

EUProNet 4.0

INF_MINF

Guczi Ádám, Francia Csaba

Felkészítő tanár: Bicskei Róbert

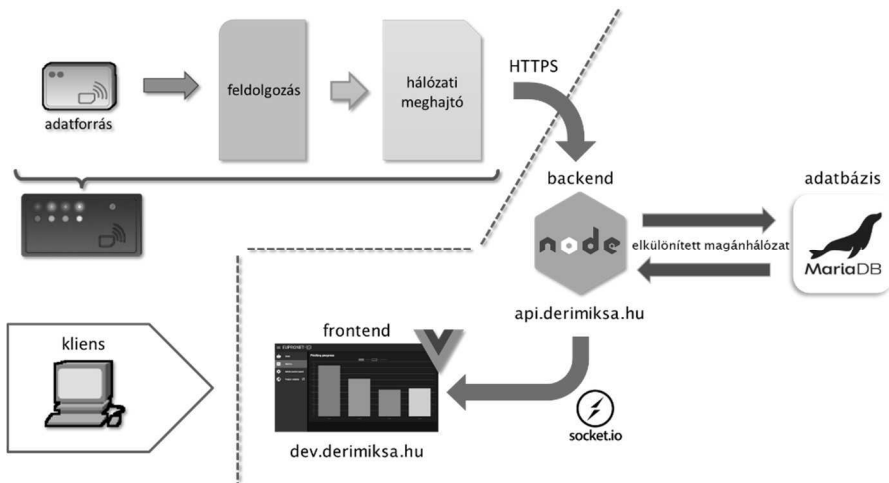
Szegedi SZC Déri Miksa Műszaki Technikum

1. Bevezetés

2019 szeptemberében indult nemzetközi projekt során egy Industry 4.0 technikákat felhasználó gyártósor fejlesztésében veszünk részt. Magyarország azt a feladatot kapta, hogy az egyes megrendelések és munkaállomások állapotát központi adatbázisban tároljuk úgy, hogy vizualizálhatók legyenek egy nyilvános weboldalon. Ezt az alapötletet stílusosan túltervezve egy valós idejű termelésfigyelő és nyilvántartó rendszer prototípusát alkottuk meg, ami bármilyen termelési ciklusra könnyedén rászabható.

2. Probléma megoldásának menete

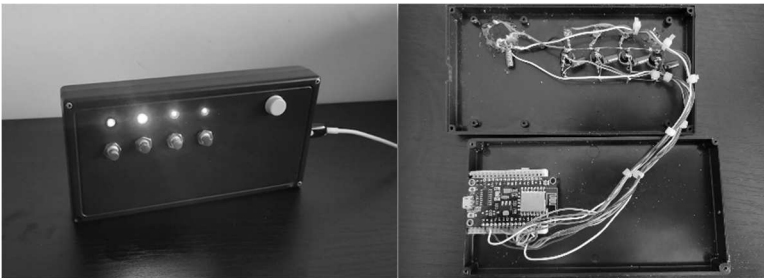
A járványhelyzetből kifolyólag enyhült a külföldiekkel való kommunikáció és jelentősen lassult a teljes projekt előrehaladása, így munkánk alul specifikált maradt. Tehát a rendszer ránk eső részének tervezésében és kivitelezésében szinte teljesen szabad kezet kaptunk. Ezáltal egy olyan rendszer fejlesztésébe kezdtünk, ami rugalmasan tud illeszkedni a többi ország munkaállomásához, ahogy azok életre kelnek.



1. ábra: Áttekintés

A munkaállomásokról lejövő adatokat egy mikrokontroller dolgozza fel és HTTPS kérésekben küldi a backend felé. Az adatbázishoz nincs közvetlen külső hozzáférés, a backend közvetítésével kerül bele adat. Amikor egy kliens a vizualizációs weboldalra csatlakozik, egy stabil websocket alapú, kétirányú kommunikáció jön létre a backenddel. A bejövő adatokból grafikonok készülnek, a változásokról a backend minden csatlakozott klienst valós időben értesít.

2.1. A doboz



2. ábra: A „munkaállomás” és belseje

A „doboz” egy ESP8266 irányította integrált vezérlőeszköz, a gyártósor egy munkaállomásának a megtestesítője. A fejlesztés során egy 3D nyomtatóról keletkezett adatfolyam közvetítőjeként képzeltük el. Egy-egy gombnyomással valamely színű korong nyomtatásának indításáról illetve sikeréről tudjuk a központi adatbázist értesíteni. A termelés állapotát így manuálisan közli, de könnyedén el lehet képzelni, hogy egy adott munkaállomás kimenetét dolgozza fel, teljesen automatizáltan.

