

SYNTHR – Omnicoin synthetic assets

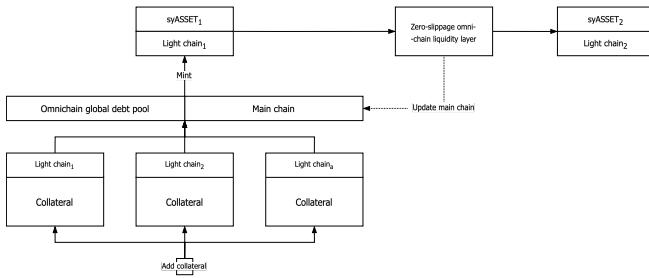
Ritwik Rudra Vithuran Karvannathasan

05/07/2022

Introduction

SYNTHR utilizes omnicoin synthetic assets to power a zero-slipage execution environment. The protocol brings together a combination of pull and push oracles, a zero-slipage omnicoin liquidity layer, and multiple independent consensus layers, providing you with a unique cross-chain solution.

1. Omnicoin global debt pool



The omnicoin global debt pool aggregates cross-chain collateral and debt balances, allowing you to add high-quality liquid collateral on multiple chains and mint omnicoin syASSETS on any chain. This enables the creation of overcollateralized debt positions with decentralized cross-chain solvency. The omnicoin global debt pool acts as the counterparty for all zero-slipage cross-chain swaps.

$$\sum \text{Collateral balance}_a \cup \sum \text{Debt balance}_a$$

> a = User 1, User 2, User 3, ...

C-ratio

$$\frac{\text{Collateral balance}}{\text{Debt balance}} \times 100$$

Collateral balance

$$\sum \$\text{Collateral}_{ab} + \$\text{Liquidation rewards}$$

> a = axlUSDC/USDC, eETH, rsETH, ...

> b = Arbitrum, Avalanche, BNB Chain, ...

Debt balance

$$\sum \$\text{syASSET}_{ab}$$

> a = syAAPL, syAVAX, syBNB, ...

> b = Arbitrum, Avalanche, BNB Chain, ...

A. Architecture

The architecture consists of multiple light chains and a main chain. The main chain exclusively hosts the aggregator contracts, enabling gas-optimized cross-chain synchronicity. This eliminates the need for any off-chain computations to save gas fees and ensures a censorship-resistant framework.

B. Hedge pool

The hedge pool issues hedge pool tokens and swaps its deposits for the current composition of the omnicoin global debt pool, ensuring delta neutrality. It rebalances these hedge pool tokens to capture the evolving composition of the omnicoin global debt pool. This enables you to generate delta-neutral yield from protocol liquidations and protects you from any sharp second-party debt balance volatility.

C. SYNTHR debt shares

Every time you mint omnicoin syASSETS, you generate personal and protocol debt. The omnicoin global debt pool, which represents overall protocol debt, works on the model of debt load sharing. This means that all users are collectively responsible for the protocol's solvency. Your SYNTHR debt shares correspond to your ownership of the omnicoin global debt pool.

Debt percentage

$$\frac{\text{SYNTHR debt shares}}{\sum \text{SYNTHR debt shares}_a}$$

> a = User 1, User 2, User 3, ...

SYNTHR debt shares

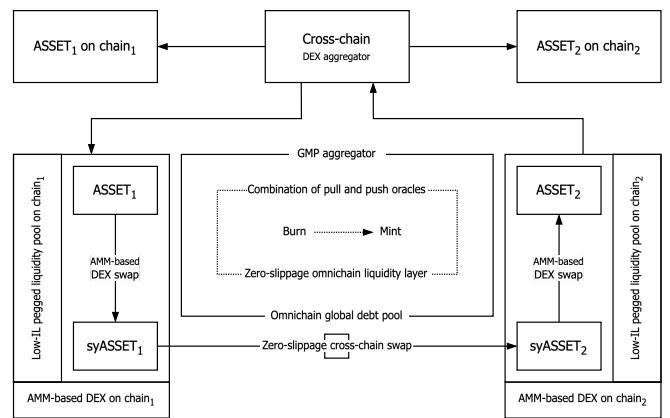
$$\frac{\text{Debt balance}}{\text{Price}_{\text{syUSD}}}$$

> $\text{Price}_{\text{syUSD}} == \1

Explanation

Let's consider an example to understand how SYNTHR debt shares work. Trader A mints 50,000 syUSD worth \$50,000 and receives 50,000 SYNTHR debt shares, while trader B mints 10 syBTC worth \$50,000 and receives 50,000 SYNTHR debt shares. The omnicoin global debt pool, worth \$100,000, consists of 50,000 syUSD, 10 syBTC, and 100,000 SYNTHR debt shares, with traders A and B each owning 50%. Let's say that the price of syBTC doubles, which means that the 10 syBTC are now worth \$100,000 and the omnicoin global debt pool is worth \$150,000, split equally between traders A and B. Trader A can exit the protocol by burning the 50,000 syUSD plus an additional 25,000 syUSD, while trader B can exit the protocol by burning 7.5 out of the 10 syBTC. Now, let's consider a different scenario in which trader A utilizes the hedge pool to ensure delta neutrality. The hedge pool swaps the syUSD for the latest composition of the omnicoin global debt pool, which is 50% syBTC and 50% syUSD. Trader A, who now effectively holds 5 syBTC and 25,000 syUSD, can exit the protocol without any losses by burning them.

