Ultimate Synthetic Delta Neutral

Jean Rausis [1], Eric Rabl [2], Blockchain Team (Lilyan Bastien, Valentin Bersier, Yoan Capron, Sami Darnaud, Nicolas Decosterd, Paul-Alexandre Tessier) [3]

[1] Co-CEO jean@ra2.tech
[2] Co-CEO eric@ra2.tech
[3] blockchain@ra2.tech

Abstract

Dollar-backed stablecoins have long been a cornerstone of the decentralized finance (DeFi) ecosystem, but they are subject to the same value depreciation as the US dollar due to inflation. Yield-bearing synthetic dollar tokens have been proposed in the past, but they suffer from centralization and lack of transparency. We present a novel protocol comprised of two DeFi products: an algorithmic yield-bearing synthetic dollar token (USDN) and a decentralized long perpetual futures trading platform. We describe the mathematical principles supporting the protocol and the interaction between both products. This research shows that a fully decentralized and transparent model can be used to support a yield-bearing synthetic dollar token while remaining economically viable and gas-efficient.

Keywords: DeFi; Blockchain; Synthetic Assets; Delta Neutral

Version 1.0 January 2025

1. Introduction

The decentralized finance (DeFi) ecosystem has long sought alternatives to fiat-backed tokens, aiming to provide users with assets that combine dollar-like stability with yield generation. However, existing solutions, particularly stablecoins, suffer from inherent flaws: they are often centralized, opaque, and yield-free for holders.

USDN aims to solve this problem by operating a fully decentralized structured product. Its architecture eliminates dependencies on centralized exchanges (CEXs) and custodial intermediaries. Instead, users interact with smart contracts to mint or redeem USDN tokens, as well as to open long perpetual positions. The underlying asset deposited to mint USDN tokens is used as liquidity for the structured product to enable leveraged trading. The first deployment of this protocol uses the Lido wrapped staked ETH (wstETH) [1] as underlying asset. This means that we combine the yield of the protocol with that of staking ETH automatically.

2. Architecture Overview

The USDN protocol is comprised of two main components: the *vault* and the *long side*, each having a balance of the underlying asset. The sum of these balances does not change unless a deposit or withdrawal is made towards the vault (see Section 3), a new long position is opened or closed (see Section 5), some protocol fees are taken, or liquidation rewards are given out to an actor of the protocol. Any change in the long side balance due to a change in the underlying asset price or funding fees is compensated by an equal but opposite change in the vault balance.

The vault holds the amount of underlying assets necessary to back the value of the USDN token (see Section 3.2). For instance, if the price of the USDN token is \$1 and its total supply is 1000 USDN, and if the price of each wstETH is \$2000, then the vault balance is 0.5 wstETH. Each deposit increases the vault balance and mints new USDN tokens.

The long side holds the amount of underlying assets corresponding to the summed value of all the long perpetual positions that exist currently. For example (ignoring fees), a newly open position with an initial collateral of 1 wstETH would increase the long side balance by 1 wstETH, because the position did not lose or gain any value yet, the asset price being the same as the entry price of the position. If the price of the underlying asset increases, the value of the position increases (with a leverage effect), and a corresponding decrease in the vault balance occurs (see Section 5.1).

When the protocol is balanced, the vault balance is exactly equal to the borrowed amount of the long side (Section 7). To incentivize this equilibrium, the protocol charges a funding fee to the side with the higher trading exposure, and rewards this amount to the other side (see Section 8).

3. USDN Token

3.1. Overview

The USDN token is a synthetic USD token designed to approximate the value of one US dollar while delivering consistent returns to its holders. Unlike a stablecoin, USDN does not claim to maintain a rigid peg to $1. Instead, it oscillates slightly above or below this reference value, supported by market forces and the protocol's innovative mechanisms.

The value of USDN comes from assets (a specific ERC20 token) stored in the protocol's vault. This can be any token for which a price oracle is available and for which the balance does not change over time without transfer. For the first release of the protocol, the token is wstETH.

