HOOKED!

REHYPED Stablecoin Liquidity on Uniswap v4

Technical Whitepaper

Version 3.0 • 2026-02-02

Abstract

Stablecoin AMMs succeed by keeping price close to parity. That stability also compresses fee yield: when price barely moves, there is less fee-paying flow, and LP returns end up leaning on incentives. HOOKED! introduces REHYPED liquidity, a fully collateralized inventory-routing design, not credit rehypothecation. The hook keeps a sized execution buffer inside the pool for immediate swaps and routes remaining LP inventory into external yield venues such as money markets via ERC-4626 vault shares. LPs therefore earn swap fees when volume is present and a baseline lending yield when markets are calm.

We implement HOOKED! as a Uniswap v4 hook. Routing happens atomically inside swap settlement, oracle observations provide TWAP-style signals, and dynamic fees are optional but bounded.

Contents

1 Introduction

Stablecoins have become one of the largest categories of on-chain assets. As of late January 2026, the total circulating stablecoin market capitalization is roughly USD 308 billion.[19]

Deep, low-slippage stablecoin liquidity on AMMs is still expensive to bootstrap. The reason is almost tautological: stable pools stay near parity, so there is little price movement to generate fee-rich order flow. In that regime, passive LPs capture relatively little volatility-driven fee flow. Issuers and pool sponsors therefore often rely on incentives (token emissions) to sustain depth.[9, 20] Incentives are not inherently the problem. The problem is that in fee-only designs the baseline can be thin, so the subsidy has to be large and persistent to keep depth where integrators want it.

1.1 Contribution

HOOKED! tackles this constraint by making *idle* stablecoin inventory productive. Instead of holding the full position as swap inventory, the hook time-shares LP capital across two places:

- a **swap buffer** that stays in-pool to clear trades immediately, and
- an **external yield leg** that deploys the remainder into money markets via ERC-4626 vault routing.

Economically, the LP return becomes the sum of a fee leg and a yield leg. When volume shows up, LPs earn swap fees. When markets are quiet, the external leg continues to pay a baseline rate. This two-leg profile is the basis for the yield-floor argument and for the incentive-efficiency discussion in later sections.

2 Background and Related Work

2.1 AMMs and stable swaps

Constant function market makers (CFMMs) such as Uniswap v2 popularized constant-product pricing.[1, 8] For stable pairs, specialized invariants such as Curve's stableswap reduce slippage near parity.[9]

2.2 Concentrated liquidity

Uniswap v3 introduced concentrated liquidity: LPs allocate liquidity only across a chosen price range (ticks), improving capital efficiency for assets that trade in a narrow band.[2, 3] Stable pairs are natural beneficiaries, but the fee constraint remains: a stable price band means fewer fee-generating price moves.

2.3 Programmable AMMs and hooks

Uniswap v4 adds **hooks**, allowing pool-specific logic to run at defined points in the pool lifecycle.[5, 6, 7] Hooks enable new pool behaviors such as dynamic fees, custom accounting, and external integrations.

2.4 Lending markets as baseline yield

Money markets such as Aave and Compound pay suppliers an interest rate driven by utilization and borrow demand.[14, 16] Aave v3 also supports **Efficiency Mode (E-Mode)** for correlated assets such as USD stablecoins, enabling higher loan-to-value ratios for stablecoin loops.[15]

3 Protocol Overview

3.1 Architecture

HOOKED! is a Uniswap v4 hook that manages stablecoin inventory across two locations: an on-pool execution buffer and one or more ERC-4626 vault positions representing external yield venues.[7, 13] Routing is **in-band**: when idle-split routing is enabled, the hook deposits excess inventory into a vault and withdraws shortfalls during the same swap settlement transaction. Separately, the hook records oracle observations (TWAP-style signals) and can apply an optional dynamic-fee schedule with a hard cap.

3.2 System components

In practice, a HOOKED! pool is a small stack of contracts:

- a Uniswap v4 pool (stable pair) with the HOOKED! hook,
- ERC-4626 vault adapters for one or more yield venues,
- an oracle coordinator for price and risk bounds,
- a dynamic-fee module,
- an optional incentives controller.

3.3 REHYPED liquidity as a state machine

From the hook's perspective, LP inventory lives in one of two buckets: the in-pool buffer used for swaps, or ERC-4626 vault shares that represent externally deployed balance. Moving between those buckets is "rebalancing." In HOOKED!, that movement is constrained by the configured buffer policy, the liquidity and withdrawal rules of the chosen external venues, and per-block rate limits designed to reduce churn.

4 Model: Two-Leg Returns and Capital Allocation

This chapter models LP returns as the sum of a fee leg and an external yield leg. We use APR to mean a simple annualized rate of return, and APY to mean the compounded return over one year (when compounding applies).

4.1 Inventory routing

Let C denote total stablecoin inventory supplied by LPs (units of stablecoin). At time t, the hook keeps an execution buffer $C_{\text{buf}}(t) = \alpha(t)C$ inside the AMM and holds the remainder as an external position $C_{\text{ext}}(t) = (1 - \alpha(t))C$ via ERC-4626 vault shares.[13] The control variable $\alpha(t) \in [0, 1]$ trades off execution robustness against external deployment.

4.2 Swap-fee revenue

Let q_t be traded notional during discrete interval t (in stable units) and ϕ_t the effective swap fee rate. Over a horizon of N intervals with total length T (in years), swap fees earned by LPs are

$$R_f = \sum_{t=1}^{N} \phi_t \, q_t \;\approx\; \phi Q, \tag{4.1}$$

where $Q = \sum_t q_t$ and the approximation holds when fees are constant.

Protocols and dashboards often report a *pool-level* fee APR relative to in-pool liquidity. For the full LP inventory C, we define the *fee contribution* to return as

$$r_f = \frac{R_f}{C} \cdot \frac{1}{T}. \tag{4.2}$$

Under a fixed routing policy where only αC is in-pool at a time, r_f can be viewed as a conversion of the pool-level fee APR into a return per unit of total LP inventory.

