

Dokumentacija SOA projekta

Za naš SOA projekat odlučili smo se da napravimo Chicago weather monitoring platform koja nadgleda i kontroliše meteorološku situaciju na tri gradske plaže (Oak, Foster, 63rd). Sistem za prikaz koristi Web dashboard koji dobija trenutne meteorološke podatke sa bilo koje od ovih plaža i isto tako obaveštenja, upozorenja i alarme kada analizator naše platforme ustanovi da se isti događaj desio. Sistem nudi mogućnost dodavanja komandi koje akuatori, softver koji kontroliše hardver vezan za neku od tri navedene plaže, (isti daje komandu hardveru da izvrši neku akciju: podizanje zastavice za dozvoljeno kupanje, zabranjeno kupanje, obaveštenje o povećanom UV zračenju, ...). Platforma je projektovana pomoću mikroservisne arhitekture koja se koristi za razvoj IoT aplikacija. Tehnologije koje su korišćene su sledeće:

- Moleculer.js za mikroservise
- NATS za message broker
- InfluxDB kao baza podataka
- Angular za Web Dashboard
- Docker

Mikroservisi

1.) Device Mikroservisi

Naša platforma je projektovana tako da svaka plaža poseduje po par senzor, akuator mikroservis npr. (fostersensor.service.js i fosteractuator.service.js). Ova senzor mikroservis ima zadatak da čita podatke sa svih hardverskih senzora koji se nalaze na toj plaži i te podatke prosleđuje data servisu, pošto ne radimo sa stvarnim senzorima realtime feed podatke simuliramo periodičnim citanjem iz fajla.

Aktuator mikroservis kontroliše skup hardvera jedne plaže (zastave, meteo alarmi, rampe za pristup plaži, sirena i slično). On ima zadatak da izvrši određenu komandu koju mu je prosledio command mikroservis, pošto ne radimo sa pravim hardverom izvršavanje komande simuliramo štampanjem navedene komande u konzoli mikroservisa.

2.) Data Mikroservis

Naša platforma poseduje jedinstveni „data“ mikroservis koji je nazvan `serializer.service.js`. On ima funkciju da primljene podatke sa sve tri plaže upisuje u svoju bazu podataka i dalje prosleđuje analyzer mikroservisu.

3.) Analyzer Mikroservis

Naša platforma poseduje jedinstveni analyzer mikroservis koji je nazvan `analyzer.service.js`. On ima funkciju da prima podatke od serializer mikroservisa i iste obrađuje pomoću heuristike koju smo stvorili za potrebe ove platforme. Ta heuristika ima zadatak da detektuje upozorenja, obaveštenja i alarme obrađujući primljene podatke. Ako se analizom ustanovi da je detektovan neki događaj asinhrono obaveštenje(notifikacija) se salje web servisu(Angular klijentu). Ova komunikacija ostvarena je pomoću message brokera. Ako ustanovi da je detektovan događaj, detektovani događaj i njegov timestamp se upisuju u analyzer bazu podataka. Takođe analyzer obaveštava command mikroservis o ustanovljenom događaju.