2.2. Backend, API kommunikáció, frontend

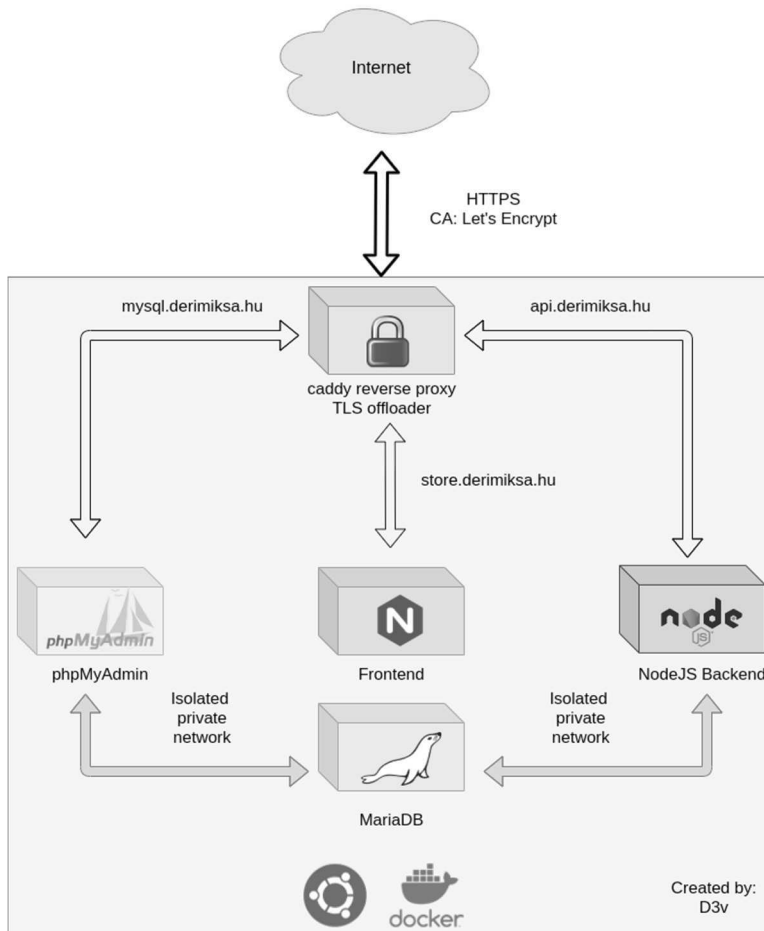
A backend egy Express.js alapú, RESTful API-val kommunikáló HTTP szerver. Kérések között állapotot nem tárol. A munkaállomásokkal API végpontokon keresztül kommunikál. Sikeres bejelentkezéskor hozzáférési tokenet küld, ami szükséges a további erőforrásokhoz való hozzáféréshez. A jelszavakat Argon2-vel titkosítja. Minden kérésről naplót, telemetriát készít, így az egyes felhasználók munkálkodása nyomon követhető.

A vizualizációs weboldal és egyben webshop jelenleg folyamatos fejlesztés alatt áll. Viszont él és virul, aktuális állapota a dev.derimiksa.hu címen megtekinthető. Vue keretrendszerrel készült, a Vuetify UI könyvtár felhasználásával. Amikor új adat érkezik a munkaállomások felől, azt a backend minden csatlakozott kliensnek továbbítja és a vizualizáció azonnal frissül.

2.3. Infrastruktúra

Ezek a szolgáltatások egy bérelt VPS szerveren Docker konténerekben futnak. A konténerek a belső hálózaton közvetlenül kommunikálnak egymással. A

bejövő kéréseket egy fordított proxy domainnév alapján irányítja a megfelelő szolgáltatáshoz. Kívül minden üzenet titkosított: ehhez a Caddy 3 havonta automatikusan igényel egy tanúsítványt, ingyenesen a Let's Encrypttől.



3. ábra: Az infrastruktúra

Az infrastruktúra felállításában és az üzemeltetésben való segítségéért külön köszönetet kell mondanunk Zsibók Márknak; nélküle nem tudtuk volna ezt a projektet talpára állítani.

3. Elért eredmények

A modulok közötti kommunikáció teljesen működőképes; a rendszer készen áll a további munkaállomások felcsatolására. A prototípus működése a

weboldalon nyilvánosan megtekinthető. Minden internetre kijutó kommunikáció titkosított, az ehhez szükséges tanúsítvány igénylése automatikusan történik. A konténerizált, moduláris felépítés gondmentes üzemeltetést, skálázhatóságot biztosít. A Gitlabon felállított CI/CD munkafolyamatok a feltöltött kódváltozásokat percekben belül automatizáltan integrálják.

A teljes forráskód elérhető a gitlab.com/Deri-Miksa/eupronet oldalon.

Felhasznált technológiák	
Virtualizációs platform	Docker
Reverse proxy, TLS offloader	Caddy
Frontend framework	Vue
Backend framework	Express
Verziókontroll	Gitlab
Mikrokontroller platform	Arduino (PlatformIO)
Adatbázis szolgáltatás	MariaDB
Fejlesztési környezet	Visual Studio Code
API kliens	Postman
Kommunikációs könyvtár	Socket.io
KDF	Argon2

1. táblázat: Használt technológiák