2. Omnicoin syASSETS



Omnichain syASSETS enable you to seamlessly move between chains without conventional bridges. The zero-slipage omnicoin liquidity layer utilizes a combination of pull and push oracles to burn and mint omnicoin syASSETS across chains, enabling zero-slipage cross-chain swaps. The combination of oracles ensures price feed accuracy and reliability, while the swaps generate protocol revenue.

Explanation

Cross-chain DEX aggregators utilize the zero-slipage omnichain liquidity layer to perform low-slipage native asset swaps.

Step 1: The cross-chain DEX aggregator utilizes a low-IL pegged liquidity pool on an AMM-based DEX to swap $syASSET_1$ for $syASSET_1$ on chain₁.

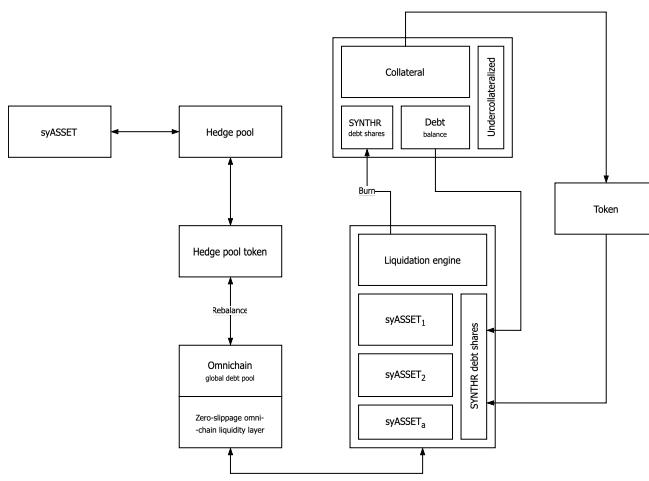
Step 2: The cross-chain DEX aggregator utilizes the zero-slipage liquidity layer to swap $syASSET_1$ on chain₁ for $syASSET_2$ on chain₂.

Step 3: The cross-chain DEX aggregator utilizes a low-IL pegged liquidity pool on an AMM-based DEX to swap $syASSET_2$ for $ASSET_2$ on chain₂.

A. GMP aggregator

The GMP aggregator utilizes multiple independent consensus layers to validate cross-chain messages, ensuring democratic and trustless cross-chain finality. This creates an operational barrier between the core contracts and relayers, preventing collusion between the two.

B. Liquidation engine



The liquidation engine liquidates undercollateralized users, preserving protocol solvency and health. It burns the SYNTHR debt shares of liquidated users and redistributes their collateral and debt balances among remaining SYNTHR debt shareholders. This requires the liquidation engine to carry out cross-chain collateral liquidations and swap them for the protocol's native token before distribution.

Flag for liquidation

$$C - \text{ratio} \leq C - \text{ratio}_{\text{Liquidate}}$$

Liquidate

$$C - \text{ratio} \leq C - \text{ratio}_{\text{Liquidate}} \wedge \text{Time remaining}_{\text{Flag for liquidation}} = 0$$

Liquidation balance

$$\frac{\text{Debt balance} - \text{Collateral balance} \times C - \text{ratio}_{\text{Minimum}}^{-1}}{1 - C - \text{ratio}_{\text{Minimum}}^{-1} \times (1 + \text{Liquidation penalty}_a)} \times (1 + \text{Liquidation penalty}_a)$$

$$> \text{Liquidation penalty} = \begin{cases} b, & a = \text{Self - liquidate} \\ c, & a = \text{Liquidate} \end{cases}$$

>> b, and c = 10%, 20%, 30%, ...

Self-liquidate

$$C - \text{ratio} \leq C - \text{ratio}_{\text{Minimum}}$$

Withdraw liquidation rewards

$$C - \text{ratio} \geq C - \text{ratio}_{\text{Minimum}} \wedge \text{Time remaining}_{\text{Escrow}} = 0$$

C. vesyUSD

Time-lock syUSD to create vesyUSD. This particular vote-escrow model protects you from any value decay at the end of your time lock and provides you with protocol governance privileges, plus bonus airdrops from ecosystem protocols and early access to ecosystem private sales.

$$\text{syUSD} \times \frac{\text{Time - lock}}{4 \text{ years}}$$

$$> \text{Time - lock} = \text{Timestamp}_{\text{Unlock date}} - \text{Timestamp}$$

Real yield

1. Farming rewards

Farm LP tokens to earn farming rewards.