4.3 External yield stream

Let i_t be the external supplier rate offered by a money market during interval t (in APR units), and let $\tilde{C}_{\text{ext}}(t)$ be the average externally deployed balance over that interval. Interest earned over the horizon is approximately

$$R_y \approx \sum_{t=1}^{N} i_t \, \tilde{C}_{\text{ext}}(t) \, \Delta t, \tag{4.3}$$

where Δt is the interval length in years. The annualized *external-yield contribution* relative to total inventory is

$$r_y = \frac{R_y}{C} \cdot \frac{1}{T}. \tag{4.4}$$

When $\tilde{C}_{\text{ext}}(t) \approx (1 - \alpha)C$ is roughly constant and rates vary slowly, $r_y \approx (1 - \alpha)\,\bar{i}$ where $\bar{i}$ is the average supplier APR.[14, 16]

4.4 Combined return and yield floor

We define r_y net of any strategy-level fees charged by the external venue (for example, vault management fees). In the reference design, inventory routing is **in-band**: deposits and withdrawals occur atomically during swap settlement when idle-split routing is enabled, and the gas cost of that settlement is paid by the transacting router (not debited from LP inventory).[74, 13]

Accordingly, the annualized return to LP inventory decomposes as:

$$r_{\text{tot}} = r_f + r_y. \tag{4.5}$$

The *yield floor* of REHYPED liquidity is the portion of Equation (4.5) that persists even when swap-fee revenue is low, namely r_y.

4.5 Settlement overhead (borne by swappers)

Although routing overhead is not an LP drag term in the reference design, it *is* a user and integrator consideration: a swap that triggers vault deposits or withdrawals consumes more gas than a vanilla swap.

Let Δg denote the incremental gas used by vault routing during a swap (for example, one ERC-4626 deposit and/or withdrawal). With gas price p_{gas} (in ETH per gas) and ETH price p_{ETH} (in USD per ETH), the incremental swap cost is approximately:

$$\Delta\text{Cost}_{\text{USD}} \approx \Delta g \, p_{\text{gas}} \, p_{\text{ETH}}. \tag{4.6}$$

Table 4.1 illustrates the order of magnitude for $\Delta g = 300{,}000$ gas (a conservative placeholder for one routing event). On recent Ethereum mainnet snapshots, gas prices have frequently been sub-gwei,[76] which makes the incremental cost small in dollar terms.

Gas price (gwei)	**ETH price**	**Cost for $\Delta g = 300{,}000$**
0.085 (base)	USD 2,500	USD 0.06
0.279 (avg)	USD 2,500	USD 0.21
0.500	USD 2,500	USD 0.38
10.0 (stress)	USD 2,500	USD 7.50

Table 4.1: Illustrative incremental settlement cost for a 300k-gas routing event. Gas price examples are from Etherscan snapshots; ETH price is rounded.[76]

In practice, most swaps can be serviced entirely from the buffer (no vault withdrawal), and routers can reduce failure probability and gas overhead by sizing trades conservatively (see Equation (5.2)). The key design point is that REHYPED liquidity enables *lower swap fees* without requiring LPs to forgo baseline yield; the marginal gas cost is a UX consideration rather than an LP return tax.

4.6 Zero-fee pools and an outperformance condition

Setting the swap fee to zero implies $R_f \approx 0$ and thus $r_f \approx 0$. A zero-fee HOOKED! pool still pays LPs when the external leg is positive:

$$r_{\text{tot}} \approx r_y. \tag{4.7}$$

Compared to a fee-only stable pool with fee return r_f^{fee} (no external yield), a zero-fee HOOKED! pool outperforms whenever

$$r_y > r_f^{\text{fee}}. \tag{4.8}$$

This is not a claim that external yield always dominates fees, but it does occur frequently in calm stablecoin markets where swap-fee yield is structurally compressed; Chapter 7 shows one year of daily observations consistent with this regime.

4.7 Incentive requirement and efficiency

If LPs require target return r^* to supply C of inventory, the annual incentive budget needed to "top up" returns can be approximated as

$$I \approx C \max(0, r^* - (r_f + r_y)). \tag{4.9}$$

For a fee-only pool with $r_y = 0$, the entire gap between r_f and r^* must be covered by incentives. HOOKED! does not argue against incentives; it makes each incentive dollar more productive by pairing it with a baseline yield leg that can persist after subsidies end.

5 Mechanism Design: Buffer Management and Rebalancing

5.1 Buffer sizing as a risk quantile

The buffer fraction α is a policy parameter: smaller buffers deploy more capital to external yield, but increase the probability that a large swap requires a vault withdrawal (or, in the worst case, fails). A useful way to think about buffer sizing is as a quantile (VaR-style) policy.

Let X be the random variable "net stable outflow that must be served immediately" over the relevant settlement window (for example, per swap or per block). A simple sizing rule is

$$C_{\mathrm{buf}} = \mathrm{Quantile}_p(X), \tag{5.1}$$

with p chosen based on the target failure probability $\varepsilon = 1 - p$. This framing matches the intuition that using an *average* flow is rarely sufficient; policies are typically tuned around the median and upper quartiles, then validated empirically.

5.1.1 Example: a 0.99-quantile policy and swap sizing

Consider a stable pair near parity where the in-range (active-band) inventory implied by current liquidity is approximately USD 50 million, split roughly evenly across tokens. With `activeBandBufferBps` set to 1,000 (i.e., $\alpha = 0.10$), the hook targets a buffer on the order of USD 2.5 million per token side. If the empirical distribution of per-block net outflow X has a 0.99-quantile of USD 2.0 million, then a 0.99-quantile policy implies that in roughly 99% of settlement windows, the buffer alone is sufficient and no external withdrawal is required.

For routers, a simple conservative trade-sizing rule is

$$q_{\max} = u_{\max} C_{\mathrm{buf}}, \tag{5.2}$$

where $u_{\max} \in (0, 1)$ is a utilization cap (for example, $u_{\max} = 0.5$ keeps half the buffer as headroom). In this toy example, $q_{\max} \approx$ USD 1.25 million. Orders larger than $q_{\max}$ can be split across venues or across multiple swaps to reduce the probability of a vault withdrawal (and, in the worst case, a withdrawal-induced revert).

Implemented policy (active-band targeting). In the current implementation, buffer management is executed **atomically inside swap settlement** when idle-split routing is enabled. The hook targets the inventory required to support the *active* tick band implied by current in-range liquidity, then keeps a configurable fraction of that active-band inventory unvaulted.[74] Operationally, this produces a dynamic $\alpha(t)$ that adapts to price and active liquidity rather than relying on a single fixed average.

5.2 Atomic vault routing during swaps

At a high level, each swap settlement does two things:

1. **Intake:** the input token received from the swap is recorded as inventory. If idle-split is enabled, the hook deposits any "excess over target buffer" into an ERC-4626 vault (external leg) and retains the remainder in the on-hook buffer.[13, 74]
2. **Payout:** the output token owed to the swap is paid first from the on-hook buffer. If the required amount exceeds the available buffer (net of fee reserves), the hook withdraws the remainder from the ERC-4626 vault and completes the payout in the same transaction.[74]

Because both legs happen within the same settlement path, "rebalancing" is not a separate background process; it is part of the swap lifecycle.

