Lending and Borrowing: High-Level Design

Version: 1.0
Catalyst Project ID: 1200142

1. Introduction

The **Cardano Lending Aggregator** aims to simplify decentralized finance on Cardano by **aggregating** and **comparing** lending and borrowing opportunities across multiple protocols. The initial milestone focuses on:

- **Integrating Liqwid Finance** (pooled lending)
- **Integrating Lenfi (V1)** (peer-to-peer lending)
- Providing a user-friendly **Yield Aggregator** page and **Positions Dashboard**

We've also evaluated other DeFi protocols on Cardano, such as FluidTokens and Levvy Finance. However, we opted to integrate Liqwid and Lenfi first due to their established track records, stable mainnet deployments, available APIs and the fact they are the highest TVL lending and borrowing protocols on Cardano.

Future expansions may include these additional protocols once their APIs/SDKs are publicly available. FluidTokens is an interesting one with what they are doing on the bitcoin side so we will revisit at a later date once they launch their next version.

This document outlines the **high-level design** of our system—covering which protocols are supported, how we index and interact with them, links to relevant APIs/contracts, and the technical structure of our backend/frontend. Our goal is to deliver a solid foundation that can be extended with more protocols and features in later phases.

---

2. Supported Protocols

2.1 Liqwid Finance

- **Protocol Type**: Pooled lending
- **Use Cases**:
    - **Supply** assets to earn yield

    - **Borrow** assets against collateral that the user has provided

- **Data Source**: Liqwid's GraphQL/REST APIs
- **Transaction Building**:
    - Liqwid provides a transaction builder API that returns a CBOR payload.
    - Our platform signs and submits this transaction via an integrated Cardano wallet.

2.2 Lenfi V1 (formerly AADA)

- **Protocol Type**: Peer-to-peer (P2P) lending
- **Use Cases**: Individualized loans with fixed terms and collateral
- **Data Source**: Lenfi APIs for listing open loans, available tokens, and terms
- **Transaction Building**: Currently read-only (no direct TX creation) for V1. We await their Lenfi V2 relaunch, which offers pooled lending and has an official SDK.

Note: Additional protocols (e.g. FluidTokens, Levvy Finance) are on our future roadmap but outside the scope of this milestone.

---

3. Data Indexing & Storage

3.1 Retrieval & Update Frequency

1. **API-Based**: For both Liqwid and Lenfi, we pull data from official APIs and blockfrost rather than directly scanning on-chain events. Future expansions may include an indexer utilizing ogmios if we need to expand the scope.
2. **Scheduling**:
    - **Live Rates & Positions**: Polled every 20 seconds (e.g., APYs, pool liquidity, outstanding loans) in line with Cardano blocks.
    - **Collateral Lists & Configuration**: Fetched once per day (or when changes are detected).

3.2 Database Storage

- **PostgreSQL**:
    - We store fetched data for fast lookups—especially important for our **Yield Aggregator** page, which needs near-instant responses when users switch tokens or switch between “Earn” and “Borrow” modes.
    - Storing snapshots in the DB also facilitates historical comparisons and user queries without hammering third-party APIs constantly.

3.3 Yield Aggregator Page

- The Yield Aggregator is a central UI component where users:
    - Select a token and a mode (Earn or Borrow).
    - View aggregated opportunities from Liqwid and Lenfi (including supply/borrow APYs, available liquidity, loan terms, etc.).
- Because we want near-real-time data and a smooth user experience, the aggregator queries our DB for pre-fetched data every time a user makes a selection.

---

4. On-Chain Transaction Handling

4.1 Overall Approach

- We primarily utilize each protocol’s **official APIs** (e.g., Liqwid’s transaction builder) instead of writing custom Plutus scripts ourselves for this milestone.
- By focusing on stable, documented endpoints, we reduce complexity and align with each protocol’s best practices.

4.2 Wallet Integration with MeshJS

- **MeshJS** is used to handle wallet interactions in the browser:
    - **Detecting** and **connecting** to user wallets (Eternl, Lace etc.)

    - **Reading** wallet balances and assets
    - **Signing** transactions returned by Liqwid's API (in CBOR format)
    - **Submitting** signed transactions to the Cardano network
- This approach leverages the standard **CIP-30** interface for web-based Cardano wallets, ensuring a familiar and secure flow for end users.

Example Flow (Liqwid "Supply")

1. The user clicks "Supply ADA" in the UI.
2. Modal pops up for the user to enter the amount to supply and the system validates that it is more than the min required by Liqwid parameters
3. The backend calls Liqwid's builder API, specifying the asset, amount, wallet UTXOs and relevant parameters.
4. Liqwid returns a CBOR transaction.
5. The frontend, via MeshJS, prompts the user's wallet to sign.
6. The signed transaction is submitted to the network.

