Architecture of the Portfolio Manager

1. Overview

The portfolio manager is a decentralized web application designed to track and display cryptocurrency assets, including token balances and NFTs, across various blockchain networks. It leverages Covalent's API for fetching wallet data and visualizes the data in a React-based frontend. This architecture includes multiple subgraphs, API integrations, and a structured data flow for efficient querying and presentation.

2. Frontend (React)

The application interface is built using **React.js**. It consists of two main components:

- Dashboard: Displays the token balances, NFTs, and native currency data.

- Graph Component: Visualizes wallet data using D3.js charts.

Key UI Components:

- Form Component: A form that allows users to input their wallet address and select a blockchain network (Ethereum, Binance Smart Chain, Polygon, or Fantom).

- Token & NFT Display: Tables showing token balances and NFTs with relevant metadata, such as contract addresses, ticker symbols, and quotes.

- Navigation and Routing: React Router is used for navigation between the main portfolio dashboard and a dedicated graph page.

3. Data Flow

1. **User Input**: The user enters their wallet address and selects a blockchain network.

2. **API Calls**: Once the "Explore" button is clicked, the app makes API requests to fetch the token balances, NFT data, and native token balances for the selected wallet.

3. **State Management**: The fetched data is stored in respective React state variables (`tokenData`, `nftData`, `nativeCurrencyData`).

4. **Update**: The state changes trigger the re-rendering of the components, updating the dashboard with the latest data from the wallet.

5. **Navigation to Graph**: When the user navigates to the graph page, the wallet address is passed as a prop to the Graph component, allowing visualization of the wallet's asset distribution.

4. Subgraphs

The application is structured in a way that could extend to use subgraphs for specific queries:

- Wallet Subgraph: Could be used to query detailed wallet transaction history and on-chain events.

- NFT Subgraph: Fetches metadata and trading history of NFTs.

- Token Subgraph: Captures the movement and holding patterns of various ERC-20 tokens.

The subgraphs provide the ability to fetch detailed historical and real-time data efficiently by indexing blockchain data.

5. Graph Component (Data Visualization)

The graph component uses **D3.js** to visualize token balance distributions. It accepts the `address` as a prop, passed from the main page when the user clicks to view graphs. This component pulls the relevant data from the state and displays the proportions of each token type held in the wallet.

6. Routing

- The app uses **React Router** for navigation:

- Main Page: Displays wallet information.

- Graph Page: Displays a chart of the wallet’s token holdings.

The `address` from the input is passed via state to the graph component, ensuring smooth navigation and consistent data flow across the app.

This modular architecture makes the portfolio manager easy to maintain, scalable for future integrations (e.g., additional blockchain networks or querying new subgraphs), and performant in fetching and displaying user data.