5.3 Active-band rebalancing and rate limits

When price crosses ticks, the active liquidity band changes. The hook may therefore rebalance buffer versus vault toward new active-band targets after settlement, subject to:

- **tolerance bands:** deviations smaller than a configured tolerance (basis points of target) do not trigger a rebalance, reducing churn,
- **rate limits:** active-band rebalancing is limited to at most once per block per pool.[74]

5.4 Worst-case outflow and external liquidity risk

Two worst-case scenarios matter in practice.

Swaps exceed the buffer. If a swap demands more output than the buffer can safely provide, the hook withdraws the shortfall from the vault. In the implementation, fee reserves are senior to user payouts (the buffer cannot dip below reserved fees), and payout is computed against the combined buffer plus vault assets.[74] This keeps accounting consistent, but it does not eliminate the need for the vault strategy to be liquid.

External venue liquidity crunch. External venues can fail to return assets on demand (for example, money markets can have insufficient available liquidity during stress). In HOOKED!'s reference vault design, if the vault cannot return the requested assets, the swap settlement reverts.[74] This is a deliberate safety choice: it prevents partial, ambiguous payouts and forces the strategy selection problem to be explicit.

Mitigations are policy-level:

- choose strategies with near-instant withdrawal guarantees (or diversify across multiple venues),
- maintain a larger buffer during periods where external liquidity risk is elevated,
- use a governance-controlled circuit breaker: disable idle-split routing and drain vault balances back to the buffer during stress.[74]

The important point is that REHYPED liquidity is fully collateralized inventory routing; it does not rely on protocol-created leverage, but it does inherit the liquidity properties of the external venues it chooses.

6 Oracles and Dynamic Fees

6.1 Oracle objectives

For stablecoin pools, oracles primarily serve two roles:

- **measurement:** producing TWAP-style signals (for example, tick cumulatives) that are harder to manipulate than instantaneous price,[4]
- **control hooks:** enabling policy responses such as dynamic fees or (optionally) routing constraints when the pool is under stress.[17]

HOOKED! includes an on-chain observation module that maintains per-pool oracle observations in a separate coordinator contract.[81] This is compatible with Uniswap-style oracle design and can be combined with external feeds such as Chainlink when desired.[17]

6.2 Dynamic fees as bounded oracle add-ons

Dynamic fees are orthogonal to REHYPED liquidity: a pool can route idle inventory externally, adjust fees dynamically, do both, or do neither. In the reference implementation, a pool's fee schedule is parameterized by:

- a **static fee** ϕ_{static} (ppm),
- a **maximum fee** $\phi_{\max}$ (ppm),
- an optional **oracle strategy** that computes an additional fee $\phi_{\text{oracle}}(t)$ based on pool context.[79, 80]

The effective swap fee is:

$$\phi(t) = \min\big(\phi_{\max},\, \phi_{\text{static}} + \phi_{\text{oracle}}(t)\big). \tag{6.1}$$

Governance surface. Governance (or a privileged administrator) sets ϕ_{static} and $\phi_{\max}$ and selects the oracle strategy for a pool.[79] The oracle strategy itself is constrained by the hard cap $\phi_{\max}$, which bounds worst-case fees.

What dynamic fees address. For stable pairs, "normal" trading near parity tends to be low-risk and price-insensitive, so fees can be kept low. When adverse selection rises (for example, when price moves away from parity or when large trades arrive relative to depth), increasing fees can protect inventory and reduce toxic flow.[18, 8]

7 Empirics: Stable Yield Level and Volatility (2024)

We use **Data S1**, a daily yield dataset for calendar year 2024, to stress-test a simple claim. Lending rates can provide a material baseline "floor" for stablecoin LPs, and the combined (fees + lending) series is less dependent on short-lived fee spikes than fees alone. Data S1 is a supplementary dataset accompanying this paper.[78]

Data availability

Data S1 is supplementary to this paper.[78] If you are reading a copy that does not include the dataset files, request Data S1 from the authors.

7.1 Dataset structure and methodology

Data S1 contains 366 daily observations for 2024 with three fields: `timestamp` (Unix seconds at UTC day boundaries), `lending` (a daily snapshot of an external stable-yield rate), and `lp` (a daily snapshot of a fee-only LP yield estimate). Both yield fields are expressed as APR levels in *percent units.*[1]

What the series represent. The purpose of Data S1 is not to forecast exact pool returns, but to illustrate an empirical pattern that is robust across stablecoin markets: lending rates tend to be smoother and present on most days, while fee-only LP yield is more bursty and dependent on short-lived volume events.

Reproducible construction (high level). A reader can reproduce a dataset of this form using public sources:

- **Lending leg (`lending`).** Query a money market supplier rate (for example, Aave v3 supply rate for a USD stablecoin) at daily frequency and convert to a simple APR level.[14, 62]
- **Fee-only leg (`lp`).** Estimate a stablecoin LP fee APR by annualizing fees earned over a window and dividing by the pool's in-range liquidity (or TVL): $r_f \approx \frac{\text{fees}}{\text{liquidity}}$.[2, 55]

Data S1 is intentionally minimal (it omits per-pool TVL), so we report unweighted summary statistics. Where TVL time series are available, a TVL-weighted analysis is a straightforward extension.

---

[1] For example, a value of 6.62 corresponds to a 6.62% APR.

