Valentina Merturi

Stefano Kals

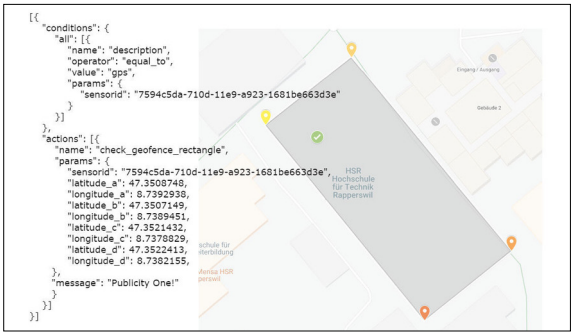
| | |
|---|---|
| Diplomanden | Valentina Merturi, Stefano Kals |
| Examinator | Prof. Beat Stettler |
| Experte | Michael Schneider, CloudGuard Software AG, Pfäffikon SZ, SZ |
| Themengebiet | System-Software |
| Projektpartner | Basile Bluntschli, WLAN-Partner.com AG, Zürich ZH, ZH |

# Fog Computing

## Rule-Engine

Postauto Shuttle Bus
Internetseite vom Postauto

netModule NB2800 Router
WLAN Partner




Geofence-Beispiel mit den entsprechenden Koordinaten und der dazugehörigen Regel
Eigene Darstellung

Aufgabenstellung: Die Firma WLAN-Partner aus Zürich betreibt mobile Router für die Postauto Flotte. Diese wurden in der Vergangenheit nur für den WLAN Internetzugang der Passagiere eingesetzt, könnten aber in Zukunft auch für die Übertragung von Echtzeit-Informationen genutzt werden. Dabei senden diverse Sensoren wie GPS, Odometer oder der Motor (wie etwa über CAN Bus) ununterbrochen und willkürlich Daten an den Router. Die Aufgabe beinhaltet es, relevante Daten abzufangen und auf dem Fahrzeug auszuwerten. Dazu sollte es möglich sein, eigene Regeln zu definieren. Fährt das Postauto beispielsweise in einen bestimmten Bereich, könnte dabei eine Aktion - in Form einer Werbung auf dem Handy, eines Alarms oder einer Ansage - ausgelöst werden. Dank dem Einblick in die Echtzeitdaten des Fahrzeuges und dem Erstellen eigener Regeln entstehen dabei unzählige Anwendungsmöglichkeiten.

Vorgehen / Technologien: Ziel dieser Arbeit ist es, dass die Sensordaten auf dem Fahrzeug jederzeit mitverfolgt und mit Hilfe spezifischer Regeln ausgewertet werden. Diese Regeln sollen dynamisch konfiguriert und angepasst werden können. Es soll für den Benutzer aber auch möglich sein zu interagieren, wenn unter Umständen mit dem Postauto etwas nicht in Ordnung ist (beispielsweise wenn der Motor überhitzt).

Für die Realisierung des intelligenten Gateways stellte WLAN Partner einen netModule NB2800 Router zur Verfügung. Die Use Cases wurden zum Teil selbst und in Zusammenarbeit mit Postauto erstellt, um unsere Arbeit darauf aufzubauen. Abgesehen davon, dass es sich um eine Linux-basierte Anwendung handelt und welche auf ARM lauffähig sein muss, überliess einem der Kunde für die Implementierung die komplette Freiheit der einzusetzenden Technologien.

Es wurde für jeden Entwickler eine Entwicklungsumgebung mittels Linux VM, Docker, Docker-Compose sowie den benötigten Abhängigkeiten aufgesetzt. Die ganze Implementierung wurde in Python3 umgesetzt. Zudem wird Mosquitto als MQTT Broker und Redis als Datenbank verwendet. Das Zusammenspiel von Mosquitto und Redis war für die Umsetzung der Regeln von grosser Bedeutung. Ausserdem werden die neuen yoyowallet Business-Rules eingesetzt. Damit war es sogar möglich, mehrere Sensor-Variablen miteinander zu vergleichen und in einer Bedingung zu prüfen.

Fazit: Es war eine Herausforderung sich in das vielschichtige und breitgefächerte Thema einzulesen. Nach dem dieser Sprung jedoch geschafft war, konnten viele Use Cases abgedeckt und die Regeln, zu den vom Kunden geforderten Anforderungen, erstellt und implementiert werden. Die Arbeit war in Bezug auf den Überbegriff Fog-Computing zudem äusserst lehrreich.

Der intelligente Gateway ist nun in einem Zustand, den man testen und weiterentwickeln kann. Die wichtigsten Funktionen konnten auch schon mittels Sensor-Simulationen getestet werden. Nun ist der nächste Schritt, echte Sensoren anzuschliessen und die gewünschten, noch nicht umgesetzten Use Cases zu realisieren.