


| | |
|---------|--|
| Status | APPROVED |
| Author | @ Vũ Minh Chiến |
| Issue |  ENIO-3515 - TDD: High-level design system architecture DONE |
| Content | <ul style="list-style-type: none"> Objectives Research High-level architecture Module details Data workflows System interactions Resource allocations |

| Related services | Approvers |
|-----------------------|---|
| Architecture Reviewer | <input checked="" type="checkbox"/> @ Đỗ Anh Ba |
| Reviewer | <input checked="" type="checkbox"/> @ Vũ Tùng Linh |
| Reviewer | <input checked="" type="checkbox"/> @ Nguyen Phuc Loi |
| Reviewer | <input checked="" type="checkbox"/> @ Trương Ngọc Giang |

High-level design system architecture

Objectives

Building a local Cardano blockchain independent of the Cardano public networks is necessary for development and testing. Here are detailed requirements and benefits for setting up such an local Cardano blockchain:

Requirements:

To meet development and testing needs, the local blockchain should include the following components:

- **Cardano node:** The core node that interacts with the blockchain.
- **Database sync data service:** A service for synchronizing data with the blockchain.
- **Database server:** A database server for storing synchronized data.
- **Cardano builder tools:**
 - **Blockfrost** - API service for interacting with Cardano blockchain data.
 - **Cardano-wallet** - tool for managing Cardano wallets and transaction submissions.

Benefits:

1. Safe Testing Environment

Avoiding Mainnet Risks: Testing on the mainnet involves real assets and transactions, which can be risky. Errors or bugs could lead to loss of real funds or other significant issues. A local blockchain provides a risk-free environment to test new code and features.

Controlled Environment: Developers can control and reset the local blockchain as needed, which isn't possible on the public networks. This allows for repeatable tests and easy recovery from errors.

2. Rapid Development Cycle

Immediate Feedback: Changes and updates can be tested immediately on a local blockchain without the delays associated with deploying to the mainnet. This speeds up the development cycle and iterative testing.

Version Control: Developers can test various versions of their smart contracts and applications quickly, allowing for rapid prototyping and debugging.

3. Custom Configuration

Tailored Environment: Developers can customize the local blockchain to mirror specific conditions or test scenarios that might not be present on the mainnet. This includes setting custom parameters, creating test tokens, and adjusting network settings.

Simulating Failures: You can simulate network failures, attacks, and other adverse conditions to test how your application or contract behaves under stress or unexpected conditions.

4. Integration Testing

Component Interaction: A local blockchain allows for testing the interaction between various components of a decentralized application (dApp), such as smart contracts, front-end interfaces, and backend services, in a controlled setting.

API Testing: Developers can test interactions with blockchain APIs without affecting the mainnet or other developers' work.

Research

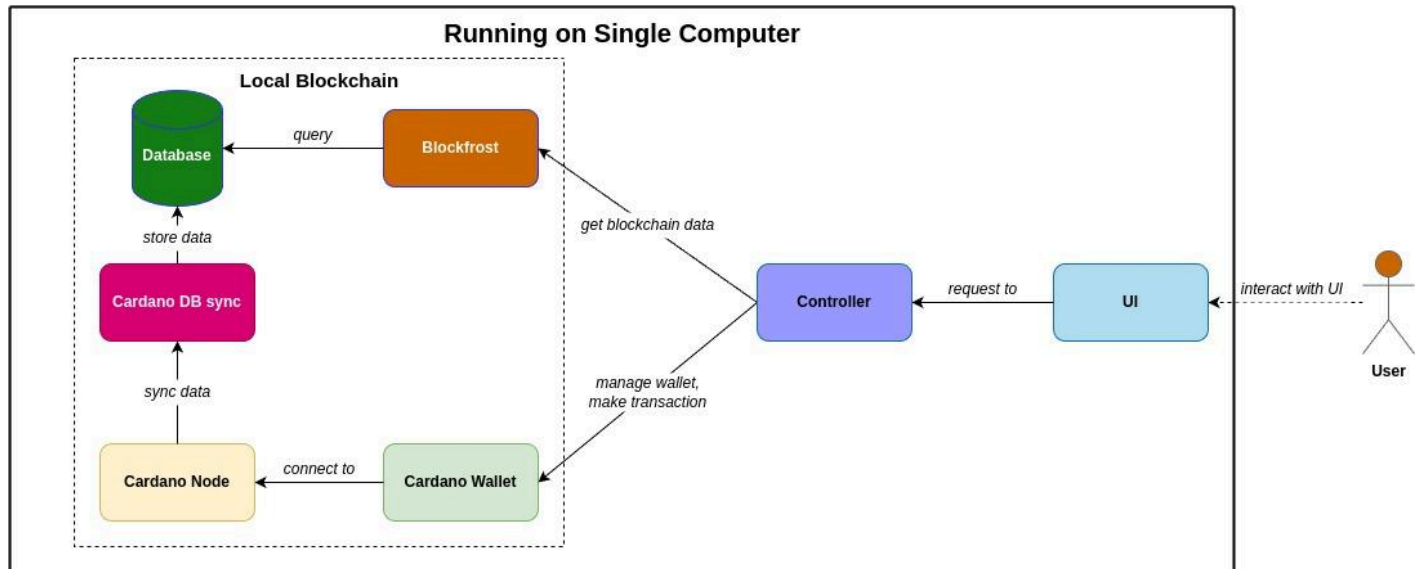
These research focuses on building a local Cardano blockchain system:

- Cardano protocol parameters and node configurations: [Protocol parameters and node configurations](#)
- Building a Cardano local Blockchain from components: [Build Cardano local Blockchain](#)

High-level architecture

Scope: This high-level design aims to build a local Cardano blockchain with a one-node architecture. To keep the HLD focused and straightforward, we have excluded aspects related to the broader Cardano network and node-to-node inter-process communication (IPC) protocols.

The following diagram outlines the interaction between the components of Cardano local blockchain system:



Module details

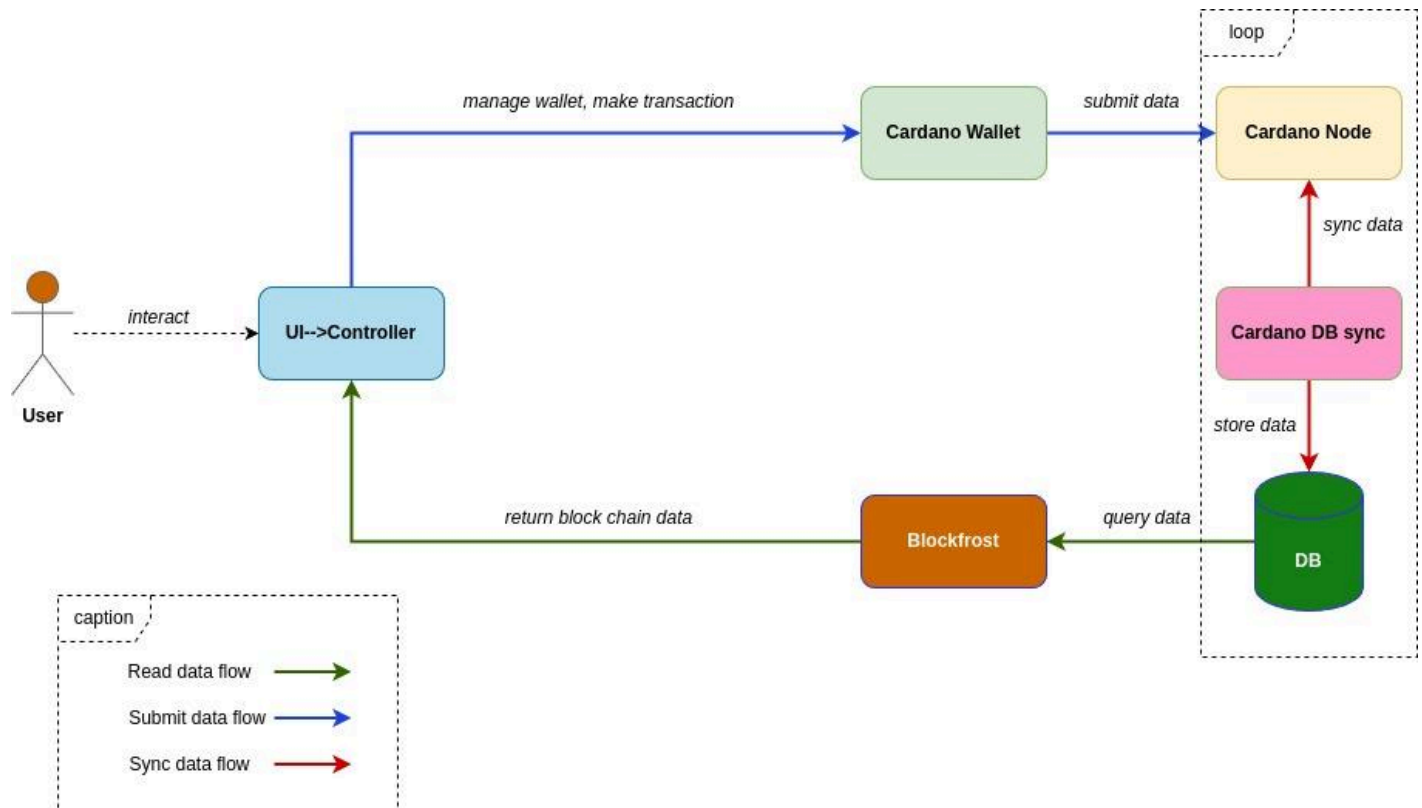
| Component | Propose | Usage | Explain |
|-----------------|---|--|--|
| Cardano node | The core software that interacts with the Cardano blockchain. It handles block creation, transaction processing, and the overall maintenance of the blockchain network. | It serves as the interface between the blockchain and its users, ensuring that blocks are validated and transactions are processed according to the Cardano protocol. | |
| Cardano DB-Sync | Cardano DB-Sync is a service that synchronizes the Cardano blockchain data with a database | It enables querying and analysis of blockchain data by maintaining a synchronized state of the blockchain in a relational database. This helps in building applications that require historical data, such as explorers or analytics tools. | |
| Database | It is a relational database system used to store data from the Cardano blockchain. | It is used by Cardano DB-Sync to store blockchain data. This enables efficient querying and reporting. | |
| Cardano-wallet | Cardano-wallet is an application that allows users to store, manage, and interact with their Cardano (ADA) cryptocurrency and other assets on the Cardano blockchain. | <p>It is designed to be used by developers to integrate wallet features into their applications. the UI service using Cardano-wallet for submit data to the blockchain, include:</p> <ul style="list-style-type: none"> Create, seed, manage wallet Build, sign, submit transaction Create native tokens and NFTs. Deploy smart contract | <p>Blockfrost features:</p> <ul style="list-style-type: none"> - Ledger, epoch, block data - Addresses, assets - Transaction details, utxos, metadata <p>Cardano Wallet features:</p> <ul style="list-style-type: none"> - Create, seed, manage wallet - Build, sign, submit transaction - Create native tokens and NFTs - Deploy smart contract <p>UI features:</p> <ul style="list-style-type: none"> - Display blockchain information - Wallet management - Make transaction |