7.2 Yield components

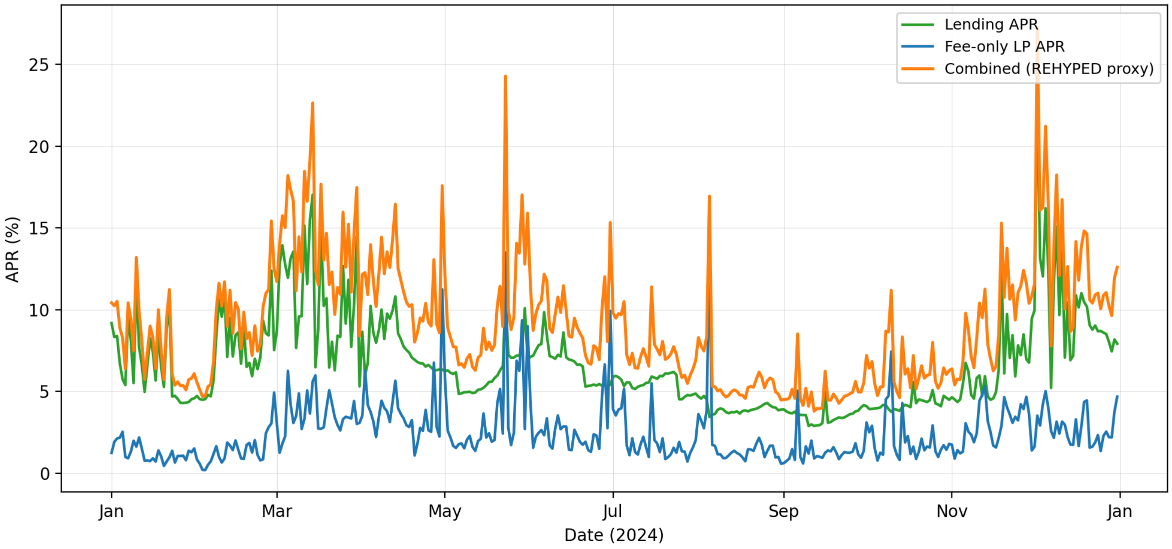

Figure 7.1: Daily yield components (2024). "Lending" is the external stable-yield leg, "LP fees" approximates swap-fee yield, and "Combined" is their sum.

7.3 Descriptive statistics

Let y_t^{lend} and y_t^{lp} be daily APR observations (in percent units) and define $y_t^{\text{comb}} = y_t^{\text{lend}} + y_t^{\text{lp}}$. Table 7.1 reports basic summary statistics.

Series	**Mean**	**Median**	**Std**	**Min**	**Max**	**CV**
Lending (y^{lend})	6.62	5.97	2.83	2.89	23.77	0.43
LP fees (y^{lp})	2.45	1.92	1.77	0.19	13.51	0.72
Combined (y^{comb})	9.07	8.39	3.75	3.79	27.29	0.41

Table 7.1: Summary statistics for daily APR observations in Data S1 (percent units). CV denotes the coefficient of variation (Std/Mean).

Three practical observations follow:

- **Lending dominates most of the time.** In Data S1, lending yield exceeds LP-fee yield on about 96% of days.
- **Fees are spikier.** The LP-fee series has a materially higher CV (0.72) than lending (0.43).
- **A combined "floor" emerges.** The combined series has a lower CV than LP fees alone (0.41 vs 0.72), consistent with the "yield floor" intuition in Chapter 4.

7.4 Volatility of yield

We operationalize yield volatility as a 30-day rolling standard deviation of APR levels (percentage points):

$$\sigma_{30}(t) = \mathrm{StdDev}(y_{t-29}, \ldots, y_t). \tag{7.1}$$

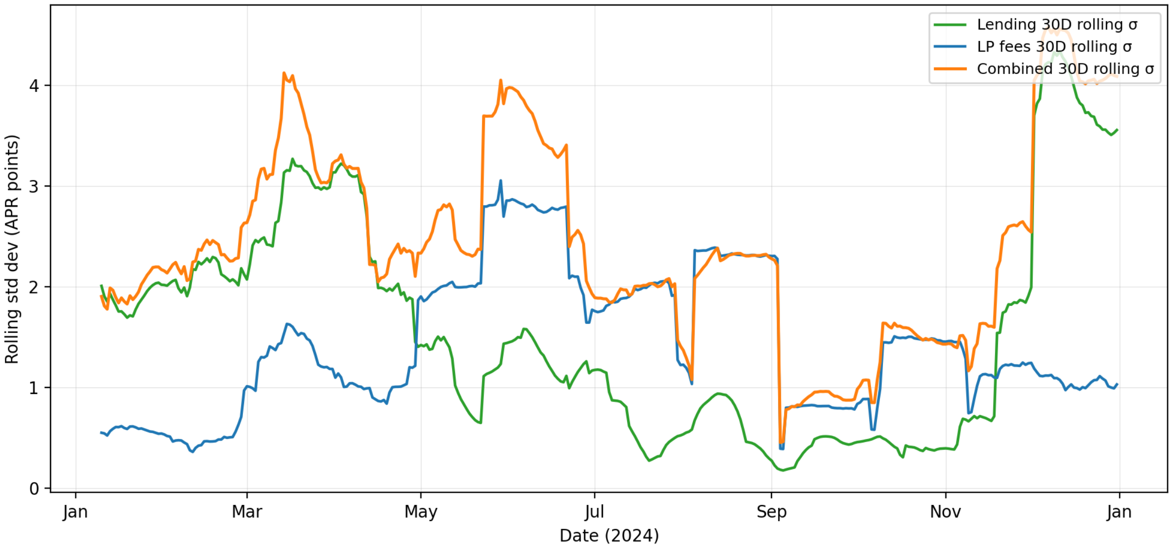

Figure 7.2: Yield volatility (30-day rolling standard deviation) for lending yield, LP-fee yield, and the combined series.

7.5 Rate changes and fee spikes

To visualize when lending rates increase or decrease versus when fee yield spikes, we plot daily changes $\Delta y_t = y_t - y_{t-1}$.

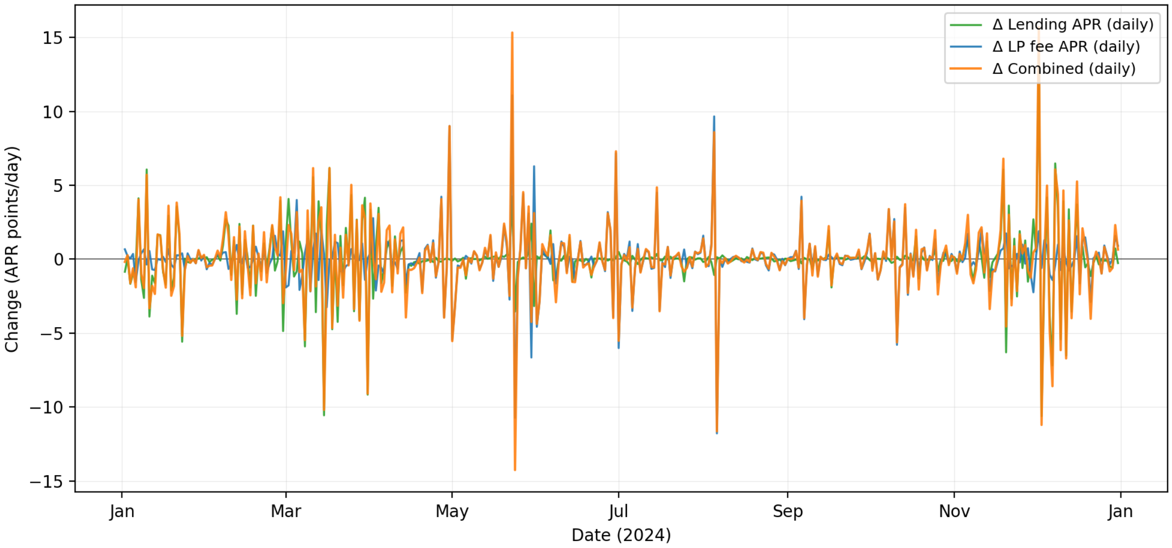

Figure 7.3: Daily changes in yield components (Data S1). Fee yield tends to move in jumps relative to lending, which is typically smoother.

