HiFive: A Multi-chain DEX Supporting a Hierarchical Transaction Architecture Order Model.

1. Overview

HiFive is a high-performance, scalable multi-chain DEX built on Layer2. It supports spot trading and perpetual contract trading, adopting the "off-chain matching, on-chain settlement" order book model.

HiFive aims to design a forward-looking trading model, targeting the development direction of Layer2 and its users, rather than the current space. By promoting the order matching paradigm, it strives to enhance the trading experience and security, offering a superior price discovery mechanism, and addressing the inherent flaws of current AMM trading, such as excessive slippage and insufficient liquidity.

HiFive's vision is to create a trading platform that perfectly balances decentralization and user experience. We believe that, with the widespread adoption of Web3 technology, decentralized exchanges will become the mainstream in the future. HiFive is committed to being a pioneer in this transformation, with the goal of becoming a significant DEX in the zkSync ecosystem. It plans to expand to Layer2 networks like Arbitrum, Polygon, etc., achieving multi-chain deployment in the future.

2. DEX Development Trends

Trading lies at the heart of the DeFi realm, serving not only as the hub for cryptocurrency transactions but also as the foundational pillar for smart contracts, tokens, and a myriad of decentralized applications. Presently, decentralized exchanges (DEX) are progressively becoming mainstream in the Web3 universe. Bolstered by blockchain technology, they have adeptly addressed issues concerning intermediary trust and eliminated single points of failure. As a notable representative, Uniswap, since its establishment in November 2018, recently recorded an accumulated trading volume surpassing 600 billion USD as of May 2023.

Emerging blockchain networks are either launching their own DEX platforms or encouraging third-party developers to construct DEXs atop their frameworks, aiming to lure a broader user base and more substantial capital inflow. DEXs not only amplify the ecosystem of these blockchains but also afford users a richer palette of trading options coupled with enhanced liquidity. With the advancements in cross-chain technology, DEXs have evolved beyond being confined to a singular blockchain, initiating support for multi-chain asset trading, thereby reinforcing their stature in the entire Web3 landscape. Looking forward, we believe that, buoyed by further innovations and technological advancements, DEXs will persist in expanding their influence, furnishing the decentralized financial ecosystem with more refined and diverse services.

3. Evolution of Trading Models

Traditionally, order books have been the most popular trading method on centralized exchanges.

An order book, a list of buy and sell asset orders sorted by price levels, provides traders with an intuitive view of the supply and demand for assets. It's the most straightforward way to depict asset supply and demand. With real-time changes in the financial market, the flexibility of the order book ensures that traders can adjust their positions in a timely manner.

In recent years, the burgeoning Ethereum trading ecosystem has necessitated decentralized trading. A trading mode called "on-chain order matching" was designed. Early platforms like ethfinex, district0x, Augur, and melonport adopted this method. "On-chain order matching" moves the order book to the blockchain, placing one or more of the four major steps – asset custody, order book maintenance, trade matching, and clearing and settlement – on-chain. However, because trade quotes are hosted on-chain, this mode means that placing and canceling orders both incur a transaction fee. Coupled with inadequate on-chain liquidity, "on-chain order matching" has not been widely continued.

Subsequently, an alternative to on-chain order books known as Automated Market Makers (AMM) was proposed. AMMs set prices based on a constant function algorithm for liquidity pools, aiming to address the inherent flaws of the order matching mode.

Despite being constrained by Ethereum's infrastructure, the AMM model still heavily depends on experienced traders to ensure that liquidity pools align with market prices. Its inherent limitations in dynamic price discovery and arbitrage activities often result in lower returns for liquidity providers. For trading users, issues such as slippage and Miner Extractable Value (MEV) have persistently plagued swaps in AMM DEXs, with high transaction costs diminishing the desire to trade. Furthermore, mainstream DEXs cannot directly support limit order functions, restricting users to swap at current prices without the ability to place orders or utilize more complex trading strategies.