4.3 Lenfi V1 (Read-Only)

- For Lenfi V1, we currently display open loans and relevant terms.
- Lenfi peer to peer loans are split into 2 types
    - **Liquidity Requests:** This is where the user submits a loan request for a token and they provide liquidity in their chosen token. They define all the parameters of the loan request including interest to be paid and duration. Health factor is determined by the amount of collateral they provide.
    - **Liquidity Deposits:** This is where the user offers a token that another user can loan out. The user providing the initial token defines the parameters they are willing to accept e.g. the collateral token expected, minimum collateral, interest and term.
- All Lenfi V1 loans are indexed by the platform but transaction creation is not planned for V1 for now.

---

5. API and Contract References

- **Liqwid**
    - **API**: https://v2.api.liqwid.finance/graphql
    - **Contract** - addr1w8arvq7j9qlrmt0wpdvpp7h4jr4fmfk8l653p9t907v2nsss7w7r4

- **Lenfi / Aada**
    - API: https://app.aada.finance/api/ & https://api.lenfi.io/api/v0.1/
    - V2 SDK: https://github.com/lenfiLabs/lenfi-sdk

Contracts and Scripts:

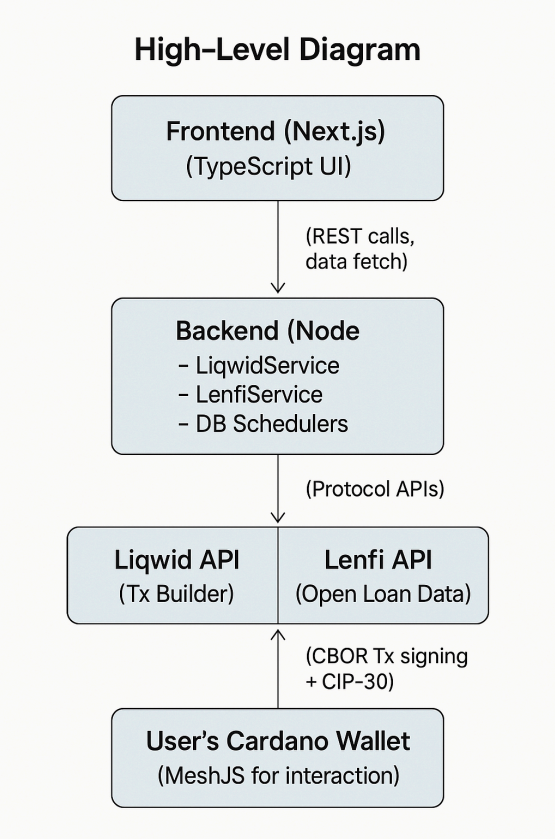
- addr1zy9940grv28qxz9k82l9gmqd80vfd8a2734e35yzsz9cqktfjcnq9fczt4qkxgec2hz6x7f38vnj8xuxywk4x4qgzh9smq5w00, //request.hs -- Request created. Lender to fund
- "addr1zykhtew0z93z6hmgu2ew7kl9puqz0wmafp0f3jypuejkwmrfjcnq9fczt4qkxgec2hz6x7f38vnj8xuxywk4x4qgzh9skq4p22, //collateral.hs -- Loan funded. Borrower to repay
- addr1zxfgvtfgp9476dhmq8fkm3x8wg20v33s6c9unyxmnpm0y5rfjcnq9fczt4qkxgec2hz6x7f38vnj8xuxywk4x4qgzh9st8q78h, //interest.hs -- Borrower repaid -- Lender to claim
- addr1zxcjtxuc7mj8w6v9l3dfxvm30kxf78nzw387mqjqvszxr4mfjcnq9fczt4qkxgec2hz6x7f38vnj8xuxywk4x4qgzh9sp92046, //liquidation.hs -- Funds were liquidated. Borrower to claim
- addr1zytwe3qhc0kf5k8yaur60cnhcxjg9zvfdnftp0rfu2czprtfjcnq9fczt4qkxgec2hz6x7f38vnj8xuxywk4x4qgzh9sgzwepc, //debt_request.hs -- Funds locked as liquidity deposits 'addr1z8tjrqy2dj5uk6her4ksltyxy2flzykktxkahzlahm9nwctfjcnq9fczt4qkxgec2hz6x7f38vnj8xuxywk4x4qgzh9st86ewu', //request.hs v2 -- Funds locked as liquidity deposits
- addr1zyc7w5n699ews00yujnhw59g4nuzykuzgl5x6nzqp49zv5tfjcnq9fczt4qkxgec2hz6x7f38vnj8xuxywk4x4qgzh9sdyxnxc, //collateral.hs v2-- Funds locked as liquidity deposits
- addr1zy6v8c7xdhftln7zk5uvt9h6jaknaxlx6hz5nkw63mpgwamfjcnq9fczt4qkxgec2hz6x7f38vnj8xuxywk4x4qgzh9sw9snf6, //debt_request.hs v2-- Funds locked as liquidity deposits
- script1xt5vpt33fm6tu3fvz65enpnlvmg6z7gle9evktmuwn3c6gjfc7p, // Pool contract
- script1nwvlaa0wnf43wzjp3xv738k6myam74dlrlh027mq20trg3ng772, // Collateral contract
- script1sqscxzh7mkzlmgf98k3tuadkds8xt3yzzj8t3jnfpypukld9xck, // Leftovers contract

