Timeswap V2

Decentralized and oracle-less fixed time preference protocol

June 2022

Ricsson Ngo
ricsson@timeswap.io

Abstract

Timeswap is a fixed time preference protocol for users to manage their ERC20 tokens over discrete time. It works as a zero liquidation fixed maturity money market and options market in one. Users can lend tokens into the pool to earn fixed yields. They can also borrow or leverage tokens against other tokens, without the fear of liquidation. Liquidity providers (different from lenders) create markets for any pair of tokens, adding liquidity, and being the counterparty to all lenders and borrowers of the protocol. In return, they earn transaction fees from both sides of the market. Timeswap utilizes a unique constant sum options specification and a duration weighted constant product automated market maker (AMM) similar to Uniswap AMM. It is designed to not utilize oracles, is capital efficient, permissionless to deploy, game theoretically sound in any state of the market, and is easy to use. It becomes the fundamental time preference primitive lego to build exotic and interesting DeFi products that need discrete time preference.

1 Introduction

Timeswap V2 is an upgrade over Timeswap V1. It has many beneficial properties compared to money markets and options markets in the industry. This makes Timeswap a superior and scalable time preference protocol for the DeFi industry.

1.1 Permissionless

In current money market protocols like Aave and Compound, creation of new lending and borrowing markets requires the permission from centralized entities or DAOs. The permissioned nature of many of these protocols are due to game theoretically unsound design which requires some semi-centralized entities to secure. This is antithetical to the decentralization ethos and democratization of finance.

In Timeswap, liquidity providers can create pools for any ERC20 pair, without permission. It is designed to be generalized and works for any pair of tokens, at any time frame, and at any market state. Each pool is isolated from each other, thus making high quality token pairs safe from less desirable token pools. By utilizing the infinite design space of ERC20 tokens and composability of smart contracts, Timeswap can offer infinite types of discrete time money market and options market.

1.2 Oracle-less

Many money market protocols usually utilize oracles for them to work. This leads to vulnerability to oracle manipulation hacks. This is a fundamental problem of using oracles, which limits decentralization, scalability, and safety of protocols. Using oracles is also expensive and cumbersome to implement for developers.

Timeswap does not use any oracles. Instead it discovers the interest rate and collateral factor through free market arbitrage similar to how Uniswap discovers spot rate. Most importantly, this makes the tokens safe and immune to oracle manipulation attacks.

1.3 Perfect Price Range

The Timeswap V1 triple variable constant product will always yield a positive interest rate no matter how large the lending transaction. It asymptotically goes to zero interest as lending transaction size grows. This is a favorable behavior. But when lending transaction size grows, the collateral factor asymptotically goes to zero, thus can go into under-collateralized territory. While it will still be arbitraged by borrowers, which will always push the collateral factor greater than one hundred percent, it will be preferable that the slippage mechanism for lending is such that the collateral factor stays above one hundred percent.

Timeswap V2 has implemented an ingenious feature where the collateral factor is always over-collateralized no matter how large the lending transactions. Under-collateralized loan by definition is a guaranteed arbitrage. By limiting price range to where it is always over-collateralized, this increases the price efficiency and lower slippage costs for both lenders and borrowers.

1.4 Self Healing

Many money market protocols require time sensitive liquidation. This means that if the price of tokens fell too low or fell too fast, such that liquidation cannot catch up, this leads to the failure of the protocol. It also usually requires centralized intervention to try to revive the protocol.

Well designed free market AMM requires the ability to self heal its state and price no matter what the market price may become or how fast it changes. Protocols like Uniswap, Balancer, and Curve have such behaviors.

Timeswap is not exclusive to this property. It does not matter how fast the spot price, interest price, and collateral factor of the pair goes down or goes up. It does not matter if it is the bear market or bull market. The arbitrage mechanism of Timeswap, will always move the AMM price to where it should be, based on the preference of the free market.

1.5 Symmetric Market

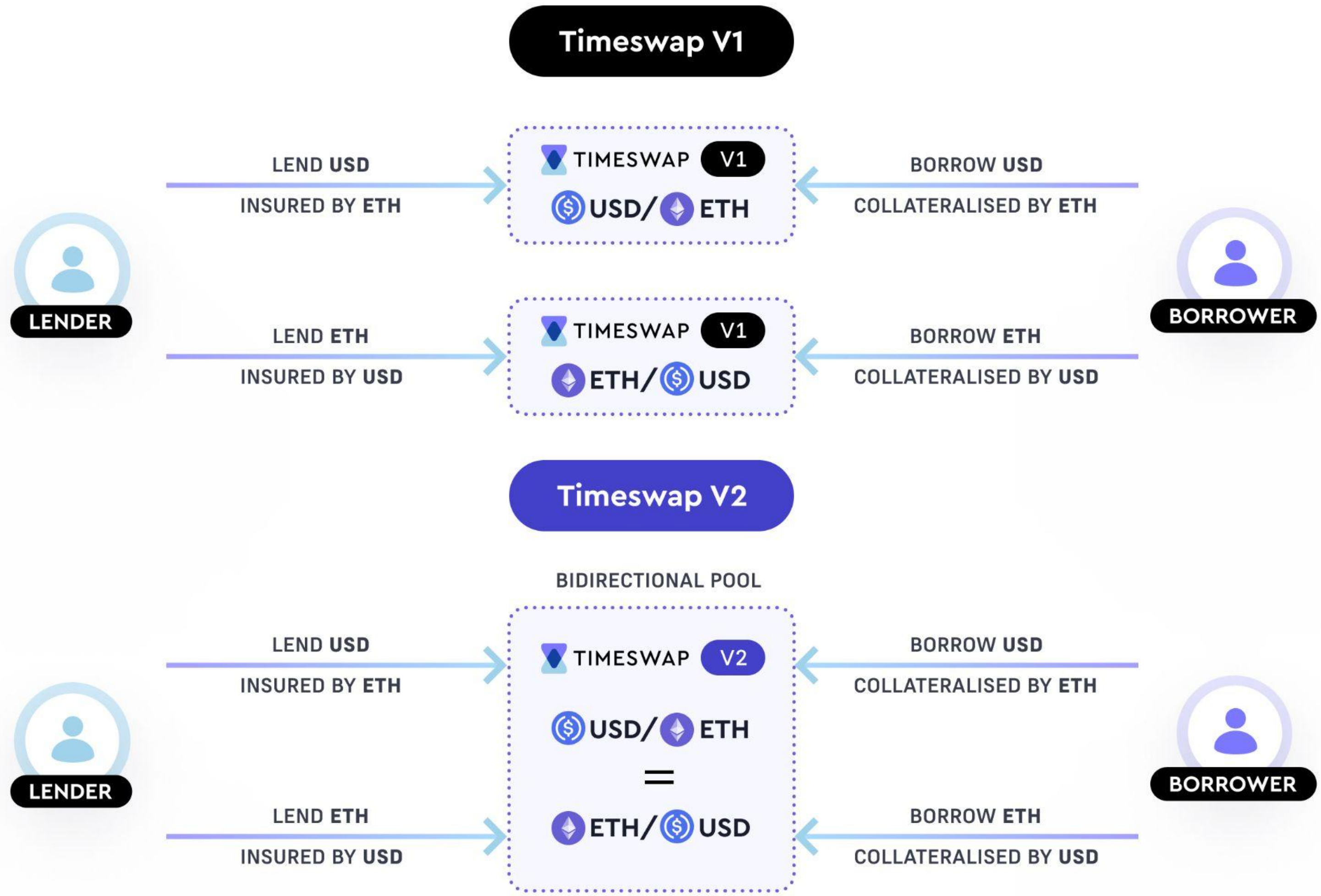
A good game theoretic sound AMM must be as symmetric as possible, where the two opposing sides of the market have opposite transactions that perfectly cancel each other. This leads to efficient pricing for the market. This also makes an elegant and easy way to close and reverse transactions and positions. Uniswap is a great example of symmetric design. Assuming no transaction fee, if traders transact with a Uniswap pool, and then transact in the opposite direction with the same transaction size, they end up with the same tokens they started with. If doing these transactions yields the user less tokens, then users get worse pricing in interacting with the pools. If it yields the user more tokens, then users can siphon funds from the pool.

Timeswap V2 has perfect symmetry for lending and borrowing. Lenders can withdraw their funds before maturity given a small penalty by making equivalent borrowing transactions, while borrowers can close their borrowing positions with discount before maturity by making equivalent lending transactions. Liquidity providers can also withdraw their liquidity before maturity. This gives magnitude order more flexibility to all the users of Timeswap V2.

1.6 Bidirectional Pool

A pair of tokens in Timeswap V1 is not bidirectional. It means that a pool with pair of token A (asset) and token B (collateral) is different from a pool with pair of token B (asset) and token A (collateral). This leads to more liquidity fragmentation. Many money market protocols and option protocols have this problem as well.

In Timeswap V2, the pools are now bidirectional, giving it greater capital efficiency and flexibility. Lenders can lend either token A and/or token B into the same pool, while borrowers can leverage on token A and/or token B in the same pool, using token A and/or token B as collateral. See figure [1] for visualization regarding bidirectionality.



Timeswap V1
LEND USD
INSURED BY ETH
TIMESWAP V1
USD/ETH
BORROW USD
COLLATERALISED BY ETH
LENDER
BORROWER
LEND ETH
INSURED BY USD
TIMESWAP V1
ETH/USD
BORROW ETH
COLLATERALISED BY USD
Timeswap V2
BIDIRECTIONAL POOL
LEND USD
INSURED BY ETH
TIMESWAP V2
USD/ETH
=
ETH/USD
BORROW USD
COLLATERALISED BY ETH
LENDER
BORROWER
LEND ETH
INSURED BY USD
BORROW ETH
COLLATERALISED BY USD

figure [1]

1.7 Gas Efficient

Many DeFi option protocols tend to use Black-Sholes formula on-chain to estimate the pricing of the option. Doing so is very calculation intensive, which will heavily increase gas cost for users. It is the reason why many option protocols cannot survive in the first layer blockchain protocols, and have to resort to deploying on second layer blockchain protocols.

Timeswap will not use the Black-Sholes formula to determine the price of the option. Instead the protocol provides the price based on a simple constant product formula very similar to Uniswap. This makes the protocol more gas efficient. This also makes it very easy for anyone to intuitively create money markets for their tokens, without the need of learning complicated financial formulas.

1.8 Past Independent AMM

Another issue of using Black-Shcoles formula on chain (depending on how it is implemented by the protocol designer) is the historical bias it may imbue on the pricing. As one can see, from protocols like Uniswap, Balancer, and Curve, they do not use price drift of the pair or historical statistical information in the AMM. The price simply follows the present decision of the users and arbitrageurs of the protocol

Timeswap is designed to be past-independent. It does not have any historical data stored in the AMM that determines price, which gives it zero past data bias, and pricing that perfectly follows the present decisions of the free market.

1.9 Capital Efficient Liquidity

In Timeswap V1 triple constant product AMM, it requires liquidity providers to add capital into the pool. It is believed that the amount of liquidity required could become more efficient.

Timeswap V2's new design improves the liquidity capital efficiency by magnitudes, making it more lucrative for liquidity providers to join the protocol. The revenue mechanics and divergent cost mechanics have also been improved to further make liquidity provision more profitable.

1.10 V1 and V2 Comparison

In Timeswap V1, users can make transactions in one pool, flexibly choosing the risk ratio of asset versus collateral for their lend and borrow positions. While this is a great feature, many of the above important characteristics prove to be hard to implement.

Timeswap V2 will remove the features of multiple non-fungible risk ratios in a single pool. This means given the same pair and maturity, there can be multiple pools of different risk ratios with separate capital liquidity. While it increases liquidity fragmentation and decreases user flexibility, the V2 improvements justify the change. This also has the extra benefit of easier financial modeling for the liquidity providers, since there is less complexity due to removing many non-fungible risk positions in a pool. To recreate the experience of having multiple risk positions for users, liquidity providers can create multiple pools of the same maturity instead, and a protocol could be built on top to manage those liquidity positions. See figure [2].

