

Internet stvari i servisa

Servisno-orijentisane arhitekture

Projekat 2020

Projektovati i implemenitrati mikroservisnu arhitekturu IoT sistema, koja se sastoji od sledećih mikroservisa:

1. **Device Microservice** – upravlja jednim ili više Senzora ili Aktuatora. Ovi uređaji su virtuelni i predstavljeni komponentama/objektima. U okviru Device Microservisa se čitaju podaci iz datoteke sa postojećim senzorskim podacima i periodično **šalju Data Microservice** mikroservisu **REST zahtevom** (PUT, POST) i **Publish na MQTT broker** na topic na koji je Data Microservice subscribe-ovan.
Device Microservice obezbeđuje API:
 - GET - Za dobijanje trenutnih parametara (meta podataka) očitavanja: tip senzora koji se očitava, vremenski period očitavanja i slanja, ili threshold kada se mora poslati nova vrednost (npr. 10% promene prethodne vrednosti)
 - PUT,POST - za set-ovanje i modifikaciju parametara očitavanja ili komande za virtuelne Aktuatore.
2. **Data Microservice** – dobija podatke sa Device Microservisa i upisuje ih u NoSQL bazu podataka. Obezbeđuje API:
 - GET – za Obezbeđuje pretraživanje podataka po različitim kriterijumima
 - PUT/POST API za REST upis podataka sa Device Servisa
 - Subscribe na MQTT broker za dobijanje podataka sa Device Servisa
3. **Analytics Microservice** – dobija (REST-POST, Subscribe) podatke sa Data Mikroservisa i analizira ih u cilju detektovanja događaja/stanja značajnih za domen aplikacije i podatke o detektovanom događaju (tip, vrednosti, lokacija, vreme, ...) skladište u sopstvenoj NoSQL bazi podataka. Za detektovanje događaja možete koristiti jednostavna if-then pravila (npr. if puls > 80 and T>38 then potencijalni COVID bolesnik), ili eventualno postojeće biblioteke za mašinsko učenje. U skladu sa tipom događaja, poslati notifikaciju na Web dashboard i zahtev za akcijom REST POST zahtevom **Command Microservicu** koji će preko Device Microservisa pokrenuti akciju na Aktuatoru.
4. **Command Microservice** – šalje komande virtuelnom Aktuatoru slanjem POST zahteva (ili Publish na Message Broker) **Device Microservice**. Obezbeđuje API:
 - GET za dobijanje liste komandi i njenih parametara
 - PUT/POST definisanje parametara novih komandi ili promena postojećih.
5. **Gateway servise** - REST API (GET, POST, PUT,...) za pristup mikroservisima u sistemu
6. **Web dashboard** – pristup sistemu preko API Gateway-a za prikaz senzorskih podataka, notifikacija/upozorenja na bitne događaje, postavljanje parametara očitavanja, postavljanje parametara komandi,...

Generalni zahtevi:

- Svaki servis treba da bude u okviru Docker container-a sa sopstvenom bazom podataka (ako je ima).
- Mikroservise implementirati korišćenjem jedne ili obe tehnologije, ASP.NET Core i NodeJS (Moleculer), a pored ovih možete koristiti i Java, Go ili Python mikroservise.
- Za asinhronu komunikaciju između mikroservisa po principu Publish/Subscribe koristiti NATS i MQTT Message broker-a, a pored njih možete i upotrebiti neki od sledećih: RabbitMQ, Vert.X, Kafka.
- Podatke koje će da generišu virtuelni senzori možete naći na sledećim linkovima (možete i samo potražiti javno dostupne IoT podatke na Internetu)
 - <https://www.datasciencecentral.com/profiles/blogs/great-sensor-datasets-to-prepare-your-next-career-move-in-iot-int>
 - <https://data.world/datasets/sensors>
 - <http://archive.ics.uci.edu/ml/datasets.php>
 - **Podatke sa kojima prijavite kao odgovor na objavu projekta na CS sajtu predmeta, jer dva tima ne mogu da rade sa istim podacima!**
- Izvorni kod projekta postaviti na GitHub i pripremiti dokument sa kratkim opisom implementiranih servisa, REST API-a (endpoint-a) i pub/sub topic-a preko kojih komuniciraju, kao i CLI komande za startovanje i testiranje servisa (curl, Postman)

Rok za izradu i postavljanje projekta na GitHub je dan ispita u junskom ispitnom roku.