Архитектурен дизајн

Crypto App е веб-апликација за визуелизација и анализа на податоци за криптовалути кои се преземаат од надворешен API и се складираат во база. Архитектурата е хибридна и ги комбинира:

- Слоевита веб архитектура (UI – Controller – Service – Repository – Database),
- Pipe & Filter подархитектура во сервисот за прибирање податоци,
- Дистрибуирана архитектура со микросервиси и
- Контејнеризација со Docker.

1. Концептуална архитектура

User / App UI

Корисникот преку веб-интерфејс (App UI) може да избере криптовалута, да гледа табели и графици со историски податоци и да добие дополнителни „insights“.

Controller

Го прима HTTP барањето од App UI (UI actions: coins, insights) и го проследува до Service слојот. Тој претставува централна точка за обработка и насочување на барањата во веб-апликацијата.

Service

Во Service слојот се наоѓа главната бизнис логика: избор на криптовалута и подготовка на податоци за приказ во UI. Service користи два репозиториуми:

- **CoinRepository** – пристап до податоци за криптовалути (симбол и име),
- **HistoricalDataRepository** – пристап до дневните OHLCV вредности.

Database

Централна PostgreSQL база во која се чуваат и тековните информации за криптовалутите и историските записи.

Data Ingestion Service

Посебна логичка компонента задолжена за прибирање на податоци од External API (CryptoCompare). Овој сервис ги обработува податоците и ги запишува во базата. Тука концептуално се применува Pipe & Filter: низ повеќе чекори (филтри) се прави преземање, чистење, нормализација и мапирање во ентитети за база.

Со ова, концептуалната архитектура го прикажува хибридниот пристап: слоевита веб-апликација за читање/приказ на податоци и одделен ingestion сервис кој работи како pipeline.

2. Извршна архитектура (runtime)

На вториот дијаграм е прикажан протокот на повици во runtime:

Корисник → App UI

Корисникот избира криптовалута или insight. UI праќа HTTP (sync) барање до Controller.

App UI → Controller → Service

Controller синхроно го повикува Service слојот.

Service → CoinRepository / HistoricalDataRepository → Database

Service синхроно ги повикува репозиториумите, а тие преку SQL/ORM комуницираат со базата. Се враќаат потребните податоци за валутата и нејзината историја.

Service → Controller → App UI

Податоците се обработуваат и се враќаат назад до UI како табела/график (без скриени асинхрони повици кон надвор).

Паралелно со ова, во background:

Data Ingestion Service → External API (Async call)

Ingestion сервисот периодично прави асинхрон повик кон External API.

Data Ingestion Service → Database (Sync)

По Pipe & Filter обработката, сервисот синхроно ги запишува новите записи во базата.

Следното корисничко барање веќе чита свежи податоци преку истата синхрона HTTP → Service → Repository → Database патека.

Така, извршната архитектура комбинира синхрон веб-слој за корисникот и асинхрон ingestion pipeline, кои се спојуваат во базата.

3. Имплементациона архитектура

Имплементационата архитектура на Crypto App е базирана на дистрибуиран пристап со повеќе микросервиси, контејнеризирани со Docker. Системот се состои од три главни контејнери: веб-апликацијата, базата на податоци и сервисот за прибирање податоци.

Веб-апликацијата (Spring Boot) е организирана според слоевита веб архитектура и ги содржи презентацискиот слој со Thymeleaf, сервисниот слој со бизнис логика и интеграцискиот слој со репозиториуми и REST контролери. Таа го опслужува корисникот и ги прикажува податоците во UI.

Во посебен контејнер работи Data Ingestion микросервисот. Овој сервис периодично прави REST повици до надворешниот CryptoCompare API, ги обработува податоците и ги запишува во базата.

PostgreSQL базата на податоци е независен контејнер кој служи како централен складишен слој за валутите и нивните историски вредности. Комуникацијата меѓу сервисите се врши преку HTTP и SQL конектори.

Ваквата имплементација обезбедува јасно раздвојување на функционалностите, независно извршување и лесно одржување, што се главните придобивки од примената на микросервисна архитектура и Docker контејнеризација.