The introduction of Ethereum's Layer 2 solutions presents a remedy for the bottleneck of on-chain settlement. Layer 2 technology primarily leverages off-chain computations while ensuring on-chain data integrity and security, facilitating efficient, low-cost transactions. With technological advancements and growing market demand, the "off-chain order matching, on-chain settlement" trading mode based on Layer 2 has gained increasing ecosystem support.

Recently, companies like Base, ZkSync Era, and Sei have proposed protocols supporting Limit Order Books (LOB). Base aims to establish a new generation decentralized exchange supported by the traditional order book familiar to traditional exchanges. They have publicly invited LOB DEX developers to join their ecosystem. From 2023 onwards, several teams are building LOB-based DEXs on Era, including Dexchange, PerpDEX, Izumi, Mes Protocol, and Clober. For the young HiFive team, even before the launch of zkSync Era 2.0, they had already constructed an order-based trading model on Era, wherein the on-chain component utilizes zkRollup and the off-chain core is a grid scheduler.

4. HiFive's Order Model

To be more specific, the well-understood order model can be divided into two types: the pure on-chain matching + settlement mode and the off-chain matching + on-chain settlement mode. In

the pure on-chain matching and settlement mode, user's placed and taken orders are directly on-chain. When an order is taken, it directly trades with the placed order on the blockchain. A representative of this model is EtherDelta. Its advantages include complete on-chain operations and a high degree of decentralization. However, its disadvantages are very low trading performance and high trading costs. Users need to pay gas fees for placing and canceling orders.

HiFive adopts the off-chain matching + on-chain settlement mode. Compared to the first model, it introduces an off-chain grid scheduler role. Users generate order forms through off-chain signatures and submit them to the inline interface of the grid scheduler. The grid scheduler is responsible for maintaining the Orderbook. Once orders are successfully matched, they are submitted to the blockchain for settlement via the grid scheduler's external interface. The advantages of this model include:

- High Efficiency: By moving the matching process off-chain and adopting a Layer 2 architecture for settlements, there is reduced confirmation time, which greatly enhances transaction performance.
- Low Transaction Costs: Users only need to pay a minimal gas fee for their entire trading behavior. Placing and canceling orders do not require additional fees. It supports the BOT order splitting and filling mechanism, resulting in almost zero slippage.
- Security: The off-chain order matching can effectively prevent arbitrage risks where bots manipulate gas fees for front-running trades. Orders are matched based on the "first come, first served" principle. Orders that arrive at the matching engine first are given priority in matching.

5. Product Features

HiFive offers functionalities including spot trading, limit orders, margin contracts, and perpetual contracts.

● Spot market single transaction:

- Directly exchange 150 base assets such as BTC, ETC, BNB, etc.

- Instant settlement.

- High-speed matching engine ensures fast transaction execution.

- Fee: 0.4%.

● Limit Order

- Users can set their desired purchase or sale price.

- The order will only be executed when the market price reaches the user-set limit price.

- Supports partial transaction functionality; the remaining unexecuted portion will continue to be listed.

- Fee: 0.4%.

● Margin Contract

- Allows users to trade by depositing a certain proportion of margin, increasing investment leverage.

- Provides automated risk management features, such as automatic position reduction and forced liquidation mechanisms.

- Supports trading between cross-leverage assets.

- Fee: Contract limit order 0.07%. Contract market order 0.15%.

- Perpetual Contract

- Users buy or sell based on their anticipated market direction.

- Ensures the contract price is closely pegged to the spot price through a funding rate mechanism.

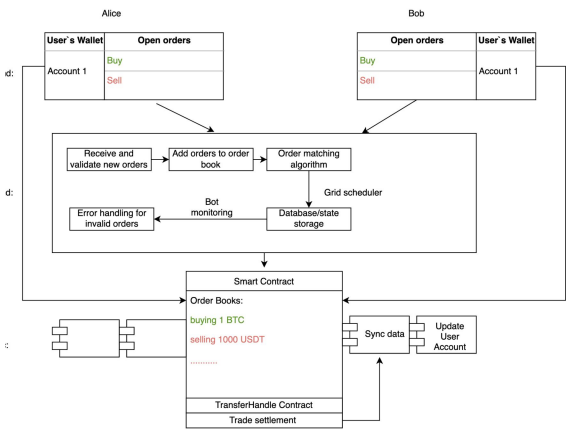
- Supports up to 20x leverage.