- **MeshJS**
    - Docs: MeshJS Dev Documentation
    - GitHub: MeshJS GitHub

If needed, specific on-chain contract addresses or Plutus script hashes can be retrieved from each protocol's doc or Cardano explorers.

6. System Architecture

6.1 High-Level Diagram



6.2 Frontend

- **Next.js** (React + TypeScript) for pages like:
    - **Yield Aggregator**
    - **Positions Dashboard**
    - **Protocol-Specific Pages** (e.g. Liqwid, Lenfi)
- **Interaction Flow**:
    - User selects mode (Earn/Borrow) and token on the aggregator page.

    - Frontend calls our backend to retrieve relevant data from the DB.
    - For Liqwid actions, transaction data is sent back to the frontend, which uses **MeshJS** to request wallet signatures.

6.3 Backend

- **Node.js** application with a set of services (e.g., `LiqwidService`, `LenfiService`) that query third-party APIs.
- **Data Handling**:
    - Schedules frequent updates (every 20 seconds for rates, daily for configurations).
    - Populates a **PostgreSQL** database with the latest protocol data.
- **Endpoints**:
    - e.g., `GET /api/aggregator` returns combined Liqwid + Lenfi data (filtered by token/mode).
    - e.g., `POST /api/liqwid/buildTx` returns CBOR for user-specific transactions.

6.4 Security & Reliability

- **User Assets**: All private keys remain in the user's wallet (MeshJS + CIP-30).
- **Transaction Logic**: Offloaded to Liqwid's official builder or (in the future) Lenfi's SDK.
- **Caching**: The database approach eases load on external APIs and ensures minimal downtime if protocols' APIs are intermittently slow.

---

7. Future Development & Extensibility

This **high-level design** is deliberately modular. Adding new protocols or advanced features—such as an in-house on-chain indexer, yield farming integrations, or advanced analytics—would involve creating additional services in the backend and new UI components in

the frontend. The same architecture (APIs + DB + wallet signing via MeshJS) can easily scale to support a broader ecosystem.