| Blockfrost | Blockfrost is a third-party API service that provides access to the Cardano blockchain data. | <p>It offers a convenient way to interact with the blockchain, the UI service using Blockfrost for get blockchain data, include:</p> <ul style="list-style-type: none"> Blockchain ledger, epoch, block data Addresses, assets Transaction details, utxos, metadata | |
| UI | The UI acts as a consumer of features provided by Cardano tools, serving as the bridge that allows users to interact with the local blockchain. | <p>It is the component that directly interacts with the user. It retrieves data from Cardano tools, the main features the UI service support is:</p> <ul style="list-style-type: none"> Display blockchain information | |

| | | | |
|------------|---|---|--|
| | | <ul style="list-style-type: none"> wallet management Make transactions for sending, create native tokens, NFTs or smart contract deployment. | |
| Controller | It is the life-cycle management and API BFF service of Cardano local blockchain | <p>Controller service is responsible for managing the lifecycle of the local blockchain, allowing users to trigger or terminate the blockchain as needed.</p> <p>On the other hand, controller service also works as an API BFF, wrapping all features of Cardano tools and forwarding requests from the UI service to the appropriate Cardano tool only when the blockchain is active.</p> | |

References:

- Cardano node: <https://github.com/IntersectMBO/cardano-node>
- Cardano DB-Sync: <https://github.com/IntersectMBO/cardano-db-sync>
- Cardano-wallet: <https://github.com/cardano-foundation/cardano-wallet>
- Blockfrost: <https://github.com/blockfrost/blockfrost-backend-ryo>
- UI: In progress
- Controller: in progress

Data workflows



1. Read data flow

- Users interact with UI services to make API requests to Blockfrost to fetch data from the Cardano blockchain. Blockfrost queries data from databases and returns the requested data in a structured format.