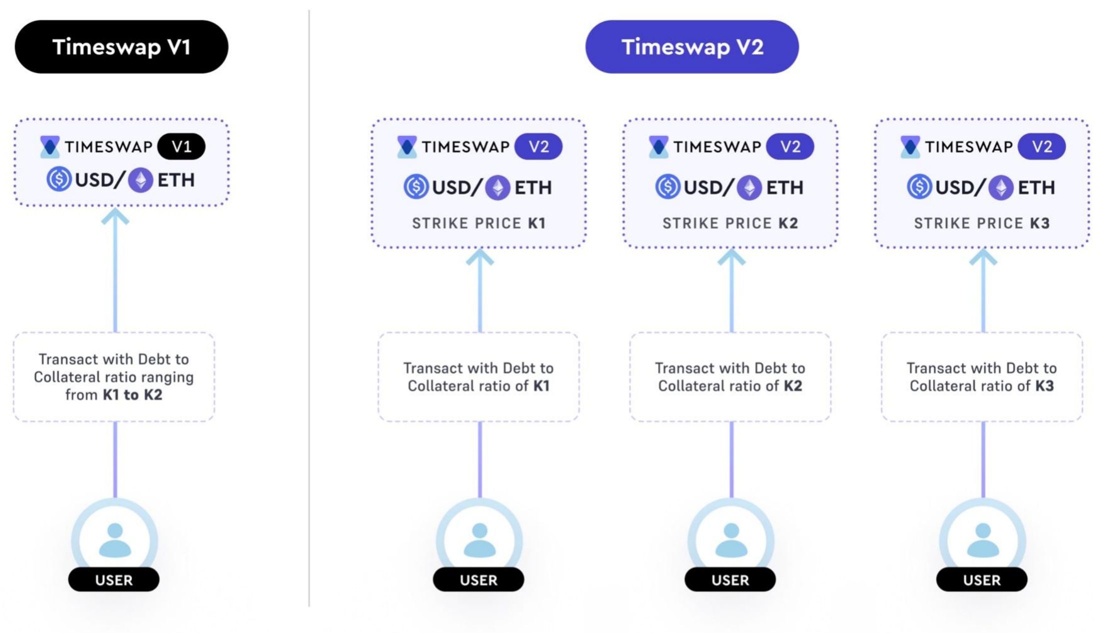


figure [2]

By removing multiple risk ratio positions in a pool. The AMM can now be simplified. Timeswap V2 implemented a unique AMM that is suitable for money markets.

Following are the differences and improvements between V1 and V2:

Characteristics	**Timeswap V1**	**Timeswap V2**
Market Permission	Permissionless for any pair of ERC20 tokens.	Permissionless for any pair of ERC20 tokens.
Oracle usage	Does not use oracles, thus safe from oracle manipulation attacks.	Does not use oracles, thus safe from oracle manipulation attacks.
Price range	Transactions can lead to under-collateralization, thus worst slippage for users.	Transactions are always over-collateralized, leading to less slippage for users.
Self healing	Can move the price to any market price, due to free market arbitrage.	Can move the price to any market price, due to free market arbitrage. More efficient given that it is never under-collateralized.
Symmetricity of market	Slightly asymmetrical, leading to less efficient pricing for users.	Perfectly symmetrical. Gives users the ability to close their lending, borrowing, liquidity provider positions before the maturity of the pool natively.
Bidirectionality	Not bidirectional, thus more liquidity fragmentation.	Bidirectional pool, increasing capital efficiency by double and giving more choices for users.
Gas efficiency	Utilizes a simple AMM for great gas efficiency.	Utilizes a simpler and more effective AMM for better gas efficiency.
Past independency	Does not use any historical data in the AMM.	Does not use any historical data in the AMM.
Capital Efficiency	Capital efficient for liquidity providers.	More capital efficient and better earning potential by more than double for liquidity providers.
Strike Flexibility	Flexible choice of strike per transaction in a pool.	Fixed strike per pool, with separate liquidity. To implement the experience of choosing multiple

		strikes, it requires liquidity providers to create multiple pools.

2 Timeswap Transaction

For simplicity and clarity, the explanations and examples use USD and ETH as the pair of tokens. Note that any pair of ERC20 tokens can work in Timeswap.

Timeswap takes advantage of any DeFi DEX to swap tokens whenever necessary for improved fluidity and capital efficiency of the transactions in Timeswap. It can freely interact with any DEX. Note that DEXs are not needed for Timeswap to function, but integrating DEXs does improve the user experience. For the paper, Uniswap DEX is used as the example DEX that Timeswap utilizes.

2.1 Pool Terminology

Every pool is denominated by these fixed parameters which are decided by the liquidity providers (not lenders) who started the pool.

- Pair of tokens being lent, borrowed, or used as collateral for eg. ETH-USD.
- Transition price (K) or Strike price denominated as Token B over Token A(need to tell which one is asset and which one is collateral
- And what ratio is k), which determines the risk profile of the lending and borrowing transactions for eg. a Transition price of 2,000 USD/ETH.
- Maturity time of the pool with a specific strike price for eg. April 30, 2023, 3:00:00 UTC.

2.2 User Profile

There are three main users of the Timeswap protocol, each having different financial goals. (Note that USD and ETH are interchangeable which means one can use any token as an asset and collateral as the Timeswap v2 pools are bidirectional)

- **Lenders** deposit USDC and receive *lending positions* for a higher USDC return after maturity, with an over-collateralized amount of ETH insurance in case they don't receive USDC returns.
- **Borrowers** deposit and lock either ETH as collateral, receive USDC as the amount borrowed, and receive *borrowing positions* where they can unlock and withdraw the ETH collateral, as long as they pay back USDC debt before maturity.
- **Liquidity providers** create Timeswap pools and add liquidity into the pools to facilitate lending and borrowing transactions. They provide both *lending positions* and *borrowing positions* into the pool. They earn transaction fees from both lenders and borrowers.

  The key difference between lenders and liquidity providers are as follows:

  - Lenders receive fixed *lending positions* (thus fixed USDC return and ETH insurance).
  - Liquidity providers have liquidity positions holding both *lending positions* and *borrowing positions*, in which the amount changes every time lenders and borrowers interact with the pool (thus they take on impermanent loss, which is why transaction fees are needed to make adding liquidity worthwhile).

2.4 Transition Price / Strike Price (K)

Transition Price or **Strike Price** determines the risk profile of a Timeswap pool. It's very important for users to understand what this price entails to fully understand the risk and reward profile of their lending and borrowing positions.
The Transition Price/Strike Price (K) is the ratio between the following:

- The number of USDC returns over the ETH insurance of *lend positions*.
- The number of USDC debt over the ETH collateral locked of *borrowing positions*.

2.5 Lender’s Perspective

Suppose that Alice wants to earn a safe yield on her 1,000 USD in a timeframe of 1 year. She chooses a USD-ETH pool with a transition price / strike price of 400 USD per ETH and maturity of 1 year. The current spot price is 2000 USD per ETH, thus she deems a transition price of 400 USD per ETH safe enough, as the possibility of the spot price going below 400 USD per ETH in 1 year is low. She deposits 1,000 USD into the protocol and receives an ERC1155 token representing lending position with returns of 1,098.76 USD and collateral insurance of 2.7469 ETH ($\frac{1098.76}{2.7469} = 400$ Transition Price or Strike Price).

Suppose after 1 year, the spot price is 1,800 USD per ETH, which is above the Transition price of 400 USD per ETH. Her ERC1155, representing the lending position, gives her 1,098.76 USD.

Alternatively, suppose after 1 year, the spot price falls to 300 USD per ETH, which is below the Transition / Strike price. Her ERC1155 token gives her 2.7469 ETH. See figure [3].

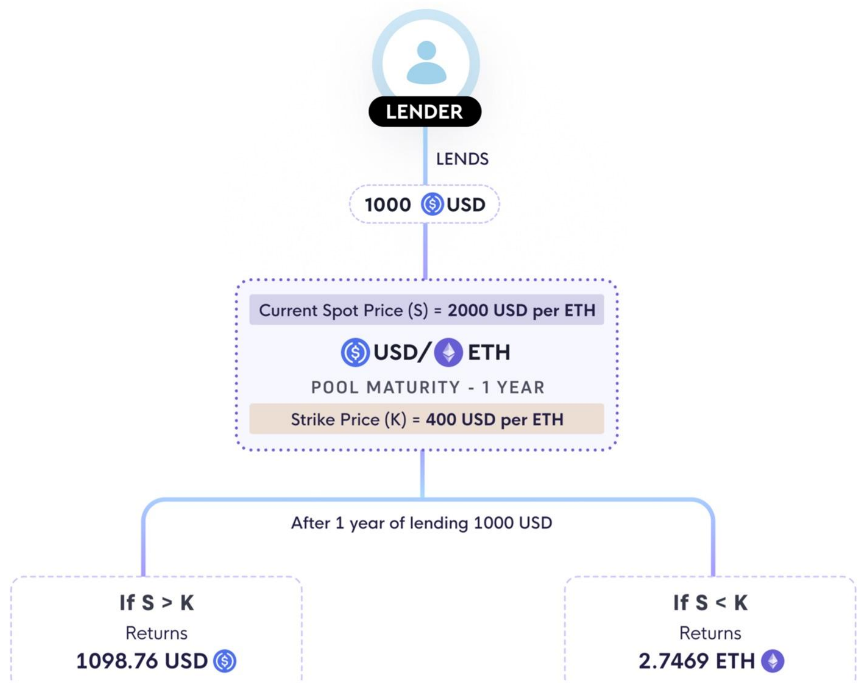


figure [3]

Suppose instead of waiting for 1 year, in 6 months, Alice wants to withdraw from her lending position earlier than the 1 year maturity. She deposits the ERC1155 token back into the protocol, and she receives 1,045 USD from the pool. As one can see, she has received USD amount smaller than 1,098.76 USD. This is due to withdrawing earlier.

NOTE: Please refer to Lending example as given in Section 3.2.5 for numerical calculations of the above example

2.2 Borrower's Perspective

Suppose that Bob wants to borrow 1,000 USD with some ETH as collateral in a timeframe of 1 year. He chooses the same USD-ETH pool with a Transition / Strike price of 400 USD per ETH and maturity of 1 year. The current spot price is also 2,000 USD per ETH. By choosing a low Transition /Strikeprice of 400 USD per ETH, he expects to deposit a greater amount of ETH as collateral, but in return he expects a lesser interest rate. He withdraws 1,000 USD from the protocol, but in return the protocol asks for 2.7469 ETH as collateral, which he deposits. He finally receives an ERC20 token representing debt of 1,098.76 USD and collateral locked of 2.7469 ETH ($\frac{1098.76}{2.7469} = 400$ Transition/Strike Price).

Since he has collateral locked denominated in ETH, and debt denominated in USD, he is hoping that the price of ETH will go up at maturity. Suppose sometime before maturity, the spot price is still 2,000 USD per ETH, but he believes that the price of ETH will suddenly go down below the Transition price of 400 USD per ETH. He can transform his borrowing position to the opposite direction. This is done by withdrawing 2.7469 ETH collateral and depositing 1,098.76 USD as the new collateral locked. His position will turn to having debt of 2.7469 ETH and collateral locked of 1,098.76 USD.

Suppose that a few days after the transformation of his borrowing position (still before the maturity), the spot price actually falls down to 200 USD per ETH, but now he believes the ETH price will climb back up again. He can transform his borrowing position back by depositing 2.7469 ETH collateral back into the protocol and withdraw the 1,098.76 USD. Bob can do this process however many times he wants.

Suppose instead of waiting for 1 year to pay the debt, in 6 months, he decides to fully close his borrowing position. He has no interest in leveraging towards ETH or towards USD anymore. He deposits the ERC1155 token back into the protocol, pays back 1,045 USD into the pool, and he receives 2.7469 ETH collateral from the pool. As one can see, he does not need to pay the full 1,098.76 USD amount, since he gets a discount for paying earlier than the 1 year maturity.