---

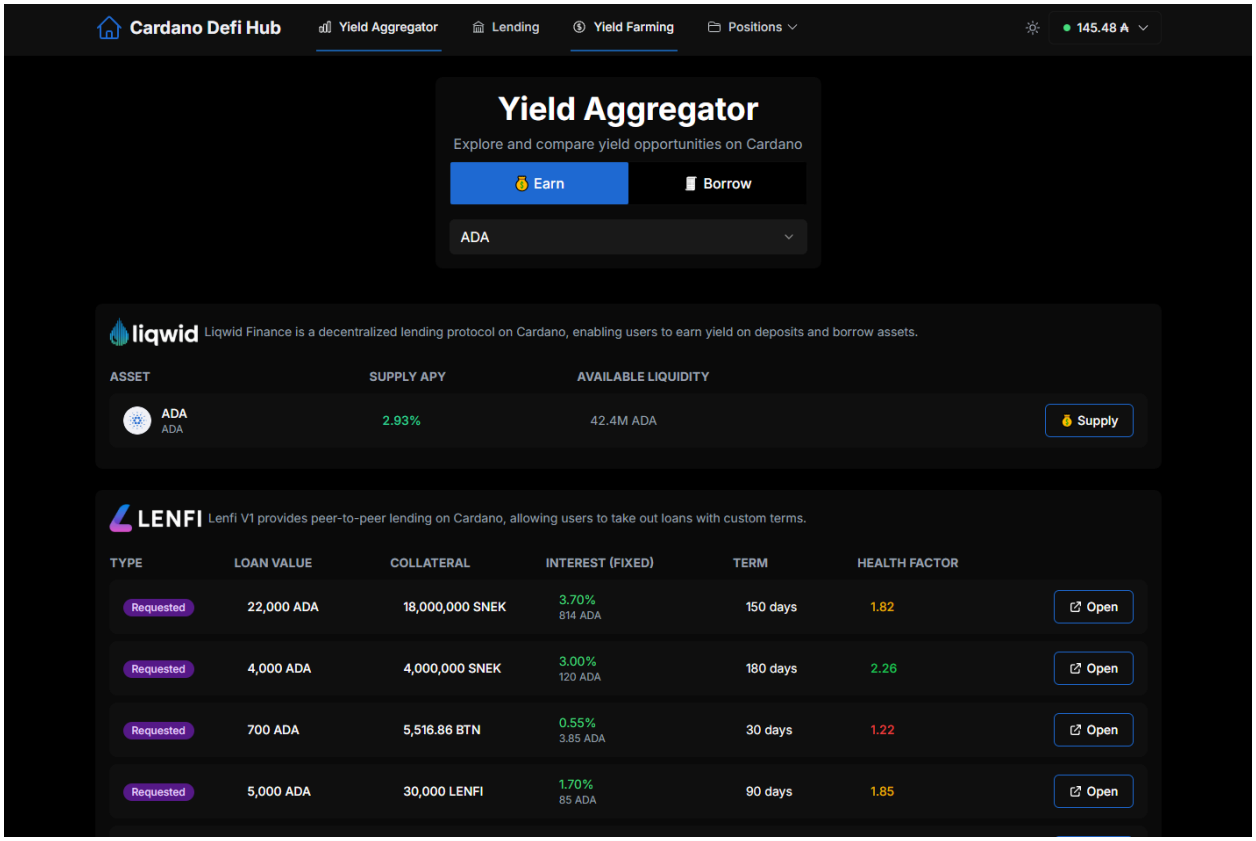
8. Conclusion

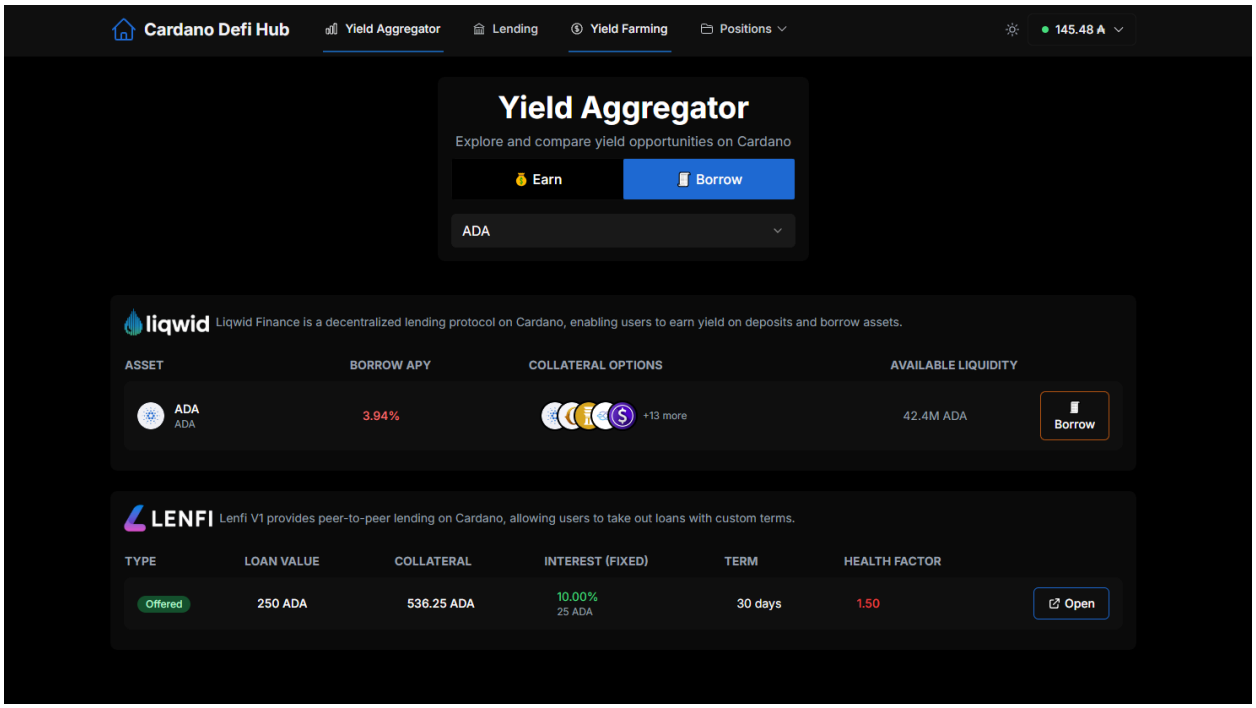
The Cardano Lending Aggregator provides a consolidated, user-friendly platform for exploring and engaging with Cardano lending protocols. By combining:

- **API-based data retrieval** from Liqwid and Lenfi
- **Frequent database updates** for near-real-time yield aggregator displays
- **MeshJS wallet integration** for secure transaction signing

