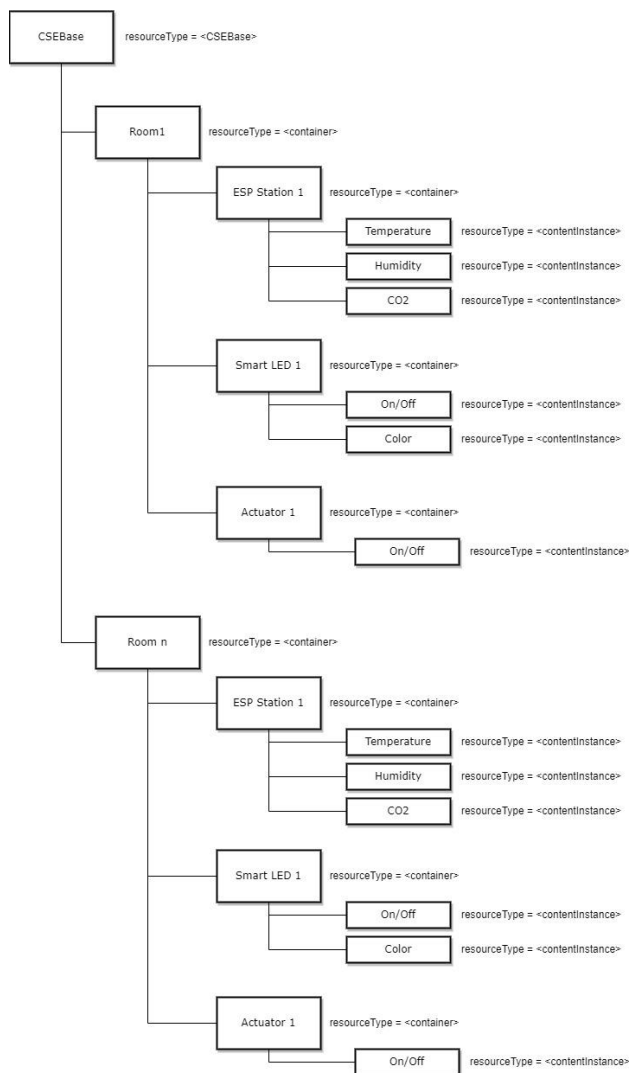


## Aktueller Stand Projekt MIO:AID

Wir entwickeln ein System zur Überwachung der Luftqualität mit OneM2M, ZigBee Sensoren und ESP Sensoren. Die ACME stellt das zentrale Element dar das über MQTT die Messdaten der Sensoren bekommt. Wir wollen nicht nur die Luftqualität messen, sondern auch abhängig davon ein Fenster mittels einem ZigBee Fenstermotor öffnen können und zudem die Luftqualität mit einer Smart LED farblich darstellen (grün, gelb, orange, rot).

Der Ablauf der Daten ist wie folgt:

Die ZigBee Sensoren und Aktoren schicken und bekommen ihre Daten von ZigBee2MQTT, dass über einen MQTT Broker an Node-RED angebunden ist. In Node-RED werden wir die Steuerungslogik implementieren und die Daten der Sensoren in für die CSE verständliche MQTT Requests konvertieren. Die Sensordaten des ESP (Temperatur, Luftfeuchtigkeit) werden vom ESP direkt mit CSE kompatiblen MQTT Messages and die CSE geschickt. Der ESP erstellt, wenn nötig, AE-Ressourcen und Container selbst. Die Daten werden in folgender Struktur gespeichert (siehe Abbildung)



Damit unsere Lösung möglichst portabel und flexibel ist, verwenden wir Docker-Compose für die CSE, MQTT, Portainer, Node-RED und ZigBee2MQTT. Der verwendete ZigBee USB Stick ist von Sonoff.

Aktuell sind folgende Arbeitspakete abgeschlossen:

- 1) Docker container für jede Software erstellt und konfiguriert. Die einzelnen Applikationen können untereinander kommunizieren
- 2) Daten von den ZigBee Sensoren in Node-RED empfangen.
- 3) Mittels Onem2m kompatiblen MQTT Requests Container in der CSE erstellen und per WebUI beobachten
- 4) Mittels eines Python Files und REST Requests AE und Container anlegen
- 5) Daten des Luftqualitätssensor an den ESP senden.
- 6) Dummydaten vom ESP über MQTT an den MQTT Broker Container schicken
- 7) MQTT Messages für das Erstellen von Container und Flexcontainer vorbereitet.
- 8) Bauteile für einen Demonstrator mit ZigBee Glühbirne, ESP und Luftqualität Sensor ausgesucht.
- 9) Erste Überlegungen für eine Sicherheitskonzept (Zertifikate, TLS, Credentials, Access Control Policies)

Wir hatten am Anfang Probleme bei der Erstellung der Flexcontainer. Mittlerweile wissen wir, dass wir die ModuleClasses und DeviceClasses über ein fcp-File importieren müssen.

Das haben wir auch schon erfolgreich gemacht und können jetzt Flexcontainer erstellen.

Was wäre aus Ihrer Sicht ein sauberer Lösungsansatz mit Flexcontainern (bzw. SDT)?

Wir dachten, dass wir drei neue DeviceClasses erstellen:

1. DeviceSmartLed
2. DeviceAirQualitySensor
3. DeviceActuator

Zudem würden wir mehrere neue ModuleClasses erstellen:

- linkQuality für die Zigbee-Signalqualität
- colorTemp für die Farbtemperatur
- colorSled mit den Datapoints hue + saturation + colorX + colorY, airquality. Datenpunkte, die für die Ansteuerung unserer SmartLED wichtig sind.
- Usw.

Wir senden Ihnen auch unser fcp-File. Sind wir bei der Namensgebung für "type" an irgendwelche Vorgaben gebunden, oder können wir das für unser Projekt frei wählen? (z.B. mio:colT für colorTemperature).

Unseren weiteren geplanten Arbeitspakete und Schritte sind:

1. Access Control Policies (z.B. damit die Devices eines Raums, nur CRUD auf ihre eigenen Ressourcen haben)
2. Subscription / Notification
3. Zertifikate und Benutzerkonzept für MQTT
4. Fertigstellung des Prototypen
5. Visualisierung mit NodeRed Dashboards
6. Dokumentation