4.) Command Mikroservis

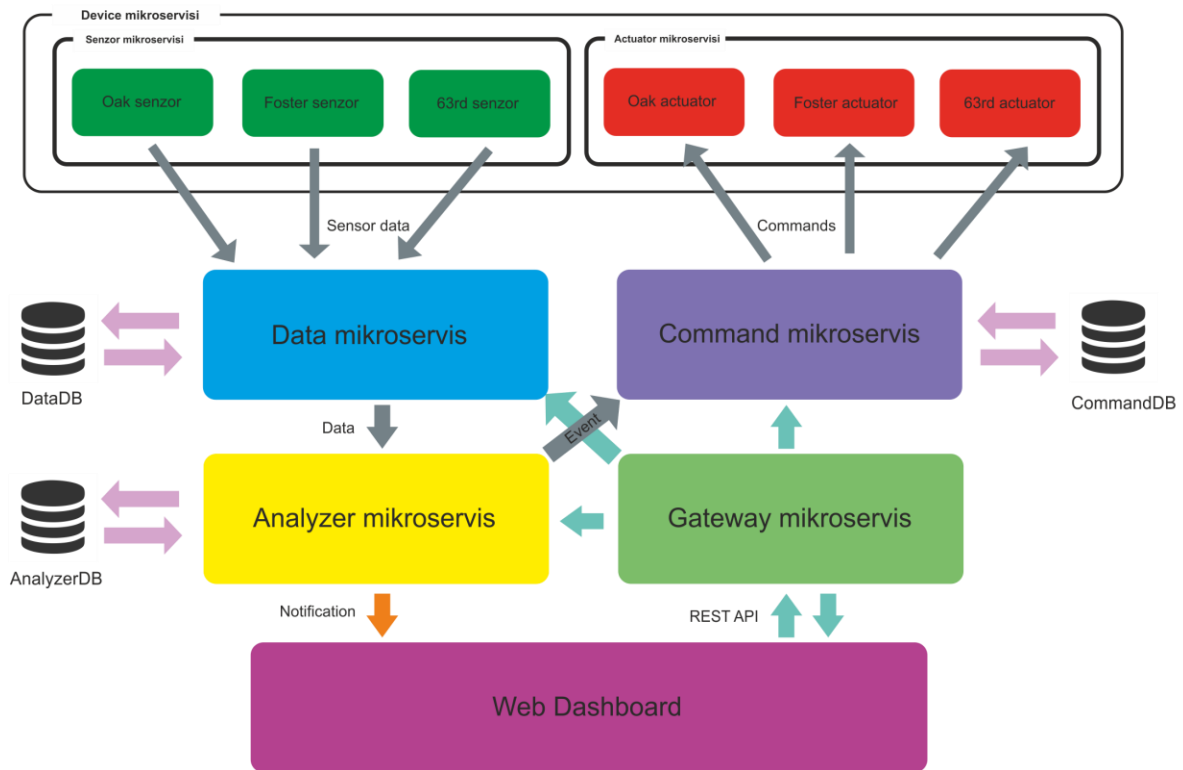
Naša platforma poseduje jedinstveni command mikroservis koji je nazvan `command.service.js`. Na osnovu događaja koje je analyzer ustanovio command mikroservis određuje vrstu komande i plažu na koju tu komandu treba poslati (kom aktuatoru). I odabranu komandu sa timestampom upisuje u sopstvenu bazu podataka. Command mikroservis ima mogućnost da na zahtev doda novu komandu u sistem

5.) Gateway Mikroservis

On služi kao „kapija“ naše platforme prema „spoljnom svetu“. Odnosno nudi REST API koje naš web klijent može da koristi. Ako želimo da uzmemo bilo koji podatak ili analizu možemo kontaktirati samo gateway api koji pomoću ugrađenog molecular transportera taj zahtev prenosi npr. Serializer mikroservisu, command mikroservisu,

6.) Web dashboard

Naša platforma poseduje web dashboard implementiran kao angular web klijent koji dobija notifikacije sa našeg sistema, koristi REST API pomoću kojeg može da čita određene podatke, dodaje novu komandu,



Message brokeri

Platforma koristi NATS message broker za komunikaciju između mikroservisa, takođe koristimo socket.io za salnje asinhronih notifikacija angular klijentu. Topici koji postoje u našem sistemu su:

1. Tri topic-a koje sensor servisi koriste za slanje izmerenih vrednosti serializer servisu.
 - foster.read,
 - oak.read
 - sixthree.read.
2. Tri topic-a koji služe da data servis salje podatke analyzer servisu, to su:
 - data.oak
 - data.foster
 - data.sixthree
3. Tri topic-a koji služe da analyzer servis posalje detektovan dogadjaj command servisu:
 - oak.command
 - foster.command
 - sixthree.command

4. Tri topic-a koji služe da command servis pošalje komandu actuator servisima:
 - actuator.oak
 - actuator.foster
 - actuator.sixthree
5. Topic za slanje asinhrono notifikacije klijentu od strane analyzer servis:
 - alert.client

Endpoint-i

Gateway mikroservis sadrži sve endpoint-e naše platforme a neki od njih su:

GET endpoint-i:

- <http://localhost:3001/allcommands>
- <http://localhost:3001/recentoak>
- <http://localhost:3001/recentfoster>
- <http://localhost:3001/recent-sixthree>
- <http://localhost:3001/all-oak>
- <http://localhost:3001/all-foster>
- <http://localhost:3001/all-sixthree>
- ...

POST endpoint-i:

- <http://localhost:3001/addnewcommand>
- <http://localhost:3001/addparam>
- ...

Pokretanje

Web dashboard se startuje pozicioniranjem CLI-a u folder `iot-clinet` projekta i kucanjem komande `npm install` i nakon toga `ng serve`. Web dashboard-u mozete pristupiti iz browser-a na adresi <http://localhost:4200>.

Mikroservisi koriste docker te se svi oni (uključujući NATS broker, InfluxDB i ostale zavisnosti) startuju pozicioniranjem CLI-a u folder `iot-services` i kucanjem komande `npm install` i nakon toga `docker-compose up --build`.