ReHold

Renat Gafarov (re@rehold.io)
Mikhail Semin (michael@rehold.io)

February 9, 2023

Introduction

The ReHold protocol is an algorithmic derivative over Uniswap V3 that allows you to create short trades with high annual returns by setting precise price ranges in concentrated Uniswap V3 Liquidity Pools (LPs). A financial product with a high annual return, but the investment is short-term (up to 24 hours).

We're the first who made the Dual Investments decentralized. Some centralized exchanges have a similar product but our protocol has a much more user-friendly interface and is adapted for beginners who don't have a lot of experience in financial products (especially derivatives).

Dual Investment

Dual Investment is a short-term staking feature offering high APRs. It's a pair of tokens, like ETH/USDT, where you pick only one and lock it for hours depending on the staking plan.

Then, you claim the locked amount plus the guaranteed yield — and it is paid out irrespective of where the price went. After the claim, an output amount will be returned to the user wallet with the yield (which is fixed after the creation of Dual Investment).

Also, you may replay your trade (reinvest) after the staking period ends instead of claiming to optimize your gas costs. After the replay, an output amount with the yield of the finished trade will be the input amount in a new Dual Investment on the same pair of tokens. If the staking plan of the finished trade has been changed, the protocol will find the most similar parameters for the user.

The staking plan includes a base ticker, quote ticker, APR, and staking period in hours. You may start a Dual Investment only with two allowed tickers: base ticker (e.g. ETH) or quote ticker (e.g. USDT).

Protocol has limits for a minimum and maximum amount of each Dual Investment to limit risks of market slippage and it depends on Uniswap V3 Liquidity Pools (LPs) trading volume.

The result of Dual Investment depends on the close price: if the close price is equal to or greater than the entry price, then the output ticker will be the quote ticker; if the close price is less than the entry price, then the output ticker will be the base ticker.

	Input Ticker = Base Ticker	**Input Ticker = Quote Ticker**
Close Price >= Entry Price	$OA_q = IA_b * EP * (1 + SP\frac{APR}{365*24})$	$OA_q = IA_q * (1 + SP\frac{APR}{365*24})$
Close Price < Entry Price	$OA_b = IA_b * (1 + SP\frac{APR}{365*24})$	$OA_q = \frac{IA_q}{EP} * (1 + SP\frac{APR}{365*24})$

IA_b – Input amount in the base ticker
IA_q – Input amount in the quote ticker
OA_b – Output amount in the base ticker
OA_q – Output amount in the quote ticker

b - Base Ticker
q - Quote Ticker

EP - Entry Price
SP - Staking Period

Staking Plan

Parameter	Type	Example	Description
Id	uint256	0	The index of the staking plan
BaseToken	address	0x2170ed0880ac9a755fd29b2688956bd959f933f8	The contract address of the base token
QuoteToken	address	0x55d398326f99059ff775485246999027b3197955	The contract address of the quote token
StakingPeriod	uint256	24	The staking period in hours
Yield	uint256	410959	The yield percent for the selected staking period
Enabled	bool	true	The boolean indicates the staking plan is active

Dual Investment

Parameter	Type	Example	Description
Id	uint256	20	The index of the trade
TariffId	uint256	3	The index of the selected staking plan
User	address	0xc36e66c063db811c025221ec43a7cdc298f8df69	The user address
BaseToken	address	0x2170ed0880ac9a755fd29b2688956bd959f933f8	The contract address of the base token
QuoteToken	address	0x55d398326f99059ff775485246999027b3197955	The contract address of the quote token
StakingPeriod	uint256	24	The staking period in hours

Yield	uint256	410959	The yield percent for the selected staking period
InputToken	address	0x55d398326f99059ff775485246999027b3197955	The contract address of the input token
InputAmount	uint256	1000000000000000000000	The input amount from the user
InputBaseAmount	uint256	0	The input from the user if the input token is the base token
InputQuoteAmount	uint256	1000000000000000000000	The input from the user if the input token is the quote token
OutputToken	Token	0x55d398326f99059ff775485246999027b3197955	The output ticker after the staking period ends
OutputAmount	Int	1004109590000000000000	The output amount after the staking period ends
InitialPrice	uint256	1634385880023995200959	The entry price of ETH/USDT from the Chainlink oracle
ClosedPrice	uint256	1659340612095943689572	The close price of ETH/USDT from the Chainlink oracle
StartedAt	uint256	1674479332	The start timestamp
FinishAt	uint256	1674565732	The close timestamp after which the user can claim or replay the trade
Claimed	bool	true	The boolean indicates the user has claimed or replayed the trade

Referral Program

The referral program works according to the revenue share model as first-touch, that is, the first inviter receives a percentage of the invitee's profit.