3.2. Price

Due to the algorithmic nature of the USDN token, its price in dollars p_{usdn} can be calculated using the following formula:

$$p_{\text{usdn}} = \frac{B_{\text{vault}} p}{S_{\text{tot}}} \tag{1}$$

where B_{vault} is the balance of assets held in the protocol's vault, p is the price of the asset token in USD, and S_{tot} is the total supply of USDN tokens.

3.3. Token Minting

USDN tokens are minted whenever a deposit is made into the protocol's vault. The amount of minted USDN a_{usdn} is calculated by dividing the dollar value of the deposited assets by the USDN price:

$$a_{\text{usdn}} = \frac{a_{\text{asset}} p}{p_{\text{usdn}}} \tag{2}$$

Taking into account (1), the minting formula can be rewritten as:

$$a_{\text{usdn}} = \frac{a_{\text{asset}} S_{\text{tot}}}{B_{\text{vault}}} \tag{3}$$

3.4. Token Burning

When assets are removed from the protocol's vault, USDN tokens are burned in proportion to the withdrawn amount, following (2). Thus, for a given amount of USDN to be burned, the corresponding withdrawn assets amount is:

$$a_{\text{asset}} = \frac{a_{\text{usdn}} p_{\text{usdn}}}{p} = \frac{a_{\text{usdn}} B_{\text{vault}}}{S_{\text{tot}}} \tag{4}$$

3.5. Yield Sources

From (1), it is clear that the USDN price is influenced by the total assets held in the protocol's vault. As such, if the vault balance increases as a result of position fees, losses from long positions, or funding fee payments, the USDN price will rise. When a certain threshold is reached, the token rebases to a price slightly above $1 by increasing the total supply and balance of each holder. This increase in balance represents the yield of the USDN token. The rebase mechanism ensures that yields do not induce a price that significantly exceeds the value of $1. There is no balance and total supply adjustment (debase) if the price falls below $1, as the yield is expected to bring it back up relatively quickly.

4. Vault

The vault manages the supply of USDN tokens. The two main actions of the vault are deposits and withdrawals.

The deposit action allows to lock assets into the vault and mint a proportional amount of USDN tokens by providing an oracle price for the asset token. It follows the formula described in Section 3.3.

The withdrawal action allows to redeem USDN tokens for an equivalent dollar amount of assets from the vault. It follows the formula described in Section 3.4.

5. Long Side

The long side manages user positions. A position is comprised of the collateral (in assets) that the user deposited, together with a leverage which allows to control a larger position than the collateral. For example, a leverage of 3 times with an initial collateral of 1 wstETH behaves like a position of 3 wstETH. The product of the leverage and the initial collateral is called total exposure. When the price of the asset reaches the liquidation price for a position, its value is considered too small for it to continue existing, and it gets closed (in a decentralized way, see Section 5.2.1). Any remaining value goes to the vault pool and forms part of the yield of USDN. The two primary actions for the long side are opening new positions and closing (partially or entirely) existing positions.

When opening a new position, the user deposits assets as collateral and indicates their desired liquidation price, which is used to calculate the position's leverage. The entry price is taken from an oracle. When closing a position, users withdraw part or the entirety of the current value of their position, including any profits and losses (PnL) resulting from the asset's price action.

5.1. Position Value, Profits and Losses

The value of a long position is determined by the current market price of the asset coupled with its total exposure and liquidation price. The position value $v(p)$ is calculated as follows:

$$v(p) = \frac{T(p - p_{\text{liq}})}{p} \tag{5}$$

where p is the price of the asset (in dollars), T is the total exposure of the position (see (15)), and p_{liq} is the liquidation price of the position.

According to this formula, the position's value increases when the asset price rises and decreases when the asset price falls. The position value is used to calculate the PnL (Δv) relative to the position's initial collateral.

To calculate the profit of a position with an initial collateral of 1 and an initial leverage of 3x, the initial position value ($p_{\text{entry}} = 3000$) is compared with the value of position at a new market price. The initial value of the position is calculated as:

$$v(p_{\text{entry}}) = v(3000) = \frac{3(3000 - 2000)}{3000} = 1 \tag{6}$$

If price of the asset increases to \$4000:

$$v(4000) = \frac{3(4000 - 2000)}{4000} = 1.5 \tag{7}$$

$$\Delta v = v(4000) - v(3000) = 0.5 \tag{8}$$

The position has a profit of 0.5 asset.