NOTE: Please refer to Lending example as given in Section 3.2.8 for numerical calculations of the above example

2.3 Liquidity Provider's Perspective

Suppose Dan wants to earn yield by market making. He believes there will be a large volume of lending and borrowing in the USD-ETH pool with a Transition price of 400 USD per ETH and maturity of 1 year. He can earn fees from the large volume of transactions. He can choose either USD, ETH, or a combination of both as liquidity to be added. He decided to deposit 1,000 USD. He receives an ERC1155 token representing his liquidity position, as well as some ERC1155 token representing either excess lending position or excess borrowing position.

Suppose after 6 months, the number of lending and borrowing transactions are large. Dan wants to withdraw his liquidity provision position. He deposits his liquidity provision ERC1155, to withdraw 1,230 USD, thus earning a decent profit.

Alternatively after 6 months, the number of lending and borrowing transactions are too few. Dan withdraws his liquidity provision position, by depositing his NFT into the pool and ERC20 token, but receives only 980 USD, thus taking some divergent loss. ***Divergent loss*** is the difference between when the LP are holding tokens in an AMM (Automated Market Maker) Liquidity Pool and just simply holding the lending and borrowing positions (i.e. HODLing) on their balance.

3 Technical Details

From this section onwards are technical details on how Timeswap works under the hood. The section above are what lenders, borrowers, and liquidity providers will experience when interacting with Timeswap through a good frontend decentralized web application. This section onwards is specifically for developers and analysts, but will be a great read for those who want to understand the inner workings of Timeswap.

The Timeswap protocol is built with two main components, with an optional third component being another DEX like Uniswap. The first component is the Timeswap Option, which is a unique option implementation never before seen in the DeFi industry. It is financially designed from first principles instead of trying to follow how TradFi implements options. The lending and borrowing positions that lenders and borrowers receive respectively is based on this options. The second component is the Timeswap Pool, which utilizes an ingenious AMM similar to Uniswap AMM. It is specifically designed to create a perfect price range specifically for lending, borrowing, and options trading purposes. The third optional component is a DEX, where swaps can make interacting with Timeswap more convenient and capital efficient.

There are two other auxiliary components namely ***Timeswap Token*** and ***Timeswap Liquidity Token*** which are ERC1155 contracts that let the protocol represent the lending positions, borrowing positions, and liquidity positions as ERC1155 tokens for better composability with other contracts.

3.1 Timeswap Constant Sum Option

The European option is the right but not the obligation to swap tokens at a predetermined strike rate at a predetermined time. For example, if a user owns an option to purchase 1 ETH with a fixed 1,000 USD in 1 year; then in exactly 1 year, the user can choose to purchase the 1 ETH paying 1,000 USD or choose not to do so. It is the fundamental building block to create any fixed maturity financial product like lending and borrowing products. There are two caveats to consider with implementing the European option in Ethereum. Firstly, options must be fully collateralized to guarantee the right to swap tokens; Any under-collateralized option cannot fully guarantee the right to swap. Secondly, token payment to swap must be deposited before the expiry of the option, since there is no exact time payment in Ethereum. This is the reason why pretty much all current options protocols in Ethereum are American options (which is the right but not the obligation to swap tokens at a predetermined strike rate before a predetermined time).

The reason why users would purchase the European option over the American option is because they do not need the time optionality. They have a strong fixed time preference. Due to not having time optionality, the European option is cheaper than the same American option counterpart.

The way to implement the European option in Ethereum is to reframe its definition. Suppose we have a European call option to purchase 1 ETH for 1000 USD in 1 year. To guarantee the right to exercise this option, 1 ETH must be locked, never to be spent or used. In 1 year, the owner can exercise the option, thus withdrawing 1 ETH, and depositing 1000 USD. The end state of the European call option at maturity is either 1 ETH is left or 1000 USD is left. Interestingly, the European put option to sell 1 ETH for 1000 USD in 1 year, also has the same end state of having 1 ETH left or 1000 USD left. American call option on the other hand can be exercised earlier than the maturity, the 1000 USD that is deposited already goes to the counterparty to utilize earlier than the maturity. So American option's end state is different from European option's end state. Given these insights, the reframing of the European option is as follows: Either 1 ETH or 1000 USD must be locked at anytime before the maturity, the owner has the right but not the obligation to withdraw 1 ETH or 1000 USD, whichever is currently locked, by depositing 1000 USD or 1 ETH to be locked respectively. The owner can do this however many times before the maturity. This has the same end state as both the European call and put option.

3.1.1 Option Implementation

Timeswap Constant Sum Option is the closest implementation of the European option (Not an American option) given the requirements of decentralization and caveats of writing smart contracts. The option is fully collateralized with either ETH or USD. When the option has locked ETH, then it gives the holder the right but not the obligation to swap USD for ETH before the maturity of the option. When the option has locked USD, then the holder can swap ETH for USD. When the swap is exercised, the token that is deposited becomes the new locked token of the option, thus giving the holder the right but not the obligation to swap the tokens back before maturity. This design gives the holder the ability to swap ETH and USD back and forth as many times as the holder wants, where the fixed ratio of the amount of ETH and amount of USD swapped is defined as the strike rate.

Another way to look at this option is that owners of the option can withdraw and deposit any of the locked tokens, as long as the resulting amount of locked tokens follow a constant sum. This can be done however many times the owners want within a fixed duration. This is the inspiration for the name Constant Sum Option.

Holding a Timeswap Constant Sum Option is defined as holding Long. There are two states of the option: **Long ETH** when ETH is locked in the contract, and **Long USD** when USD is locked in the contract. *Long positions are borrowing positions in Timeswap.* There is no concept of long call or long put in the Timeswap option, as they are both the two states of an option.

Being the counterparty to the option is defined as holding Short. They receive either ETH or USD whichever is left locked in the contract at the expiry of the option. *Short positions are lending positions in Timeswap.* Again, there is no distinction between short call and short put for the same reason as above. If there are multiple parties holding long, and they decided differently on leaving ETH and USD in the contract. The short holders simply get pro rata of both ETH and USD.

Because of the game theoretically sound design, anyone can mint the option, where an equivalent number of long positions and short positions are minted, as well as the correct amount of ETH or USD is locked.

Let K be the strike/transition price of the option. Also for convenience, let S be the spot price of a DEX. All explanations onwards will use these two prices.

Following is an example of a Timeswap Constant Sum Option with possible transactions that users can make:

A pair of ERC20 tokens follow a specific format where token0 is always the ERC20 tokens that has a smaller address size than token1. *The strike price/transition price (K) is always defined in token1 over token0.* Suppose there is a Constant Sum Option contract with pair ETH/USD (ETH is token0, and USD is token1), strike/transition price of K USD per ETH, and maturity in 1 year.

There are three different ERC1155 tokens from this contract: **Long ETH, Long USD, and Short**. Long ETH is denominated as ETH, for example, 100 Long ETH means that there are 100 ETH locked in the contract. Same for Long USD which is denominated in USD. For Short it can either be denominated in ETH or USD, depending on the value of K. In the smart contract, K is stored as an UQ128.128 uint256 value. If $K \geq 1$ then Short is denominated as ETH (token0). If $K < 1$ then Short is denominated as USD (token1). **For simplicity in the examples onward supposed that $K \geq 1$.**

Any user/contract can mint 1 Long ETH and 1 Short, when it locks 1 ETH into the option contract before maturity. See figure [4].

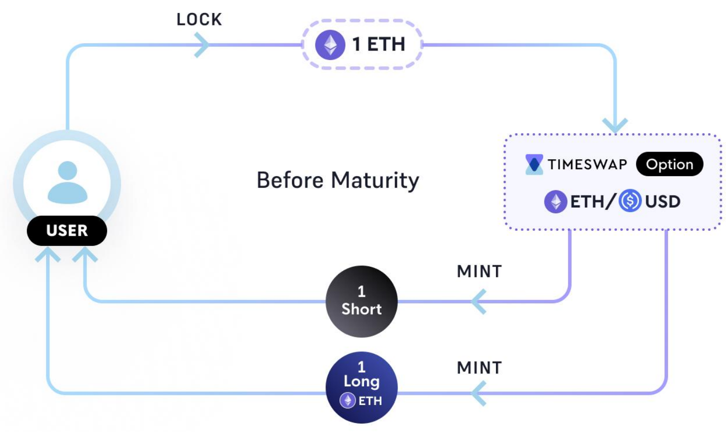


figure [4]

Any user/contract can mint K Long USD and 1 Short, when it locks K USD into the contract before maturity. This can only be called before maturity. See figure [5].

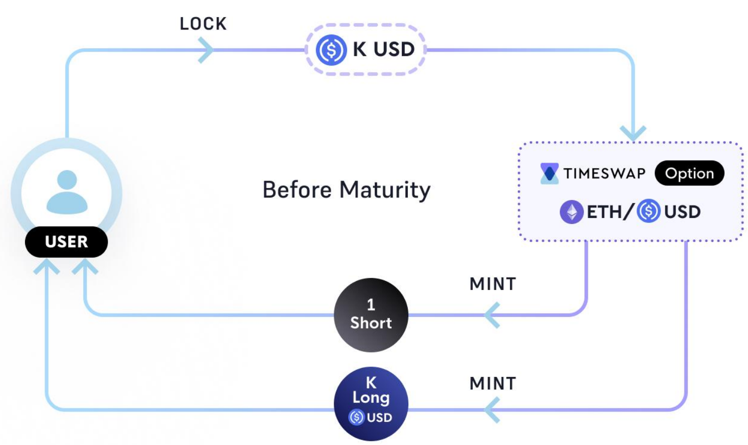


figure [5]

When a contract/user owns 1 Long ETH, it can lock K USD to withdraw 1 ETH before maturity, then the 1 Long ETH turns into K Long USD. This can only be called before maturity. See figure [6].

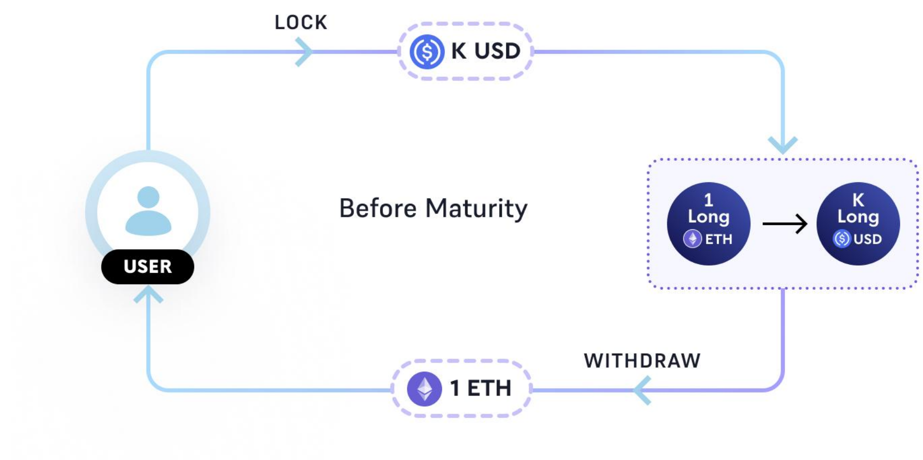


figure [6]

When a contract/user owns K Long USD, then it can lock 1 ETH to withdraw K USD before maturity, then the K Long USD turns into 1 Long ETH. This can only be called before maturity. See figure [7].