2. Submit data flow

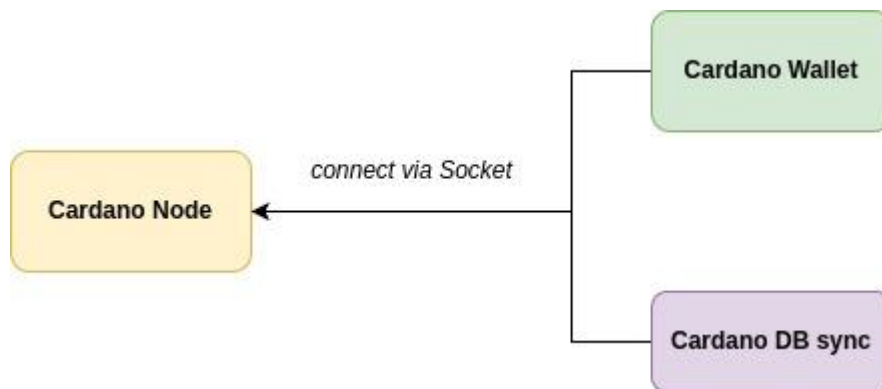
- Cardano-Wallet Foundation manages wallet operations and transactions. Handles wallet creation, key management, and transaction signing, facilitates the submission of signed transactions to the Cardano node and provides functionalities to query wallet balances and transaction history.

3. Sync data flow

- Cardano DB-Sync continuously synchronizes data from the Cardano Node and stores it in a DB. DB-Sync queries the Cardano Node for blockchain data, including transactions, blocks, and addresses. The data is transformed from the node's raw format into a structured format suitable for relational storage. The transformed data is inserted into a DB for efficient querying and analysis.

- Relational database stores the blockchain data in a structured, relational format, stores data such as blocks, transactions, and address information that DB-Sync extracts from the Cardano Node, provides an interface for querying data, enabling tools and applications to retrieve historical and current blockchain information efficiently.

System interactions



1. Cardano Node and Cardano tools

- Cardano node socket is used by various Cardano tools and services to communicate with a running Cardano node. It is essentially an IPC (Inter-Process Communication) that allows different components, like wallets, CLI, Koios, Kupo ... to interact with the node to perform operations like querying blockchain data, submitting transactions, or checking the status of the network.
- The node socket configuration is set up in the socket file of the Cardano node. For more details, please refer to this research: [Build Cardano node](#)

2. Cardano tools and Users/Applications

Blockfrost

- HTTP/REST API: Blockfrost provides a RESTful API that users and applications can query via HTTP requests. This is link API docs: <https://docs.blockfrost.io/>

Cardano wallet:

- HTTP/REST API: Cardano wallet provides a RESTful API to user can manage wallet and make transactions. This is link API docs: <https://cardano-foundation.github.io/cardano-wallet/api/edge/>

Resource allocations

Team Members and Roles

- **Project Manager:** Ba Đỗ Anh
- **Team leader:** Lợi Nguyễn Phúc
- **UX/UI designer:** Ly Ngô Khánh
- **Backend Developer:** Giang Trương Ngọc, Vũ Minh Chiến
- **Frontend Developer:** Trần Việt Anh, Trương Thị Huyền Trâm

Resource Allocation Plan

| Member | Task/Phase |
|-----------------------|--|
| Đỗ Anh Ba | Create project plan, timelines, and milestones |
| Ngô Khánh Ly | UX/UI Designer for UI service |
| Nguyễn Phúc Lợi | <ul style="list-style-type: none">• Manage team, break task• Responsible for local blockchain deployment for testing• Technical support |
| Trương Ngọc Giang | <ul style="list-style-type: none">• Research smart contract deployment• Technical support |
| Vũ Minh Chiến | <ul style="list-style-type: none">• Research Build Cardano blockchain with components• Research Cardano protocol parameters and node configurations• Build Cardano local blockchain by Docker• Design high-level system structure |
| Trần Việt Anh | Technical support |
| Trương Thị Huyền Trâm | Build Ui service |