Swapnet Whitepaper [DRAFT]

Teddy Woodward and Jeff Wu
{teddy, jeff}@swapnet.exchange

Introduction

Decentralized finance (DeFi) is an exciting, emerging ecosystem that reimagines the traditional financial system with the use of blockchain technology. Today, in DeFi, there are protocols that create stable assets (MakerDAO), decentralized exchange (0x, Uniswap) and decentralized lending and borrowing (Compound). In this paper, we introduce Swapnet, a protocol to facilitate decentralized interest rate derivative trading on the Ethereum mainnet. With Swapnet, a user will be able to hedge their interest rate exposure potentially for years in the future. Lenders will be able to fix the rate at which they lend their assets, borrowers will be able to fix their borrowing costs, and speculators will be able to position themselves to take advantage of expected changes in future interest rates. This will give DeFi users financial certainty and enable them to plan and invest for the long term.

Swapnet can be implemented as a set of Ethereum smart contracts without any centralized components. Additionally, trading on Swapnet removes counterparty default risk without sacrificing capital efficiency. Decentralization, zero counterparty risk and capital efficiency are essential to the long-term success of the protocol.

Pseudonymity on blockchain makes it difficult to coerce a party to honor its obligations. Other DeFi projects tend to mitigate counterparty risk by requiring any financial obligation to be over-collateralized. This removes the risk that a party does not pay the debts that it previously agreed to pay, but it is capital-intensive. In the context of an interest rate derivatives market, a protocol that mandates over-collateralization would destroy much of the utility that such a market could provide. Over-collateralization would force down potential returns on capital and limit the economic appeal of interest rate swaps. Swapnet would then likely struggle to attract liquidity and provide little value to the greater DeFi ecosystem.

In order to provide capital efficiency, Swapnet introduces a novel collateralization system. Namely, the protocol will require partial collateralization and be able to liquidate an account's positions if it is at risk of default without affecting the counterparties to this liquidated account. In order to achieve this, Swapnet must be able to quantify the value of an account's portfolio and its "riskiness". We detail a risk and valuation framework that monitors account collateralization levels as well as a liquidation protocol for under-collateralized accounts that redistributes an under-collateralized accounts obligations in order to keep risk within the system unchanged.

We also introduce a governance token that will act as a backstop to the system, ensuring that all trades in our system will be settled. This token will convey rights and obligations to its holders that will incentivize them to act in the best interests of the overall system. Transaction fees will be used to ensure that the overall value of the token grows with the success of Swapnet.

Account Structure

An account is created when a user moves collateral into an escrow smart contract. This collateral balance is used to back the account's portfolio of trades, determine which new trades the account is allowed to place and which tokenized trade pools (tPools) the account can bid on. An account may add collateral or remove free collateral at any time, however, it may not remove collateral such that its trades become undercollateralized.

An account's required collateral is determined by the composition of its trade portfolio. This account structure incentivizes users to post all their collateral to one account in order to maximize collateral netting between their trades. Collateral netting (discussed further in the Risk and Pricing Framework section) is critical for capital efficiency.

In our system, once trades are written to mainnet they are immutable and *guaranteed to settle*. Accounts can hedge their previous trades by entering into new trades that offset their existing positions.

Tradable Instruments

The set of tradable instruments in this protocol is open and extendable, pending affirmation from the system's governance token holders. Any new instrument introduced to the protocol must conform with a specification so that it fits within the protocol's overall framework. The protocol will launch with two tradable instruments, an interest rate swap on Compound lending rates called the cSwap and a tokenized bundle of cSwaps called a tPool.

Non-Fungible Swap Tokens (cSwap)

cSwaps allow users to hedge the Compound lending rate associated with a given token for a specified period of time. A cSwap is defined by five things: its reference token (Dai, USDC), a start block (B_S), an end block (B_E), an annualized fixed rate (K), and a notional size (N). A cSwap entitles its holder to a cash flow to be exchanged at the maturity of the swap equal to the difference between the interest on the fixed rate K and the interest on the realized floating rate between blocks B_S and B_E. The fixed interest is known upfront, this is:

$$I_{Fixed} = K * N * (B_E - B_S) / 2102400$$

The floating interest can only be known retrospectively. It is:

$$I_{Float} = (CompoundExchangeRate_{B_E} / CompoundExchangeRate_{B_S}) - 1) * N$$

