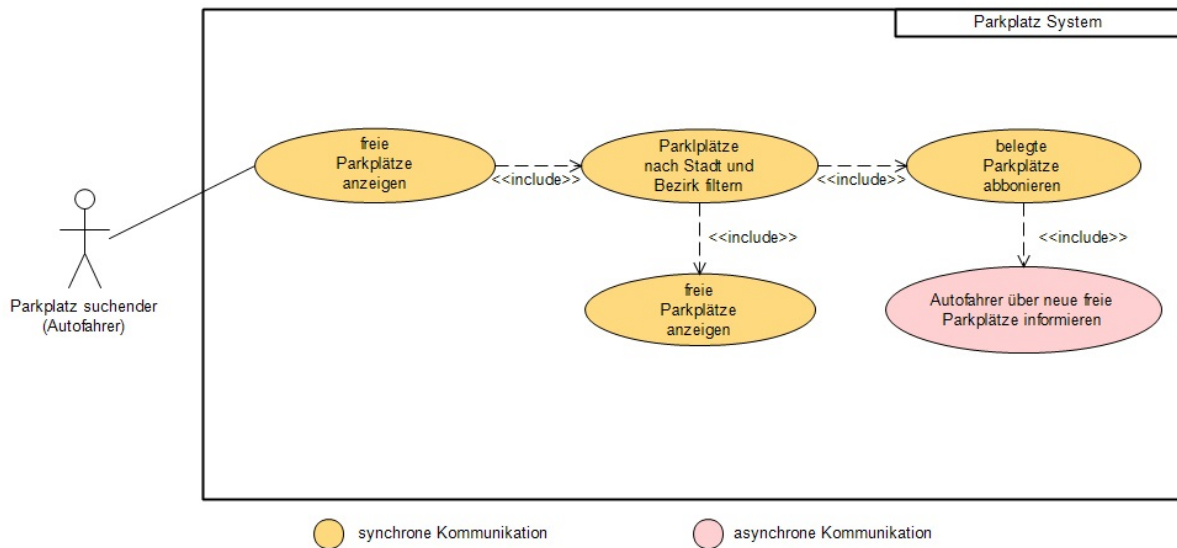


Abbildung 1: Kommunikationsmodell