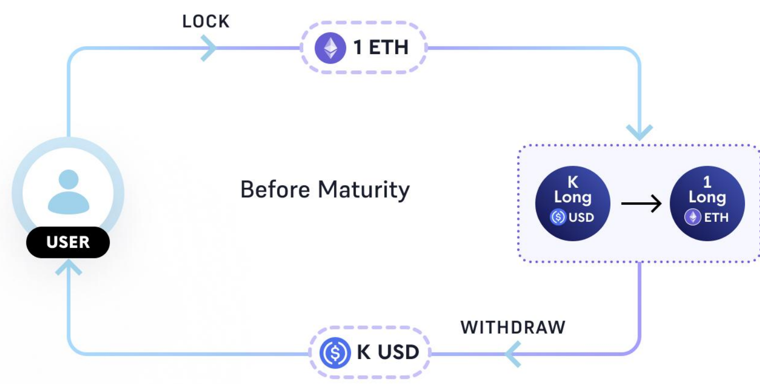


figure [7]

When a contract/user owns 1 Long ETH and 1 Short, it can burn both to withdraw 1 ETH before maturity. See figure [8].

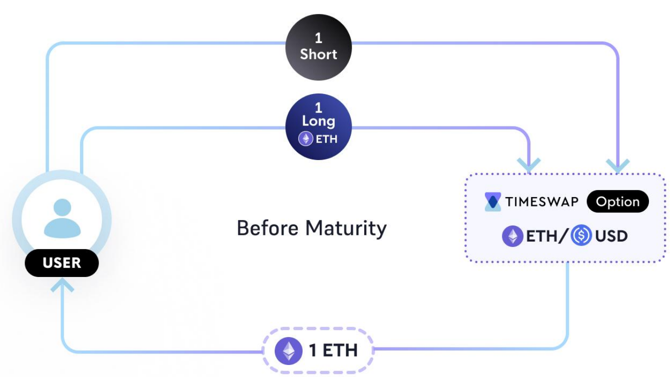


figure [8]

When a contract/user owns K Long USD and 1 Short, it can burn both to withdraw K USD before maturity. See figure [9].

figure [9]

When a contract/user owns 1 Short, it can either receive 1 ETH or K USD (or pro rata of both) after maturity, based on the decisions of holders of Long ETH and Long USD. See figure [10].

figure [10]

The Timeswap Constant Sum Option is a subset of Timeswap Option, which are options specification where owners have the right but not the obligation to withdraw and deposit tokens from the option at any time and at however many times the owners want, for a fixed duration, as long as the resulting locked tokens follow a specific formula.

The Timeswap Option family is designed to be the option primitives in Ethereum. Constant Sum Option is the base layer that Timeswap money market is built on.

3.1.2 Option Analysis

All Short, Long ETH, and Long USD have positive values, as all of them are fully collateralized by ETH and USD. Due to the ability to freely mint and burn these option tokens, 1 Long ETH plus 1 Short is financially equivalent to 1 ETH. Similarly, K Long USD plus 1 Short is financially equivalent to K USD (K is the strike price of the option).

$$Short + Long\ ETH = ETH \quad (1)$$

$$Short + K \cdot Long\ USD = K \cdot USD \quad (2)$$

Put-Call Parity is a very important principle regarding the value relationship between a European call option and a European put option. Both equation (1) and (2) combined, replacing ETH with its spot price equivalent of USD ($ETH = S \cdot USD$)

$$Long\ ETH + K \cdot USD = K \cdot Long\ USD + S \cdot USD \quad (3)$$

This perfectly aligns with the Put-Call Parity. Therefore, when borrowers receive either Long USD or Long ETH, but want the other position, they can use the Timeswap option mechanics (see section 3.1.1) to transform it to the position they want, exactly following the Put-Call Parity.

Due to equations (1) and (2), the value of the options must be

$$Short \leq min(ETH, K \cdot USD) \quad (4)$$

$$Short \leq min(S, K)USD \quad (5)$$

since if $Short > min(ETH, K \cdot USD)$, then $Long\ ETH < 0$ and/or $Long\ USD < 0$, which is a contradiction to the equations (1) and (2), therefore proving equation (4) by contradiction.

Furthermore, due to all the equations (1), (2), and (4), the possible price range between Short and Long USD/Long ETH leads to the following

$$\frac{Short}{min(K\ Long\ ETH, Long\ USD)} \in [0, \infty) \quad (6)$$

Another important insight due to equations (1), (2), and (4) is that the value of these tokens follow

if $K < S$, then K $Long\ USD < 1\ Long\ ETH$ (7)

if $K > S$, then K $Long\ USD > 1\ Long\ ETH$ (8)

Note that if $K = S$, then $K\ Long\ USD = 1\ Long\ ETH$, as they can be interchanged with each other with no loss or gain.

These are important price behaviors that are taken into account when designing the Timeswap AMM formula.

3.2 Timeswap Pool

A Timeswap pool is a contract that gives the user the ability to swap Long ETH and/or Long USD for Short, and vice versa.

For lending and selling options, the contract/user locks either ETH or USD in Timeswap Options contract to mint Short and Long ETH/Long USD (See section 3.1.1). The pool receives the Long ETH/Long USD minted, then transfers more Short to the contract/user. This is done in one single transaction.

For borrowing and buying options, the contract/user locks either ETH or USD in Timeswap Options contract to mint Short and Long ETH/Long USD. The pool receives the Short minted, then transfers more Long ETH/Long USD to the user. This is done in one single transaction.

For adding liquidity, the contract/user locks either ETH or USD in Timeswap Options contract to mint Short and Long ETH/Long USD. The pool receives both Short and Long ETH/Long USD, following a duration weighted constant product AMM ratio (explained more in the next section). If more Short is added than Long ETH/Long USD, then the contract/user will have some excess Long ETH/Long USD in its balance. Similarly, if more Long ETH/Long USD is added than Short, then the contract/user will have some excess Short in its balance. In the case it is the first to add liquidity, it can add at any ratio of Short and Long ETH/Long USD it wants, thus initializing the interest rate of the pool. This is also done in one single transaction.

As one can see, all three products are fundamentally the same transactions with the difference in how tokens are transferred. A DEX is used to make these transactions more fluid, whenever necessary.

Concentrated liquidity system can be added into Timeswap pool AMM for higher capital efficiency. It will be implemented for future versions. The focus is to see the effectiveness of the base AMM of Timeswap.

3.2.1 Timeswap Automated Market Maker

A Timeswap pool uses the Duration Weighted Constant Product automated market maker (AMM) similar to Uniswap. It is designed specifically for pricing of Timeswap options. Money market transactions will be shown as an example of interacting with the pool.

First, a Timeswap option smart contract must be deployed, which has a parameter of a pair of tokens (ETH/USD is the same as USD/ETH), the maturity in Unix timestamp, and a fixed strike rate between ETH and USD (Which token from the pair is first or second is arbitrary and will not affect the logic of the contract).

The Uniswap protocol can be used in tandem with Timeswap to make more fluid transactions.

A Timeswap pool has a parameter of a Timeswap option, thus inheriting the parameters of the option. A pool holds three reserves that determines the price of swapping option positions.

Let x be the reserves of Long ETH, converted to the same denomination of Short following K ratio, where K is the strike price, if needed.
Let y be the reserves of Long USD, converted to the same denomination of Short following K ratio, if needed.
Let z be the reserves of Short per second for the duration of the option.
Let d be the duration of the pool, thus dz is the total number of Short in the pool.
Let L be the square root of the constant product of the AMM. ($k = L^2$)
Let I be the marginal interest rate per second of the Short per total Long.

$$(x + y)z = L^2 \ (9)$$

$$\frac{z}{x+y} = I \ (10)$$

If $K \geq 1$, then $y = \frac{y^*}{K}$, where y^* is the reserves of Long USD and x is the reserves of Long ETH. If $K < 1$, then $x = Kx^*$, where x^* is the reserves of Long ETH and y is the reserves of Long USD. ***(Note that for examples onward, suppose*** $K \geq 1$***)***

3.2.2 Rebalance Transaction

One fundamental mechanics of the Timeswap pool other than the lend, borrow, and add liquidity functions is the rebalance function. Details are as follows:

Please note that for all the examples below K is the strike price of a pool.

Let Δx be the number of Long ETH deposited into the pool.
Let Δy be the number of Long USD divided by K withdrawn from the pool.

$$(x + y + \Delta x - \Delta y)z = L^2 \ (11)$$

Let Δx be the number of Long ETH withdrawn from the pool.
Let Δy be the number of Long USD divided by K deposited into the pool.

$$(x + y - \Delta x + \Delta y)z = L^2 \quad (12)$$

As one can see, ignoring z which does not change in this context, it is a constant sum formula. Anyone can withdraw Long ETH as long as the equivalent multiple of K Long USD is deposited. Similarly, anyone can withdraw Long USD as long as equivalent multiple of $\frac{1}{K}$Long ETH is deposited.

Constant sum formula in DeFi is used as a swap for pairs of tokens that should have the same value, for example, between two USD stablecoins. But in Timeswap, constant sum is used for a very different purpose.

Due to equations (7) and (8), the rebalance function incentivizes arbitrageurs to withdraw all the Long ETH or Long USD, whichever has higher value, and replace it with K converted equivalent of the other which has lower value. Therefore, it is expected that Timeswap pool will either have only Long ETH or only Long USD in it, whichever has lower market value. See both figure [11] and figure [12] for the diagram of the inner transactions for rebalancing.

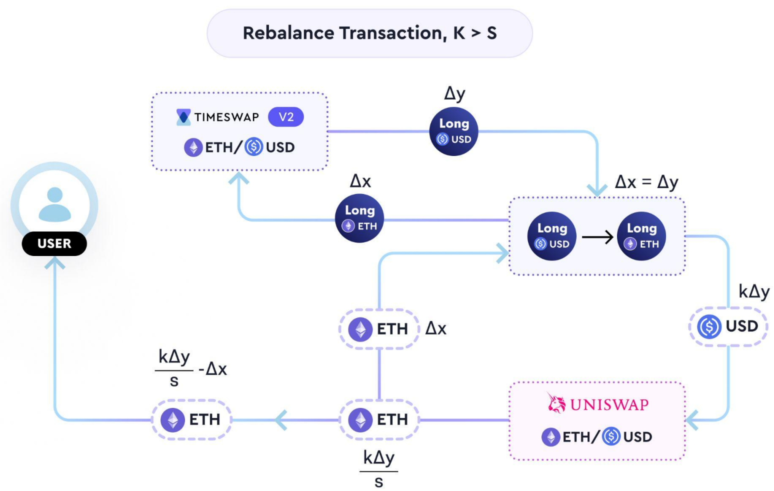


figure [11]

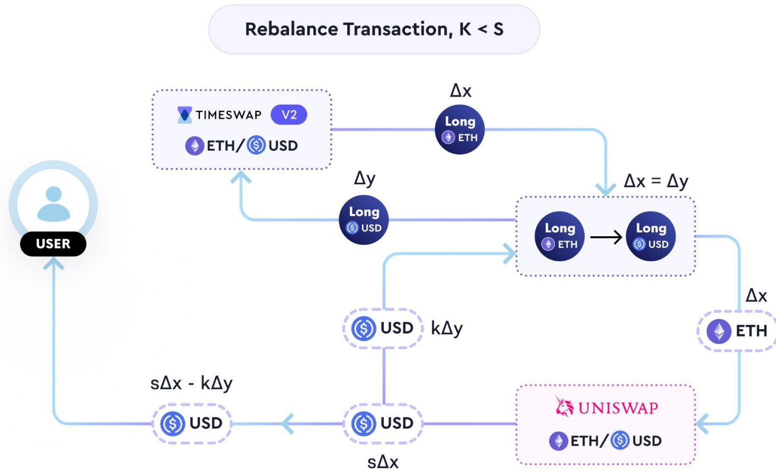


figure [12]

When $K < S$, then it is expected that $x = 0$, and only Long USD will be in the pool. Thus equation (9) can be simplified to

$$yz = L^2 \quad (13)$$

While when $K > S$, then $y = 0$, and only Long ETH will be in the pool. Thus equation (9) can be simplified to

$$xz = L^2 \quad (14)$$

As one can see, both equations (13) and (14), are constant product formulas, same as Uniswap's AMM formula ($xy = L^2$), without the concentrated liquidity (which will be added in the future iteration of Timeswap). Due to using the same formula, both equations (13) and (14) will have the same behavior as a Uniswap pool (except for the additional duration mechanism of z in Timeswap).