- Fee: Contract limit order 0.07%. Contract market order 0.15%.

6.Specification: Protocol Features

6.1 WorkFlow

The following outlines the general processing flow of HiFive's off-chain matching and on-chain settlement.



HiFive Architecture Diagram

1. The system establishes a grid scheduler for a specific trading pair (e.g., ETH/USDT).

2. The system receives ETH/USDT quotes (from 3 oracle sources).

3. The system initiates the market-making Bot monitoring mechanism.

4. Users sign and authorize the grid scheduler.

5. Users deposit.

6. Users initiate orders (check balance with CheckAsset) and access the grid scheduler.

7. The grid scheduler matches buy and sell parties in terms of volume and price (BOT is introduced).

8. After a successful match, the grid scheduler communicates the trading status to zksync.

9. Zksync executes the transfer.

10. Zksync sends the transaction information to the main network for settlement.

11. The transaction is finally settled, and the status is updated.

6.2 Transaction Message Format

Each order contains transaction parameters and signed data messages. The trading message format for HiFive is as follows:

Name	Data Type	Description
version	address	Address of the Exchange smart contract This address will change each time the protocol is updated
maker	address	Address originating the order
taker	address	Address permitted to fill the order
tokenA	address	Address of an ERC20 Token contract.
tokenB	address	Address of an ERC20 Token contract.
ValueA	uint256	Total uits of tokenA offered by maker.
valueB	uint256	Total units of tokenB requested by maker.
expiration	nuint256	Time at which the order expires (seconds since unix epoch).

1. version :Address of the Exchange smart contract. This address changes every time the protocol is updated.

2. maker :Address originating the order. Address of the maker.

3. taker : Address permitted to fill the order. Address that is allowed to execute the order.

4. tokenA : Address of an ERC20 Token contract.

5. tokenB : Address of an ERC20 Token contract.

6. valueA : Total units of tokenA offered by maker.

7. valueB : Total units of tokenB requested by maker.

8. expiration: Time at which the order expires (seconds since the unix epoch).

Packet information in HiFive message format, JSON data structure (example)

```
{
   'order": {
      "version": "01a2b3c4d5e6f... ",
      "maker":  "0x7a8b9c0d1e2f... ",
      "taker":  "0x3c4d5e6f7a8b... ",
      "tokenA": {
         "address":"05e6f7a8b9cOd...",
         "value":1000
      },
      'tokenB": {
         "address": "0x9c0d1e2f3c4d..."
         "vaue": 500
      },
      "expixation":1677645600
   }
}
```

6.3 Market making mechanism

In HiFive, the order processing mechanism involves interactions between market makers (liquidity providers) and users. Specifically, the market maker mechanism plays a crucial role in the context of perpetual contracts and margin contracts. The HiFive market maker mechanism facilitates asset trading by providing buy and sell prices, executing transactions programmatically. Users do not directly trade with each other; this is similar to the mechanism in stock trading.

The main purpose of introducing market makers is to improve capital efficiency. If users quote each other, it may lead to a large bid-ask spread in the market, resulting in dispersed liquidity. To avoid this situation, the HiFive platform introduced the role of market makers, with market-making robots providing tight quotes

to keep the order price range effective, thereby reducing the spread.

For perpetual contracts, which are unique contracts without expiration and settlement dates, continuous liquidity is required to ensure the smooth operation of the market. Market makers from HiFive play a crucial role here. They not only provide continuous bid and ask prices for perpetual contracts, but also ensure that the market can maintain its liquidity and stability even in cases of high leverage.

