Nodi che tengono traccia dell’energia generata dal generatore, il loro livello di carburante e la loro temperatura. Dati campione sono mandati dai monitor usando MQTT e CoAP ad un collector che li salva in un db. Il collector gestisce anche le anomalie.

Ogni monitor 🡪 un IoT device nRF52840 USB Dongle

I monitor hanno tre sensori 🡪 temperatura, livello carburante, energia generata (+ sistema di allarme che accende i led)

MQTT broker sviluppato su ubuntu sulla VM

Collector (java) 🡪 riceve dati dai dongles, salva nel db, gestisce anomalie

**MQTT NODE**

Nodi pubblicano i dati in base ai toopics: energy\_generated, fuel\_level and temperature

Mentre i monitor pubblicano su alarm\_ID

* topic\_name 🡪 {"n": "fuel\_level", "v": "746", "u": "L", "id": "2"}
* alarm\_NODE\_ID 🡪 per ricevere allarmi dal collector. Il collector manda NO\_ERROR, FUEL\_LEVEL\_ERROR and TEMPERATURE\_ERROR.

**CoAP node**

Si registra al collector usando una GET al server. Espone un set di risorse: energy\_generated, fuel\_level and temperature

* /topic\_name 🡪 {"n": "temperature", "v": "58", "u": "C", "id": "3"}
* /alert 🡪 per ricevere messaggi di allarme dal collector. Accetta POST/PUT request contenendo il tipo di anomalia (ex: value=2 temperature, value=0 disattivare, value=1 basso livello di carburante)

**Collector**

Interagisce sia con MQTT che CoAP, salva i dati in MySQL e determina anomalie

* /resoruce\_name 🡪 ricevere dati dai monitor
* /alarm\_NODE\_ID 🡪 notificare i nodi delle anomalie

Il CoAP collector: viene istanziata una coap collector class per ogni possibile sensore. Ogni volta che il collector riceve un campione, controlla il valore e nel caso invia una POST request all’ /alarm