One notable AMM behavior from Uniswap is the price range of $[0, \infty)$ between the pair being swapped. This will be the same case for Timeswap as well for equation (13) and (14)

$$\text{if } K < S, \text{ then } \frac{y}{z} \in [0, \infty) \text{ (15)}$$
$$\text{if } K > S, \text{ then } \frac{x}{z} \in [0, \infty) \text{ (16)}$$

This price range aligns perfectly with the expected price range of Short and Long ETH/Long USD positions in equation (6).

3.2.3 Rebalancing Example

Supposed there are 100 Long ETH and 100,000 Long USD ($x = 100$, $y = 100$) in a 1,000 USD per ETH strike rate pool. Suppose again the current spot rate is 1,800 USD per ETH. This is the case where $K < S$. An arbitrageur will call the rebalance function, which will have the contract do the following transactions:

1. Withdraw 100 Long ETH ($\Delta x = 100$) from the Timeswap pool.
2. Using the Timeswap option mechanics (see section 3.1.1) to turn 100 Long ETH to 100,000 Long USD by depositing 100,000 USD and withdrawing 100 ETH.
3. The 100 ETH withdrawn from the 100 Long ETH is swapped through the DEX, thus receiving 180,000 USD. The 100,000 USD that was deposited to turn 100 Long ETH to 100 Long USD, comes from the 180,000 USD from transacting with DEX.
4. The 100,000 Long USD ($\Delta y = 100$) is then deposited back into the Timeswap pool.

The end result is as follow:

- The arbitrageur has a total profit of 80,000 USD for making this transaction.
- The Timeswap pool will now have 200,000 Long USD ($x = 0, y = 200$).

Suppose that the spot price decreased to 500 USD per ETH suddenly. An arbitrageur will call the rebalance function, which will have the contract do the following transactions:

1. Withdraw 200,000 Long USD ($\Delta y = 200$) from the Timeswap pool.
2. Using the Timeswap option mechanics (see section 3.1.1) to turn 200,000 Long USD to 200 Long ETH by depositing 200 ETH and withdrawing 200,000 USD.
3. The 200,000 USD withdrawn from the 200,000 Long USD is swapped through the DEX, thus receiving 400 ETH. The 200 ETH that was deposited to turn 200,000 Long USD to 200 Long ETH, comes from the 400 ETH from transacting with DEX.
4. The 200 Long ETH ($\Delta x = 200$) is then deposited back into the Timeswap pool.

The end result is as follow:

- The arbitrageur has a total profit of 200 ETH for making this transaction.
- The Timeswap pool will now have 200 Long ETH ($x = 200$, $y = 0$).

As one can see, it is expected rebalancing arbitrage happens whenever spot price crosses the strike price of the Constant Sum Option. So it is expected that as long as spot rate is not equal to the strike rate, one of x or y will be zero.

3.2.4 Lend Transaction

For lenders to get Short (lending positions) by depositing ETH and/or USD, the contract does the following:

Let Δx be the number of Long ETH deposited into the pool.
Let Δy be the number of Long USD divided by K deposited into the pool.
Let Δz be the number of Short per second for the duration of the pool withdrawn from the pool.
Let d be the duration of the pool, thus $d\Delta z$ is the total number of Short withdrawn from the pool.

$$(x + y + \Delta x + \Delta y)(z - \Delta z) = L^2 \quad (17)$$

To get Short from this pool, Long ETH/Long USD is required to be deposited into the pool. As specified in Timeswap option mechanics (See section 3.1.1), the contract will first use the ETH and/or USD deposited by the lender to mint equivalent Short and Long ETH/Long USD. Then deposit the Long ETH/Long USD into the pool.

But to maximize the number of total Short minted and received from the pool, the contract decides if Long ETH or Long USD will be minted and deposited, based on the equations (7) and (8). Lenders do not care about the value of Long ETH/Long USD minted, since it will be sent to the pool anyway. Thus, the contract will mint Long ETH/Long USD whichever costs less for the lenders.

If $K < S$, the contract will mint Long USD, thus $\Delta x = 0$. Equation (17) can be simplified to

$$(x + y + \Delta y)(z - \Delta z) = L^2 \quad (18)$$

In this case, if the lenders choose to deposit in USD, the contract will directly mint Δy Short and $K\Delta y$ Long USD, thus will require $K\Delta y$ USD deposited (see section 3.1.1). Then the $K\Delta y$ Long USD is deposited into the pool to withdraw $d\Delta z$ Short from the pool.

But if the lenders choose to deposit in ETH, the contract will first swap the ETH to USD through Uniswap. Then mint Δy Short and $K\Delta y$ Long USD, thus will require $\frac{K\Delta y}{S}$ ETH deposited (see section 3.1.1). Then the Δy Long USD is deposited into the pool to withdraw $d\Delta z$ Short from the pool.

The final result for both case of depositing USD or ETH is the lenders receiving $\Delta y + d\Delta z$ Short. See both figures [13] and [14].

Note that the Timeswap Pool contract does not know the spot price (S), instead it tries both cases of not swapping through Uniswap and swapping through Uniswap, and see which maximizes the amount of Short lenders receive.

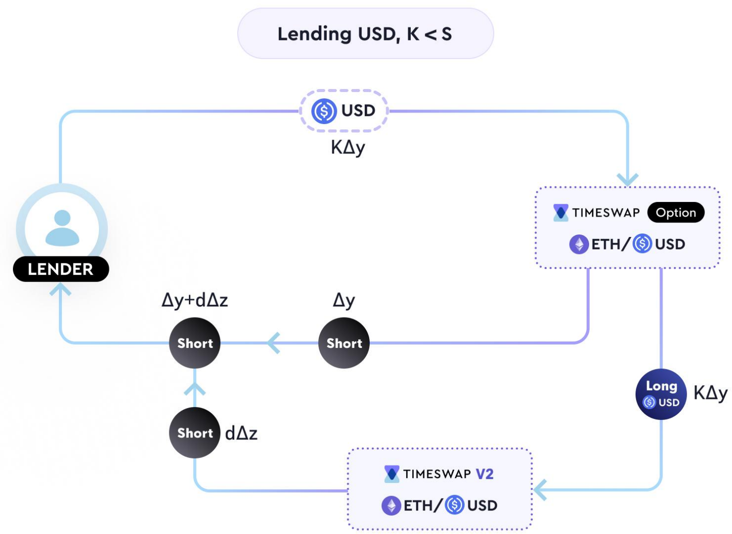


figure [13]

If $K > S$, the contract will mint Long ETH, thus $\Delta y = 0$. Equation (17) can be simplified to

$$(x + y + \Delta x)(z - \Delta z) = L^2 \quad (19)$$

In this case, if the lenders choose to deposit in ETH, the contract will directly mint Δx Short and Δx Long ETH, thus will require Δx ETH deposited (see section 3.1.1). Then the Δx Long ETH is deposited into the pool to withdraw $d\Delta z$ Short from the pool.

But if the lenders choose to deposit in USD, the contract will first swap the USD to ETH through Uniswap. Then mint Δx Short and Δx Long ETH, thus will require $S\Delta x$ USD deposited (see section 3.1.1). Then the Δx Long ETH is deposited into the pool to withdraw $d\Delta z$ Short from the pool.

The final result for both case of depositing USD or ETH is the lenders receiving $\Delta x + d\Delta z$ Short. See both figures [13] and [14].

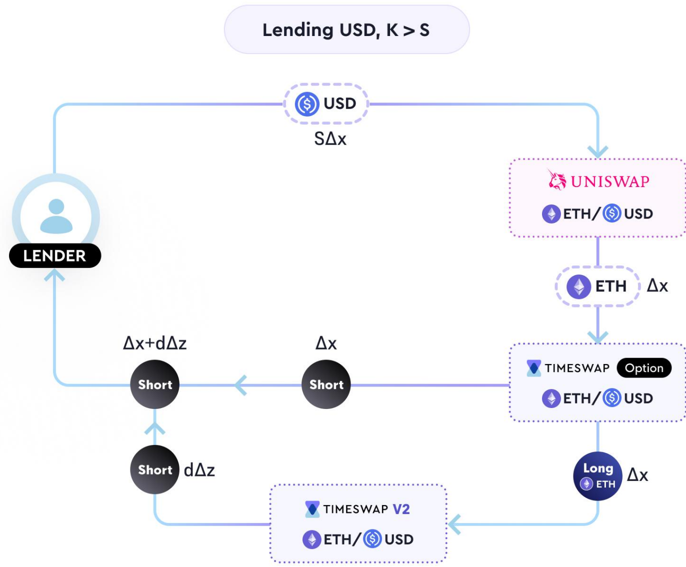


figure [14]

3.2.5 Lending Example

Case 1.1 Supposed there are 160,000 Long USD and 20 Short in a 800 USD per ETH strike pool that matures in 1 year ($x = 0$, $y = 200$, $z = \frac{20}{31557600}$ since 1 year has 31557600 seconds). Suppose again there is a DEX where it has a spot swap rate of 2000 USD per ETH. A lender wants to lend 1,000 USD, thus calling the lend function, which will have the contract do the following transactions:

1. Using the Timeswap option mechanics (see section 3.1.1), deposit 1,000 USD to mint 1,000 Long USD and 1.25 Short.
2. The minted 1,000 Long USD ($\Delta y = 1.25$) is deposited into the Timeswap pool.

3. Following the constant product AMM equation (16), $\Delta z = 0.00000000393$. Thus, 0.1242 Short ($31557600\Delta z \approx 0.1242236025$) is withdrawn from the Timeswap pool.
4. Both 1.25 Short minted and 0.1242 Short from the pool goes to the lender.

The end result is as follows:

- The lender has a total of 1.3742 Short. The Short amount represents 1,099.36 USD return, if the spot price stays above 800 USD per ETH. It also represents 1.3742 ETH remuneration, if the spot price falls below 800 USD per ETH.
- The lender has USD annual percentage rate (APR) of 9.936%, and ETH collateral bond position (CBP, similar to collateral debt position CDP) of 274.84%.

Case 1.2 Suppose given the same example above, the lender wants to lend 0.5 ETH instead. The contract will do the following transactions:

1. 0.5 ETH will be swapped to 1,000 USD through a DEX.
2. Same steps 1 to 4 as the above example when the lender wants to deposit 1,000 USD.

The end result is the same with a slight reframing:

- The lender has ETH annual percentage rate (APR) of 174.84%, and USD collateral bond position of 109.936%

Case 2.1 Suppose given the same first example, but instead the spot swap rate is 600 USD per ETH instead. A lender wants to lend 1.25 ETH, thus calling the lend function, which will have the contract do the following transactions:

1. Using the Timeswap option mechanics (see section 3.1.1), deposit 1.25 ETH to mint 1.25 Long ETH and 1.25 Short.
2. The minted 1.25 long ETH ($\Delta x = 1.25$) is deposited into the Timeswap pool.
3. Following the constant product AMM equation (17), $\Delta z = 0.00000000393$. Thus, 0.1242 Short ($31557600\Delta z \approx 0.1242236025$) is withdrawn from the Timeswap pool.
4. Both 1.25 Short minted and 0.1242 Short from the pool goes to the lender.

The end result is as follows:

- The lender has a total of 1.3742 Short. The Short amount represents 1,099.36 USD return, if the spot price rises above 800 USD per ETH. It also represents 1.3742 ETH return, if the spot price stays below 800 USD per ETH.
- The lender has ETH annual percentage rate (APR) of 9.936%, and USD collateral bond position (CBP, similar to collateral debt position CDP) of 146.58%.

Case 2.2 Suppose given the same example above, the lender wants to lend 750 USD instead. The contract will do the following transactions:

1. 750 USD will be swapped to 1.25 ETH through a DEX.
2. Same steps 1 to 4 as the above example when the lender wants to deposit 1.25 ETH.