Interpretation for buffer policy. A buffer policy should be robust to the fast component (fee-driven flow and inventory shocks) while allowing the slow component (lending rates) to provide baseline yield. In practice this means: keep enough buffer to service bursts of demand without frequent vault exits, and treat external routing as an opportunistic use of *idle* inventory rather than a hard dependency for every swap.[74]

Limitations. Data S1 does not include pool TVL time series, so we report unweighted APR observations. TVL-weighted analysis is a straightforward extension when per-pool capital series are available.

8 Bootstrapping Liquidity: Evidence From Unichain Incentives

8.1 Why incentives decay into “mercenary” liquidity

Liquidity mining is effective at creating a burst of TVL, but historically struggles with retention after rewards end. This is discussed directly in Uniswap governance commentary around incentive programs.[20]

8.2 Case study: Uniswap v4 USDC and USD$_{\text{T}}$0 on Unichain

Uniswap incentive programs can bootstrap liquidity quickly, but retention after incentives end is often low. To quantify this pattern, we reconstruct the 2025 UNI incentive schedule and the USDC/USD$_{\text{T}}$0 pool TVL time series on Unichain.[75]

Across 141 incentive days (2025-04-15 through 2025-09-02), the campaign paid a total of USD 2,901,470.[75] Pool liquidity peaked at USD 163,189,773 (2025-05-08).[75] By the final incentive day, TVL was USD 20,311,259, and by 2025-10-17 it had fallen to USD 1,340,040.[75] As of late January 2026, third-party trackers report roughly USD 785,000 of liquidity remaining in the pool.[77]

The pool can be viewed in the Uniswap interface.[70]

8.2.1 What the numbers imply

- **Incentive intensity.** Total spend of USD 2.90 million over 141 days implies an average of about USD 20,600 per day, with higher spend during the early high-emission phase.[75]
- **Retention (peak-to-residual).** Residual liquidity of USD 0.785 million is about 0.48% of the USD 163.19 million peak.[75, 77]
- **Retention (end-of-incentives-to-residual).** Relative to USD 20.31 million at the end of incentives, the residual is about 3.86%.[75, 77]
- **Cost per retained dollar.** USD 2.90 million of incentives versus USD 0.785 million of residual TVL implies about USD 3.70 of incentives per USD 1 of remaining liquidity.[75, 77]
- **Fees versus incentives.** The reconstruction estimates total swap fees of USD 412,800 over the same window, so fees covered about 14.23% of incentive spend.[75]

These figures are directionally consistent with a common pattern in liquidity mining: incentives can purchase rapid TVL growth, but retention after incentives end is often low. Even without taking a position on whether incentives are “good” or “bad,” the observed decay highlights a practical constraint for stablecoin issuers: sustaining deep liquidity via fee-only returns can require persistent subsidy.[21]

Figure 8.1: USDC/USD$_{\text{T}}$0 pool TVL over the Unichain incentive period (shaded) and the post-incentive decay, reconstructed from the campaign schedule and pool TVL time series.[75]

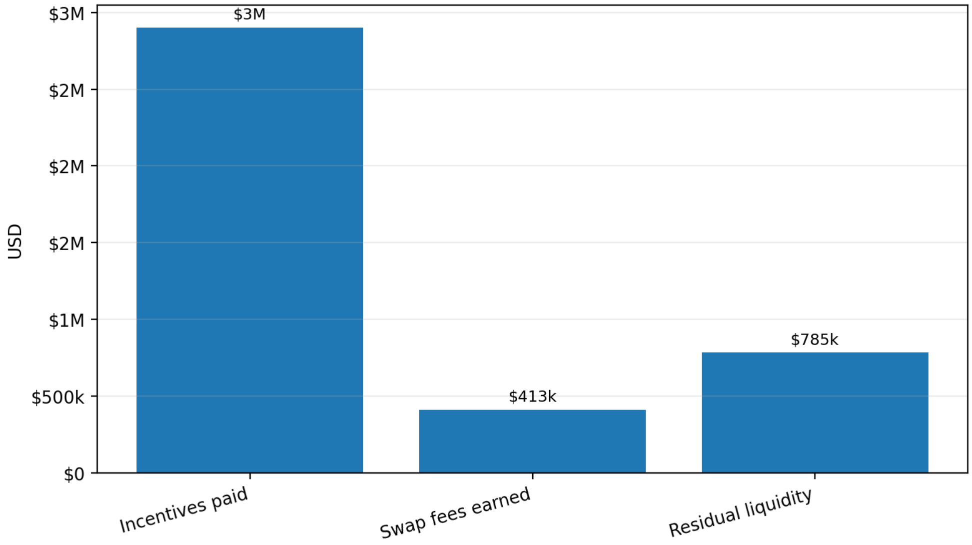

Figure 8.2: Incentives paid versus swap fees earned during the 2025 Unichain campaign, with current pool liquidity as a residual reference point.[75, 77]

8.3 How HOOKED! improves incentive efficiency

HOOKED! changes retention economics because the pool continues to pay a baseline yield after incentives end. A separate Gauntlet retrospective notes that a hook-enabled, inventory-routing pool on USDC/USDT can deliver higher base yield than a vanilla pool by adding an external yield leg.[24]

In other words, incentives can be used to bootstrap a position that remains productive after the subsidy ends. This is exactly the “yield floor” mechanism formalized in Chapter 4.

9 Emerging Stablecoins: The DEX-Lending Flywheel

9.1 The two-sided liquidity problem

A new stablecoin needs two kinds of liquidity:

- **transactional liquidity**: deep DEX pools for swaps at low slippage so users can actually spend and trade the asset,
- **balance-sheet liquidity**: large money-market supply so users can borrow the stablecoin (or borrow against it) to build leverage positions and integrate it into DeFi portfolios.

