Introduktion och syfte

Målet är att sätta sig in i hur vi kopplar ihop flera delar av ett lite mer komplext API-system bestående av:

a)      En embedded-enhet som skickar värden via en MQTT-broker

b)     En API-server som lyssnar på embeddedenhetens MQTT-data och producerar ett API

c)      En API-gateway som låter klienter konsumera API-serverns API

d)     En klient som konsumerar APIt

[En bild som visar text, skärmbild, Teckensnitt, diagram

Automatiskt genererad beskrivning](https://storage.omniway.se/quilluploads/3f7fc6f9f3e046446e85cede1f95b7c9.png)

Uppgift

Uppgiften går ut på att knyta ihop flera delar i en längre kedja av oberoende system.

Ni ska som grupp designa och implementera ett system som producerar och konsumerar data enligt mönster ovan.

Beståndsdelar

-         Klient: En enkel klient som regelbundet begär information från ett API. Kan vara t.ex. en konsoll-applikation eller något mer grafiskt.

-         API-gatway: En applikation som skickar information mellan Klient och API-server, med lite features som caching, rate limiting, etc.

-         API-server: En Flask-applikation som läser av MQTT-data och publicerar detta via ett API

-         MQTT-server som kommunicerar MQTT-data mellan publisher och subscriber

-         Embedded-enhet som läser av mätdata från en sensor och skickar den till en MQTT-broker

Av dessa skall ni implementera dessa själva:

-         Klient: Ni ska skriva en applikation som läser ett API och visar datan

-         API-server: Ni ska skriva en Flask-applikation som läser MQTT-meddelanden och gör dem tillgängliga via ett API

-         Embedded-enhet med sensor: Ett enkelt program som skickar MQTT-meddelanden med sensordata. Det går bra att använda färdiga exempel från Embedded-kursen, eller exempel från Internet (T.ex. från Wokwi)

o  Ni kan välja att även ha flera enheter som producerar flera olika mätdata

System som används utan att skriva själva:

-         API-gateway: Använd en färdig produkt. Produkten behöver installeras och konfigureras av er.

-         MQTT-gateway: Använd förslagsvis en som ni har bekantskap med sedan innan, t.ex HiveMQ.

Bedömning

I denna inlämning kan man få betygen G eller VG. Betyget grundas på följande kriterier:

·        Självständigt arbete

·        Struktur och funktion på lösningen

·        Svårigheten i de problem ni har valt att lösa

·        Demonstration av att ni förstår bra arbetssätt och strukturerad kommunikation.

·        Kvalite på presentation

Jag tar hänsyn till alla punkterna ovanför när jag gör bedömning av betyg.

Redovisning

Källkoden ska vara pushad till ett eget repository på GitHub. Uppgiften ska genomföras i grupper om ca 3 personer och fullständiga namn ska finnas med i README.md i repots main-branch (master).

Inlämningen sker genom att en länk till gitrepot skickas in av en av personerna i Omniway.

(Den andra personen behöver också skicka in, men det kan vara tomt.)

Uppgiften skall även redovisas för läraren, antingen privat eller i storgrupp. Här skall ni kort redogöra för vad er lösning gör, samt en kort demo.