

PLATFORMA ZA TRGOVINU ENERGIJOM SA MIKROSERVISNOM ARHITEKTUROM I AI ANALIZOM TRŽIŠTA **ENERGY TRADING PLATFORM WITH A MICROSERVICES ARCHITECTURE AND AI MARKET ANALYSIS**

Aleksa Popov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PRIMENJENO SOFTVERSKO INŽENJERSTVO

Kratak sadržaj – U radu je prikazana realizacija aplikacije za automatizovano trgovanje električnom energijom, koja koristi treniran AI model za predikciju cena. Rešenje je implementirano upotrebom mikroservisne arhitekture, a celokupna aplikacija je bazirana na Docker kontejnerima.

Ključne reči: *Mikroservis, AI model, Docker, trgovina energijom*

Abstract – *The paper presents the implementation of an application for automated electricity trading, which uses a trained AI model for price prediction. The solution is implemented using a microservices architecture, and the entire application is based on Docker containers.*

Keywords: *Microservice, AI model, Docker, energy trading*

1. UVOD

Tržište električne energije se danas menja velikom brzinom. Ključni pokretač ovih promena je razvoj pametnih energetske mreže (Smart Grid) koje omogućavaju dvosmernu komunikaciju između proizvođača i potrošača. U skladu sa ovakvim sistemom, korisnici više nisu samo pasivni potrošači, već mogu i sami da proizvode i skladište energiju, na primer, pomoću solarnih panela i kućnih baterija. Ova nova dinamika dovodi do čestih i naglih promena u cenama energije, čime dolazi do sve veće potrebe za automatizovanim sistemima koji će nam olakšati praćenje promena.

Cilj ovog rada je osmišljen da zadovolji potrebe nastale novom dinamikom trgovine, kreiranjem softverske platforme koja automatizuje proces trgovine energijom. Platforma koristi veštačku inteligenciju da predvidi kretanje cena i na osnovu tih predikcija donosi odluke o kupovini ili prodaji, olakšavajući korisnicima da ostvare uštedu ili profit. Za osnovu sistema izabrana je mikroservisna arhitektura, koja omogućava da se kompleksan problem razloži na manje delove. Na taj način, stvoren je skalabilan temelj za buduća unapređenja i dodavanje novih funkcionalnosti.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Bošković, doc.

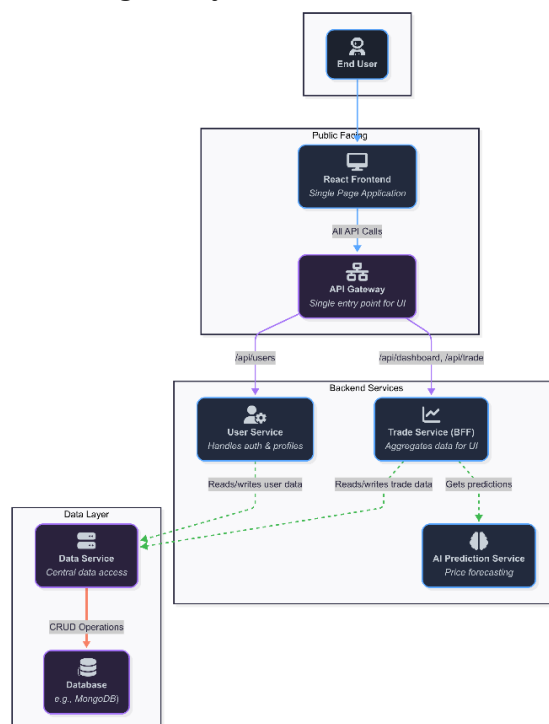
2. IZBOR ARHITEKTURE SISTEMA

Izbor prave arhitekture je ključan korak u izradi svakog softvera, koja će uticati na svaku dodatnu promenu na sistemu u budućnosti. Za ovaj projekat, postojale su dve glavne opcije pristupa: monolitnog i mikroservisnog.

Monolitna arhitektura podrazumeva da se sve komponente aplikacije nalaze unutar jedne, jedinstvene celine. Iako je ovakav pristup jednostavniji za početni razvoj, on sa sobom nosi veliki broj nedostataka kako sistem raste. Svaka pa i manja izmena zahteva ponovno testiranje i implementaciju celokupne aplikacije. Još jedna velika mana je da je ceo sistem vezan za jednu tehnologiju.

Zbog ovih nedostataka, za ovaj projekat je odabrana mikroservisna arhitektura [1]. Ona podrazumeva razbijanje sistema na skup manjih, nezavisnih servisa, gde je svaki servis odgovoran za jedan specifičan poslovni zadatak. Ovaj pristup je doneo par ključnih prednosti. Fleksibilnost tehnologije, koja je bilo od ključnog značaja za AI servis, nezavisnost i lakše održavanje što omogućava da se svaki servis razvija, testira i ažurira nezavisno i skalabilnost.

