



Prof. Dr.-Ing. Manuel Duque-Antón

Robin Tretter, B. Sc.
<robin.tretter@hs-kl.de> (Raum O228, Tel. -5357)

Praktische Aufgaben: Verteilte Systeme

Aufgabe 3 - Entwicklung einer verteilten Anwendung

In dieser Aufgabe soll eine verteilte Anwendung mithilfe von MQTT entwickelt werden. Die konkret zu lösende Anwendung wird nicht vorgegeben, sondern muss von dem Studierenden selber formuliert werden. Dabei sollten allerdings folgende Randbedingungen erfüllt sein. Mindestens eine Ausgabe bzw. ein Aktor und mehrere Eingaben bzw. Sensoren sollten in der Anwendung vorgesehen und implementiert werden. Die Anwendung soll aus mehreren Komponenten bestehen, welche auf mehreren Geräten ausgeführt werden sollen. Hierfür werden Raspberry Pi, ein Switch und die benötigten Netzkabel zur Verfügung gestellt. Für die verteilte Anwendung soll eine Architekturskizze erstellt werden, die den Kommunikationsablauf zwischen den einzelnen Komponenten darstellt. Zur Realisierung der verteilten Anwendung dürfen auch fertige Komponenten eingesetzt werden, z.B. Datenbanken (SQLite [1], InfluxDB [2]), MQTT Broker (Eclipse Mosquitto [3]...), usw. Im Vordergrund dieser Aufgabe soll der verteilte Charakter der entwickelten Anwendung stehen, also die Kommunikation und Zusammenarbeit der einzelnen Komponenten untereinander.

Zur Verfügung gestellte Hardware:

- 4x Raspberry Pi 3B+ inkl. Netzteil, Gehäuse und SD-Karte, je nach Gruppengröße oder Aufgabe
- 1x 5 Port Ethernet Switch
- 4x Netzkabel 2m

Zu Beachten:

- Es soll eine Architekturskizze, die den Kommunikationsablauf der verteilten Anwendung darstellt angefertigt werden.
- Zur Realisierung der verteilten Anwendung dürfen auch fertige Komponenten eingesetzt werden, z.B. Datenbanken (SQLite, Influx-DB), MQTT Broker (Eclipse Mosquitto...)
- Im Vordergrund soll der verteilte Charakter, also die Kommunikation zwischen den einzelnen Komponenten, der Anwendung stehen.
- Es soll mindestens eine Ausgabe bzw. Aktor und mindestens zwei Eingaben bzw. Sensoren geben.
- Als Programmiersprache kann zwischen Python und Java gewählt werden.

Testen und Überprüfen der Implementierungen

Zur Überprüfung und Testierung stellen die Teilnehmer*innen ihre verteilte Anwendung in einer kurzen Präsentation dar. Die Präsentation soll eine Beschreibung der Architektur (Architekturskizze) und der eingesetzten Komponenten inklusive ihrer Rollen (z.B. Pi A -> MQTT Broker), sowie eine Live-Demo beinhalten.

Beispiel Smart Thermometer

Als Beispiel für eine mögliche Anwendung soll ein Smart Thermometer dienen. Dieses kann aus mehreren Raspberry Pi's bestehen (vgl. Abbildung 1).

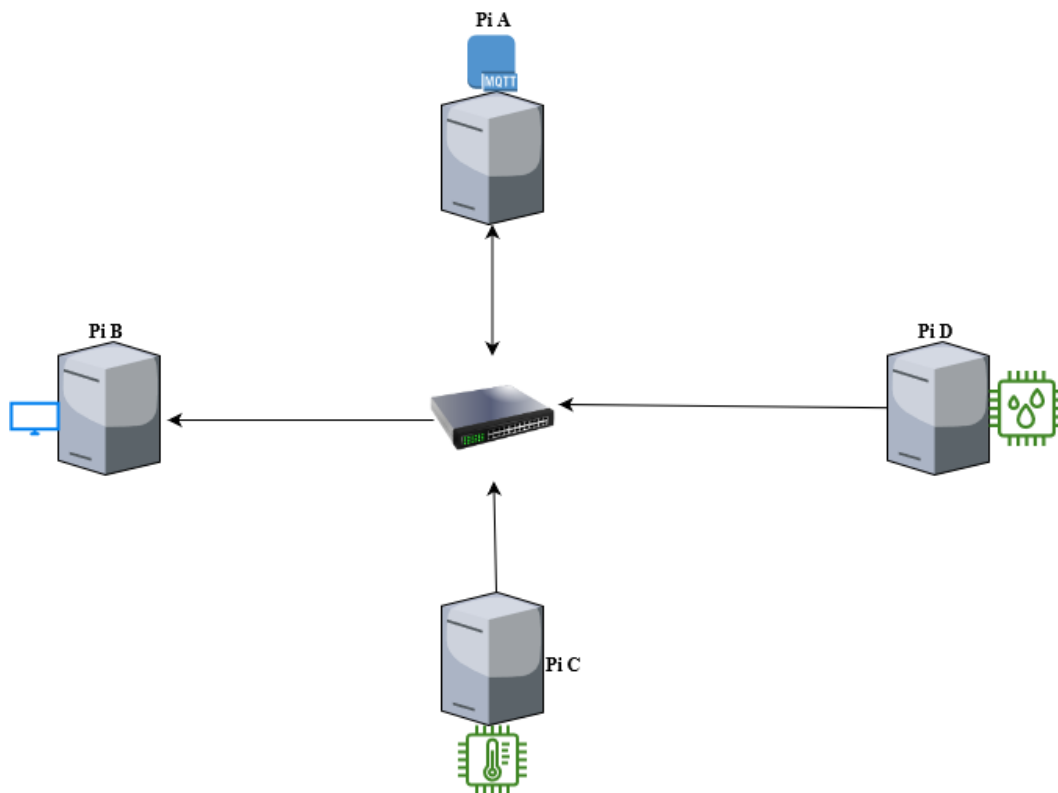


Abbildung 1: Beispielarchitektur eines Smart Thermometers

Der erste Pi (Pi A) stellt den MQTT Broker dar. Dieser dient als zentraler Knotenpunkt für alle Daten. Konkret werden alle Daten an die Smart Ausgabe (Pi B) weitergeleitet, damit diese ausgegeben werden können. Die beiden Pi's C und D stellen die Sensoren dar. Hierbei übernimmt Pi C die Rolle eines Thermometers, und Pi D die Rolle eines Hygrometers. Diese Pi's senden ihre Daten entweder zu einem festen Intervall oder bei jeder Änderung um ein bestimmtes Delta an den Broker.

Literatur

- [1] SQLite Consortium: *SQLite*. <https://www.sqlite.org/index.html>. Version: 03 2022
- [2] InfluxData: *InfluxDB*. <https://www.influxdata.com>. Version: 03 2022
- [3] Eclipse: *Mosquitto*. <https://mosquitto.org>. Version: 03 2022