HiFive's market makers, following a programmatic approach, break down larger orders into smaller and more manageable sizes to facilitate easier executions and reduce slippage (the difference between the executed price and the expected price). HiFive market makers employ an order splitting algorithm to decompose large trade orders into multiple smaller sub-orders. This is done to enhance liquidity, optimize order book matching, and reduce slippage, thereby narrowing the gap between the transaction execution price and the user's expected price. This strategy helps in enhancing the price stability of the market and offers users a superior trade execution quality.

HiFive order splitting algorithm

O Represents a large order.

- V_O Number of sub-orders O.
- n Number of split sub-ordersr i.
- v_i rading volume for each sub-order i.
- $P_{expected}$ Expected trading price
- · $P_{actual,i}$ Actual transaction price for the sub-order i.
- S as a Point, Defined as $P_{actual,i} - P_{expected}$

HiFive The order splitting algorithm of HiFive market maker can be described as follows:

1.When receiving a large order O, First, determine V_O.

2.To determine the appropriate number of sub-orders based on market depth and liquidity parameters n.

3. $i = 1$ TO n, Assigning a trading volume to each sub-order. As v_i.

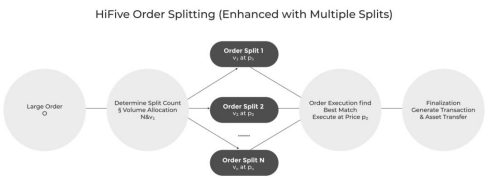
4.For each sub-order, find the best match in the order book and execute at the closest available price $P_{expected}$.

5.Calculate the actual execution price. $P_{actual,i}$ and Slippage S.

6.By optimizing the matching and execution strategies of sub-orders, aim to minimize slippage as much as possible, thereby reducing overall slippage.

7.When a matching order is found, the exchange will generate a trade and settle it on the blockchain.

8 After settlement is complete, the assets will be transferred.



Market Maker Bot Configuration： The JSON configuration file provides detailed setting options for HiFive's market-making robot, typically including the following settings:

- cryptowatchApiKey: Cryptowatch API key used for obtaining real-time market data.

- ethPrivKeys: Array of Ethereum private keys used for relaying transactions and market-making.

- ChainId: The ID of the HiFive blockchain.

- WsUrl: WebSocket URL.

- ethereumRPC: Ethereum RPC address used for interaction with the Ethereum blockchain.

Currency pair settings:

In the configuration file, the settings for each trading pair are represented by a JSON object. Here's an explanation of each field:

- priceFeedPrimary: The primary price feed specified by its Cryptowatch ID.

- priceFeedSecondary: The secondary price feed specified by its Cryptowatch ID. Some trading pairs may not have a secondary price feed.

- slippageRate: The allowable slippage rate, expressed as the maximum permissible difference between the execution price and the expected price.

- maxSize: The maximum trading size for the trading pair.

- minSize: The minimum trading size for the trading pair.

- minSpread: The minimum spread for the trading pair.

- active: Indicates whether the trading pair is active. If set to true, the market maker will operate on this trading pair.

Trading pair settings:

Here is an example of the settings for the ETH-USDC trading pair:

Primary price source: Cryptowatch: 6631

Secondary price source: Cryptowatch: 588

Slippage rate: $1\times10-5$

Maximum trading size: 100

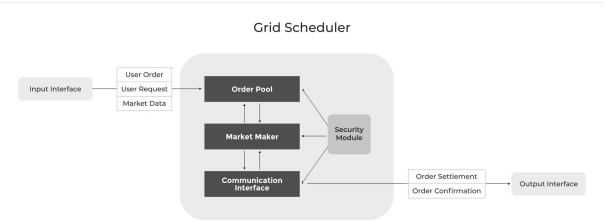
Minimum trading size: 0.0003

Minimum spread: 0.0005

Activation status: true

6.4 Grid Scheduler

HiFive's grid scheduler serves as an off-chain aggregation and matching system. Its primary function is to gather, organize, and optimize user orders, and then interact with market-making robots and other off-chain resources to provide the best possible trading conditions.