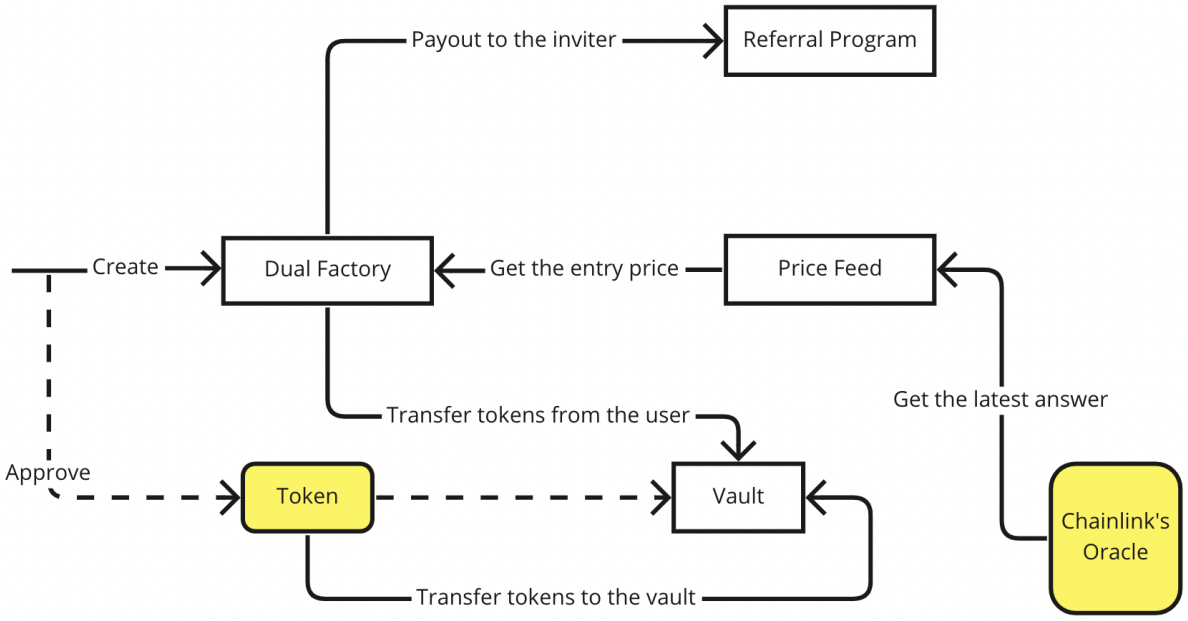
1. User opens the interface via someone's referral URL
   e.g. https://rehold.io/j/A3SA
2. ReHold converts the inviter code into the inviter address and sets *X-Referral-Data* as an array of the inviters' addresses to the cookies of the invitee user:
   a. If the invitee doesn't have *X-Referral-Data* in his cookies, then only one (current) inviter address with an expiration timestamp will be set;
   b. If the invitee has *X-Referral-Data* in his cookies, then the inviter address with an expiration timestamp will be added to the end of the array (the size of the array is limited to **10** inviters' addresses);
   c. The inviter's expiration timestamp will be updated again after the invitee opens the interface via the referral URL. If the inviter's address is expired at the moment when the invitee opened the interface, then the inviter's address will be set to the end of the array;
   d. The *X-Referral-Data* cookie can be tied to the invitee's browser and IP address, in which case all devices connected to the interface from this IP will also receive the same cookies.
3. User connects his wallet to the protocol and should sign a message to verify his address. When a user has signed the message, the protocol will store the inviter's address forever and will not override it after a user will open someone's other referral URL later.
4. User starts a Dual Investment and sends transaction to the blockchain with inviter's address:
   a. The protocol will use the first non-expired inviter's address and which is not equal to the current user address;
   b. If there're no valid inviters' addresses, a user will be marked as non-referral forever.
5. The revenue share percentage is the same for all but could be overridden for the specific inviter in some cases.
6. The inviter's earning is always in USDT and equal to the invitee's Dual Investment yield multiplied by the revenue share percentage;
7. The inviter can claim his earnings at any time without any limits and restrictions.

Architecture

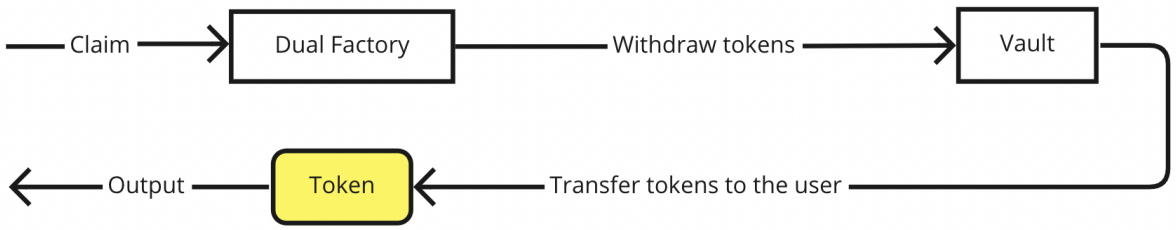
The protocol has 4 well-optimized smart contracts to process users' trades, transfer tokens between users and the protocol, pay earnings from the referral program to inviters, and interact with official Chainlink oracles.