Projects therefore run two incentive programs: one to rent DEX TVL and one to attract money-market supply.

The problem is not that incentives are "bad." The problem is that stablecoin fee-only pools can have structurally low fee return when price stays near parity, which forces large incentive budgets just to maintain baseline depth.

9.2 Money markets can hold billions in stablecoins

On major money markets, stablecoin supply can reach the scale of billions of dollars, meaning the lending leg is not a niche market. For example, on Aave v3 Ethereum, stablecoin supply is reported in the multi-billion dollar range across USDT and USDC.[25] Aave also supports high-efficiency stablecoin borrowing and supplying via E-Mode, increasing capital efficiency for correlated stable assets.[15]

9.3 The flywheel: why DEX depth and lending supply reinforce each other

DEX depth improves a stablecoin's transactional utility: lower slippage reduces the cost of adoption for integrators, arbitrage, and payment-like use cases. Money-market supply improves a stablecoin's balance-sheet utility: it enables borrowing, leverage loops, and integration into structured products.

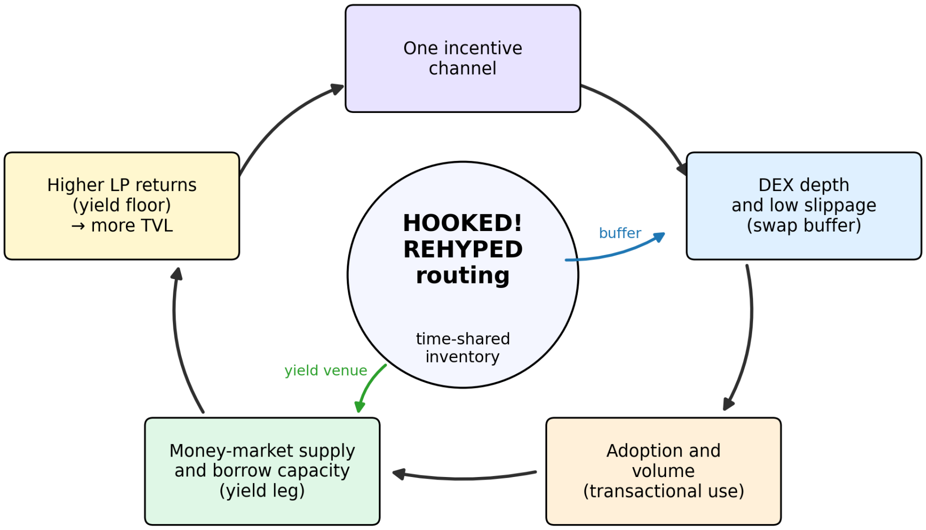

Figure 9.1: DEX and lending flywheel for emerging stablecoins under REHYPED liquidity. A single incentive channel bootstraps pool TVL. Inventory is time-shared between the swap buffer (to provide tight execution) and external yield venues (to grow money-market supply). Improved depth and borrowability reinforce adoption and volume, supporting higher LP returns and attracting more TVL.

These are complements. When a stablecoin is easy to trade, it is easier for borrowers to enter and exit positions. When it is easy to borrow at scale, it becomes more likely to be used as collateral and settlement, which increases overall circulation and trading volume.

9.4 Example: Ethena USDe

Ethena publicly lists integrations with money markets such as Aave as part of its ecosystem.[26] Aave governance has discussed USDe price feed handling and risk controls, reflecting how stablecoin listings become protocol-level concerns when supply is large.[28] Ethena governance has also proposed deploying part of its backing assets into Aave to increase capital efficiency.[27] In parallel, Morpho governance has discussed incentives for USDe and sUSDe markets.[29]

Where HOOKED! fits. A HOOKED! pool can offer USDe transactional depth via concentrated liquidity while routing idle inventory into money markets. This means an incentive aimed at DEX depth can simultaneously increase money-market supply, improving borrowability and leverage adoption without running two separate incentive channels. The stablecoin is not used twice at the same instant; rather, the same LP-owned inventory is time-shared between the two venues as market conditions demand.

9.5 Example: Usual Money USD0

Usual has documented reward programs that target both DEX liquidity (for example, Curve LP activity) and lending or vault activity (for example, Morpho vaults).[31, 30, 32] This “split incentives” pattern is common for emerging stablecoins: DEX depth and lending supply are purchased separately.

9.6 What HOOKED! changes

With REHYPED liquidity, one incentive channel can do double duty:

- incentivize the DEX pool to attract TVL, and
- automatically route idle inventory into lending markets, increasing stablecoin supply where borrowers can access it.

This creates a tighter economic relationship between AMMs and lending markets:

- deeper DEX pools reduce slippage and improve price discovery for the lending market (better entry and exit),
- larger lending supply increases stablecoin usefulness and circulation, feeding back into DEX volume.

For stablecoin issuers, the practical result is improved capital efficiency of incentives: subsidized liquidity remains productive even when swap-fee revenue is low because the external yield leg can continue to pay a baseline return, and the same deployed TVL can simultaneously strengthen money-market depth.

10 Security Considerations

10.1 Primary risks

Key risks include:

- **Smart contract risk** in the hook and vault adapter code paths (including ERC-4626 integrations).
- **External venue risk:** lending markets and strategies can change parameters, pause withdrawals, or hit liquidity constraints, and REHYPED settlement is atomic (external withdrawal failure $\Rightarrow$ swap revert).[14]
- **Oracle and observation integrity.** Oracles are not used to set execution prices or to value deposits; they are used for measurement and (optionally) bounded fee modulation. Manipulation would primarily affect fee setting or circuit-breaker triggers within configured limits, not create balance-sheet insolvency.[4, 17]
- **User execution risk:** MEV and transaction ordering are primarily risks for traders and routers, not protocol solvency. Integrators should apply standard protections (trade splitting, private orderflow, simulation, and fallback routing).[18]

10.2 Mitigations (high level)

Mitigations include conservative venue whitelisting, caps and rate limits, bounded fee parameters (Section 6.2), and the ability to disable idle-split routing (circuit breaker) so inventory remains in the buffer.[74]

11 Conclusion

Stablecoin AMMs are fee-constrained by design: stability reduces the price movement that generates fee-rich flow. HOOKED! makes stablecoin liquidity productive by pairing swap execution with external stable yield, creating a return floor and changing the economics of incentives.