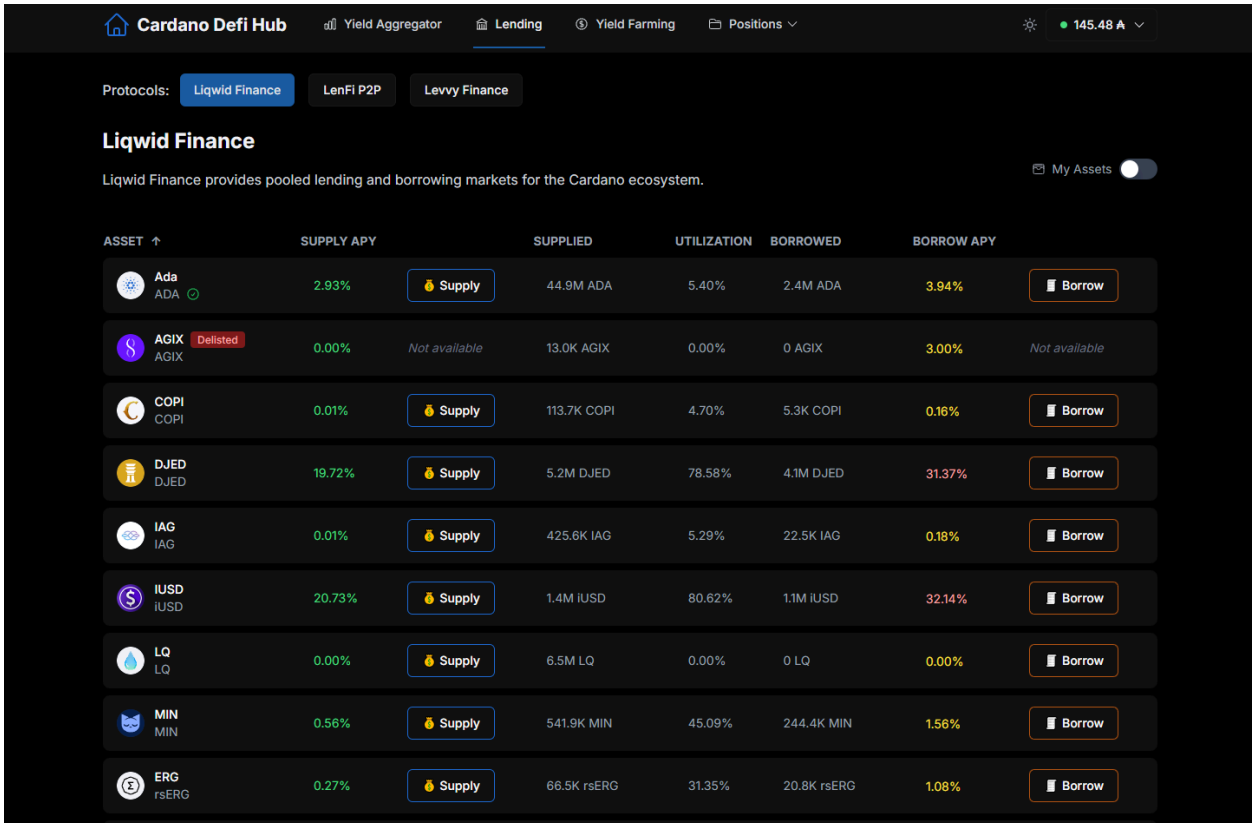
We are delivering a simple, straightforward experience for DeFi users. This high-level design meets the initial Catalyst milestone requirements and positions us to expand into more protocols and features in subsequent phases.

