Aergo Decentralized Exchange

Yum Protocol

Introduction

The blockchain enables the creation of decentralized communities built around concepts like fungibility and the transfer of value. For these activities to thrive, protocols that facilitate the seamless exchange of digital assets in a secure and efficient manner are essential.

Without a protocol (or a set of them) that allows for trustless and decentralized asset swaps, users would be forced to engage in OTC or P2P trades, which poses a significant risk to the safety and integrity of all participants.

AERGO, with its low operational costs and high speed, offers both regular and advanced users a unique opportunity to manage real and synthetic assets in ways that few platforms can match. To meet this growing need, Yum Protocol emerges, poised to become the backbone of decentralized exchange on the AERGO network.

Design concepts

To get the protocol going, a traditional and well tested Automatic Market Making implementation would be deployed.

How Does a Traditional AMM Model Work?

Automated Market Makers (AMMs) are decentralized exchange mechanisms that facilitate the trading of digital assets without the need for an order book. Instead of having buyers and sellers placing orders, AMMs use a mathematical formula to determine the price of assets on the platform.

The formula commonly used is: $x \cdot y = k$

Where:

- **x** stands for the quantity of the first token in the liquidity pool (e.g., Token A).

- **y** stands for the quantity of the second token in the pool (e.g., Token B).

- **k** is a constant that stays the same before and after every transaction (known as the "constant product").

In an AMM model, users can trade assets by adding or removing tokens from a liquidity pool. When someone trades tokens in the pool, the quantities of tokens **x** and **y** adjust, which in turn changes the price. This is because the product **k** must always remain constant, thus creating a price shift based on the new ratio of the assets.

Liquidity and LP Tokens

Liquidity in AMMs comes from users who want to earn a yield on their assets by providing them to the liquidity pool. In return for contributing liquidity, users earn a portion of the transaction fees generated by trades in that pool. This incentive encourages more users to add liquidity, which enhances the trading experience by reducing slippage and providing better price stability.

When users provide liquidity, they receive a Liquidity Provider (LP) token that represents their share of the pool. The LP token serves as a deposit receipt and entitles the holder to withdraw their proportionate share of the assets in the pool at any time, along with a share of the gained fees.

How Are LP Tokens Calculated?

Liquidity Provider (LP) tokens represent a user's share of the liquidity pool. When a user deposits tokens into a pool, they get LP tokens as a receipt, indicating their contribution to the total liquidity. These LP tokens entitle users to withdraw their share from the pool, along with a portion of the trading fees collected.

LP Token Calculation Process:

1. **First-Time Liquidity Addition (Empty Pool):**

   - If the pool is empty, the first person to add liquidity sets the initial ratio between the two tokens.

   - For instance, if they add 100 Token A and 200 Token B, they might receive a fixed number of LP tokens (say, 1000 LP tokens) representing their 100% ownership of the pool.

2. **Adding Liquidity to an Existing Pool:**

   - When adding liquidity to a pool with existing tokens, the amount of LP tokens given is proportional to the user's contribution relative to the current liquidity.

   - For example, if the pool has 1000 Token A and 2000 Token B, and a user adds 100 Token A and 200 Token B (increasing the total liquidity by 10%), they will receive 10% of the total LP tokens in circulation.

3. General Formula for Issuing LP Tokens:

   - The amount of LP tokens issued can be calculated as:

   LP tokens received = (tokens added / total liquidity) * total LP tokens in circulation

   - This ensures that the LP tokens accurately reflect the user's share of the pool.

Why Are LP Tokens Important?

LP tokens allow liquidity providers to withdraw their share of the pool at any time. Additionally, they enable providers to earn a portion of the fees generated by trades in the pool. Each time a trade occurs, a small fee (e.g., 0.3%) is collected and distributed to liquidity providers in proportion to their pool share.

This system aligns rewards with the amount of liquidity each user provides, ensuring that the token ratios in the pool remain consistent as users add or remove liquidity.

Frontend for Yum Protocol

The frontend for Yum Protocol will be designed to be as lightweight and straightforward as possible, ensuring a smooth user experience and accessibility from anywhere. The interface will allow users to execute transactions seamlessly, providing essential functionalities such as token swapping, liquidity provision, and portfolio management.

To enhance decentralization and resilience, the frontend will be deployable on decentralized storage solutions such as IPFS, ensuring that the application stays accessible even if traditional web hosting services are unavailable. This approach guarantees a robust, decentralized, and user-friendly experience, aligning with the core principles of blockchain technology.

Technical implementation

Yum Protocol will consist of three main smart contracts: **Factory Contract**, **Pair Contract**, and **Router Contract**. A **front-end** will be provided for the users to interact with these contracts. Below is a description of the role and functioning of each:

1. Factory Contract

The Factory Contract is the core of managing trading pairs and liquidity. Its functions include:

- **Creating token pairs:** It allows the creation of new token pairs (e.g., AERGO/USDT). Each pair will have its own liquidity contract (Pair Contract), which will be managed by the Factory.
- **Registering pairs:** Once a pair is created, it is stored in the Factory, allowing users to check which pairs are available on the DEX.
- **Permission control:** The Factory ensures that each token pair has a unique liquidity contract and prevents the creation of duplicates.

The Factory Contract acts as the “central administrator” that keeps a record of all available pairs in Yum Protocol, facilitating the creation and management of liquidity pools.

2. Pair Contract

The Pair Contract stands for each individual liquidity pool. It handles the central logic of the swap and liquidity provision. Its functions include:

- **Managing token reserves:** The Pair Contract keeps two token reserves (for example, token A and token B). These reserves are updated each time a user swaps tokens or adds/removes liquidity.
- **AMM formula (x * y = k):** The Pair uses the constant product formula to determine the exchange rates between the tokens. Here, x and y represent the amounts of the two tokens in the pool, and k is a constant value that does not change after an operation.
- **Issuing LP tokens (Liquidity Provider tokens):** When a user adds liquidity to the pool, the contract issues LP tokens proportional to the amount added. These LP tokens represent the user's share in the pool and can be used to claim their portion of the reserves plus the accumulated fees.
- **Calculating fees:** The Pair Contract charges a small fee on each transaction. Part of this fee accumulates in the pool to reward liquidity providers.

3. Router Contract

The Router Contract is the main interface for users and simplifies interactions with the other contracts. Its functions include:

- **Token swapping:** It helps the exchange of one token for another by interacting with the Pair Contracts to execute the operation under the AMM formula.
- **Adding and removing liquidity:** It provides functions for users to add or remove liquidity from a pool. The Router interacts with the Pair Contracts to manage the reserves and issue/burn LP tokens.
- **Support for multiple swaps:** It allows for complex swaps that involve multiple pairs, helping exchange routes through different pools to optimize exchange rates.