The end result is the same with a slight reframing:

- The lender has USD annual percentage rate (APR) of 46.58%, and ETH collateral bond position of 109.936%

3.2.6 Lending Analysis

As shown in figure [15], is the payoff diagram of a lender interacting with a pool. The x-axis is the spot price at maturity of the pool. The y-axis represents profit and loss of the lender, depending on the spot price at maturity. The horizontal line in the middle represents the amount of USD investment by the lender, thus represents the break even point. The diagonal line represents the profit and loss of holding an equivalent amount of ETH at the beginning of the transaction. As one can see, if the spot price of ETH is greater at maturity, then the holder of ETH will get USD profit. On the other hand, if the spot price of ETH is lesser at maturity, the holder of ETH will get USD loss.

Holding a lending position, shows a fixed yield as long as the spot price at maturity stays above the strike price. Another important insight, is that the lending position has a higher profit or less loss when the spot price is lesser, compared to holding ETH. This makes a lending position much safer than holding ETH. But in return, lending position sacrifice higher yield, when the spot price is greater.

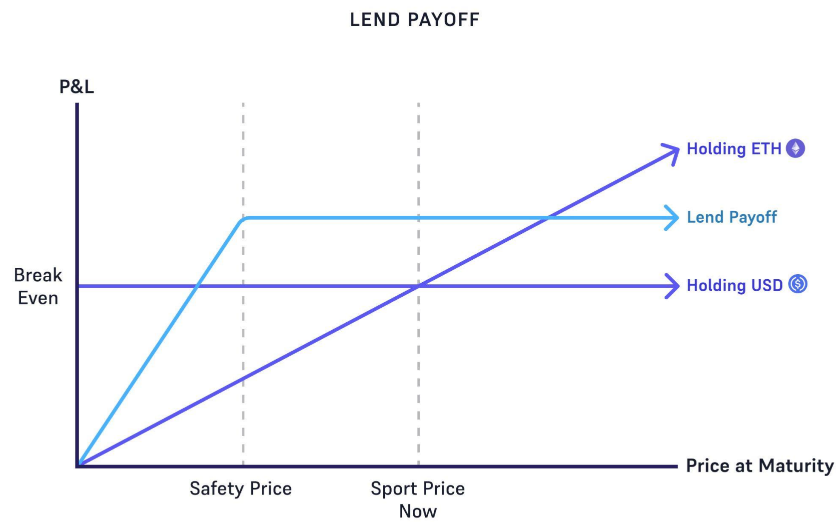


figure [15]

Another promise of the protocol is that no matter how large the lending transaction, the user will always get a positive interest rate and over-collateralization of the opposing collateral.

When $K < S$, from equation (18), lender deposits $K\Delta y$ USD as principal. If the lender chooses to deposit in ETH, the contract first swaps it to $K\Delta y$ USD as well. In return, lender receives $\Delta y + d\Delta z$ Short. This amount represents $K(\Delta y + d\Delta z)$ USD return and $\Delta y + d\Delta z$ ETH insurance return.

The interest amount is simply the difference of USD return and the principal in USD

$$K(\Delta y + d\Delta z) - K\Delta y > 0 \text{ (20)}$$

since $d\Delta z > 0$, thus the interest return is always positive.

The collateral bond position (CBP, similar to CDP) on the other hand is the ratio between the insurance return in USD over the USD principal.

$$\frac{S(\Delta y+d\Delta z)}{K\Delta y} > 100\% \ (21)$$

since $K < S$, then $K\Delta y < S\Delta y$. Together with $d\Delta z > 0$, the CBP is always over-collateralized.

When $K > S$, from equation (19), lender deposits Δx ETH as principal, converting to USD, it is $S\Delta x$ USD. Similarly for when the lender chooses to deposit in USD. In return, lender receives $\Delta x + d\Delta z$ Short. This amount represents $K(\Delta x + d\Delta z)$ USD return and $\Delta x + d\Delta z$ ETH insurance return.

The interest amount in this case is as follows

$$K(\Delta x + d\Delta z) - S\Delta x > 0 \ (22)$$

since $K > S$, then $K\Delta x > S\Delta x$. Together with $d\Delta z > 0$, the interest returns are always positive.

The CBP in this case is as follows

$$\frac{S(\Delta x+d\Delta z)}{S\Delta x} > 100\% \ (23)$$

since $d\Delta z > 0$, thus CBP is always over-collateralized.

3.2.7 Borrow Transaction

For users to get Long ETH or Long USD from depositing ETH and/or USD, the contract does the following:

Let Δx be the number of Long ETH withdrawn from the pool.
Let Δy be the number of Long USD divided by K withdrawn from the pool.

Let Δz be the number of Short per second for the duration of the pool deposited into the pool.
Let d be the duration of the pool, thus $d\Delta z$ is the total number of Short deposited into the pool.

$$(x + y - \Delta x - \Delta y)(z + \Delta z) = L^2 \quad (24)$$

To get Long ETH/Long USD from this pool, Short is required to be deposited into the pool. As specified in Timeswap option mechanics (See section 3.1.1), the contract will first use the ETH and/or USD deposited by the borrower to mint equivalent Short and Long ETH/Long USD. Then deposit the Short into the pool.

But to maximize the number of Long ETH/Long USD (whichever the borrower wanted) minted and received from the pool, the contract will decide to withdraw Long ETH or Long USD, based on the equations (7) and (8). The Long ETH or Long USD minted (See section 3.1.1), is based on what the position the borrower wants to hold. But the contract will always withdraw Long ETH or Long USD from the pool, whichever is valued higher. If the Long USD/Long ETH withdrawn from the pool is not what the borrowers wanted, it will be transformed from Long ETH to Long USD, or from Long USD to Long ETH through the Timeswap option mechanics (See section 3.1.1), where borrowers have to shoulder the cost as well.

But due to the rebalance arbitrage mechanism (See section 3.2.2), only either Long ETH or Long USD will be in the pool, whichever is valued less. Thus the contract has no choice to withdraw whatever Long ETH/Long USD exist in the pool. But it will still always try to withdraw the Long ETH/Long USD that is valued higher, if ever it is in a pool, for better pricing for borrower.

if $K < S$, the pool will only have Long USD in it, due to rebalance arbitrage, thus $x = 0$ and $\Delta x = 0$. Equation (24) can be simplified to

$$(y - \Delta y)(z + \Delta z) = L^2 \quad (25)$$

In this case, suppose the borrower wants to leverage towards ETH (Holding Long ETH), borrowing USD, and depositing ETH as collateral. The contract first mints $d\Delta z$ Short and $d\Delta z$ Long ETH, thus requires $d\Delta z$ ETH from the borrower. Then the $d\Delta z$ Short is deposited in the pool to withdraw $K\Delta y$ Long USD (since there is no Long ETH in the pool). Finally, the contract transforms the $K\Delta y$ Long USD to Δy Long ETH, using the Timeswap option mechanics (See section 3.1.1) by depositing Δy ETH coming from the borrower and withdrawing $K\Delta y$ USD to be given to the borrower. In total, the borrower deposits $\Delta y + d\Delta z$ ETH as collateral; the borrower receives $K\Delta y$ USD as amount borrowed and $\Delta y + d\Delta z$ Long ETH as a borrowing position. This is shown as a diagram in figure [16].

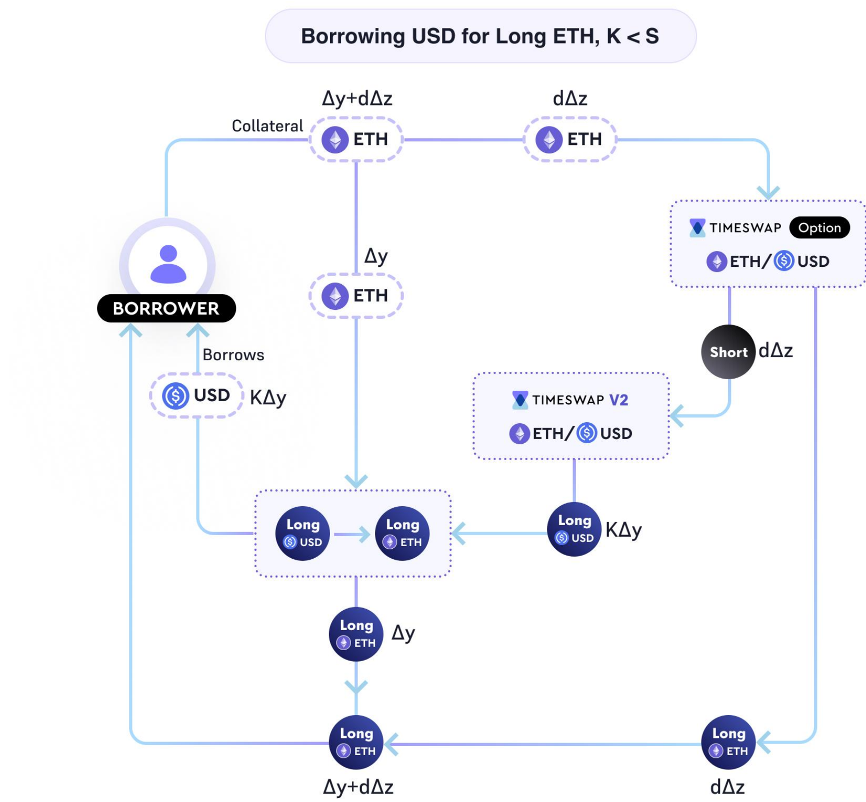


figure [16]

In the same case, suppose the borrower wants to leverage towards USD (Holding Long USD), borrowing ETH, and use USD as collateral. The contract first mints $d\Delta z$ Short and $Kd\Delta z$ Long USD, thus requiring $Kd\Delta z$ USD from the borrower. Then the $d\Delta z$ Short is deposited in the pool to withdraw $K\Delta y$ Long USD. Finally, the contract uses Uniswap to swap $K\Delta y$ USD coming from the borrower to $\frac{K\Delta y}{S}$ ETH which also goes to the borrower. In total, the borrower deposits $K(\Delta y + d\Delta z)$ USD as collateral; the borrower receives $\frac{K\Delta y}{S}$ ETH as amount borrowed and $K(\Delta y + d\Delta z)$ Long USD as a borrowing position. This is shown as a diagram in figure [17].

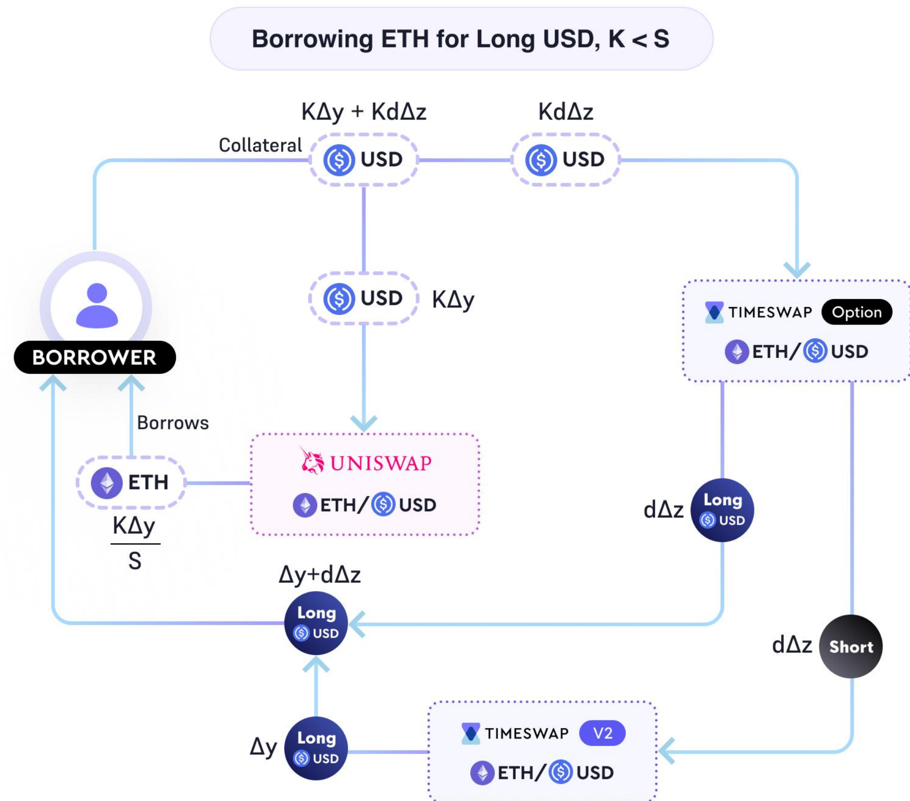


figure [17]

if $K > S$, the pool will only have Long ETH in it, due to rebalance arbitrage, thus $y = 0$ and $\Delta y = 0$. Equation (24) can be simplified to

$$(x - \Delta x)(z + \Delta z) = L^2 \; (26)$$

In this case, suppose the borrower wants to leverage towards ETH, borrowing USD, and depositing ETH as collateral. The contract first mints $d\Delta z$ Short and $d\Delta z$ Long ETH, thus requiring $d\Delta z$ ETH from the borrower. Then the $d\Delta z$ Short is deposited in the pool to withdraw Δx Long ETH. Finally, the contract uses Uniswap to swap Δx ETH coming from the borrower to $S\Delta x$ USD which also goes to the borrower. In total, the borrower deposits $\Delta x + d\Delta z$ ETH as collateral; the borrower receives $S\Delta x$ USD as amount borrowed and $\Delta x + d\Delta z$ Long ETH as a borrowing position. This is shown as a diagram in figure [18].