Input Interface:

- Collects trade requests from users.

- Provides real-time market data to users, such as the latest buy/sell quotes, order depth, trade volume, etc.

Order Pool:

- Stores pending HiFive user orders.

- Categorizes HiFive orders, such as by asset type, trade volume, expected trade price, etc.

Order Matching and Optimization Engine:

- Aggregates similar orders.

- Utilizes a greedy algorithm to find the best matches between buyers and sellers to minimize slippage and enhance trade speed.

- Dynamically adjusts order priorities based on market depth and market maker quotes.

Interface with Market-Making Robots:

- Transmits optimized orders to market-making robots for execution.

- Gathers real-time quotes from market-making robots and updates the order matching and optimization engine based on these quotes.

Communication Module:

- Utilizes WebSocket, a secure and efficient protocol, to engage in real-time communication with users and market-making robots.

- Provides order status updates, such as executed, partially executed, pending, etc.

Security Module:

- Validates all incoming orders to prevent malicious orders or attacks.

- Employs TLS encryption technology to ensure secure communication with users and market-making robots.

Output Interface:

- Once orders are fully matched off-chain, updates their status to "pending settlement" and sends them to the blockchain for final settlement.

- Provides users with trade confirmations and receipts.

The workflow of HiFive grid scheduler is as follows:

Users send trade requests to the grid scheduler through the input interface.

The scheduler stores orders in the order pool and initiates processing using the order matching and optimization engine.

The scheduler interacts with market-making robots, fetching real-time quotes and further optimizing order matching based on these quotes.

Once matching buyers and sellers are found, the scheduler sends the trade details to market-making robots for execution.

After the trade is completed off-chain, the scheduler updates the order status to "pending settlement" and sends it to the blockchain for final confirmation and settlement.

Users receive trade confirmations and receipts through the output interface.

6.5 Signature authorization

In Fifive, not only is it necessary to verify the source of an order, but it is also crucial to ensure that the order maker has authorized the exchange smart contract to transfer tokens on their behalf. Fifive accomplishes this through the use of the Metaapprove function. The authorization process involves the following steps:

```solidity
tokenContract.approve(exchangeAddress, amount);
```

Token Approval：

The maker first needs to call the approve function of the token contract, granting permission for the smart contract to transfer their tokens on their behalf.

Order Placement:
Once authorized, the maker can place an order, specifying the quantity of tokens they want to sell.

Order Execution:

When the other party (Taker) wishes to execute the order, the HiFive smart contract first checks whether sufficient authorization has been obtained. This verification is accomplished by checking the allowance function.

```solidity
uint256 allowed = tokenContract.allowance(maker, exchangeAddress);
if (allowed < orderAmount) throw;
```

uint256 allowed = tokenContract.allowance(maker, exchangeAddress);

If the authorization is sufficient, the HiFive smart contract will invoke the “transferFrom” function to actually transfer the tokens.

6.6 On-chain order processing (Layer2).

After the matched trades are summarized, the grid scheduler will generate a zkSNARK proof, which is sent to the zkSync ecosystem. This zkSNARK proof contains the following information:

Order Details:
- Addresses of the buyer and seller: Used to verify the identities of both parties in the transaction.
- Transaction amount: The specific quantity of the trade between the buyer and seller.
- Type of assets being traded: For example, ETH, DAI, USDC, etc.

Order Status:
- Whether it has been fully matched or partially matched.
- If partially matched, how much remains unmatched.

Price and Timestamp:
- The execution price of the transaction.
- Timestamps for order submission and matching to ensure timely - processing of orders.

Leverage and Margin Information (if applicable to leveraged trading):
- Leverage ratio.
- Amount of margin.
- Any fees or adjustments related to the margin.

Signatures and Public Keys:
- Signatures of the buyer and seller used to verify the authenticity of the order.
- Public keys used for functions like ecrecover to verify the signatures.