The key consequence is not only higher yield, but improved **incentive efficiency**: subsidized TVL can remain useful after incentives end because a portion of that TVL is routed into lending markets, where it supports borrowing, leverage, and real on-chain activity.

A Notation

Symbol	**Meaning**
C	Total LP-supplied stablecoin inventory (units of stablecoin).
α	Buffer fraction, $C_{\text{buf}} = \alpha C$.
Q_t	Cumulative traded notional through time t.
$\phi(t)$	Swap fee rate through time t.
L	Pool TVL (stable units).
r_f	Fee-based APR.
r_y	External yield APR (net of strategy fees).
Δg	Incremental gas used by routing during a swap.
p_{gas}	Gas price (ETH per gas).
p_{ETH}	ETH price (USD per ETH).

B Glossary (Selected)

TVL	Total value locked: the dollar value of assets deposited in a protocol.
Liquidity mining	Paying token incentives to LPs to rent TVL.
Money market	An on-chain lending market where suppliers earn interest paid by borrowers.
REHYPED liquidity	Fully collateralized inventory routing: LP-owned stablecoins are time-shared between a Uniswap v4 execution buffer and external yield venues accessed via ERC-4626 vault shares. The protocol itself does not create leverage; it reallocates inventory between venues over time.
ERC-4626	A tokenized vault standard used to represent deposits into yield strategies.[13]
Hook	A Uniswap v4 extension point that runs custom logic during pool lifecycle events.[7]
Mercenary liquidity	TVL that appears only while incentives are high and exits quickly when incentives end.

Bibliography

[1] Uniswap Labs. *Uniswap v2 Core.* https://github.com/Uniswap/v2-core.

[2] Adams, Zinsmeister, Robinson, Salem. *Uniswap v3 Core.* https://github.com/Uniswap/v3-core.

[3] Uniswap Labs. *TickMath (Uniswap v3).* https://docs.uniswap.org/contracts/v3/reference/core/libraries/TickMath.

[4] Uniswap Labs. *Oracle Library (Uniswap v3).* https://docs.uniswap.org/contracts/v3/reference/core/libraries/Oracle.

[5] Uniswap Labs. *Uniswap v4.* https://github.com/Uniswap/v4-core.

[6] Uniswap Labs. *Uniswap v4 Core Repository.* https://github.com/Uniswap/v4-core.

[7] Uniswap Labs. *Uniswap v4 Documentation.* https://docs.uniswap.org/contracts/v4/overview.

[8] Angeris and Chitra. *Improved Price Oracles: Constant Function Market Makers.* https://arxiv.org/abs/2003.10001.

[9] Egorov. *StableSwap Invariant.* https://curve.fi/files/stableswap-paper.pdf.

[10] Balancer Labs. *Balancer Whitepaper.* https://balancer.fi/whitepaper.pdf.

[11] Ethereum Foundation. *EIP-20: ERC-20 Token Standard.* https://eips.ethereum.org/EIPS/eip-20.

[12] Ethereum Foundation. *EIP-2612: permit Extension for ERC-20.* https://eips.ethereum.org/EIPS/eip-2612.

[13] Ethereum Foundation. *EIP-4626: Tokenized Vault Standard.* https://eips.ethereum.org/EIPS/eip-4626.

[14] Aave. *Aave v3 Overview.* https://aave.com/docs/aave-v3/overview.

[15] Aave. *Efficiency Mode (E-Mode).* https://aave.com/docs/aave-v3/overview#efficiency-mode-e-mode.

[16] Compound Labs. *Compound: The Money Market Protocol.* https://compound.finance/documents/Compound.Whitepaper.pdf.

[17] Chainlink. *Chainlink Data Feeds Documentation.* https://docs.chain.link/data-feeds.

[18] Daian, Goldfeder, Kell, Li, Zhao, Juels. *Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges.* https://arxiv.org/abs/1904.05234.

[19] DeFiLlama. *Stablecoins Circulating.* https://defillama.com/stablecoins.

[20] Uniswap Governance. *[Governance Proposal] Uniswap Unleashed: Unichain and Uniswap v4 Liquidity Incentives.* https://gov.uniswap.org/t/governance-proposal-uniswap-unleashed-unichain-and-uniswap-v4-liquidity-incentives/25250.

[21] Uniswap Governance (community comment). *Unichain and Uniswap v4 Liquidity Incentives* (discussion of incentive spend versus fee generation). https://gov.uniswap.org/t/governance-proposal-uniswap-unleashed-unichain-and-uniswap-v4-liquidity-incentives/25250/20.

[22] RootData (citing Uniswap Foundation data). *Data: Driven by liquidity incentives, Unichain TVL soared to over 100 million USD* (Apr 17, 2025). https://www.rootdata.com/news/275782.

[23] Dexscreener. *USDC / USD$_T$0 on Unichain (Uniswap v4).* https://dexscreener.com/unichain/0x77ea9d2be50eb3e82b62db928a1bcc573064dd2a14f5026847e755518c8659c9.

[24] Gauntlet. *Incentives Generated $33B in Volume and Foundational Liquidity for Unichain: 3-Month Retro.* https://www.gauntlet.xyz/resources/gauntlet-generated-33b-in-volume-for-unichain.

[25] Aavescan. *Aave v3 Ethereum: Live Rates and Supplied Amounts.* https://aavescan.com/.

[26] Ethena. *Ecosystem.* https://ethena.fi/ecosystem.

[27] Ethena Foundation Governance. *Proposal: Ethena Supplying USDtb to Aave market* (May 12, 2025). https://gov.ethenafoundation.com/t/proposal-ethena-supplying-usdtb-to-aave-market/572.

[28] Aave Governance. *sUSDe and USDe Price Feed Update* (proposal). https://app.aave.com/governance/v3/proposal/?proposalId=262.

[29] Morpho Governance Forum. *Ethena <> Morpho Blue: Incentives for the USDe and sUSDe markets* (Mar 13, 2024). https://forum.morpho.org/t/ethena-morpho-blue-incentives-for-the-usde-and-susde-markets/510.

[30] Usual Money. *Usual Launches New Vault with MEV on Morpho* (Jul 25, 2024). https://usual.money/blog/morpho-vault.

[31] Usual Money Docs. *Pills Campaign Rules.* https://docs.usual.money/archive/usual-airdrop/pills-campaign-rules.

[32] Usual Money. *Ending the USD0/++ Liquidity Transition.* https://usual.money/blog/finalizing-the-usd0-liquidity-transition.

[33] Morpho. *Reward Campaigns on Morpho.* https://docs.morpho.org/build/rewards/concepts/reward-campaigns.

