Arhitectura aplicatiei

Arhitectura este elementul cuprinde activitatile pe care utilizatorul le poate realiza in cadrul aplicatiei, functionalitatile acesteia, serverele pe care se bazeaza si blockchain-urile accesibile prin acestea.

Graphical user interface

Description automatically generated with medium confidence

Arhitectura este de tip REST - Representational state transfer si este un stil creat pentru a ghida implementarea efectiva a aplicatiei, dar si a design-ului acesteia. Prin intermediul arhitecturii de tip REST, fiind bazata pe protocolul HTTP, se respecta standardele aplicatiilor informatice de pe web, facilitand comunicarea intre acestea.

Un request de tip HTTP este realizat de catre client catre asa numitul “host”, bazat pe un server. Scopula cestui request este de a accesa resursele de pe server. Pentru a realiza un request, clientul se foloseste de componenta unui URL - Uniform Resource Locator care include informatiile necesare pentru a accesa resursa. In general, un request este impartit in trei parti: o linie de request, un set de campuri pentru antet (header) si un corp care este optional.

Table

Description automatically generated

Linia de request (Request Line)

Linia de request este prima linie din request-ul realizat. Ea contine cel putin aceste trei componente:

1. O metoda – acestea ii indica serverului c ear trebui sa faca cu request-ul. De exemplu, serverului i se cere sa trimita resursa – metodele de GET si POST.
2. Calea componentei – aceasta indica unde se afla resursa pe server.
3. Numarul versiunii de HTTP – arata specificatia HTTP cu care clientul a incercat sa faca request conform protocolului.

Antetele HTTP ( HTTP headers)

Antetele HTTP sunt scrise pentru a oferi informatii destinatarului despre expeditor si despre modul in care expeditorul incearca sa comunice cu destinatarul. Aceste antete sunt alcatuite din nume si valoare.

Corpul request-ului (Body)

Continutul corpului mesajului ( a request-ului) poate fi denumit corp de mesaj sau corp de entitate. Acesta este continul real al mesajului. Corpul mesajului conține corpul entității, care poate fi în starea sa originală sau poate fi codificat într-un fel pentru transport, cum ar fi împărțit în bucăți (codare de transfer în bucăți). Corpurile de mesaje sunt adecvate pentru unele metode de solicitare și nepotrivite pentru altele. Astfel ca o cerere cu metoda POST, care trimite date de intrare către server, are un corp de mesaj care conține datele. O cerere cu metoda GET, care cere serverului să trimită o resursă, nu are un corp de mesaj.

Prin intermediul acestui stil de arhitecturi, aplicatia comunica cu cele patru server din cadrul acesteia: WalletConnect, Moralis, Newsdata si Coingecko. In primul rand, pentru ca utilizatorul sa se poata conecta la aplicatie, acesta trimite un request de tip POST serverului WalletConnect, care odata cu verificarea datelor, stabileste o conexiune la distanta, care permite o comunicare sigura intre portofelele virtuale si aplicatiile de tip Dapp (aplicatii decentralizate). Datele trimise sunt encriptate simetric printr-o cheie comuna intre cele doua entitati. Conexiunea este initializata de catre una dintre entitati prin deep link și se stabilește atunci când cealalta entitate aprobă această cerere.

Deep links sau “link-urile profunde” sunt link-uri care trimit utilizatorii catre un alt site sau catre o alta aplicatie instalata. Acestea sunt folosite pentru a trimite utilizatorii direct către anumite locații din aplicație, economisind utilizatorilor timp și energie pentru a localiza ei înșiși o anumită pagină - îmbunătățind semnificativ experiența utilizatorului. Conectarea profundă face acest lucru specificând o schemă URL personalizată (linkuri universale iOS) sau o adresă URL de intenție (pe dispozitivele Android) care deschide aplicația utilizatorului dacă aceasta este deja instalată.

Astfel, procesul prin care utilizatorul se poate conecta la portofelul virtual in aplicatia de tip Dapp este realizat prin intermediul API-ului WalletConnect, care la randul sau, prin intermediul Deep link-ului, redirectioneaza utilizatorul din aplicatia dezvoltata in alta aplicatie, o aplicatie de portofel virtual deja existent, cum ar fi Metamask sau Binance, prin care i se cer credentialele de conectare, verificarea amprentei sau a Face ID-ului. Odata cu aceasta verificare, utilizatorului i se cere semnarea unei tranzactii care atesta ca acesta a fost de acord sa permita conectarea portofelului prin intermediul API-ului WalletConneect si inca o tranzactie prin care isi exprima acordul ca API-ul Moralis sa verifice jetonul de conectare pe parcursul sesiunii in cadrul aplicatiei dezvoltate.

In cadrul aplicatiei, utilizatorul poate adauga fonduri in portofelul sau virtual bazat pe una dintre retelele de Testnet a unuia dintre blockchain-urile Ethereum, Smart Chain, Mumbai sau Avalanche Fuji. Adaugarea de fonduri se realizeaza prin Deep Linking, utilizatorul fiind redirectionat catre unul dintre robinetele, numite in practica Faucet, robinetul fiind specific retelei alesei. Aceasta retea este identificata prin intermediul id-ului de lant (chain).

Ca functii generale, prin intermediul aplicatiei se pot transfera fonduri, se poate verifica istoricul tranzactiilor si a fondurilor existente. Astfel, toate aceste procese sunt bazate pe API-ul Moralis, care are injectat Web3, pentru a avea acces la blockchain. Pentru le putea realiza, trebuie trimise id-ul lantului, dar si adresa portofelului virtual.

Pentru a prelua datele, aplicatia trimite solicitari (request-uri de tip GET) catre API-urile Newsdata si Coingecko, de fiecare data cand utilizatorul ajunge pe unul dintre ecranele de grafice sau de stiri. Astfel, odata preluate se trimit datele inapoi catre frontend pentru a fi formatate si afisate.