Other Possible Metadata:
- Such as the unique ID of the order, identifiers for potential matching engines, and so on.

After receiving a valid zkSNARK proof on zkSync, on-chain pre-settlement will take place as follows:

On-chain submission:

- The ProofVerifierContract contract of zkSync will accept this zkSNARK proof and verify its validity.

- If the proof is valid, the ProofVerifierContract contract will record the results of this transaction (such as asset transfers, updates to account balances, etc.) in the on-chain state.

- Since the on-chain processing only requires handling proofs and results, rather than processing every specific transaction, it significantly reduces on-chain data volume and processing costs.

On-chain Submission and Margin Trading:

- When zkSync contracts receive zkSNARK proofs for margin orders, they not only verify their validity but also examine the logic related to margin.

- During each transaction settlement, the smart contract will first update the state of the on-chain order book and then check the trader's margin situation. If a trader's margin is insufficient, their order may be rejected, or their position may be forcibly liquidated.

```solidity
updateOrderBook(order);
checkMarginRequirements(user);
```

When traders want to increase their margin or adjust their positions, they can directly perform these operations within the on-chain order book.

```solidity
adjustMargin(user, amount);
adjustPosition(user, newPosition);
```

Security and Data Availability:

HiFive's on-chain order book provides traders with a secure, transparent, and real-time trading environment. All transactions and margin information are recorded on the blockchain, which means that even in the event of technical issues or other problems, traders can safely recover their assets.

At the same time, because all information is recorded on the blockchain, zkSNARK technology ensures the authenticity and integrity of this information.

WebSocket API

In HiFive, a WebSocket API interface is provided for real-time price updates, real-time order book updates, and personal trade notifications. The message structure and response

methods have been optimized for WebSocket usage. The basic structure of the WebSocket API interface is as follows: { "op": "operation", "args": ["list", "of", "args"] }

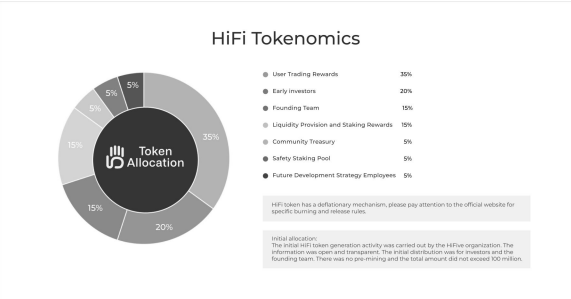
For example, to request the top 50 orders in the order book for BTC to USDT, you can send the following WebSocket message:

{ "op": "getOrderBook", "args": ["BTC", "USDT", "top50"]}

When on-chain transactions occur and result in changes in asset prices, the backend implementation listens to on-chain transaction events. Using existing transaction data and the order book, it calculates new asset prices. Subsequently, through a WebSocket server, the new price data is broadcasted to all connected clients. Order updates:[chainId,orderId,status,txHash,remaining/error]

Based on this, developers can easily construct and parse messages of this structure, ensuring that WebSocket communication is clear and consistent. This structured approach also makes API expansion and maintenance easy.

7.Token Model



HiFive issues the HiFi token with a total supply of 100 million, and it will never be increased.

Distribution of HiFi tokens is as follows:

-User Trading Rewards: 35% of the total HiFi supply.

-Liquidity Provision and Staking Rewards : 15% of the total HiFi supply(5%staking,10% liquidity provision).

-Community Treasury : 5% of the total HiFi supply.

-Safety Staking Pool : 5% of the total HiFi supply.

-Early investors : 20% of the total HiFi supply.

-Founding Team： 15% of the total HiFi supply.

-Future Development Strategy Employees：5% of the total HiFi supply.

HiFi tokens have a deflationary mechanism. Please refer to the official website for specific burn and release rules.

Initial Allocation:

The initial HiFi token generation event is organized by HiFive, with transparent and publicly disclosed information. The initial allocation is for investors and the founding team, with no pre-mining, and the total supply does not exceed 100 million tokens.