The payer of the swap is entitled to $I_{Float} - I_{Fixed}$ $at\ time\ B_E$. The receiver of the swap is entitled to the opposite, $I_{Fixed} - I_{Float}$ $at\ time\ B_E$.

cSwap tokens will be represented as ERC-721 NFTs. The _tokenId will encode the features of the cSwap: reference token, start block, maturity length in blocks, fixed rate, notional size, and a further unique id. Each cSwap token will also reference the _tokenId of its counterpart: the token that represents the other side of the trade. A cSwap token may be owned by an account or a tPool contract. The transfer functions will check if the recipient has enough free collateral to accept the token and revert if they do not. When cSwap tokens are created, they also must call Compound in order to get the rate at the start block. If the start block is not the same block as when the transaction is mined, then a function similar to the settlement function (discussed in the Token Settlement section) must be used to record the start rate.

cSwap tokens will have a settlement function that can only be called after maturity at the specified block height. At settlement, the owed amount of underlying collateral will be transferred between the two parties and the token will be destroyed.

Tokenized Trade Pools (tPool)

A bundle of cSwap NFTs can be turned into a fungible ERC-20 token called a tPool. Each tPool token represents a proportional ownership claim to the entire bundle of cSwap NFTs contained inside of it. It also requires its holder to have the required amount of collateral in their account while they hold it as the holder is liable for any losses resulting from the underlying cSwap NFTs.

An account can manually trigger the tokenization of a bundle of cSwaps or this tokenization can be triggered by the liquidation protocol described later in this paper. Once triggered, a new tPool token will be created and the ownership of all the underlying cSwap NFTs will be transferred to this new contract address. This contract will not have the ability to transfer the cSwap tokens from its own control.

The tPool contract is settled by calling the settlement function in each of the underlying cSwaps. When the underlying cSwap is settled, the payouts are distributed from the counterparty to all the tPool token holders in proportion to their holdings. Once all underlying cSwaps have been settled, the tPool contract can be destroyed.

Similar to the cSwap token, the tPool token’s transfer function will check the recipient’s account for the required amount of free collateral by calculating the net collateralization requirement of all its underlying cSwaps.

Token Settlement

Settlement is a challenge faced by tokens that rely on a value or values that are associated with the Ethereum state at a particular block (such as the Compound exchange rate at a cSwap's maturity block). In Ethereum, we have no native method for retrieving arbitrary data at a specific, historic block from within a smart contract; we can only query the state of smart contracts at the current block. This, combined with the inability to guarantee transactions being mined in a specific block, means that settling tokens at a precise block in the future is not trivial.

However, because the data we require for token settlement has already been written to the blockchain, we can sidestep many of the challenges with off-chain oracles that other projects face. In this section we will outline a few approaches to solving this on-chain oracle problem with varying degrees of security. Swapnet will likely adopt some combination of the approaches listed here which provides users the right, but not the obligation, to settle their trades with the highest level of security.

Signatures from Counterparty

Tokens can settle at any arbitrary value if a transaction is submitted with valid signatures from both parties. For cSwaps this requires two signatures, for tPools it would require signatures from all token holders. It's unlikely that this will be implemented initially for tPools.

Proof of Stake Settlement

One construction is to simply have a group of validators (or perhaps governance token holders in our case) stake tokens in order to gain the ability to settle tokens. Tokens can only be settled by some threshold of validators signing a transaction attesting to a specific value, such as "5% was the lending rate of Compound at block N".

It will be trivial to observe if there is malicious activity in this transaction as anyone can inspect the historical block. In the case of a bold, malicious attack, an honest party can provide a fraud proof in the form of a SNARK that will slash the validators' token holdings. We should also require that the aggregate amount at stake for the validators signing the transaction is greater than the value of the token settlement in order to prevent the scenario where it can become profitable to act dishonestly.

The benefit of this approach (as opposed to simply using SNARKs outlined in the next section) is greatly reduced gas costs; about 15k gas for threshold signatures according to this blog post.

SNARK Settlement

For fraud proofs, swaps with a very large settlement value, or the very paranoid we also propose a method of using a SNARK to generate a compact Merkle proof of a value in the Ethereum blockchain at a specified block height.

The SNARK would verify the function f(u, w) where u is the input and w is the witness. In our case, u = (block hash, block number, value) and w = Hash(block header values, merkle path from state root to contract address, merkle path from storage root to value).