If price of the asset decreases to \$2500:

$$v(2500) = \frac{3(2500 - 2000)}{2500} = 0.6 \tag{9}$$

$$\Delta v = v(2500) - v(3000) = -0.4 \tag{10}$$

The position has a loss of 0.4 asset.

5.2. Liquidation

The risk associated with leveraged trading is that a position can be liquidated. A liquidation occurs when the value of the collateral is insufficient to repay the borrowed amount (with a margin). In this situation, any remaining value from the position is credited to the vault, and the owner of the position loses their collateral.

Liquidations are an essential part of the protocol and should be performed in a timely manner. If a liquidation is executed too late (when the current asset price is much below the liquidation price of the position), the effective position value is negative and would skew the calculations of the funding fees for other position owners (Section 8). Additionally, a negative position value at the time of its liquidation would make it hard to reward the liquidator without incurring a loss to the vault's balance.

Note that thanks to the algorithmic nature of the PnL calculations (see Section 5.1), there is no "bad debt" when a liquidation occurs too late, because an amount equal and opposite to the negative position value was already credited to the vault side, and can be used to repay the debt in the long side automatically.

5.2.1. Liquidation Rewards

To ensure positions are liquidated as soon as possible, and thus reduce the risk of negative effects on the protocol, executing liquidations is incentivized with a reward paid out to the liquidator.

The reward is mainly derived from the gas cost of a liquidation transaction. The formula is divided into two parts. The first component is based on the gas cost and depends on the number n of liquidated positions (in practice, positions are grouped into buckets and liquidated in batches, see Section 9):

$$r_{\text{gas}} = \gamma(g_{\text{common}} + n g_{\text{pos}}) \tag{11}$$

where γ is the gas price (in native tokens per gas unit), g_{common} is the constant part of the gas units spent in the transaction and g_{pos} is the amount of gas unit spent for processing each position.

The sum $(g_{\text{common}} + n g_{\text{pos}})$ is roughly equal to the total gas used by the liquidation transaction.

The gas price γ is the lowest value between the block base fee [2] (with a fixed margin added to it to account for an average priority fee) and the effective gas price that the liquidator defined for the transaction.

The second component of the reward formula takes into account the total exposure of each liquidated position i, and the price difference between their liquidation price and p, the effective current price used for the liquidation:

$$r_{\text{value}} = \sum_{i=1}^{n} \frac{(P_i - p)T_i}{p} \tag{12}$$

where P_i is the liquidation price of the position, p is the asset price at the time of liquidation, and T_i is the total exposure of the position. As the price difference grows (meaning the remaining position value diminishes), the incentive grows as well, ensuring the profitability of executing liquidations regardless of the current gas price.

The resulting rewards in native tokens are calculated as follows:

$$r = \mu \cdot r_{\text{gas}} + \nu \cdot r_{\text{value}} \tag{13}$$

where μ and ν are fixed multipliers that can be adjusted to ensure profitability in most cases.

6. Trading Exposure

The trading exposure of the vault side is defined as (B_{vault} being the vault balance):

$$E_{\text{vault}} = B_{\text{vault}} \tag{14}$$

The trading exposure of the long side E_{long} is defined as:

$$T_i = c_i l_i \tag{15}$$

$$E_i = T_i - v_i \tag{16}$$

$$T_{\text{long}} = \sum_i T_i \tag{17}$$

$$B_{\text{long}} = \sum_i v_i \tag{18}$$

$$E_{\text{long}} = \sum_i E_i = T_{\text{long}} - B_{\text{long}} \tag{19}$$

where T_i is the total exposure of a long position i (defined as the product of its initial collateral c_i and initial leverage l_i), E_i is the trading exposure of a position (defined as its total exposure minus its value v_i), T_{long} is the total exposure of the long side, and B_{long} is the long side balance. The long side trading exposure can be interpreted as the amount of assets borrowed by the long side position owners.

As the price of the asset increases, B_{long} increases and the trading exposure of the long side decreases. Inversely, as the price of the asset decreases, B_{long} decreases and the trading exposure of the long side increases.

