

## **Verteilte Systeme – Anforderungsanalyse**

1. Sensor, zentraler Steuerrechner, Übermittlung der Daten von Sensor an Steuerrechner per UDP, Messung des Sensors, Ausgabe der verarbeiteten Daten
2. HTTP-Server und Schnittstelle implementieren, Befehle verarbeiten + Daten empfangen und abspeichern
3. Server implementieren, Daten der Sensoren schicken per MQTT, Testausfälle durchführen, Daten speichern
4. per RPC Schnittstelle Daten übermitteln, Client kann auf Server zugreifen und Daten auslesen, Messwerte werden zwischen allen Herstellerservern Synchronisiert

### **Aufgabe 1:**

Es gibt mehrere simulierte Sensoren pro Fahrzeug, welche zufällige Werte per UDP asynchron an den Zentralserver übermitteln. Dort werden diese in einem Standardcontainer gespeichert und auf der Konsole ausgegeben. Jedes Fahrzeug und jeder Sensor im Fahrzeug hat eine Eindeutige ID.

Programmiersprache: c++

### **Aufgabe 2:**

Der Zentralserver verfügt über eine HTTP-Schnittstelle, über welche per Browser alle Daten eines Fahrzeugs, alle Daten eines Sensors eines Fahrzeugs oder alle Daten aller Fahrzeuge abgerufen werden. Dies geschieht indem die HTTP GET anfrage verarbeitet wird.

Programmiersprache: c++

### **Aufgabe 3:**

Die Messwerte der Sensoren in den Fahrzeugen aus Aufgabe 1 werden per MQTT an beliebig viele Herstellerserver verteilt. Alle Herstellerserver speichern alle empfangenen Messwerte ab. Da jeder Hersteller mindestens 3 Server hat, simulieren wir Ausfälle einzelner Server. Falls dies geschieht empfangen die weiterhin funktionstüchtigen Server trotzdem die Nachrichten der Fahrzeuge damit keine Messdaten verloren gehen. Bei Verlust von Messdaten bevor sie erfolgreich per MQTT veröffentlicht wurden hilft dies jedoch nicht.

Programmiersprache: Python

### **Aufgabe 4:**

Die gespeicherten Daten der Herstellerserver von Aufgabe 3 werden mittels XML-RPC an die Versicherungen übermittelt. Ein Herstellerserver überprüft seine Daten bevor er sie zurückgibt auf Vollständigkeit. Dazu wird per MQTT eine Anfrage an alle anderen Herstellerserver geschickt, welche dann alle gespeicherten Messwerte zu angefragtem Sensor erneut veröffentlichen. Damit kann sichergestellt werden, dass selbst nach dem Ausfall eines Herstellerservers dieser alle Messwerte an die Versicherung liefern kann.

Programmiersprache: Python