The SNARK prover would prove that the entire Merkle path from the value to the storage root, up to the state trie root and into the block header matches the block hash at the specified block number. All of this data in the witness is “hidden” from the smart contract via a hash. Ethereum smart contracts will be able to validate this SNARK proof by retrieving the block hash of a block up to 256 blocks before the current block. This, along with the value provided to the contract and the witness, will be inputs into the verify function. According to some estimates, this verification would cost 400k gas.

High Liquidity Blocks

Some block heights may emerge (either by convention or incentives) as times when many instruments mature. One example of this are the periodic cSwaps we discuss in the Price and Risk Framework section. Token holders of instruments that mature at these “high liquidity blocks” can share the cost of a settlement oracle. A settlement oracle at one of these block heights will simply store the settlement data on chain. This data will be provided by one of the methods outlined above (or potentially other methods) for instruments to retrieve when their settlement function is eventually called. This oracle may accept a fee for providing an honest value.

Conversely, these blocks are also high leverage points for attackers who want to disrupt the system by providing false oracle data. Token holders are ultimately responsible for calling the settlement function of their tokens in a timely manner. We have provided a few potential implementations of feeding data into that settlement function and we anticipate other methods will also emerge.

It will ultimately be the role of governors, developers, and the market to ensure that Swapnet’s settlement methodology is secure.

Trading Methodologies

Swapnet will support on-chain requests for quotes and signed trades that are negotiated on off-chain venues.

Request for Quote

In the Request for Quote (RFQ) model, a requester publishes a proposal for a trade to the RFQ smart contract. Bidders submit signed multisig bids for the requester’s proposal off chain. These bids can be sent in any acceptable off chain method: email, P2P messaging over whisper, IRC, etc. As part of the cSwap system we will provide a default, secure channel for bids, however, users would not be limited to our method.

The requester chooses the bid they want, adds their signature to the multisig transaction, and submits it to the RFQ smart contract in order to finalize the trade. Two cSwaps representing the positions of the requester and the winning bidder are created with a reference to each other. Each cSwap will be owned by its respective account. The original request is removed from the RFQ contract.

Bidders may specify a number of blocks that their bid is valid for, preventing the requester from holding onto the bid (and therefore the bidder's potential free collateral) indefinitely. The protocol will provide a view function for the requester to verify that the bidder has enough free collateral to make good on their bid. It will be incumbent on the bidder to perform this verification before publishing the bid to the RFQ smart contract (the contract will revert the transaction if the bidder does not have enough free collateral).

There is the possibility of a race condition if a single bidder "double bids" their free collateral to different requesters, in which case, only the first requester who has their transaction mined on the blockchain will make a successful trade. We believe this is an acceptable trade off for increased privacy in the bidding process as well as decreased gas costs for the bidders (they do not have to submit their bids on chain). Ultimately, we believe this will result in more bids and greater liquidity in the cSwap system.

Off-Chain Trading

Participants can also negotiate trades off-chain in a centralized exchange, over a messaging service, or other trading venue. These off-chain trades can be submitted to the RFQ smart contract as a request that is fulfilled immediately by a signed bid. The smart contract will do the same check as in the RFQ process (indeed it would be the same smart contract) to check if both parties are eligible to enter into that trade (i.e. if they have the necessary collateral). If they are, the protocol will update both accounts' current active trade pools to reflect the new trade.

Risk and Pricing Framework

This section will discuss how we price and measure the risk of an arbitrary cSwap. The principles of this protocol's pricing methodology are simple and extendable to any interest rate derivative product. To price any derivative, this protocol first derives an expectation for any future exchanges of value (cash flows) associated with an instrument or portfolio. A cSwap, for example, has a single cash flow that occurs on the instrument's maturity block. After deriving the expectation of all the cash flows in an instrument or portfolio, pricing these instruments requires discounting these future expected cash flows to the present. This is known as taking the present value (PV) of a future cash flow.

Discount Rates

Each currency on Swapnet will be linked to a discount rate curve. In order to calculate the net present value (NPV) of a future cash flow (such as the cash flow from a cSwap at maturity), we will need to reference a discount rate. Each currency on Swapnet will be evaluated independently to choose the appropriate discount rate (the observable rate which most closely approximates that currency's risk-free rate). The discount rate oracle that each currency uses will be defined by governance token holders when they onboard that currency onto Swapnet.