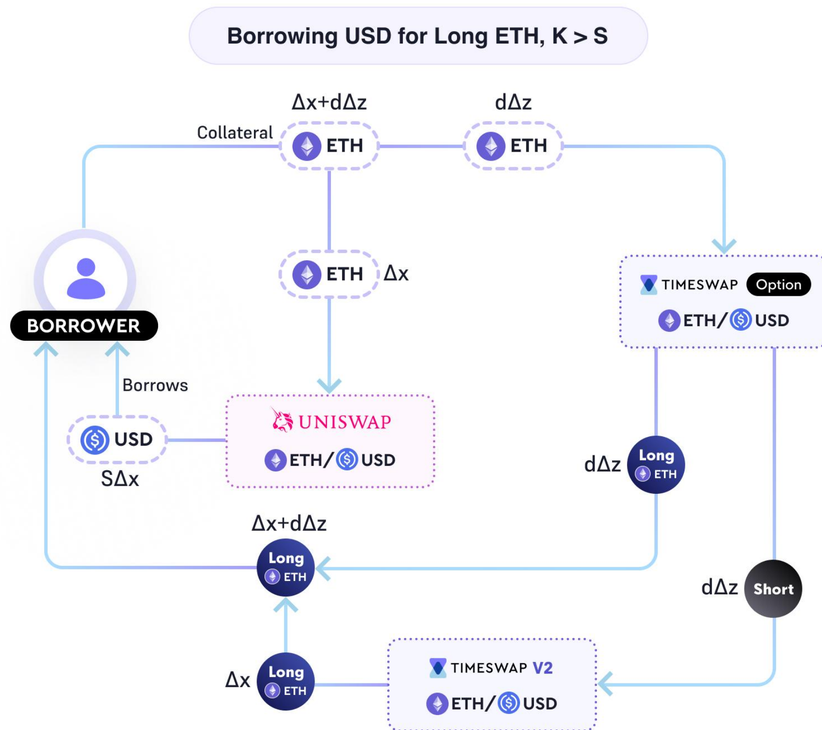


figure [18]

In the same case, suppose the borrower wants to leverage towards USD, borrowing ETH, and use USD as collateral. The contract first mints $d\Delta z$ Short and $Kd\Delta z$ Long USD, thus requiring $Kd\Delta z$ USD from the borrower. Then the $d\Delta z$ Short is deposited in the pool to withdraw Δx Long ETH. Finally, the contract transforms the Δx Long ETH to $K\Delta x$ Long USD, using the Timeswap option mechanics (See section 3.1.1) by depositing $K\Delta x$ USD coming from the borrower and withdrawing Δx ETH to be given to the borrower. In total, the borrower deposits $K(\Delta x + d\Delta z)$ USD as collateral; the borrower receives Δx ETH as amount borrowed and $K(\Delta x + d\Delta z)$ Long USD as a borrowing position. This is shown as a diagram in figure [19].

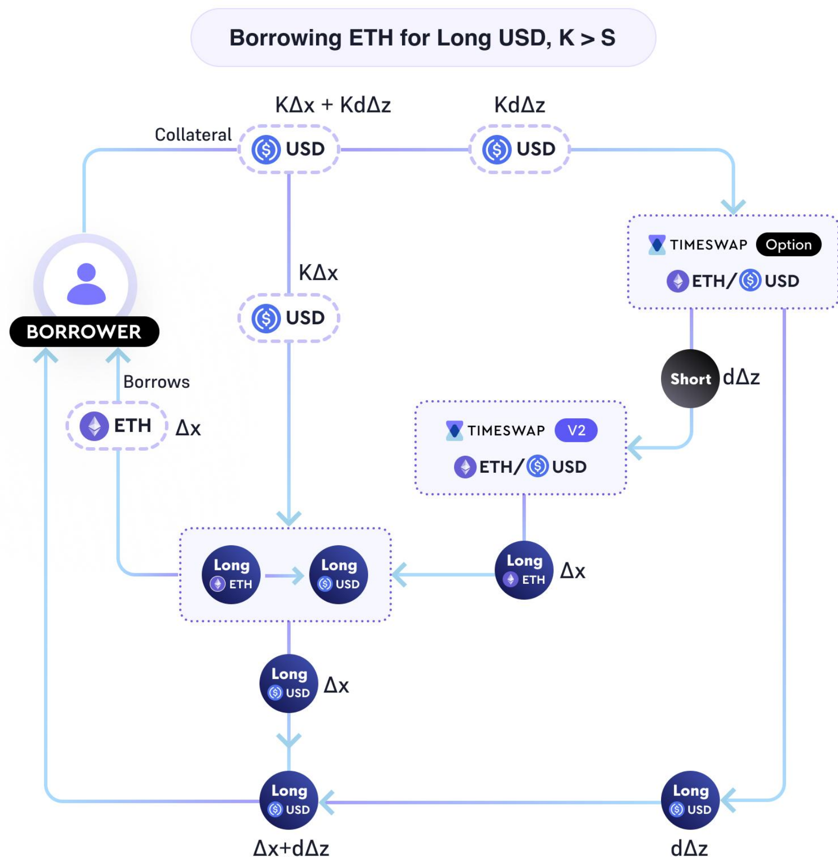


figure [19]

3.2.8 Borrowing Example

Case 1.1 Supposed there are 160,000 long USD and 20 Short in a 800 USD per ETH strike pool that matures in 1 year ($x = 0$, $y = 200$, $z = \frac{20}{31557600}$ since 1 year has 31557600 seconds). Suppose again there is a DEX where it has a spot swap rate of 2000 USD per ETH. A borrower wants to borrow 1,000 USD, leveraging towards ETH, thus calling the borrow function, which will have the contract do the following transactions:

1. The contract withdraws 1,000 Long USD ($\Delta y = 1.25$) from the Timeswap pool. Following the constant product AMM equation (25), $\Delta z = 0.00000000393$. Thus, 0.1242 Short ($31557600\Delta z \approx 0.1242236025$) is required by the Timeswap pool.
2. Using the Timeswap option mechanics (see section 3.1.1), deposits 0.1242 ETH to mint 0.1242 Long ETH and 0.1242 Short, to which the 0.1242 Short is deposited in the pool.
3. Using again the Timeswap option mechanics, deposits 1.25 ETH to transform 1,000 Long USD to 1.25 Long ETH, thus withdrawing 1000 USD.
4. The 1000 USD withdrawn from 1,000 Long USD, goes to the borrower.
5. Both the 1.25 ETH used to transform Long USD to Long ETH, and the 0.1242 ETH from minting, are both from the borrowers.

The end result is as follows:

- The borrower has a total of 1.3742 Long ETH. The Long ETH amount represents 1,099.36 USD debt. It also represents 1.3742 ETH collateral locked. When the spot price stays above 800 USD per ETH, then the borrower will pay the debt to withdraw the collateral locked. When the spot price falls below the safety price, then the borrower will default.
- The borrower has USD annual percentage rate (APR) of 9.936%, and ETH collateral debt position (CDP) of 274.84%.

Case 1.2 Suppose given the same example as above, but instead, a borrower wants to borrow 1.25 ETH, leveraging towards USD, thus calling the borrow function, which will have the contract do the following transactions:

1. The contract withdraws 1,000 Long USD ($\Delta y = 1.25$) from the Timeswap pool. Following the constant product AMM equation (25), $\Delta z = 0.00000000393$. Thus, 0.1242 Short ($31557600\Delta z \approx 0.1242236025$) is required by the Timeswap pool.
2. Using the Timeswap option mechanics (see section 3.1.1), deposits 99.44 USD to mint 0.1242 Long USD and 0.1242 Short, to which the 0.1242 Short is deposited in the pool.
3. Swap 2,500 USD through Uniswap to get 1.25 ETH, which goes to the borrower.

4. Both the 2,500 USD from swapping through Uniswap and the 99.44 USD from minting, are both from the borrowers.

The end result is as follows:

- The borrower has a total of 1,099.36 Long USD. The Long USD amount represents 1,099.36 USD collateral. It also represents 1.3742 ETH return. When the spot price falls below 800 USD per ETH, then the borrower will pay the debt to withdraw the collateral locked. When the spot price stays above the safety price, then the borrower will default.
- The borrower has ETH annual percentage rate (APR) of 174.84%, and USD collateral debt position (CDP) of 109.936%.

There are other cases as well, but they all follow very similar processes as above cases.

3.2.9 Borrowing Analysis

Holding a borrowing leverage towards ETH position, shows a complete loss when the spot price falls below the strike price. Another important insight is that the profit and loss grows faster than holding ETH when the spot price at maturity is higher. Therefore, the borrower wants the spot price to increase as high as possible at maturity. This is what it means to leverage on ETH.

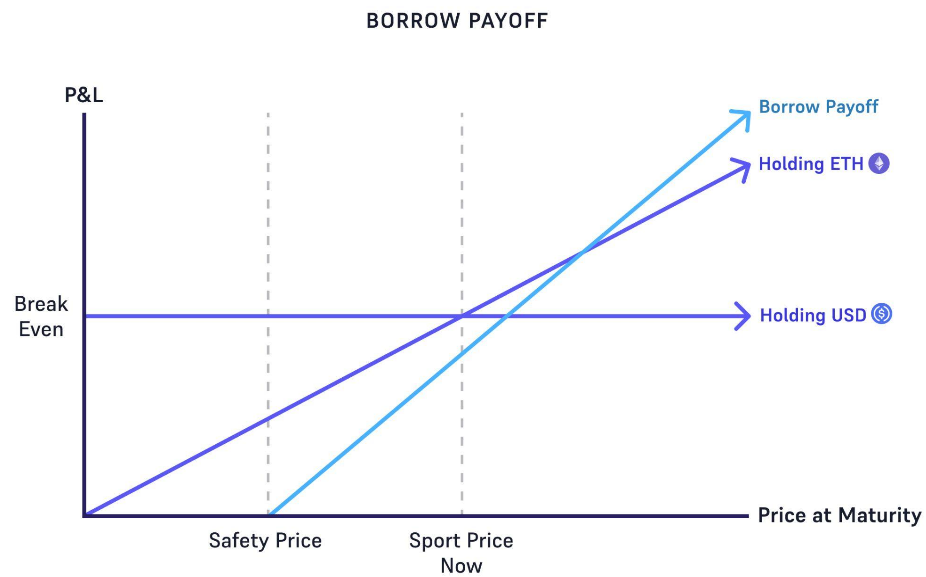


figure [20]

Holding a borrowing leverage towards USD position, shows a complete loss when the spot price stays above the strike price. But when the spot price at maturity falls below the strike price, suddenly the borrower's profit and loss start increasing tremendously. Therefore, the borrower wants the spot price to fall as much as possible at maturity. This is what it means to leverage oppositely on ETH (Or leverage towards USD).