Token Allocation Mechanism:

HiFive will utilize token allocation rewards to incentivize early participation by DEX participants and provide ecological funding for the project's ongoing development. These rewards provide additional incentives to entities performing critical network functions, apart from fees. The network benefits greatly from incentivizing early participation by creating a reliable distributed agent network as soon as possible. Additionally, the HiFi burning mechanism will allow HiFive to maintain a stable token supply over time.

Token Locking:

Token locking helps align the incentives of token holders with the long-term growth of the HiFive ecosystem. Most HiFi tokens are subject to locking at the time of issuance.

Governance Functionality:

HiFi tokens are used for standard governance functions, such as chain security and the ability to participate in technical governance by voting to upgrade runtime code in proportion to one's stake in the protocol. HiFi holders participate in governance through an on-chain voting mechanism for proposals: runtime upgrade proposals are voted on by token holders. Approved proposals are programmatically enacted on-chain.

8.Release Planning

HiFive V1 Release: Before the second quarter of 2024

Focus on framework development and feature building, including the implementation of the HiFive order model architecture (including on-chain and off-chain matching platforms); implementation of spot DEX, limit orders, margin contracts, and perpetual contracts.

HiFive V2 Release: Before the second quarter of 2025

Focus on DEX ecosystem development, expanding HiFive's supported blockchains to Layer 2 networks such as Arbitrum and Polygon, achieving multi-chain deployment; customized trading functionalities (HOOK module), and more strategy optimizations.

HiFive V3 Release: First quarter of 2026

A DAO-driven full ecosystem DEX, independently customized DEX blockchain operations; an aggregated DEX ecosystem; dedicated to becoming the infrastructure of the Web3 world.

9.Team

Jean De Dieu MANISHIMWE | CTO

Seasoned Chief Technology Officer with 8 years of experience in the cryptocurrency market, a deep understanding of blockchain technology, and expertise in innovating high-security solutions. Leads project deliveries with a strategic vision and practical approach, earning recognition from clients and stakeholders. Transforms concepts into practical solutions and is committed to driving the rapid development of the industry.

Rwandan internet entrepreneur, Rwibutso Ivan, specializes in promoting digital currencies, particularly Bitcoin. He effectively engages with potential clients through social media, email, and advertising to drive the adoption of digital currencies. He is dedicated to building trust and helping people understand how digital assets integrate into daily life, with a strong belief in the future of cryptocurrency IT marketing.

Patocs Ricsi | COO

Ricsi is a cryptocurrency market consultant with rich experience. Since 2018, he focuses on diverse projects, strengthens communication teams, assists project communities, and actively fosters interaction between companies and token holders. Proficient in blockchain technology and market analysis, he provides strategic guidance, enabling project success and industry leadership.

Rwibutso Ivan | Co-founder

Mukimbili Noah | Co-founder

A top-tier technical advisor with 4 years of extensive experience in the cryptocurrency market, proficient in blockchain technology, and comprehensive cryptocurrency market analysis. Holds an outstanding track record in providing strategic advice and wise decision-making, offering reliable guidance for teams and organizations to navigate with confidence and precision in the fast-paced industry.

Bengeduz Tímár | Technical Adviser

Bengeduz Tímár, a Chief Technology Officer, is a seasoned expert in the crypto market, specializing in blockchain and innovative security solutions. He leads teams to ensure precise project deliveries and leaves a lasting impact. Since 2017, he actively participates in multiple projects, serving as both internal and external consultant, excelling in marketing and sales, and holding various leadership roles.

Horváth Fanni has been immersed in the realm of blockchain technology and the potentials of web3 since 2019. A proficient manager, she excels in orchestrating the team's efforts behind the scenes and adeptly resolving ongoing challenges. Her priority lies in ensuring that those who place their trust in the team reflect upon the project with enduring satisfaction, even as time progresses.

Horváth Fanni | Co-founder