2.1. Način organizacije mikroservisa



Slika 1. Arhitektura sistema sa prikazom komunikacije

Unutar mikroservisne arhitekture, primenjeno je nekoliko ključnih obrazaca kako bi se sistem efikasno sagradio. On je podeljen na servise po njihovoj poslovnoj funkciji. Postoje četiri servisa koja izvršavaju glavne funkcije ovog sistema. Vizualan prikaz ove arhitekture, kao i način na koji servisi međusobno komuniciraju, dat je na Slici 1.

- User Service: upravlja korisničkim nalogima, procesima registracije i prijave, kao i izdavanjem sigurnosnih JWT tokena [4]
- Trade Service: sadrži ključnu poslovnu logiku, upravlja stanjem baterije i donosi odluke o trgovini.
- AI Service: funkcioniše kao specijalizovana komponenta koja na zahtev isporučuje predikcije cena.
- Data Service: centralna tačka za pristup bazi podataka, obezbeđujući potrebne podatke svim ostalim servisima.

“Baza podataka kao servis” (DaaS) [5]: Mikroservis-data simulira ulogu DaaS provajdera. Umesto da svaki servis direktno pristupa bazi podataka, svi zahtevi za čitanje ili pisanje podataka idu isključivo preko ovog servisa. Iako predstavlja odstupanje od “čistog” mikroservisnog pravila gde svaki servis ima svoju bazu, ovaj pristup je odabran kao savršeno rešenje za početnu fazu projekta. Postoje mnoge prednosti u korišćenju paterna Shared Database [6] za razvoj nove velike aplikacije koja uključuje mikroservise.

3. KORIŠĆENE TEHNOLOGIJE I ALATI

Za realizaciju ovog projekta korišćena je kombinacija modernih tehnologija, gde je svaka odabrana tako da na najbolji način odgovori na specifične zahteve.

Frontend – Korisnički interfejs je izgrađen pomoću React biblioteke, koja omogućava kreiranje dinamičkih i reaktivnih komponenti. Za upravljanje globalnim stanje aplikacije, korišćen je Redux.

Backend – Osnovu za User, Trade i Data servise čini Node.js, platforma izabrana zbog svoje efikasnosti u obradi mrežnih zahteva. Na njoj je korišćen Express.js framework za lako kreiranje REST API endpointa. Za AI servis odabran je Python zbog velike dostupnosti biblioteka, dok je Flask [3] poslužio kao web framework.

Baza podataka – Svi podaci se čuvaju unutar MongoDB, popularnoj noSQL bazi podataka. Za komunikaciju drugih servisa i baze korišćen je Mongoose, alat koji olakšava definisanje modela podataka i rad sa njima.

Veštačka inteligencija – Model za predikciju cena je implementiran pomoću Scikit-learn biblioteke.

Kontejnerizacija – Ceo sistem je orhanizovan pomoću Docker [2] platforme, gde je svaki mikroservis zapakovan u sopstveni izolovani kontejner. Za pokretanje i međusobno povezivanje svih kontejnera u lokalnom okruženju korišćen je Docker Compose, čime je postignuta laka prenosivost i konzistentnost sistema.

4. SPECIFIKACIJA SISTEMA

Ovo poglavlje detaljno opisuje funkcionalnosti sistema iz korisničkog i sistemskog ugla.

4.1. Korisničko iskustvo i funkcionalnost sistema

Korisnik po prvom ulasku na aplikaciju ima pristup početnoj strani, gde se nalazi cena energije kao i istorija cena, i takođe može da pristupi predikcijama i strategijama trgovine. Pre korišćenja glavne funkcionalnosti, potrebno je prvo kreiranje profila. Nakon uspešne registracije, kao i prilikom svake sledeće prijave, korisnik unosi svoje kredencijale. Ukoliko je autentifikacija uspešno prošla, korisniku se generiše JWT (JSON Web Token) [4]. Ovaj token služi kao digitalni ključ koji korisniku omogućava siguran pristup svim delovima platforme.

Glavna kontrolna tabla, koja je centralno mesto za sve aktivnosti, dizajnirana je da bude pregledna i intuitivna, a njen izgled je prikazan na slici 2. Na tabeli se ističe grafikon koji vizualizuje istorijke, ali i predviđene cene električne energije dobijene od AI servisa. Pored ovog grafa korisnik ima i opciju da pogleda predikcije za ceo sledeći dan na stranici Market.

Battery stranica korisniku omogućava da registruje svoje baterije i definiše parametre za njih. Svaka baterija pored svojih vrednosti ima i strategiju trgovine kao i stanje u kom se nalazi. Pored pasivnog praćenja, korisnik ima mogućnost I da aktivno upravlja svojim resursima. Platforma nudi izbor predefinisanih strategija tgovanja koje korisnik može da dodeli svojim baterijama. Svaka strategija definiše različit pristup trgovini. I van strategija korisnik može da upravlja svojim baterijama manuelno.

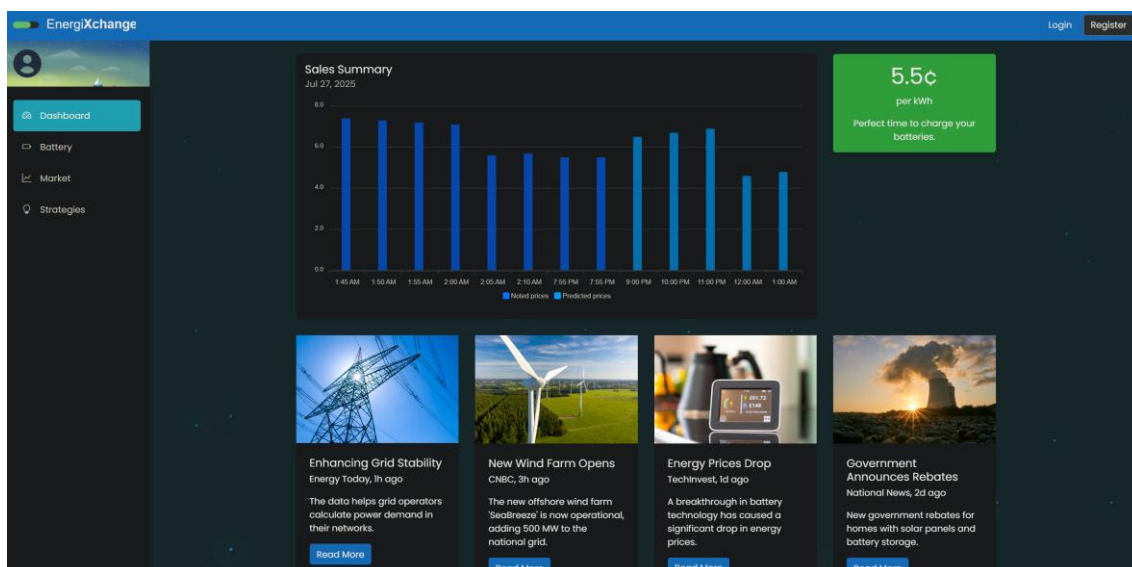
Ipak, ključna prednost platforme leži u njenom automatskom radu. Jednom kada korisnik odabere strategiju, sistem u pozadini preuzima upravljanje. Ovo je najvažniji sistemski proces, nevidljiv za korisnika, gde Trade Service neprestano analizira predikcije cena dobijene od servisa i u skladu sa aktivnom strategijom, on samostalno donosi odluke o najboljem trenutnom stanju baterije.

Sve spomenute automatske akcije se beleže kao transakcije i ažuriraju stanje korisnikovog virtualnog novčanika. Svi ovi podaci su vidljivi na user stranici, i korisnik može uvek da doda sredstva na spomenut novčanik. Uplata je ovde obezbeđena Google reCaptcha-om v3 [8].

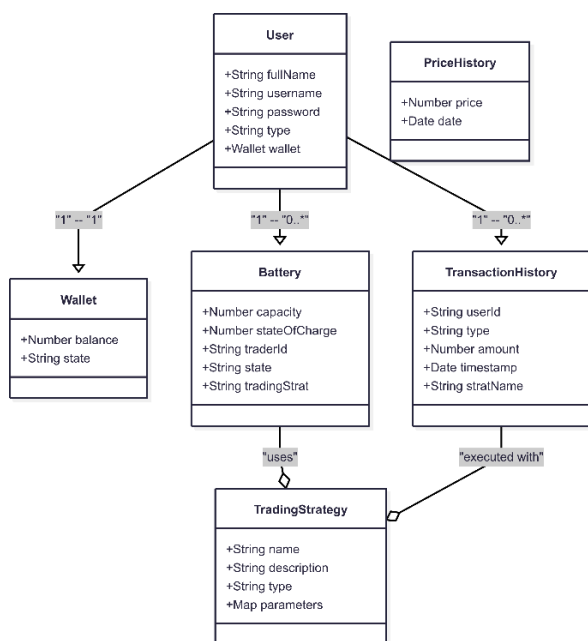
4.2. Dizajn model podataka

Osnovu sistema čine Mongoose modeli koji definišu ključne entitete I njihove međusobne veze. Statička struktura ovih modela, koja predstavlja temelj celog sistema, detaljno je prikazana na klasnom dijagramu na slici 3.