1. **Dual Factory** – the contract for users' operations (such as creating, claiming, or replaying trades), and managing staking plans.
2. **Referral Program** – the contract for inviters' earnings and stats, mapping invitees to inviters, and earnings claiming.
3. **Price Feed** – the contract for interacting with Chainlink price feeds, getting historical and current prices for supported pairs of tokens.
4. **Vault** – the contract for transferring tokens between the protocol and the user, also the storage for users' deposits.

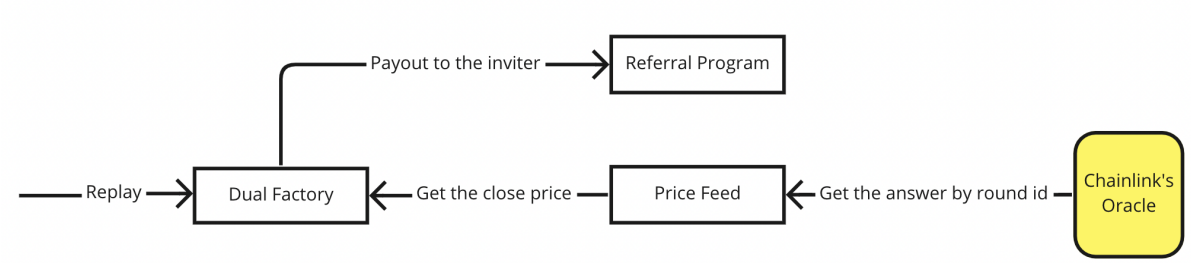
Create Dual Investment



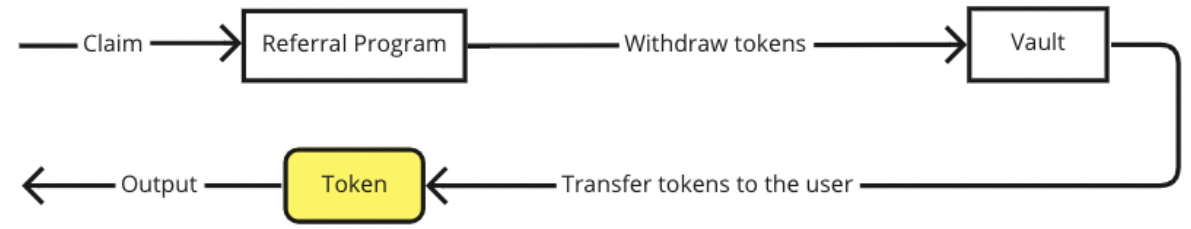
Claim Dual Investment



Replay Dual Investment



Claim Referral Earnings



Execution

The protocol earns from providing liquidity in Uniswap V3 Liquidity Pools (LPs). The latest version of Uniswap has Introduced the concentrated liquidity feature: when you allocate liquidity in a limited price range of the pool curve, you will earn all the fees paid by users trading in this range. ReHold's innovative algorithm allows us to choose the best moment and price range to provide liquidity.

Uniswap V3[1]

The defining idea of Uniswap v3 is that of concentrated liquidity: liquidity bounded within some price range. In earlier versions, liquidity was distributed uniformly along the $x \cdot y = k$ reserves curve, where x and y are the respective reserves of two assets X and Y, and k is a constant. This is simple to implement and allows liquidity to be efficiently aggregated, but means that much of the assets held in a pool are never touched. Having considered this, it seems reasonable to allow LPs to concentrate their liquidity to smaller price ranges than $(0, \infty)$. We call liquidity concentrated to a finite range a position. A position only needs to maintain enough reserves to support trading within its range, and therefore can act like a constant product pool with larger reserves (the virtual reserves) within that range.