[34] Curve Finance Docs. *Liquidity Gauges.* https://docs.curve.finance/dao/curve_dao/overview/

[35] Curve Finance Docs. *Vote-Escrow (veCRV).* https://docs.curve.finance/dao/vecrv/.

[36] Uniswap Labs. *Uniswap v2 Documentation.* https://docs.uniswap.org/contracts/v2/overview

[37] Uniswap Labs. *Uniswap v3 Documentation.* https://docs.uniswap.org/contracts/v3/overview.

[38] Aave. *Aave Protocol Documentation.* https://aave.com/docs.

[39] Aave. *UiPoolDataProvider (View Contracts).* https://aave.com/docs/aave-v3/smart-contracts/view-contracts.

[40] Aave. *FAQ: Supply caps and reserve limits.* https://aave.com/faq.

[41] Merkl. *Merkl API and Integration Docs.* https://docs.merkl.xyz/.

[42] Aera. *Aera Vaults.* https://aera.finance/.

[43] OpenZeppelin. *ERC-20 Implementation.* https://github.com/OpenZeppelin/openzeppelin-contracts.

[44] Ethereum Foundation. *EIP-712: Typed structured data hashing and signing.* https://eips.ethereum.org/EIPS/eip-712.

[45] Chainlink. *Chainlink: A Decentralized Oracle Network* (whitepaper). https://chain.link/whitepaper.

[46] MakerDAO. *DAI and the Maker Protocol Documentation.* https://docs.makerdao.com/.

[47] Financial Stability Board. *Regulation, Supervision and Oversight of "Global Stablecoin" Arrangements.* https://www.fsb.org/work-of-the-fsb/financial-innovation-and-structural-change/crypto-assets/global-stablecoin-arrangements/.

[48] Bank for International Settlements. *Annual Economic Report (Stablecoins and tokenised money).* https://www.bis.org/publ/arpdf/ar2023e.htm.

[49] Curve Finance Docs. *LP Tokens in Curve Pools.* https://docs.curve.finance/stableswap-exchange/stableswap-ng/pools/lp_token/.

[50] Morpho. *Morpho Documentation.* https://docs.morpho.org/.

[51] Ethena. *Documentation.* https://docs.ethena.fi/.

[52] CoinMarketCap. *Ethena USDe.* https://coinmarketcap.com/currencies/ethena-usde/.

[53] CoinMarketCap. *Top Stablecoin Tokens by Market Capitalization.* https://coinmarketcap.com/view/stablecoin/.

[54] Visa. *Stablecoins and the future of onchain finance.* https://corporate.visa.com/en/solutions/crypto/stablecoins/stablecoins-and-the-future-of-onchain-finance.html.

[55] DeFiLlama. *DeFi Metrics.* https://defillama.com/metrics.

[56] Aavescan. *Supply change dashboard.* https://aavescan.com/base-v3/supply-change.

[57] Uniswap Governance. *Forse Analytics Retrospective Discussion* (linked in Uniswap governance thread). https://gov.uniswap.org/.

[58] Uniswap Labs. *Hooks overview (Uniswap v4).* https://docs.uniswap.org/contracts/v4/concepts/hooks.

[59] Uniswap Labs. *Dynamic fees (Uniswap v4).* https://docs.uniswap.org/contracts/v4/concepts/dynamic-fees.

[60] Uniswap Labs. *Oracle concepts (Uniswap).* https://docs.uniswap.org/concepts/protocol/oracle.

[61] Ethereum Foundation. *EIP-4626 Rationale and References.* https://eips.ethereum.org/EIPS/eip-4626.

[62] Aave. *Protocol Subgraphs.* https://github.com/aave/protocol-subgraphs.

[63] Uniswap Labs. *Uniswap v3 Mathematics.* https://uniswap.org/whitepaper-v3.pdf.

[64] Curve Finance. *Curve Whitepaper and Docs.* https://docs.curve.finance/.

[65] Compound Labs. *Compound Documentation.* https://compound.finance/docs.

[66] Aave. *Security and Audits.* https://aave.com/security.

[67] Uniswap Labs. *Security and Audits.* https://docs.uniswap.org/contracts/v4/overview/security.

[68] Merkl. *Merkl App.* https://app.merkl.xyz/.

[69] Uniswap Labs. *Uniswap Interface.* https://app.uniswap.org/.

[70] Uniswap Labs. *Unichain USDC/USD$_T$0 pool (interface).* https://app.uniswap.org/explore/pools/unichain/0x77ea9d2be50eb3e82b62db928a1bcc573064dd2a14f5026847e755518c8659c9.

[71] Dexscreener. *Dexscreener Documentation.* https://docs.dexscreener.com/.

[72] Velodrome Finance. *Solidly: The first ve(3,3) AMM (reference implementation).* https://github.com/velodrome-finance/solidly.

[73] HOOKED! Contributors. *HOOKED! Pitch Deck.* Internal document (shared with partners), 2026.

[74] HOOKED! Contributors. *HOOKED! Hook Con-*

tract (Hook.sol) and Vault Routing Implementation. (Source code repository / internal release.)

[75] Majestic Labs. *Uniswap USDC/USD$_T$0 Unichain Campaign: Incentive Schedule, TVL Series, and Fee Coverage Reconstruction*. Supplementary analysis accompanying this whitepaper (file: `Uniswap TVL Projection.pdf`).

[76] Etherscan. *Ethereum Gas Tracker*. https://etherscan.io/gastracker.

[77] GeckoTerminal. *USDC/USD$_T$0 on Unichain (Uniswap v4) Pool Page*. https://www.geckoterminal.com/unichain/pools/0x77ea9d2be50eb3e82b62db928a1bcc573064dd2a14f5026847e755518c8659c9.

[78] Majestic Labs. *Data S1: Daily Stablecoin Yield Dataset (2024)*. Supplementary data accompanying this whitepaper (files: `stable-yields-2024.tsv`, `stable-yields-2024.csv`).

[79] HOOKED! Contributors. *HookConfigRegistry.sol (Pool fee and range configuration registry)*. Source code (internal release).

[80] HOOKED! Contributors. *CFMMSwap.sol (Virtual pool swap and dynamic fee logic)*. Source code (internal release).

[81] HOOKED! Contributors. *HookOracleCoordinator.sol (Oracle observation coordinator)*. Source code (internal release).

[82] HOOKED! Contributors. *HookVault.sol (ERC-4626 vault and strategy routing)*. Source code (internal release).