$$\text{Distributable farming rewards} \times \frac{\text{Staked balance}}{\sum \text{Staked balance}_a} + \alpha$$

$$> \alpha = \text{Unclaimed distributable farming rewards}$$

$$>> \text{Stake LP token} V \text{Unstake LP token} \xrightarrow{\text{Assigns}} \alpha$$

$$> a = \text{User 1, User 2, User 3, ...}$$

$$> \text{Distributable farming rewards} = \text{Share of token emissions}_{\text{Per block}}$$

2. Liquidation rewards

The liquidation engine distributes liquidation rewards.

$$\text{SYNTHR debt shares} \times (\alpha_{\infty} - \alpha)$$

$$> \alpha_{\infty} = \alpha_{\infty-1} + \frac{\text{Distributable liquidation rewards}}{\sum \text{SYNTHR debt shares}_a} + \beta$$

$$>> \alpha = \text{Infinitely increasing arbitrary variable}$$

$$>>> \text{Escrow} V \text{Burn} V \text{Mint} \xrightarrow{\text{Assigns}} \alpha_{\infty}$$

$$> \beta = \text{Unclaimed distributable liquidation rewards}$$

$$>> \text{Burn} V \text{Mint} \xrightarrow{\text{Assigns}} \beta$$

$$> a = \text{User 1, User 2, User 3, ...}$$

$$> \text{Distributable liquidation rewards} = \frac{\text{Liquidation balance}}{\text{Price}_{\text{Token}}}$$

3. Minting rewards

Mint omnichain syASSETS to earn minting rewards.

$$\text{SYNTHR debt shares} \times (\alpha_{\infty} - \alpha)$$

$$> \alpha_{\infty} = \alpha_{\infty-1} + \frac{\text{Distributable minting rewards}}{\sum \text{SYNTHR debt shares}_a} + \beta$$

$$>> \alpha = \text{Infinitely increasing arbitrary variable}$$

$$>>> \text{Mint} V \text{Burn} V \text{Withdraw minting rewards} \xrightarrow{\text{Assigns}} \alpha_{\infty}$$

$$> \beta = \text{Unclaimed distributable minting rewards}$$

$$>> \text{Mint} V \text{Burn} \xrightarrow{\text{Assigns}} \beta$$

$$> a = \text{User 1, User 2, User 3, ...}$$

$$> \text{Distributable minting rewards} = \text{Share of protocol revenue}_{\text{Per epoch}}$$

4. vesyUSD rewards

Create vesyUSD to earn vesyUSD rewards.

$$\text{Distributable vesyUSD rewards} \times \frac{\text{vesyUSD}}{\sum \text{vesyUSD}_a}$$

$$> a = \text{User 1, User 2, User 3, ...}$$

$$> \text{Distributable vesyUSD rewards} = (\alpha + \beta)_{\text{Per epoch}} + \gamma$$

$$>> \alpha = \text{Share of token emissions}$$

$$>> \beta = \text{Share of protocol revenue}$$

>> $\gamma = \sum$ Unclaimed distributable vesyUSD rewards_a

>>> a = User 1, User 2, User 3, ...

Revenue distribution

The protocol distributes all of its revenue to its stakeholders based on their contributions, actively fostering a strong sense of community and propagating continued engagement and collaboration among its participants. It also creates a sense of collective long-term alignment.

Revenue generation

1. Core

All zero-slipage swaps directly contribute to protocol revenue.

2. SDKs

SDKs enable one-click deployments of synthetic asset protocols with custom configurations. 30% of revenues from all SDK deployments flow into SYNTTHR, which are either distributed to stakeholders or utilized towards buying back and burning the protocol's native token.

Security

The GMP and oracle aggregators ensure high levels of transaction security and mitigate front-running risks. Comprehensive insurance, periodic bug bounty programs, regular external audits by third-party firms, and rigorous internal audits intensify overall protocol security.

SYNTH utility

1. Collateral

Add the protocol's native token as collateral.

2. Farming rewards

Earn farming rewards in the protocol's native token.

3. Flagger fees

Earn flagger fees in the protocol's native token.

$$\frac{a}{\text{Price}_{\text{Token}}}$$

> a = \$10, \$15, \$20, ...

4. Liquidation rewards

Earn liquidation rewards in the protocol's native token.

5. Liquidator fees

Earn liquidator fees in the protocol's native token.

$$\frac{a}{\text{Price}_{\text{Token}}}$$

> a = \$10, \$15, \$20, ...

6. vesyUSD rewards

Earn vesyUSD rewards in the protocol's native token.

References

Breidenbach, C. Cachin, B. Chan, A. Coventry, S. Ellis, A. Juels, F. Koushanfar, A. Miller, B. Magaarian, D. Moroz, S. Nazarov, A. Topliceanu, F. Tram'er, and F. Zhang, **Chainlink 2.0: Next steps in the evolution of decentralized oracle networks**

H. Lambur, **Uma**: A decentralized financial contract platform

Pyth Data Association, **Pyth Network**: A first-party financial oracle

S. Liu and I. Lee, Mirror Protocol

Synthetix system documentation