7. Imbalance

The protocol is at its optimum when it is balanced, which means its imbalance is zero. The imbalance is defined as the relative difference between the trading exposures of both sides (Section 6):

$$I = \begin{cases} \frac{E_{\text{long}} - E_{\text{vault}}}{E_{\text{long}}} & \text{if } E_{\text{vault}} < E_{\text{long}} \\ \frac{E_{\text{long}} - E_{\text{vault}}}{E_{\text{vault}}} & \text{else} \end{cases} \tag{20}$$

From (20), we can see that the imbalance is positive when the long side has a larger trading exposure. We can also see that the imbalance is bounded by $[-1, 1]$.

8. Funding

To incentivize depositors in the protocol side with the lowest trading exposure, the protocol charges a funding fee to the largest side, which is paid to the smaller side. The fee for a time interval Δt (in seconds) starting at instant t_1 and ending at t_2 is defined as:

$$F_{\Delta t} = F_{t_1,t_2} = E_{\text{long}_{t_1}} f_{\Delta t} \tag{21}$$

where $E_{\text{long}_{t_1}}$ is the trading exposure of the long side at the beginning of the interval and $f_{\Delta t}$ is the funding rate for that interval.

The funding rate for that interval is calculated as:

$$\begin{aligned} f_{\Delta t} &= f_{t_1,t_2} \\ &= \frac{t_2 - t_1}{86400} \left(s \operatorname{sgn}\left(I_{t_1}\right) I_{t_1}^2 + \sigma_{t_0,t_1} \right) \end{aligned} \tag{22}$$

where s is a scaling factor that can be tuned, $\operatorname{sgn}(I)$ is the signum function[1] applied to the imbalance I_{t_1} (20) at instant t_1 and σ is a skew factor (see Section 8.1). The denominator of the fraction refers to the number of seconds in a day, which means that $f_{t-86400,t}$ is the daily funding rate for the period ending at t. It can be observed that the sign of the funding rate matches the sign of the imbalance so long as the σ term is zero, thus a positive imbalance (more long trading exposure) results in a positive funding rate in that case. If the σ term is largely negative, the funding rate could be negative even if the imbalance is positive.

Note that the funding rate is calculated prior to updating the skew factor (which itself depends on the daily funding rate), so the skew factor is always the one calculated for the previous time period.

At the end of the funding period Δt, the vault and long side balances are updated as follows (ignoring PnL):

$$\begin{aligned} B_{\text{vault}_{t_2}} &= B_{\text{vault}_{t_1}} + F_{t_1,t_2} \\ B_{\text{long}_{t_2}} &= B_{\text{long}_{t_1}} - F_{t_1,t_2} \end{aligned} \tag{23}$$

8.1. Skew Factor

In traditional finance, funding fees are usually positive and serve as a kind of interest rate on the amount borrowed by the long side. This means that fees should ideally not be zero even if the protocol is perfectly balanced.

The dynamic skew factor σ is introduced to ensure that the funding rate matches the market's accepted interest rate when the protocol is balanced. This factor is calculated as an exponential moving average of the daily funding rate. For a time interval Δt, the skew factor is updated as follows:

$$\begin{aligned} \sigma_{\Delta t} = \sigma_{t_1,t_2} &= \alpha f_{t_2-86400,t_2} + (1-\alpha)\sigma_{t_0,t_1} \\ &= \frac{\Delta t}{\tau} f_{t_2-86400,t_2} + \frac{\tau - \Delta t}{\tau} \sigma_{t_0,t_1} \end{aligned} \tag{24}$$

where α is the smoothing factor of the moving average, τ is the time constant of the moving average, $f_{t_2-86400,t_2}$

[1]The signum function returns -1 if the sign of its operand is negative, 0 if its value is zero, and 1 if it's positive.

is the daily funding rate for the last day, and σ_{t_0,t_1} is the previous value of the skew factor.

Because this factor is summed with the part of the funding rate which is proportional to the imbalance in (22), it shifts the default funding rate value when the protocol is balanced. In practice, if the imbalance remains positive (more trading exposure in the long side) for some time, the daily funding rate will keep increasing. When the funding fees become too important for the long side position owners, they will be incentivized to close their positions, which will decrease the imbalance. When the imbalance reaches zero, the daily funding rate will stop increasing, and maintain its current value thanks to the skew factor. This ensures that the market finds its own daily funding rate which is deemed acceptable by the protocol actors.