Periodic cSwaps

We will construct these discount rate curves by dividing our time horizon into discrete regular chunks called "periods". Each period will have an associated cSwap whose start and end blocks match the start and end blocks of that period. These special cSwaps are called periodic cSwaps. The mid-rate of a periodic cSwap represents the market's expectation for the average rate over the course of that period, and it can be conceived of as the "fair price" for the cSwap. By splitting up the time horizon into periods, each with market-determined mid-rates, we can build a discount rate curve that gives us an expectation of the risk-free rate at any point on our time horizon. With this discount rate curve we can infer the mid-rate of any cSwap that lies within our time horizon. The division of the time horizon into discrete periods also provides Swapnet the ability to represent any cSwap's risk in a standardized format. This standardized risk format enables Swapnet to aggregate the risk of an arbitrary portfolio of cSwaps into a single, standard set of figures.

Initially the protocol will launch with a time horizon of one million blocks, and four periods of 250,000 blocks. This means that, at block t, this protocol will only support cSwaps that mature at a block less than t + one million. Governance token holders will be able to change the time horizon and the periodic cSwap cadence through voting. Users can trade swaps that mature at any specific blocks they choose that lie within the time horizon.

We believe that the market will benefit from some degree of standardization and the protocol will encourage users to trade periodic cSwaps as opposed to idiosyncratic cSwaps. This will lead to greater liquidity and more competitive markets. We will initially incentivize trading in periodic cSwaps over idiosyncratic cSwaps, but over the long term we expect that active incentivization will not be necessary to promote activity in periodic cSwaps.

The list of active periods will be stored in the valuation smart contract and queried whenever a cSwap is created. This list will automatically extend on a rolling basis as periodic cSwap time horizons mature.

cSwap Pricing

Here we will formalize the calculation of a cSwap's NPV. First some notation.

B_c = current block number
$X(n)$ = Compound exchange rate at block n
$I(a, b)$ = Non-Annualized Compound lending rate between block a and b (a < b)
$A(a, b)$ = Annualized Compound lending rate between block a and b (a < b)
2102400 = the approximate number of Ethereum blocks in a year

P^i = The set of periodic cSwaps where i represents the period
${B_S}^i$ = the start block for P^i
${B_E}^i$ = the end block for P^i
$M(i)$ = the mid rate of period i, defined as the mid point between the bid and offer rates for the periodic cSwap P^i.

Note the following:

- ${B_S}^i = {B_E}^{i-1}$
- the formula for annualizing an interest rate is:

$$A(a, b) = 1 + (I(a, b) - 1) / ((b - a) / 2102400)$$

- interest rates are composable via multiplication:

$$I(s, e) = I(s, c) * I(c, e) = (X(c) / X(s)) * (X(e) / X(c)) = X(e) / X(s) \ \ where\ s \leq c \leq e\ and\ s \neq e$$

A given period P^i can either be a future period or the current period. If it is the current period, then $B_c > {B_S}^i$. In this case we take $M(i)$ to be the implied mid-rate from B_c to ${B_E}^i$, otherwise we will use the entire period and consider $M(i)$ to be the implied mid-rate from ${B_S}^i$ to ${B_E}^i$. Formally:

If $B_c < {B_S}^i$ for P^i (a future period):

$$Define\ s = {B_S}^i,\ e = {B_E}^i\ and\ s < e$$
$$I(s, e) = M(i) = X(e) / X(s)$$

If $B_c \geq {B_S}^i$ for P^i (the current period):

$$Define\ s = {B_S}^i,\ e = {B_E}^i,\ c = B_c,\ \ s \leq c \leq e\ and\ s \neq e$$
$$I(c, e) = M(i) = I(s, e) / I(s, c) = (X(e) / X(s)) / (X(c) / X(s)) = X(e) / X(c)$$

The method by which we will infer a mid-rate for an arbitrary cSwap within the protocol's time horizon relies upon the composability of interest rates and the existence of observable mid-rates $M(i)$ for each period within the time horizon. For the purposes of our initial launch we will make the assumption that the expected marginal, per-block lending rate is constant throughout a period. This assumption will allow us to use $\{M(i)\}$ to infer the mid-rate over any space of time within the time horizon.

