

# **Verteilte Systeme**

**WS2021/22  
MOORE & GIMBEL**

## **Protokoll**

von

Daniel Chenani  
Leonie Zeiler

# Aufgabe 1a - UDP Sockets

## Implementierung

- Simuliert wird ein Ein- und Ausliefern von Produkten und Inventarisierung mit Hilfe der Sensoren. Dabei soll jeder der Sensoren ein eigenständiger Prozess sein. Es sollen mindestens 80% der Pakete erfolgreich ankommen.
- Die Informationen der Sensoren sollen sich kontinuierlich ändern und mit Hilfe von UDP permanent an das Lager geschickt werden. Das Lager gibt dann die empfangenen Pakete mit folgenden Informationen aus: IP, Sensor-Typ, Port. Diese Infos werden dann in einem Log-File gespeichert.
- Da es sich um den ersten Entwurf handelt und ein Proof of Concept ist, werden bestimmte Daten und deren Vollständigkeit außer Acht gelassen.

## Tests

### Test 1 - Warenhaus empfängt Daten von mehreren Sensoren

- Es sollen 10 Datensätze über ein UDP-Socket von einem Sensor an ein Warehouse gesendet. Danach wird die LOG-Datei des Sensors mit der LOG-Datei des Warehouse verglichen.
- Es sollen per UDP-Socket mindestens 80% der Pakete erfolgreich ankommen. Wenn ein Paket angekommen ist, soll die Datei vollständig sein.

#### Erwartung

- Unsere Daten werden empfangen und die LOG-Dateien stimmen überein.

#### Ergebnis

- Unsere Daten wurden vollständig empfangen und die LOG-Dateien stimmen überein.

### Test 2 - Mehrere Warenhäuser empfangen Daten von mehreren Sensoren

- Es sollen 10 Datensätze über UDP-Sockets von zwei Sensoren an ein Warenhaus und von zwei weiteren Sensoren an ein weiteres Warenhaus gesendet werden. Wenn ein Paket angekommen ist, soll die Datei vollständig sein.

#### Erwartung

- Unsere Daten werden empfangen und die LOG-Dateien stimmen überein.

#### Ergebnis

- Unsere Daten wurden vollständig empfangen und die LOG-Dateien stimmen überein.

# Aufgabe 1b - TCP Sockets

## Test 1 - HTTP-Server und GET-Requests

- Der HTTP-Server wird von warehouse1 im Browser aufgerufen. Dadurch wird ein GET-Request an den Server gesendet. Dieser antwortet dann mit einer Anzeige als HTML-Seite.

### **Erwartung**

- HTTP-Server sollte den GET-Request korrekt verarbeiten. Die HTML-Seite sollte korrekt dargestellt werden.

### **Ergebnis**

- HTTP-Server hat den GET-Request korrekt verarbeitet. HTML-Seite wurde korrekt dargestellt.

## Test 2 - HTTP-Server gibt eine Liste aller gesendeten Daten zurück

- Die Sensoren senden jeweils 10 Datensätze an das warehouse. Die aufgelisteten Daten werden abgeglichen.

### **Erwartung**

- Die Vollständigkeit sollte gegeben sein und die Daten sollten mit den geloggten Daten übereinstimmen.

### **Ergebnis**

- Die Vollständigkeit ist gegeben und die Daten stimmen mit den geloggten Daten überein.