9. Liquidation Ticks

As previously stated in Section 5.2, the long side positions are grouped by liquidation price into buckets for efficient liquidation. Each bucket is called a liquidation tick and is identified by its number, ranging from −322 378 to 980 000.

The tick number of a bucket containing positions can be used to calculate the price at which those positions can be liquidated, and this price includes a penalty, such that the position value is greater than zero when it can be first liquidated. This penalty ψ_i denominated in number of ticks, is stored in the bucket's metadata and allows to calculate the *theoretical* liquidation price (price at which the position value is zero) of a position in tick i by retrieving the price of the tick number $(i - \psi_i)$.

For instance, if the penalty for tick 1000 is 100 ticks (≈1.005%), then the theoretical liquidation price of a position in tick 1000 is the price of tick 900. But the position can be liquidated as soon as the price falls below the price of tick 1000.

9.1. Tick Spacing

To improve gas performance when iterating over the ticks, we only consider ticks that are multiples of a tick spacing λ for the liquidation buckets. The range of valid ticks is:

$$R \Rightarrow \left\{ \lambda k \mid k \in \mathbb{Z} : \left\lceil \frac{-322378}{\lambda} \right\rceil \leq k \leq \left\lfloor \frac{980000}{\lambda} \right\rfloor \right\}$$

For the Ethereum implementation, the tick spacing is defined as 100 ticks (the price difference between two consecutive usable ticks is ≈1.005%).

9.2. Unadjusted Price

At the core of the tick system is the equation that dictates the conversion from a tick number to a price that we qualify as "unadjusted". This equation is:

$$\varphi_i = 1.0001^i \tag{25}$$

where φ_i is the unadjusted price for the tick i. From this formula, we can see that the unadjusted price increases by 0.01% for each tick. This allow to represent a wide range of prices with a small number of ticks. In practice, because of the tick number range described above, prices ranging from \$0.000 000 000 000 01 to ≈\$3 tredecillion ($\approx 3.62 \times 10^{42}$) can be represented.

9.3. Adjusted Price

However, because of the funding fee mechanism, the liquidation price of a position can change over time. If the funding fee is positive, a position's collateral is slowly eaten away, which in turn increases its liquidation price. The "adjusted" price P_i of a tick is thus calculated as:

$$P_i = M\varphi_i \tag{26}$$

where M is a multiplier that represents the accumulated effect of the funding fees. Interestingly, all ticks are affected by the funding fees in the same way, which means that the multiplier M is the same for all ticks (see Section 9.3.1).

Since it would be imprecise to represent the multiplier M as a fixed-precision number in the implementation, we derive an accumulator A from the equations below.

The value of a tick i can be derived from (5):

$$v_i = \frac{T_i}{l_i} = \frac{T_i\left(p - P_{i-\psi_i}\right)}{p} \tag{27}$$

where T_i is the total exposure of the positions in the tick, l_i is the effective leverage of the positions in the tick (at the current price), p is the current price of the asset and $P_{i-\psi_i}$ is the theoretical liquidation price of the positions in this tick (ψ_i being the penalty of the tick). By using (26), we can rewrite this equation as:

$$v_i = \frac{T_i\left(p - M\varphi_{i-\psi_i}\right)}{p} \tag{28}$$

With the range of valid ticks R defined in Section 9.1, we define the following invariant (analog to (18) for the ticks):

$$B_{\text{long}} = \sum_{i \in R} v_i \tag{29}$$

which means that the balance of the long side must be equal to the sum of the value of each tick i. We can now combine (17), (28) and (29), then solve for M:

$$M = \frac{p\left(\sum_{i \in R} T_i - B_{\text{long}}\right)}{\sum_{i \in R}\left(T_i \varphi_{i-\psi_i}\right)} = \frac{p\left(T_{\text{long}} - B_{\text{long}}\right)}{A} \tag{30}$$

$$A = \sum_{i \in R}\left(T_i \varphi_{i-\psi_i}\right) \tag{31}$$

where A is an accumulator that can easily be updated when a position is added or removed from the long side.