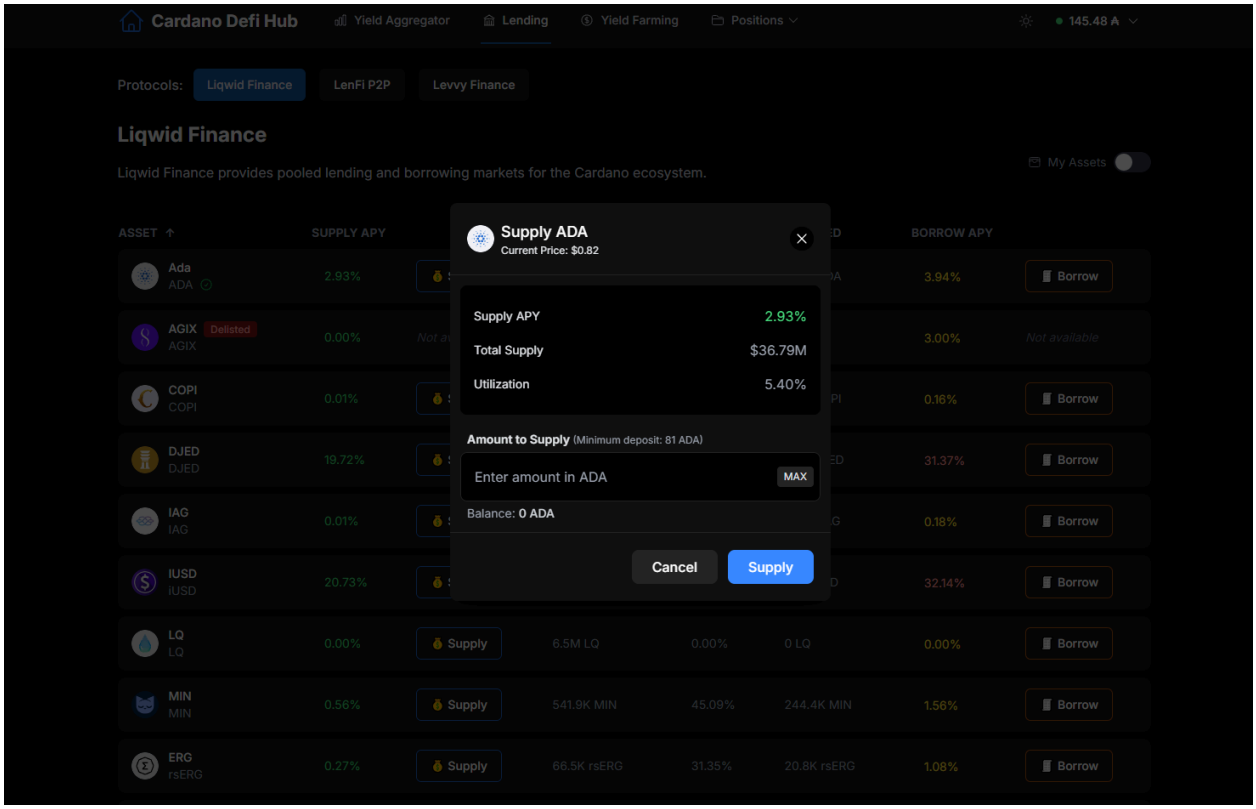
Screenshot of the working yield aggregator page

