David Mihal (author of GitHub - dmihal/uniswap-liquidity-dao) is ready to start working on a incentivizable UNIv3 pool for wstETH/ETH pair.

Uniswap V3 presents significant improvements in capital efficiency, by allowing liquidity providers to specify a price range to add liquidity at. However, this increased complexity makes it more difficult for users to add liquidity to a pool, and adds additional challenges for liquidity mining/“yield farming” systems.

This proposal will create a simple pool wrapper token on top of the Uniswap V3 wstETH/WETH pool. The goal is to provide a user experience similar to Uniswap V2, while retaining the improved capital efficiency of V3.

Specification

The pool will allow users to deposit wstETH and WETH in a balanced amount and receive LP tokens. The deposited wstETH & WETH will be added to the 0.05% fee pool on Uniswap v3. Liquidity will be distributed across two different ranges: 80% of liquidity will be allocated to the 0.95-1.01 range, while 20% will be allocated to the 0.9-1.03 range.

Users may withdraw their liquidity at any time, redeeming their LP tokens for the underlying liquidity in the pool.

Fee compounding

The pool contract will contain a rebalance function. When called, this function will execute the following logic:

1. Claim all trading fees accrued by both positions
2. Out of claimed fees, attempt to deposit 20% to the wide range & 80% to the tight range
3. With any remaining balance, swap through Uniswap to balance the allocation
4. Deposit the remaining funds into the tight range

Since this function is public and contains a swap, it may be vulnerable to front-running. Therefore, the function will compare the current tick against the TWAP price and fail if the price has moved too quickly.

Implementation

This pool will be based on the existing codebase from theuniswap-liquidity-dao repository.

However, there will be simplifications due to the removal of functionality for adjusting ticks & fee pools. The two liquidity ranges will be hard-coded at the construction of the pool and remain unchangeable.

Additional functionality will be required to allow the pool to handle multiple liquidity ranges. Furthermore, the rebalancing mechanism needs to be completed (the current implementation is very rudimentary) and the TWAP-powered frontrunning protection needs to be built.

The code will be thoroughly tested and documented (commit 81b681a726cae2facbb2a6fd68372c52f0e83ab8

has a good level of testing and documentation, as a benchmark).

A deployment script will be provided.

Compensation

A grant of 7,000 LDO is being requested from the Lido Ecosystem Grant Organization for the development, testing and deployment of this contract. 2,400 will be provided up-front, with the remainder provided at completion.

Future Work

The following items are outside the scope of this grant proposal.

Zaps

Contract to allow the user to deposit/withdraw using stETH & ETH, instead of wstETH & WETH.

LEGO Parameters

By grant size and importance it’s a boulder; so the LEGO council will have to vote on that. My personal opinion is that it’s a very good proposal with the amount of funding fully adequate for the combination of project scope, skill, and domain knowledge David has.