Let us price an arbitrary cSwap C with the usual attributes $B_S,\ B_E,\ N,\ K$. First we need to forecast the expected cash flow to be exchanged at maturity. In order to do this, we need to find the expected floating interest of C : $I_{Float}\ =\ (I(B_E, B_S)\ -\ 1)\ *\ N$, (we already know the expected fixed interest of C : $I_{Fixed}\ =\ K\ *\ N\ *\ (B_E - B_S) / 2102400$).

We can write $I(B_E, B_S)$ as the product of each periodic mid rate $M(i)$ weighted by the number of blocks in period i that fall between B_E and B_S ($b(i)$). This weighted rate can be defined as follows:

$$IR_{weighted}(i) =\ 1 +\ (b(i)\ / 250,000\ *\ (M(i)\ -\ 1))$$

Now we can calculate I_{Float} . Note that if the swap started previous to the current block, we can directly calculate the interest rate that has already been realized by querying compound.

If $B_s\ <\ B_c$ (i.e the start block is before the current block):

$$Define\ s = B_s\ ,\ c = B_c,\ e = B_e\ \ where\ s\ \leq\ c\ \leq\ e\ and\ s\ \neq\ e$$

$$I(s, e) =\ (X(c) / X(s))\ * \prod_i IR_{weighted}(i)$$

$$I_{Float}\ =\ (I(s, e)\ -\ 1)\ *\ N$$

If ${B_s}^C \geq\ B_c$ (i.e the start block is greater than or equal to the current block):

$$Define\ s = {B_s}^C,\ e = {B_e}^C\ where\ s\ <\ e$$

$$I(s, e)\ =\ \prod_i IR_{weighted}(i)$$

$$I_{Float}\ =\ (I(s, e)\ -\ 1)\ *\ N$$

Now that we have I_{Float} we have the expectation for the final cash flow F^C . We'll write it from the perspective of the receiver of the swap.

$$F^C = I_{Fixed}\ -\ I_{Float}$$

The last step in pricing C is discounting this cash flow to the present, so we need to find the discount factor for time B_E. This discount factor is $1 / I(B_C, B_E)$, and we can imply this rate the same way that we implied $I(B_S, B_E)$ above.

$$DF_{B_E} = 1 / I(c, e) = 1 / \prod_i IR_{weighted}(i)$$

Now we have the NPV:

$$NPV(C) = F^C * DF_{B_E}$$

cSwap Risk

Because the NPV of any cSwap can be determined directly from the set of implied periodic cSwap mid-rates $\{M(i)\}$, it follows in more or less straightforward fashion that the riskiness of a cSwap can be measured by taking a series of partial derivatives: $d\,NPV / d\,M(i)$ for each $M(i)$. In other words, the risk of an arbitrary cSwap is described by measuring the change of the NPV resulting from a change in each mid-rate $M(i)$, all other periodic cSwap mid-rates held equal. Standard practice is not to derive this value analytically b ut to measure it empirically: we observe the change in a cSwap's NPV given a one basis point change in the underlying mid-rate $M(i)$. This is also known as DV01 (dollar-value of a one basis point move). So the risk of a cSwap, or a portfolio of cSwaps, will be represented by a set of DV01 figures - one for each periodic cSwap within the protocol's time horizon. DV01 will be the base unit of measurement for risk and collateralization requirements, discussed in the next section.

Collateralization and Liquidation

Each account will require a certain amount of collateral to back its trades. This protocol calculates required collateral by aggregating all of the cSwaps and tPools and returning a single figure that represents the amount of collateral necessary to back the account's portfolio as a whole. Central to this methodology is the idea of risk and collateral netting - that the collateral required to back a portfolio of trades is less than or equal to the sum of the collateral necessary to back each constituent trade when considered independently.

Collateral Requirements

The protocol has to do a variety of collateral calculations during the course of business in order to ensure that accounts are adequately collateralized and behaving within the collateral ratio requirements of the protocol. The system will do a collateral check in the following circumstances:

- To verify whether an account is eligible to place a proposed trade
- To verify whether an account is eligible to transfer a trade to another account
- To put an account into liquidation

The protocol will transform a set of risk figures (a risk ladder) into a collateral requirement first by determining the amount of collateral required for each individual risk period, summing up those required collateral figures, and then applying any collateral netting considerations.