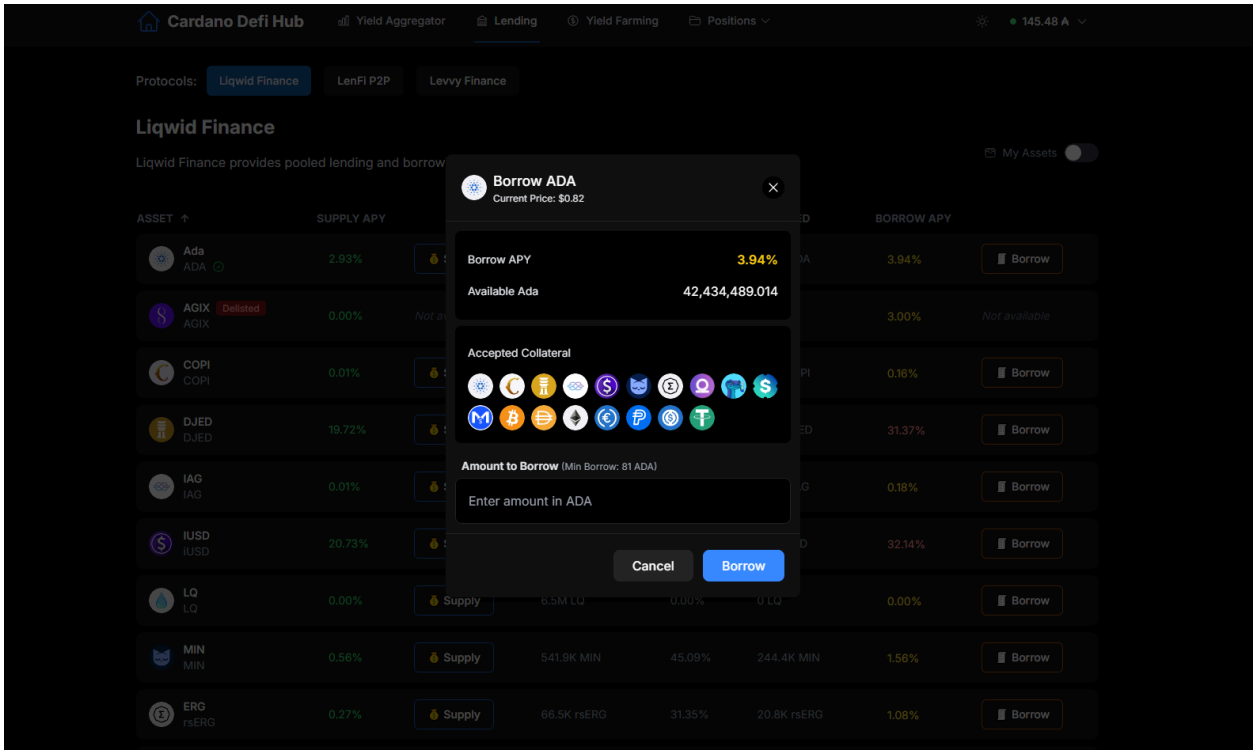


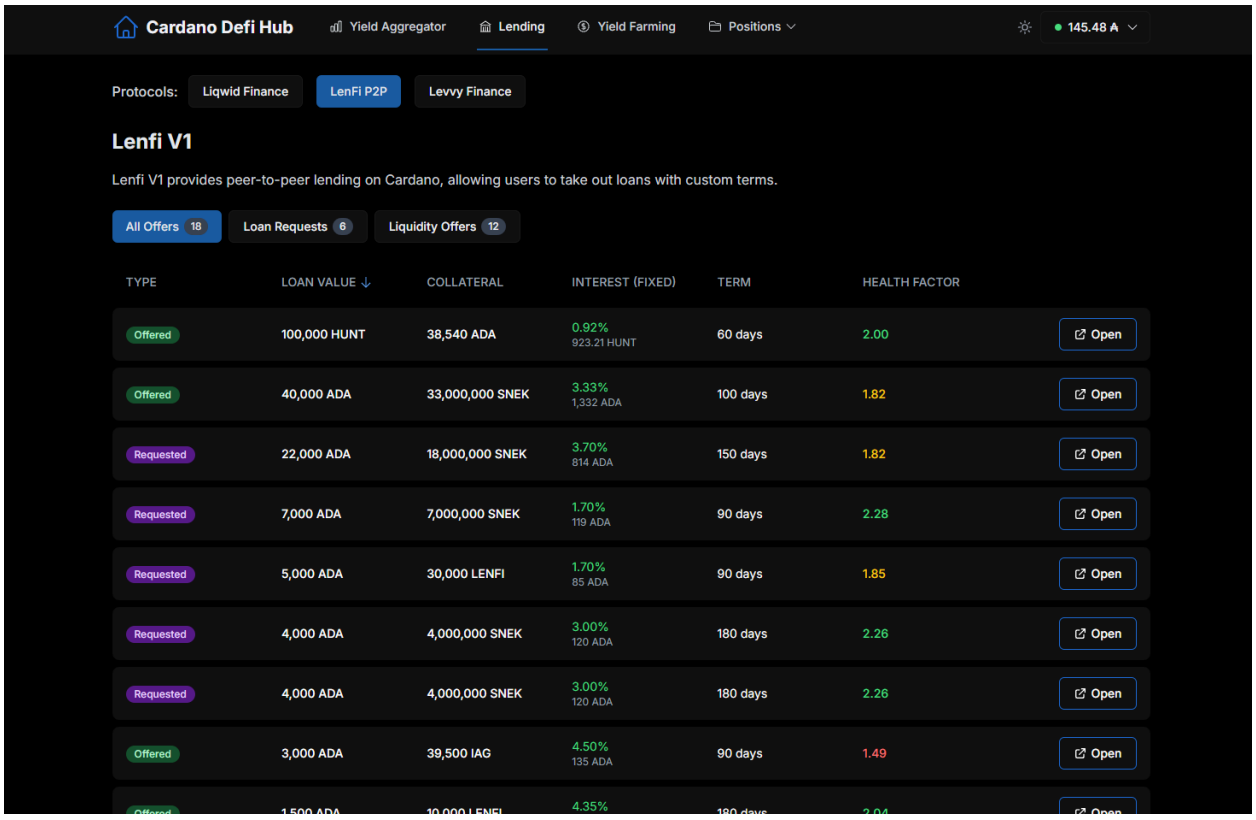
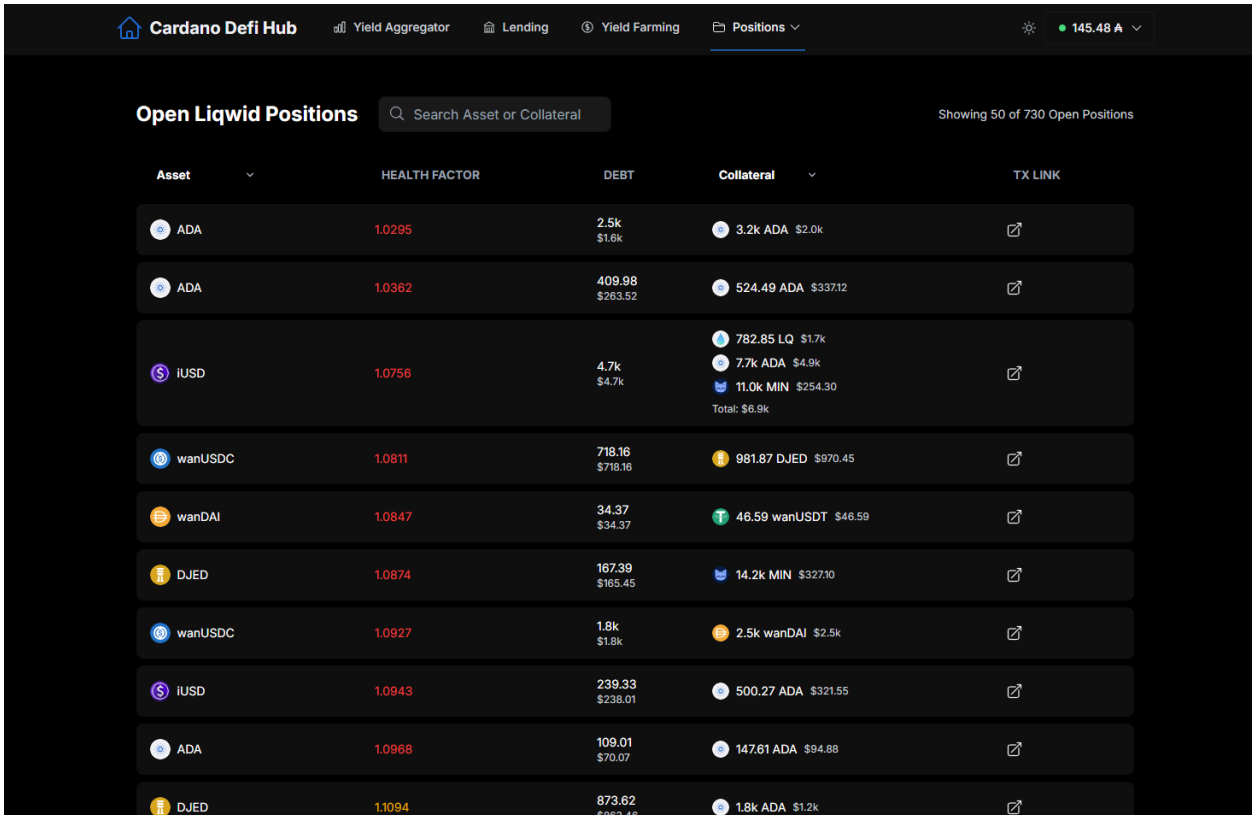


