## 



## 

## 

**Poročilo o strukturi projekta in pričakovanih težavah**

pri predmetu Sistemska Programska Oprema

## 

## 

## 

## 

**Člani skupine:**

Nik Terglav

Luka Lamprečnik

Simona Zhirova

Marija Jovanova

# **Struktura projekta in vključitev MQTT Brokerja**

## Naš projekt temelji na sistemu za upravljanje naprav (klima, okna, radio in vzvratna ogledala) z ročnimi gestami, ki jih zazna kamera in jih interpretira s pomočjo umetne inteligence. Projekt je razdeljen na več neodvisnih komponent, ki komunicirajo prek lahkega sporočilnega protokola MQTT.

### **1. Arhitektura:**

## Producerji – komponente, ki zaznajo ročne geste (kamera + AI skripta). Ko zaznajo gesto, prek MQTT pošljejo sporočilo z ukazom (npr. "okna/odpri").

## Broker (Mosquitto) – osrednja točka sistema, ki sprejema sporočila od producerjev in jih razpošlje ustreznim porabnikom. Teče kot ločena Docker storitev.

## Consumerji – komponente, ki so naročene na določene teme (topics) in ob prejemu sporočila izvedejo akcijo (npr. zmanjšanje glasnosti radia ali odpiranje okna).

### **2. Vključitev MQTT brokerja:**

## Uporabljamo odprtokodni Mosquitto MQTT broker, ki teče kot samostojna Docker storitev.

## Producer in consumer komponente se nanj povežejo znotraj skupnega virtualnega omrežja (ZeroTier VPN).

## Vsaka komponenta komunicira z brokerjem prek IP naslova znotraj VPN-ja in standardnega porta “1883”.

### **3. Razlogi za izbiro MQTT:**

## Podpira lahko sporočanje in nizko porabo virov, kar je idealno za prototipne IoT rešitve.

## Omogoča preprosto razširljivost – z dodajanjem novih tem (topics) ali naročnikov brez sprememb v obstoječem sistemu.

# **Težave pri dograjevanju (širjenju) sistema**

Pri nadgradnji in širjenju trenutnega sistema, ki temelji na MQTT komunikaciji prek VPN (ZeroTier), lahko pričakujemo več možnih izzivov. Nekateri izmed njih so tehnične narave, drugi pa so povezani z organizacijo razvoja in dolgoročno vzdrževanjem sistema.

### **1. Usklajevanje naslovov (IP, topic)**

Pri dodajanju novih naprav ali komponent, ki komunicirajo prek MQTT, je treba poskrbeti za:

* pravilno nastavitev **ZeroTier IP naslovov**,
* sinhronizacijo **topic-ov**, da vsi moduli komunicirajo na ustreznih kanalih.

Napake v teh nastavitvah pogosto povzročijo tiho nedelovanje sistema, saj MQTT ne vrne očitnih napak pri napačnih topic-ih.

### **2. Varnost (trenutno ni prednostna)**

Ker projekt razvijamo v zaprtem omrežju (VPN prek ZeroTier) in sodelujejo le štirje člani ekipe, posebne varnostne rešitve (TLS, avtentikacija, ACL) trenutno niso vključene. V primeru, da bi se sistem v prihodnosti uporabljal v širšem okolju ali kot del produkta, bi bilo treba razmisliti o dodatnih zaščitah.

### **3. Spremljanje delovanja**

Ker v projektu sodelujemo štirje člani, trenutno ne pričakujemo potrebe po uvedbi dodatnih diagnostičnih orodij, kot sta Prometheus ali Grafana. V primeru večjega sistema ali uporabe v produkciji bi bila takšna orodja priporočljiva za nadzor nad delovanjem MQTT brokerja in posameznih komponent.

### **4. Zanesljivost in skalabilnost**

MQTT broker (Mosquitto) trenutno teče kot en sam servis – brez redundance. V primeru večjega števila uporabnikov ali napake na brokerju sistem neha delovati. Razmislek o:

* **obremenitvenem testiranju**,
* **load balancingu** ali
* **rezervnih instancah**

je pomemben za dolgoročno stabilnost.

### **5. Organizacija kode in sprememb v ekipi**

Ker več članov ekipe dela na istem repozitoriju, obstaja nevarnost konflikta v skupnih datotekah (kot je “docker-compose.yml”). Dogovor o pravilih sodelovanja in verzioniranja je ključen za širjenje sistema brez motenj.