Recall that a risk ladder produces a single DV01 figure per period. A positive figure means the account will make money if rates move up and lose money if rates move down. A negative figure means the opposite. An example risk ladder might look like this:

Period 1: + 500 Dai
Period 2: - 800 Dai
Period 3: + 300 Dai
Period 4: + 100 Dai

For this example risk ladder, the account will make 500 dai if the period 1 mid-rate goes up by one basis point (bp) and it will lose 500 dai if the period 1 mid-rate goes down by one bp. It will lose 800 dai if the period 2 mid-rate goes up by one bp (this account's period 2 risk is the opposite direction of its period 1 risk) and make 800 if it goes down by one bp, and so on.

Swapnet will launch with the following method for calculating an account's required collateral C_R. But first some notation:

$\{P^i\}$ = The set of periods within the protocol's time horizon.
$\{R^i\}$ = The set of risk figures (denominated in DV01) per period i.

$$C_R = \sum_i \left|R^i\right| * 100$$

With this calculation method, our example risk ladder would require 50,000 Dai + 80,000 Dai + 30,000 Dai + 10,000 Dai = 170,000 Dai. Note that this initial construction ignores any collateral netting considerations.

It is worth taking a moment to discuss future upgrades to Swapnet that would introduce risk and collateral netting. Broadly speaking, collateral netting will fall into two categories: netting between different periods on a particular rate curve (like the long position in Period 1 vs the short position in Period 2 in the above example), and netting between positions in different rate curves (like a long position in the Compound Dai lending rate curve vs a short position in the DYDX Dai lending rate curve). The central idea is that risks in two different instruments can offset each other to some extent if the instruments are correlated. The degree of any collateral netting consideration will be voted on and approved by governance token holders.

Account Liquidation

The protocol's liquidation routine is designed to ensure the protection of the system's integrity as a whole while eliminating counterparty risk for any individual participant. The procedure achieves this aim by forcibly paring down the risk of an account whose available collateral has fallen below its required collateral threshold. When an account falls into collateral shortfall, an external participant can call a smart contract to force an account's liquidation. The contract will verify that the account is indeed in default and prevent it from creating new trades (in fact, it would not be able to due to the checks that the RFQ contract puts in place). An incentive payment set by governance token holders will be made to this external participant for successfully triggering an account liquidation to ensure liquidations are triggered in a timely manner.

The liquidation contract will then put a certain portion of the account's entire portfolio (more detail on the determination of this portion later) of cSwaps and tPools up for auction. Mechanically, Swapnet will convert all of the account's cSwaps into a single tPool. At this point, the account's aggregate holdings will be fully represented by a set of tPool tokens. Swapnet will then seize the specified percentage of each series of tPool tokens held by the account and auction them off as a single bundle.

Swapnet has to auction off enough of the account's holdings to be sure that the conclusion of the auction will bring the account to healthy collateralization levels even if the auction does not go as expected. The following example illustrates the nature of this challenge and how Swapnet's design handles it.

Take an account with a collateral balance of 100 Dai and a portfolio of trades with an NPV of 50 Dai and required collateral of 200 Dai. This account is in shortfall because it has 150 Dai of value (balance + portfolio NPV) backing a portfolio of trades that requires 200 Dai of collateral. We can solve for the smallest percentage of the account's holdings that must be liquidated in order to bring it back into equilibrium because the portfolio's NPV and required collateral scale linearly.

$$Define\ c\ =\ collateral\ balance,\ r\ =\ required\ collateral,\ p\ =\ percentage\ to\ be\ liquidated$$

$$c\ +\ NPV\ =\ (1-p)\ *\ r$$

Plugging in and re-arranging values, we get:

$100\ +\ 50\ =\ (1-p)\ *\ 200$
$200p\ +\ 150\ =\ 200$
$200p\ =\ 50$

$p = 1/4$

Simply enough, we derive that Swapnet should liquidate a quarter of this account's portfolio. This would leave the account in perfect equilibrium. The account's collateral balance would be 112.5, its portfolio NPV would be 37.5, and it would require 150 collateral. But considering the implicit assumptions we have made in this calculation introduces complications. We can't be sure that the realized auction price will come in exactly in line with Swapnet's valuation of the portfolio - indeed we should expect that not to be the case. Swapnet's internal valuation framework is an approximation which may occasionally fall out of line with the market's valuation. Furthermore, auction participants will likely demand some discount to the fair price. These uncertainties introduce the risk that the liquidated account remains in collateral shortfall even after a portion of its holdings are auctioned off. To account for this, Swapnet will add in a buffer to its calculation. A simple version of the above function that includes this buffer might look like this:

$$c + NPV = (1-p) * r * (1 - buffer)$$

In practice, this calculation will be slightly more complex and consider the composition of an account's portfolio in more detail. Different interest rate curves, and different points along these curves, may be more or less liquid and thus differently susceptible to mis-valuation by Swapnet. A more nuanced liquidation calculation would consider the granular makeup of a portfolio and apply a more accurate buffer.