---

[1] Uniswap V3 operation description is taken from the original whitepaper (https://uniswap.org/whitepaper-v3.pdf)

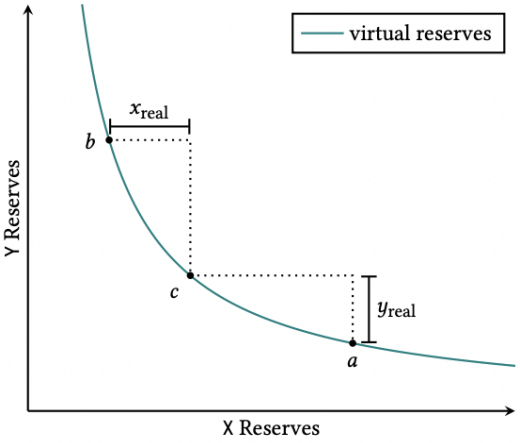


Figure 1: Simulation of Virtual Liquidity

Specifically, a position only needs to hold enough of asset X to cover price movement to its upper bound, because upwards price movement1 corresponds to depletion of the X reserves. Similarly, it only needs to hold enough of asset Y to cover price movement to its lower bound. Fig. 1 depicts this relationship for a position on a range [*pa*, *pb*] and a current price *pc* ∈ [*pa*, *pb*]. *x*real and *y*real denote the position's real reserves. When the price exits a position's range, the position's liquidity is no longer active, and no longer earns fees. At that poInt, its liquidity is composed entirely of a single asset, because the reserves of the other asset must have been entirely depleted. If the price ever reenters the range, the liquidity becomes active again. The amount of liquidity provided can be measured by the value *L*, which is equal to $\sqrt{k}$. The real reserves of a position are described by the curve:

$$(x + \frac{L}{\sqrt{p_b}})(y + L\sqrt{p_a}) = L^2$$

This curve is a translation of a formula such that the position is solvent exactly within its range (Fig. 2).

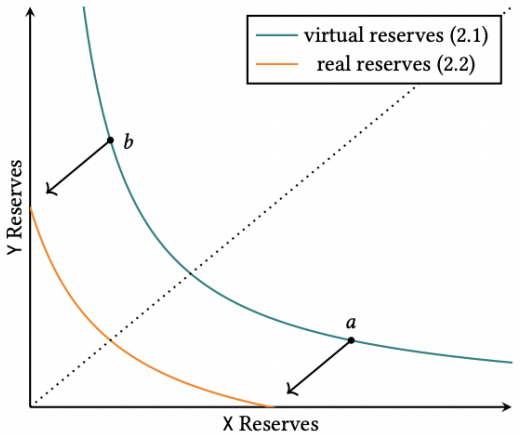
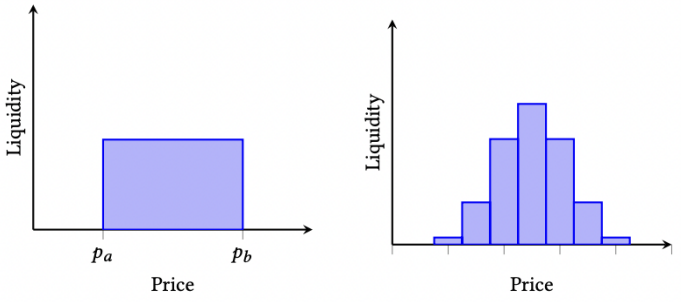


Figure 2: Real Reserves

Liquidity providers are free to create as many positions as they see fit, each on its own price range. In this way, LPs can approximate any desired distribution of liquidity on the price space. Moreover, this serves as a mechanism to let the market decide where liquidity should be allocated. Rational LPs can reduce their capital costs by concentrating their liquidity in a narrow band around the current price, and adding or removing tokens as the price moves to keep their liquidity active.



(II) A single position on $[p_a, p_b]$ **(III)** A collection of custom positions

ReHold

The protocol defines which blockchain (or a few blockchains, if the input amount is large) will bring the greatest yield in Uniswap V3 Liquidity Pools (LPs). The protocol's algorithm picks supported blockchains by Uniswap – Ethereum, Polygon, Optimism, or Arbitrum and splits the Dual Investment input amount between the most suitable Liquidity Pools (LPs).

e.g. ¼ Ethereum + ¼ Polygon + ¼ Optimism + ¼ Arbitrum.

If selected Uniswap V3 Liquidity Pools (LPs) don't have enough volumes or the expected fee earnings are small and the quote ticker of Dual Investment is a stablecoin, then the protocol will try to use different stablecoins to provide liquidity.

To mitigate the risks of price differences between stablecoins, the protocol uses ReHold's stablecoins liquidity (such as USDT/USDC, BUSD/USDT, and others).

To mitigate the risks of imbalance Uniswap V3 Liquidity Pools (LPs) the protocol limits TVL for each Dual Investment and TVL of each pair of tokens (e.g. ETH/USDT).

If the earnings from providing liquidity are greater than the Dual Investment yield to a user, then the rest is a profit for the protocol. If not, the protocol will use its own liquidity to pay a user's yield. The protocol has a mechanism for managing staking plans in different market conditions to avoid frequent losses, it will disable a pair of tokens or update the APR to suitable.