Finally, the adjusted price of a tick P_i can be calculated as:

$$P_i = M\varphi_i = \frac{\varphi_i p\left(T_{\text{long}} - B_{\text{long}}\right)}{A} = \frac{\varphi_i p E_{\text{long}}}{A} \tag{32}$$

9.3.1. Multiplier Proof

As proof that multiplier M is the same for all ticks, consider a position with a current value v_0 and a liquidation price φ that was not subject to funding fees. The liquidation price for a current asset price p is derived from (5):

$$\varphi = p_{\text{liq}}(p) = \frac{p(T - v_0)}{T} \tag{33}$$

where T is the total exposure of the position. If the funding rate is f, the funding fee for the position is:

$$F = fE = f(T - v_0) \tag{34}$$

where E is the trading exposure of the position. The new position value v_1 is then:

$$\begin{aligned} v_1 &= v_0 - F = v_0 - f(T - v_0) \\ &= v_0(1+f) - fT \end{aligned} \tag{35}$$

The new adjusted liquidation price P is thus:

$$\begin{aligned} P &= \frac{p(T - v_1)}{T} = \frac{p(T - v_0(1+f) + fT)}{T} \\ &= \frac{p(T(1+f) - v_0(1+f))}{T} \\ &= (1+f)\frac{p(T - v_0)}{T} = (1+f)\varphi = M\varphi \blacksquare \end{aligned} \tag{36}$$

From this result, we can see that all ticks are affected by the funding rate in the same way.

10. Conclusion

While existing yield-bearing synthetic dollar tokens and stablecoins suffer from centralization and opacity, this paper demonstrates that it is possible to implement a fully decentralized protocol which is gas-efficient and generates consistent yield for its users. This novel algorithmic approach solves the main challenges faced by the constrained on-chain nature of decentralized applications and provides a robust, secure and transparent model for the creation of a synthetic dollar token.

11. Glossary

funding fee: A fee charged to the side with the higher trading exposure, and rewarded to the other side. The fee is calculated by multiplying the funding rate per day by the long side trading exposure, and then normalized to the elapsed duration. 1, 2, 3, 4, 5

liquidation price: The liquidation price is the threshold at which a position's value becomes too low to sustain. If the market price hits this level, the position can be liquidated to limit further losses. 2, 3, 5

***PnL* – profits and losses**: Profits and losses are the gains and losses of a position. They are calculated as the difference between the current position value and its initial value. 2, 3, 4

rebase: A rebase is a mechanism that adjusts the total supply of a token to maintain a target price. The rebase factor is calculated as the ratio of the current price to the target price. The balance of each holder is adjusted proportionally. 2

total exposure: The total exposure of a position is the product of the position's initial collateral and initial leverage. 2, 3, 4, 5, 6

trading exposure: The trading exposure of the vault side is equal to the vault balance. The trading exposure of the long side is equal to the total exposure of all long positions, minus the long side balance. It represents the borrowed part of all long positions. 1, 4, 5, 6

***wstETH* – Lido wrapped staked ETH**: wstETH is a wrapped version of the tokenized staked ETH from Lido. 1, 2

12. Acknowledgment

Thank you to everyone who made this project possible in any way: Fecri Al-Fakeri, Brice Belkadi, Julien Chiappinelli, Carl Cronsioe, Mikael Cruchon, Antoine Duez, Diogo Estima, Lorena Feijoo, Alfred Gaillard, Esteban Gonzalez, Vicente Guedes, Matthieu Guerra, Marylou Kronauer, Simon Lussi, Sebastien Mansour, Abdel Mecheri, Bilel Oural, Flavio Roth.

References

[1] Lido, "wstETH." [Online]. Available: https://docs.lido.fi/contracts/wsteth/

[2] "EIP-1559: Fee market change for ETH 1.0 chain." [Online]. Available: https://eips.ethereum.org/EIPS/eip-1559

[3] Uniswap, "Uniswap v3 Whitepaper." [Online]. Available: https://app.uniswap.org/whitepaper-v3.pdf