Even after applying a buffer, it is still possible that the auction fails to bring an account to a healthy level of collateralization. In this case the account will continue to be eligible for liquidation. If the account is eventually fully liquidated, Swapnet will cover any value shortfall by inflating and selling the governance token.

Liquidation Auction Mechanism

Liquidations will be settled by an on-chain auction. When an account goes into liquidation, it is locked for the duration of the auction; the account cannot add or withdraw collateral or enter into new trades.

Once a portion of a liquidated account's tokens are put up for auction, any account may submit a sealed bid. When the sealed bid is submitted, the liquidation auction contract will do a free collateral check to ensure that the account can take custody of the tokens if it wins the auction. At a specified block height, bids will no longer be accepted. Bidders will then reveal their bids to the contract and the contract will settle the auction to the winning bid.

This method provides us with a secure, straightforward way to liquidate positions in Swapnet; a very important component of maintaining a stable system. Initially, the gas costs for free collateral checks will not be significant but we can imagine a scenario where portfolios are complex enough that the gas costs associated with collateral checks begins to disincentivize bidders from participating in these auctions. We are also aware that this mechanism requires active participation; this may also caused reduced bidding in these auctions if accounts are not constantly monitoring the system. Should these issues prove problematic, we will consider upgrading this mechanism.

Governance

Swapnet’s success depends on a number of ongoing decision processes which cannot be codified because they are not objective by nature. So, Swapnet will require consistent administrative engagement. Decisions regarding collateral netting between correlated instruments will always be ambiguous. We can’t arrive at the right answer to these questions analytically because there is no right answer. Instead, we can create a system of incentives that will ensure that these decisions are made in a way that prioritizes Swapnet’s integrity and long-term success by issuing a governance token.

Transaction fees establish an explicit link between Swapnet’s growth and long-term success, and the appreciation of the Swapnet governance token (SWAP). The foremost outcome we want to avoid is a set of governors who make irresponsible decisions regarding collateralization parameters in order to increase the value of their stakes in the short term by growing the system as quickly as possible while ignoring risks to systemic stability. To counteract this incentive, we use the value of the governance token as the system’s ultimate backstop. If the stability of the system is tested and payment holes begin to emerge due to irresponsible collateralization parameters, it is the governance token holders themselves who foot the bill. This acts as a strong disincentive to decision-making that threatens Swapnet’s integrity. We believe these fundamental countervailing forces will lead to decisions that emphasize Swapnet’s long-term success and maintain systemic stability.

Obligations of SWAP Token Holders

This section will outline the areas where SWAP token holders need to make decisions. The list is long, but votes need not be frequent.

Contract Upgrades

Smart contracts will need to be upgraded to address critical bugs or add new features. Governance token holders must approve these contract upgrades. One such upgrade may be a change to how mid rates are interpolated in the mark to market valuation of swaps, for example.

Collateral Onboarding

Over time, Swapnet will onboard additional assets that can be used as collateral as well as potentially introduce new instruments. Token holders must vote on these new assets as well as set their discount rates, collateralization requirements, and other parameters.

Settlement Oracles and Auctioneers

When settling swaps and liquidation auctions, Swapnet utilizes off-chain validators to provide services to the network. These actors may be required to stake a number of SWAP tokens and/or be elected into their positions. These requirements will be decided by token holders.

Periodic Swaps and High Liquidity Blocks

Token holders will decide on the cadence of periodic swaps and the overall time horizon for each interest rate curve.

Transaction Fees

Token holders will vote on where transaction fees should be implemented and at what rate. Fees can be levied on new trades and used as incentives in the liquidation process. These are the primary methods by which value will accrue to the governance token.

Collateralization Rates and Liquidation Buffers

Token holders will set required collateralization rates and netting considerations. Token holders will also set liquidation buffer parameters.