

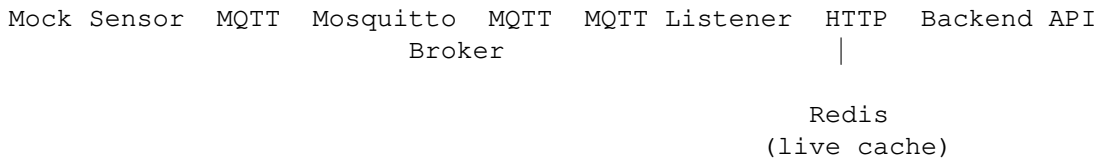
Primus MQTT Listener

Mikroserwis nasłuchujący dane z czujników IoT dla systemu **Primus Inter Pares 2026**. Odpowiada za przetwarzanie odczytów temperatury i wagi w czasie rzeczywistym oraz generowanie alertów.

Pełna dokumentacja projektu: [Primus Docs](#)

Architektura

Asynchroniczny serwis Python działający jako pomost między warstwą IoT a backendem.



Realizacja Wymagań Specyfikacji

5c. Monitoring Wagi (Nieautoryzowane Pobranie)

- Implementacja: [listener.py](#)
- Logika:
 1. Pobiera ostatnią prawidłową wagę z Redis (`Weight:{designation}:{row}:{col}`).
 2. Porównuje z aktualnym odczytem z sensora.
 3. Jeśli różnica $> \text{TOLERANCE}$ (0.5 kg) i nie ma aktywnej transakcji (`Expected-Change:...`), generuje alert `WEIGHT`.

5a/5b. Monitoring Temperatury

- Logika:
 1. Pobiera definicje regałów z backendu (zakresy `temp_min`, `temp_max`).
 2. Dla każdego odczytu sprawdza, czy temperatura mieści się w zakresie.
 3. Przy przekroczeniu generuje alert `TEMP`.

Funkcjonalności

Funkcja	Opis
MQTT Subscription	Nasłuchuje na topiku <code>sensors/+</code> (wszystkie regały)
Redis Cache	Zapisuje aktualne odczyty (<code>rack:{id}:live</code>) z TTL 60s
Alert Throttling	Nie wysyła tego samego alertu częściej niż raz na 60s
Auto-Login	Automatyczna autoryzacja do backendu z odświeżaniem tokena
Weight Sync	Przy starcie synchronizuje <code>expected weights</code> z backendu

Stos Technologiczny

Kategoria	Technologia
Język	Python 3.11
MQTT	Paho MQTT Client
Cache	Redis
HTTP	Requests
TLS	Obsługa certyfikatów dla MQTT i API

Uruchomienie

Docker (Zalecane)

Serwis uruchamiany automatycznie jako część stosu [primus-infra](#):

Lokalnie

Wymagane zmienne środowiskowe: