

## Дизајн и архитектура на софтвер

### Домашна работа 2

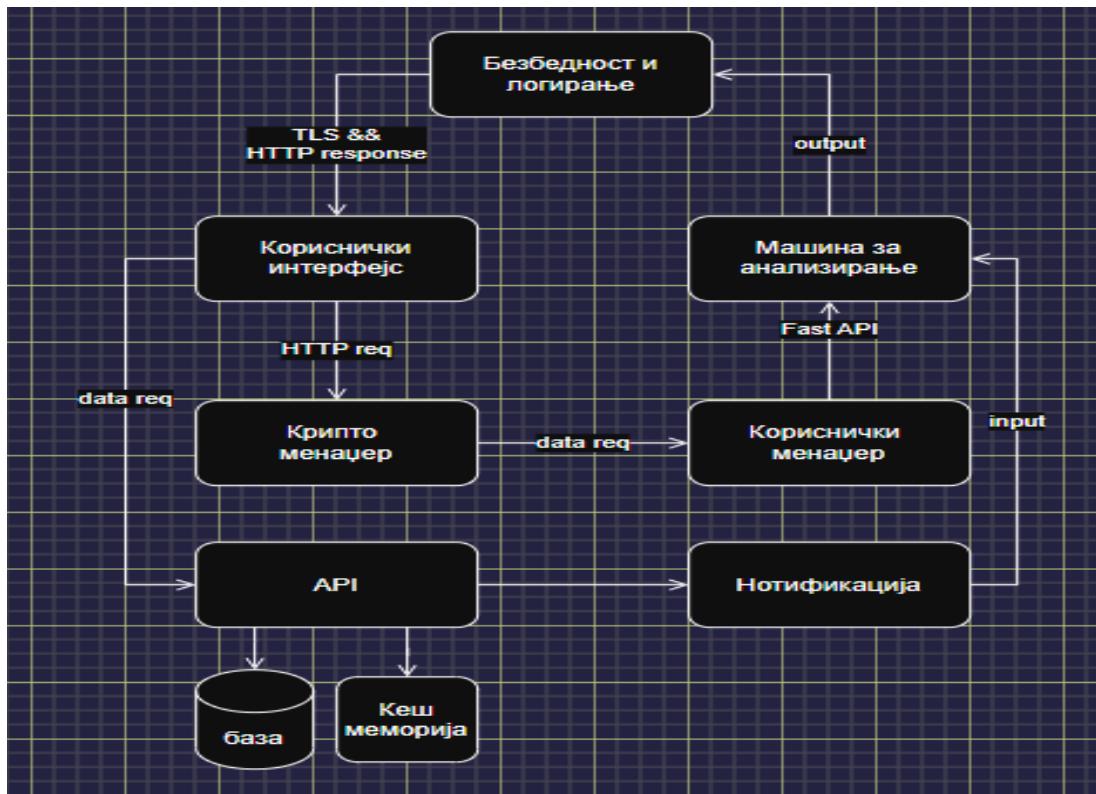
### Архитектура на крипто апликацијата

Во оваа домашна задача ќе ја опшириме апликацијата за анализа на крипто берзата која собира историски податоци за највредните 1000 активни криптовалути од берзите:

- **CoinGecko**: база што се користи за обработка на податоците од последната година; доколку нема доволно податоци за некоја извлечена валута, тогаш го користиме
- **Binance**: база што се користи за обработка на податоците од последните 10 години.

**Концептуална архитектура** - ги прикажува сите логички компоненти и нивните односи, како течат информациите низ системот.

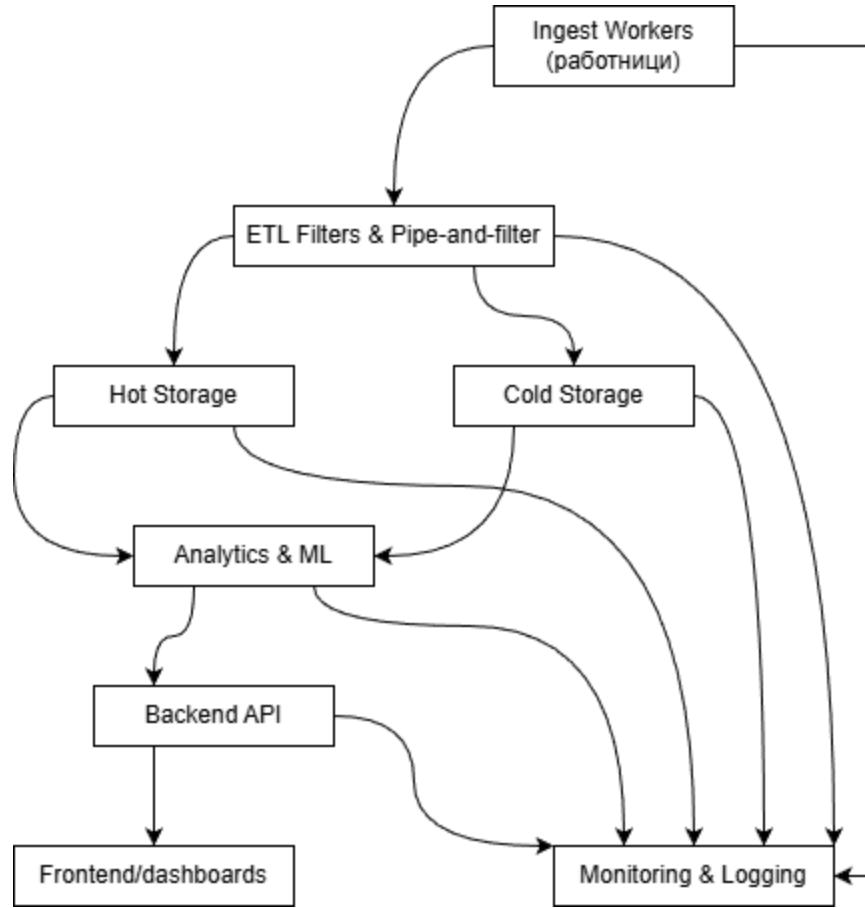
Овде спаѓаат и изворите на податоците, како API-њата на крипто берзите, историските датотеки и јавните датасети. ETL или pipe-and-filter слојот е редослед на трансформации и филтри за чистење, нормализирање и агрегација. Во архитектурата се вклучени складиштата за податоци за брзи упити и за долгорочко складирање. Исто така се вклучени и агрегирањето, статистичките пресметки, и ML моделите. Backend API сервисот е прикажан како интерфејс за пристап до податоците и аналитичките резултати. Frontend делот служи за визуелизација и интеракција со корисникот.



- **Безбедност и логирање:** ги следи сите активности во системот, обезбедува TLS енкрипција, генерира HTTP одговори и логови, следи излез од аналитичката машина.
- **Кориснички интерфејс:** прима HTTPS одговори, праќа кориснички HTTP барања.
- **Крипто менаџер:** обработува криптографија (шифрирање/десифрирање) и врши проверка на автентификација/токени.
- **Кориснички менаџер:** управува со корисничките податоци, валидира, обработува, и проследува податоци.
- **Машина за анализирање:** обработува податоци, модели, анализи, AI/ML, и генерира резултати.
- **API:** ги прима податоците од крипто менаџерот, поврзува бизнис логика & база & кеш.
- **База:** долготочно складира податоци.
- **Кеш меморија:** кешира чести податоци за побрз систем.
- **Нотификација:** испраќа известувања (email, sms, push).

**Извршна (логичка/runtime) архитектура** - ги опишува компонентите и нивните функционалности за време на извршување.

Опфатени се и ingest работниците кои собираат податоци во реално време или во batch режим. ETL филтрите за процесирање и трансформација на податоци преку различни чекори припагаат и во овај дел. Прикажани се hot (за брзи упити, се чуваат дневни и субдневни агрегирани податоци) и cold (историски податоци, овозможува backfill и анализа на податоци од минатото) storage системите, каде се чуваат податоците за краткорочна и долготочна употреба. Analytics и ML компонентите се вклучени за пресметки, предикции, и backtesting. API сервисите служат за обработка на барања од корисниците и други сервиси. Frontend делот е претставен како потрошувач на API резултати за визуелизација. Извршната архитектура ги опфаќа и комуникациските канали како секвенца на пораки или стримови на настани. Вклучено е и мониторирање (следење) и логирање на сите компоненти за да се следи текот на податоците, интеракцијата помеѓу сервисите и како системот оперира.



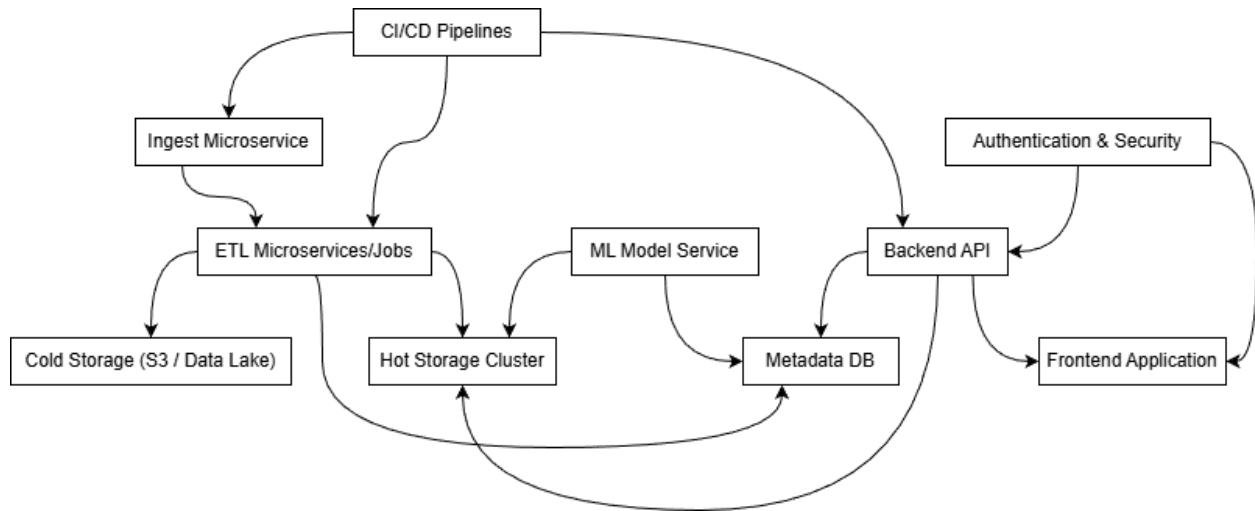
- **Ingest Workers (работници)**: ги собираат податоците од крипто берзите користејќи REST API или WebSocket → ги вчитуваат и доставуваат податоците во ETL процесот.
- **ETL Filters & Pipe-and-filter**: компоненти кои ги трансформираат, чистат, нормализираат, и агрегираат raw (“сирови” т.е. неизменети) податоците, подготвувајќи ги за складирање и анализа.
- **Hot Storage**: база на податоци за брзи упити, каде се чуваат дневни и субдневни агрегирани податоци т.е. база од податоци за краткорочна употреба.
- **Cold Storage**: база за долгорочно чување на историски податоци, овозможува backfill и анализа на податоци од минатото т.е. база од податоци за долгорочна употреба.
- **Analytics & ML**: сервис кој обработува податоци за аналитика, изведува ML (machine learning) модели, предикции, кластерирација (групирање на слични објекти во помали групи т.е. “кластери”), и backtesting.
- **Backend API**: сервис кој им овозможува на frontend апликациите и на други системи пристап до податоците и аналитичките резултати.

- **Frontend/dashboards:** кориснички интерфејс кој прикажува графици, dashboards, и извештаи за цените и анализа (OHLCV open high low close volume) на криптовалути.
- **Monitoring & Logging:** компонента која го следи системот, собира metrics (систем/стандарт за мерење), логира настани и грешки за лесно одржување и откривање на проблеми.

**Имплементациска (физичка) архитектура** - опфаќа конкретни технологии, сервиси и физички компоненти. Описува кои сервиси постојат, на кој јазик се реализирани и како се развиваат.

Вклучува ETL микросервиси, работници за ingest, аналитички и ML модули. Складиштата се конкретно дефинирани, на пример TimescaleDB или ClickHouse за топло складиште и S3/Parquet за ладно складиште. API и backend сервисите се опфатени со конкретни ендпоинти и протоколи. Frontend е дефиниран како SPA со React или друга технологија.

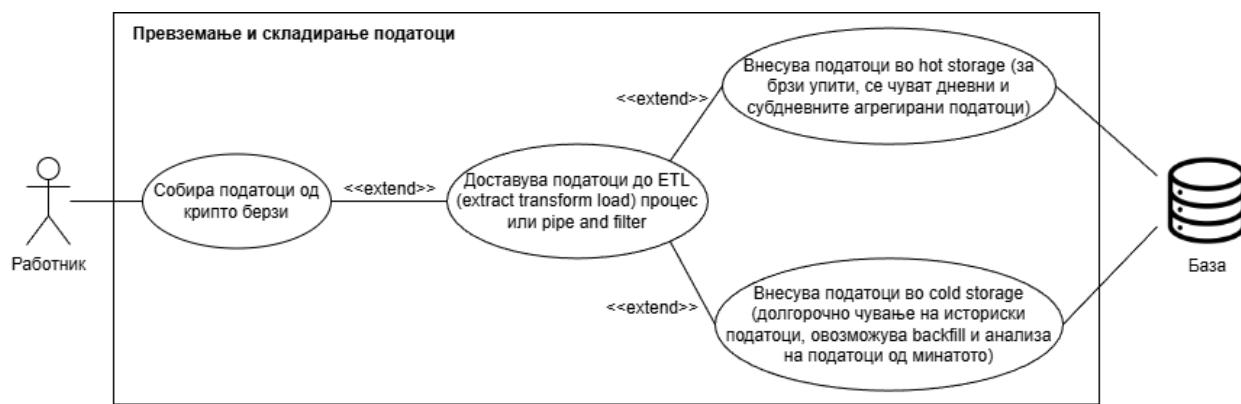
Имплементациската архитектура покажува како логичките делови од системот се реализираат во вистински, функционален сервис. Таа ги опфаќа начините на поставување на сервисите, како се следат и управуваат верзиите на апликацијата, како се штитат чувствителните податоци и како се следи и бележи што се прави во системот. Покажува и начини за подобрување на перформансите преку кеширање и како системот може да расте ако има повеќе податоци или корисници (скалирање).



- **CI/CD Pipelines:** автоматизација на build, тестирање и deployment на микросервисите и другите компоненти.
- **Ingest Microservice:** микросервис кој се поврзува со берзите, собира податоци и ги испраќа кон ETL сервисите.

- **ETL Microservices/Jobs:** микросервиси или batch jobs кои ги трансформираат податоците т.е. ги чистат, нормализираат, и агрегираат пред да се зачуваат.
  - **Cold Storage (S3 / Data Lake):** (cold/долгорочно) складиште за историски податоци, обично во Parquet или Delta формати за долгогодишна анализа.
  - **Hot Storage Cluster:** (hot/краткорочна) група (“кластер”) на бази (на пример TimescaleDB или ClickHouse) за брзо читање и запис на дневни податоци.
  - **ML Model Service:** сервис за тренирање на модели и предикција, користи податоци и од hot и од cold storage.
  - **Metadata DB:** база која чува информации за симболи, парови, берзи, верзии на податоци, и data lineage (процес на следење на животниот циклус на податоците од потеклото до дестинацијата, документирање на сите трансформации и движења).
  - **Backend API:** сервис кој нуди ендпоинти за барања од frontend апликацијата и други сервиси, управува со упити и логика.
  - **Frontend Application:** веб апликација која прикажува dashboards, графици, и извештаи за корисниците.
  - **Authentication & Security:** механизми за безбедност, управување со тајни, OAuth2/JWT authentication, encryption, и audit логирање.

**Use Case дијаграм за како се одвива превземањето и складирањето на податоци:**



**Сценарио:** Преземање и складирање податоци.

1. Работникот собира податоци од крипто берзи.
  2. Ги доставува податоците до ETL процесот или pipe-and-filter.
    - 3.1 Ги внесува податоците во hot storage за краткорочна употреба.
    - 3.2 Ги внесува податоците во cold storage за долгорочна употреба.
  4. Податоците се складираат во база.