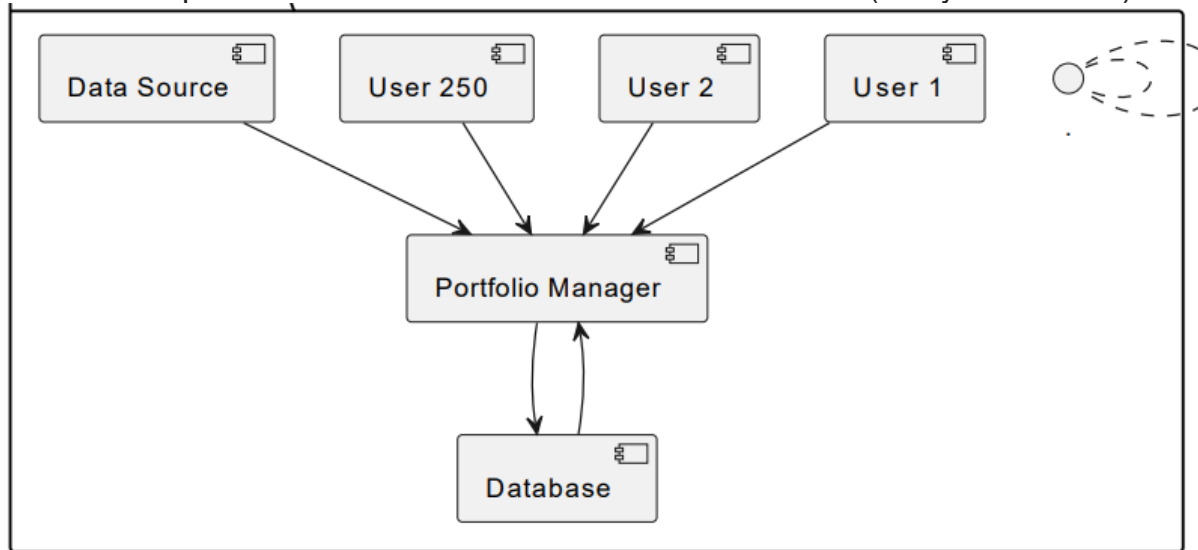


High-Level Design for Scalable Portfolio Management Platform

This document outlines a high-level design for a platform that creates, calculates, and maintains user portfolios. The platform is designed to be scalable and reliable, catering to 250 users who each have at least one asset account (stocks or mutual funds). It offers real-time portfolio updates, meaning users can see their portfolio value at any time. Prices for assets are collected from various sources and reflected in the user's portfolio as soon as the data becomes available (every 10 minutes).



System Components:

The platform is divided into distinct services for better organization, scalability, and maintainability:

1. **User Interface:** This is the web or mobile application that users interact with. It allows them to view their portfolios, manage accounts, and potentially perform other actions related to their investments.
2. **User Service:** This service handles user accounts, including registration, login, authentication, and authorization. It ensures that only authorized users can access their portfolio information.
3. **Asset Service:** This service stores and manages user asset data. This includes information like the symbols of stocks or mutual funds the user holds, as well as the quantity of each asset they own.

4. **Pricing Service:** This service retrieves and processes asset pricing data from external sources. It may need to interact with multiple data providers depending on the variety of assets users hold.
5. **Portfolio Service:** This service is responsible for calculating the total value of a user's portfolio. It takes the asset holdings information from the Asset Service and combines it with the current asset prices retrieved by the Pricing Service.
6. **Cache:** To improve performance and reduce load on the database, the platform utilizes a cache. This temporary storage holds frequently accessed data, such as asset prices. By storing recent price information, the system can retrieve it quickly without needing to query the Pricing Service every time.

Data Flow and User Interaction:

1. A user requests to view their portfolio information through the User Interface.
2. The User Interface interacts with the User Service to authenticate the user and ensure they have permission to access their portfolio data.
3. The User Service, upon successful user verification, retrieves user asset data from the Asset Service.
4. The Asset Service retrieves current asset prices either from the Pricing Service (if not already available in the cache) or directly from the cache if the data is recent enough.
5. The Portfolio Service receives the asset holdings and pricing data. It calculates the total value of the user's portfolio by multiplying the quantity of each asset by its corresponding price and summing the results.
6. Finally, the User Interface receives the updated portfolio information from the Portfolio Service and displays it to the user.

Scalability and Reliability:

- **Horizontal Scaling:** Each service can be independently scaled horizontally by adding more servers to handle increased user load or data processing

demands. This allows the platform to accommodate a growing number of users without compromising performance.

- **Redundancy:** Critical services like the User Service and Pricing Service can be implemented with redundancy. This means having multiple servers running the same service, ensuring that the platform remains available even if a single server fails.
- **Asynchronous Processing:** Tasks like price updates can be processed asynchronously. This means they don't block user requests while waiting for price data to be retrieved or processed. Users can continue to interact with the platform while price updates happen in the background.
- **Caching:** Utilizing a cache significantly reduces the load on the database by storing frequently accessed data like asset prices. This improves the platform's responsiveness and overall performance.
- **Monitoring and Alerting:** Implementing a monitoring and alerting system is crucial for identifying potential issues proactively. This allows for timely intervention and helps maintain service continuity.

Conclusion:

This high-level design provides a solid foundation for a scalable and reliable portfolio management platform. The modular architecture, efficient data flow, and focus on robust infrastructure enable the platform to effectively serve 250 users with real-time portfolio updates. Future enhancements could involve features like portfolio analysis tools and integration with additional data sources to provide users with even more insights into their investments.