Due to the rebalance arbitrage mechanism (see section 3.2.2), the user will always receive an interest rate that is positive, and an over-collateralization requirements.

When $K < S$, from equation (25), borrower leveraging towards ETH deposits $\Delta y + d\Delta z$ ETH as collateral, and borrows $K\Delta y$ USD, with a debt of $K(\Delta y + d\Delta z)$ USD.

The interest amount is simply the difference of USD debt and the principal in USD

$$K(\Delta y + d\Delta z) - K\Delta y > 0 \text{ (27)}$$

since $d\Delta z > 0$, thus the interest return is always positive.

The collateral debt position (CDP) on the other hand is the ratio between the collateral locked in USD over the USD principal.

$$\frac{S(\Delta y+d\Delta z)}{K\Delta y} > 100\% \text{ (28)}$$

since $K < S$, then $K\Delta y < S\Delta y$. Together with $d\Delta z > 0$, the CDP is always over-collateralized.

When $K < S$, from equation (25), borrower leveraging towards USD deposits $K(\Delta y + d\Delta z)$ USD as collateral, and borrows $\frac{K\Delta y}{S}$ ETH, with a debt of $\Delta y + d\Delta z$ ETH.

The interest amount is simply the difference of ETH debt and the principal in ETH

$$\Delta y + d\Delta z - \frac{K\Delta y}{S} > 0 \text{ (29)}$$

since $1 > \frac{K}{S}$, thus $\Delta y > \frac{K\Delta y}{S}$. Together with $d\Delta z > 0$, the interest return is always positive.

The collateral debt position (CDP) on the other hand is the ratio between the collateral locked in ETH over the ETH principal.

$$\frac{\frac{K(\Delta y+d\Delta z)}{S}}{\frac{K\Delta y}{S}} = \frac{(\Delta y+d\Delta z)}{\Delta y} > 100\% \text{ (30)}$$

similar to the reason of equation (28), the CDP is always over-collateralized.

When $K > S$, from equation (26), borrower leverage towards ETH deposits $\Delta x + d\Delta z$ ETH as collateral, and borrows $S\Delta x$ USD, with a debt of $K(\Delta x + d\Delta z)$ USD.

The interest amount is the difference of USD debt and the principal in USD

$$K(\Delta x + d\Delta z) - S\Delta x > 0 \text{ (31)}$$

since $K > S$, thus $K\Delta x > S\Delta x$. Together with $d\Delta z > 0$, the interest return is always positive.

The collateral debt position (CDP) on the other hand is the ratio between the collateral locked in USD over the USD principal.

$$\frac{S(\Delta x+d\Delta z)}{S\Delta x} = \frac{\Delta x+d\Delta z}{\Delta x} > 100\% \text{ (32)}$$

Since $d\Delta z > 0$, the CDP is always over-collateralized.

When $K > S$, from equation (26), borrower leveraging towards USD deposits $K(\Delta x + d\Delta z)$ USD as collateral, and borrows Δx ETH, with a debt of $\Delta x + d\Delta z$ ETH.

The interest amount is simply the difference of ETH debt and the principal in ETH

$$\Delta x + d\Delta z - \Delta x = d\Delta z > 0 \text{ (33)}$$

as shown, the interest return is always positive.

The collateral debt position (CDP) on the other hand is the ratio between the collateral locked in ETH over the ETH principal.

$$\frac{\frac{K(\Delta x+d\Delta z)}{S}}{\Delta x} = \frac{K(\Delta x+d\Delta z)}{S\Delta x} > 100\% \text{ (34)}$$

since $K > S$, thus $K\Delta x > S\Delta x$. Together with $d\Delta z > 0$, the CDP is always over-collateralized.

3.2.10 Liquidity Provision Transaction

Liquidity providers are the counterparty to both lenders and borrowers, similar to how Uniswap's liquidity providers work. Timeswap liquidity providers can create new Timeswap pools choosing whatever parameters they want or add more liquidity to existing Timeswap pools. Liquidity providers deposit both Long ETH/Long USD and Short into the pool, which requires them to only lock ETH/USD to mint Long ETH/Long USD and Short. Since $x + y$ and dz are usually not equal, liquidity providers will usually have some excess Long ETH/Long USD or excess Short they hold in their balance.

Due to the duration mechanism of Short side of the AMM, z short positions are returned to the liquidity providers every second.

When lenders or borrowers interact with the pool, liquidity providers take on divergent loss of the change in the AMM ratio of Long and Short, again very similar to how Uniswap divergent loss works. It is expected that the amount of Long and Short liquidity providers will receive after lenders and borrowers interact with the pool may be different from the time the liquidity provider deposited Long and Short. This is the reason why liquidity providers have the incentive to try to set the initial $x + y$ and dz parameter to be as close to the market price as much as possible to minimize divergent loss when creating a new pool. When adding liquidity into an existing pool, they simply must deposit both long and short positions such that I (marginal interest rate) stays the same.

Let Δx be the number of Long ETH deposited into the pool, converted to the same denomination as Short, if needed.
Let Δy be the number of Long USD deposited into the pool, converted to the same denomination as Short, if needed.
Let Δz be the number of Short per second for the duration of the pool deposited into the pool.
Let d be the duration of the pool, thus $d\Delta z$ is the total number of Short deposited into the pool.
Let l be the amount of ERC1155 liquidity tokens received

$$(x + y + \Delta x + \Delta y)(z + \Delta z) \quad (35)$$

$$\frac{z+\Delta z}{x+y+\Delta x+\Delta y} = I \quad (36)$$

$$\sqrt{(\Delta x + \Delta y)\Delta z} = l \quad (37)$$

Similarly to lending transactions, the liquidity providers should deposit Long ETH or Long USD whichever is cheaper into the pool, based on the equations (7) and (8).

There are two cases, regarding the $x + y$ and dz parameter that affects the result of liquidity provision.

Suppose the current pool has parameter $x + y > dz$, or that the liquidity providers chose it as such when creating the pool. This is usually the case for Timeswap pools with short duration and/or the interest rate per second are small. This will require liquidity providers to deposit more Long ETH/Long USD than Short. This means that if liquidity providers mint equivalent amounts of Long ETH/Long USD and Short by depositing ETH/USD (see section 3.1.1), then they will have some excess Short on their balance (not in the pool).

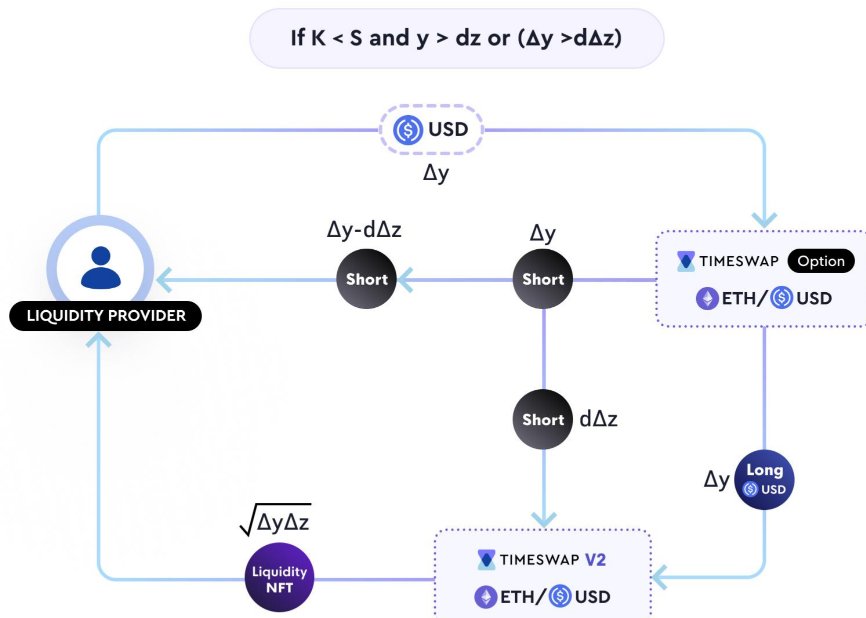


figure [21]

On the other hand, suppose the current pool has parameter $x + y < dz$. This will require liquidity providers to deposit less Long ETH/Long USD than Short. Similar to the above case, they will have some excess Long on their balance.

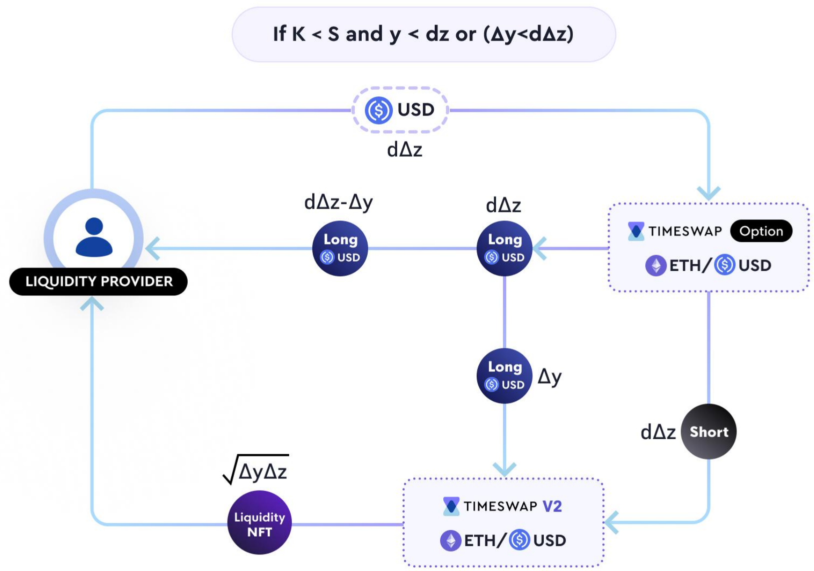


figure [22]

3.2.11 Liquidity Withdrawal Transaction

Unlike in V1 where liquidity providers can only withdraw after the maturity of the pool, in V2, liquidity providers can now withdraw liquidity even before the maturity of the option.

Let Δx be the number of Long ETH withdrawn from the pool.
Let Δy be the number of Long USD withdrawn from the pool.
Let Δz be the number of Short per second for the duration of the pool withdrawn from the pool.
Let d be the duration of the pool, thus $d\Delta z$ is the total number of Short withdrawn from the pool.
Let l be the amount of ERC1155 liquidity tokens burnt.

$$(x + y - \Delta x - \Delta y)(z - \Delta z) \ (38)$$

$$\frac{z-\Delta z}{x+y-\Delta x-\Delta y} = I \ (39)$$

$$\sqrt{(\Delta x + \Delta y)\Delta z} = l \ (40)$$

This gives liquidity providers more flexibility to manage their position risk in real time. So suppose that a liquidity provider owns 10% of the total supply of liquidity tokens, then the liquidity provider can essentially withdraw 10% of Long and Short positions out of the pool.

3.2.12 Transaction Fee

The Timeswap protocol charges a transaction fee to both lenders and borrowers as a revenue for the liquidity providers, therefore the goal of liquidity providers is to earn enough transaction fee to cover for possible cost of divergent loss. The transaction fee is charged unto the short withdrawn of every lending transaction (which is $d\Delta z$) and long withdrawn of every borrowing transaction (which is $x + y$). Due to this, liquidity providers earn fees in terms of Short and Long positions. The protocol also charges a protocol fee which is a small proportion of the transaction fee earned by liquidity providers, which goes to the Timeswap DAO as revenue.

4 Closing Remarks

Timeswap is designed to be the fundamental time preference protocol in Ethereum. This creates the ability for anyone to manage the time and risk value of their tokens at a specific timeframe and risk expectation. It is the primitive that anyone can use to create any fixed time payoff.