Pages for each individual protocol






Cardano Defi Hub
Yield Aggregator
Lending
Yield Farming
Positions
145.48 ₳
Protocols:
Liqwid Finance
LenFi P2P
Levvy Finance
Liqwid Finance
Liqwid Finance provides pooled lending and borrow
My Assets
ASSET
SUPPLY APY
BORROW APY
Ada
ADA
2.93%
3.94%
Borrow
AGIX
Delisted
AGIX
0.00%
3.00%
Not available
COPI
COPI
0.01%
0.16%
Borrow
DJED
DJED
19.72%
31.37%
Borrow
IAG
IAG
0.01%
0.18%
Borrow
IUSD
iUSD
20.73%
32.14%
Borrow
LQ
LQ
0.00%
Supply
0.00%
Borrow
MIN
MIN
0.56%
Supply
541.9K MIN
45.09%
244.4K MIN
1.56%
Borrow
ERG
rsERG
0.27%
Supply
66.5K rsERG
31.35%
20.8K rsERG
1.08%
Borrow
Borrow ADA
Current Price: $0.82
Borrow APY
3.94%
Available Ada
42,434,489.014
Accepted Collateral
Amount to Borrow (Min Borrow: 81 ADA)
Enter amount in ADA
Cancel
Borrow


Cardano Defi Hub
Yield Aggregator
Lending
Yield Farming
Positions
145.48 ₳
Protocols:
Liqwid Finance
LenFi P2P
Levvy Finance
Lenfi V1
Lenfi V1 provides peer-to-peer lending on Cardano, allowing users to take out loans with custom terms.
All Offers 18
Loan Requests 6
Liquidity Offers 12
TYPE
LOAN VALUE
COLLATERAL
INTEREST (FIXED)
TERM
HEALTH FACTOR
Offered
100,000 HUNT
38,540 ADA
0.92%
923.21 HUNT
60 days
2.00
Open
Offered
40,000 ADA
33,000,000 SNEK
3.33%
1,332 ADA
100 days
1.82
Open
Requested
22,000 ADA
18,000,000 SNEK
3.70%
814 ADA
150 days
1.82
Open
Requested
7,000 ADA
7,000,000 SNEK
1.70%
119 ADA
90 days
2.28
Open
Requested
5,000 ADA
30,000 LENFI
1.70%
85 ADA
90 days
1.85
Open
Requested
4,000 ADA
4,000,000 SNEK
3.00%
120 ADA
180 days
2.26
Open
Requested
4,000 ADA
4,000,000 SNEK
3.00%
120 ADA
180 days
2.26
Open
Offered
3,000 ADA
39,500 IAG
4.50%
135 ADA
90 days
1.49
Open
4.35%

Liqwid Finance open positions