Kao što dijagram ilustruje, User model je centralni entitet koji sadrži osnovne informacije o korisniku I njegovom novčaniku. On je direktno povezan sa Battery modelom, omogućavajući da jedan korisnik poseduje više baterija, kao i sa TransactionHisotry modelom, koji čuva zapise o svim njegovim finansijskim aktivnostima. TradingStrategy model povezan je sa baterijom.



Slika 2. Izgled glavne kontrolne table aplikacije



Slika 3. Klasni dijagram

5. MODEL ZA PREDIKCIJU CENA

Za potrebe ovog projekta izabran je model Linearne Regresije [7] implementiran pomoću Scikit-learn biblioteke. Iako postoje kompleksniji modeli za vremenske serije, ovaj model je odabran iz nekoliko praktičnih razloga. On predstavlja odličnu polaznu osnovu i veoma je brz za treniranje. Takođe rezultati su lako razumljivi, što znači da se može jasno rezumeti kako svaka ulazna karakteristika utiče na krajnju predikciju. Za prototip sistema čiji je cilj dokazivanje koncepta, ove prednosti su bile presudne.

5.1. Proces treniranja

Proces rada modela se odvija u dve faze:

- **Faza treniranja:** U ovoj fazi, model se trenira na istorijskim podacima. Ulazni podaci su numeričke karakteristike, a ciljna vrednost koju model uči da predvidi je cena. Pozivom fit metode, model pronalazi matematičku formulu, odnosno linearnu vezu između ulaznih karakteristika i cene. Jednom istreniran model, odnosno naučeni koeficijenti, čuvaju se u .pkl fajlu pomoću Joblib biblioteke.
- **Faza predikcije:** Kada Trade Service zatraži predikciju, učitava se sačuvani model iz .pkl fajla. Zatim se kreiraju buduće vremenske oznake i na njima se primenjuje isti proces inženjeringa karakteristika. Tako pripremljeni podaci se prosleđuju modelu, koji pomoću naučene formule izračunava i vraća predviđene cene za naredni dan.

6. ZAKLJUČAK

Na osnovu iznetih rezultata, može se zaključiti da primena mikroservisne arhitekture predstavlja efikasno rešenje za razvoj kompleksne platforme za trgovinu energijom. Pokazalo se da je ovakav pristup značajno fleksibilniji od monolitnog, naročito u domenu kombinovanja tehnologija. Korišćenje pragmatičnog obrazca kao što je Daas se pokazalo kao dobar kompromis koji je ubrzao početni razvoj kao i testiranje aplikacije. Potrebno je u budućnosti istražiti da li je ovakav model pouzdan i za globalnu upotrebu ili je "tradicionalni" pristup ipak bolji.

U daljem radu, primarni fokus bi se mogao staviti na unapređenje prediktivnog modela. Istraživanje upotrebe primene naprednijih modela, poput LSTM (Long Short-Term Memory) neuronskih mreža, koje su specijalizovane za analizu vremenskih serija. Ovo bi bilo od velikog značaja za preciznost same platforme, a samim tim i profitabilnosti iz automatske trgovine i pouzdanosti celog sistema.

Još jedan ocigledan korak jeste hostovanje platforme na

nekom od cloud servisa i detaljno testiranje performansi u realnom okruženju, što bi potvrdilo spremnost sistema za praktičnu upotrebu.

4. LITERATURA

- [1] Sam Newma – Building Microservices: Designing Fine-Grained Systems
- [2] Docker - <https://www.techtarget.com/searchitoperations/definition/Docker>
- [3] Flask - <https://flask.palletsprojects.com/en/stable/>
- [4] JWT Token - <https://www.geeksforgeeks.org/web-tech/json-web-token-jwt/>
- [5] DaaS - <https://www.mongodb.com/solutions/use-cases/data-as-a-service>
- [6] Pattern: Shared database - <https://microservices.io/patterns/data/shared-database.html>
- [7] Linear Regression - <https://www.geeksforgeeks.org/machine-learning/ml-linear-regression/>
- [8] reCAPTCHA - <https://developers.google.com/recaptcha/docs/v3>

Kratka biografija:



Aleksa Popov rođen je u Zrenjaninu 1998. god. Završio je osnovne akademske studije na Fakultetu tehničkih nauka 2021. godine, smer Primijenjeno softversko inženjerstvo. Nakon toga upisao je iste godine master studije, smer Primijenjeno softversko inženjerstvo.

Kontakt: popovaleksa123@gmail.com