Ithil

A generalised leveraged investment strategies protocol

V1.0.0 - February 9, 2022

Abstract

Ithil aims at introducing undercollateralised leveraged strategies in DeFi - a game changer for traders, liquidity providers and other protocols who can now rely on a variety of investment products to address their needs. Modular and upgradable at its core, Ithil offers users and other protocols leveraged interactions with the DeFi space, enabling an entirely new range of trading opportunities, acting as an open box financial instrument open to everyone. Simplicity is the key quality of Ithil, allowing everybody from the DeFi novice to sophisticated experts to access many investment strategies.

1 Introduction

The DeFi has recently seen an incredible growth and expansion in the user base, seeing the first institutional players come in too. Built on the premises of being an open space where composability and cross-contamination of protocols sits at its core, access to DeFi is getting more difficult as protocols grow in complexity and require extensive research from their users to come up with truly earning strategies. It is commonly seen how LPs on yield aggregators like Beefy Finance need to often move their liquidity to the newest strategies when the previous ones reach a breakdown point where the APY dramatically falls to 0. Similarly, uneducated users who just want a *quick and dirty* solution may find themselves losing their capital even when using supposedly "safe" blue-chip protocols like Uniswap.

1.1 What is Ithil

At its core, Ithil is built around the concept of under-collateralised lending, allowing a leverage greater than 1. Leveraged investments are a concept that has been around for centuries, allowing to invest more than one's own available liquidity. Leverage makes it possible for users to take advantage of profitable market opportunities even if they don't have immediate access to the funds required. However, it comes with the risk of the trader losing multiples more than in the case of a classical, non-leveraged investment. As always, there is no free lunch.

Ithil enables leveraged interactions with other DeFi protocols in a composable way, offering to its users a curated set of customisable investment strategies to choose from. For example, margin trading using an external dex such as Uniswap or Kyber, or leveraged staking on Yearn, are possible on Ithil using the resident liquidity as principal. In every strategy, the trader pays a *collateral* to be able to use an amount of the protocol's liquidity higher than the collateral itself to open a *position*; such collateral is refunded together with the earnigns obtained when such position is closed.

In case a leveraged position is incurring a loss which would cause the protocol's liquidity to be at risk, Ithil uses a decentralised network of bots to liquidate it and prevent impaired loans from becoming detrimental to the protocol itself. Liquidators are compensated when a losing position is closed by them.

Compared to other standard lending protocols [CITAZIONE], the under-collateralization of loans forces borrowed funds and tokens obtained as a result of the investment to never leave the protocol. When a token is exchanged for another one through an external protocol, such as a dex or a staking one, the obtained tokens are transferred to the strategy contract: only at the closure of the position, the profit and loss is computed and the consequent transfers are performed.

1.2 Why is Ithil different

There are many existing protocols [CITAZIONE] trying to go beyond traditional overcollateralised lending but fail to offer a seamless user experience and remain limited to the realm of a few highly-educated and wealthy DeFi users [CITAZIONE]. Moreover, unlike most protocols [CITAZIONE] which only use

stablecoins-backed loans, Ithil uses *value-agnostic vaults*. This means that the ultimate goal of a vault with a given ERC20 token, is to increase its balance in *that token*; therefore, the vault's dollar value is an immaterial parameter. In this way, Ithil can provide loans backed by any ERC20 token, gaining in this way also the capacity of being a profitable staking protocols for tokens which are difficult to stake elsewhere [CITAZIONE/ESEMPIO]. The precise list of tokens supported by Ithil is community-based, and its rational is preventing malicious tokens to be injected into the protocol.

2 The Vault

As mentioned, leverage is made possible by using the protocol's liquidity. The contract in which liquidity is stored and managed is the *vault*: liquidity providers (LPs) can stake their assets in the vault to provide the necessary liquidity, which will be then used by investors, who will pay fees that will be distributed back to LPs. At the eyes of LPs, Ithil is therefore a staking protocol, in which any asset can be staked to generate APY. This is more flexible than other protocols [CITAZIONE] in which only stablecoins or a restricted class of assets can generate income.

On the technical side, the various investment strategies borrow funds from the vault through the `borrow` function, and repay the loan when the position is closed calling the `repay` function. Since these functions will move the vault's funds, they are not callable by anyone but the strategies themselves, whose addresses are stored in an array within the vault to implement whitelisting. Such array `strategies` can be modified only by governance vote, to allow a new strategy (or disallow an old one) to use the vault's liquidity.

When a strategy is called, the margin is transferred as a form of payment to the vault or to the strategy, depending on the particular strategy used. [DIAGRAMMA]. The strategy then borrows from the vault the amount of capital required to perform the desired strategy and calls the relevant external protocols. When a position is closed, the vault's loan and relative fees are repayed by the strategy. See the strategies dedicated section for more details.

Ithil's vault is *value-agnostic*: once an amount of `XYZ` tokens are staked, we do not track the relative value (in USD or an underlying peg) of such treasury; instead, we simply assure the generation of more `XYZ` tokens.

The vault is responsible for the following tasks:

- Handling staking and unstaking.
- Calculation of the interest rate applied when taking a loan.
- Lending tokens to one of Ithil's supported strategies.
- Keeping account of the net loans and insurance reserve balance.

The vault also contains a list of *whitelisted tokens* approved by the governance that can be provided by LPs or borrowed by the traders. When LPs stake liquidity into the vault, they receive `iTokens`, Ithil's interest-bearing tokens representing liquidity put to work in the vault and can be used on other protocols.

2.1 Fee Redistribution

Ithil generates income in the form of fees paid by the traders. Such fees are then redistributed to the LPs proportionately on how much liquidity they provided by accruing value in the vault and increasing the backing of the yield-bearing `iToken`: LPs will only get rewards in the specific token they provided [DIAGRAMMA]. When tokens are lent for a strategy, the vault calculates the fixed fees and the interest rate applied to the loan. The fixed fees are token-specific, community-based percentages of the amount which goes out of the vault, and are the direct compensation to LP for providing their liquidity to Ithil.

In all cases, fees are considered when deciding whether to liquidate a position, so that the payment is assured for any trader's profit or loss scenario. In particular, a *risk factor* β is set for every position, such that the position is liquidated if the trader's payback goes below $\beta\%$ of the margin. See Section [SECTION] for further details.

2.1.1 Ithil's LP tokens

When an investor deposits an amount D_{TKN} of ERC20 tokens `TKN` in Ithil's vault, an amount M_{iTKN} of LP tokens `iTKN` are minted and transferred to the investor. Letting S_{iTKN} be the total supply of `iTKN`, B_{TKN}, L_{TKN}, T_{TKN} and I_{TKN} the vault's `TKN` balance, net loans, treasury balance (see Section [SECTION]) and insurance reserve balance respectively, then the amount of tokens to be minted is calculated as follows:

$$M_{\text{iTKN}} = \frac{S_{\text{iTKN}}}{B_{\text{TKN}} + L_{\text{TKN}} - T_{\text{TKN}} - I_{\text{TKN}}} D_{\text{TKN}} \tag{1}$$

Similarly, an investor who owns a quantity M_{iTKN} of LP tokens `iTKN`, can redeem them any time to get D_{TKN} `TKN`, still calculated using Equation (1); in this case, the vault will burn the investor's tokens. In the particular case in which the total balance and net loans of the vault is zero (vault's initialization for the particular token), then we mint wrapped tokens in a $1:1$ ratio.

The quantity

$$\sigma_{\text{TKN}} := \frac{D_{\text{TKN}}}{M_{\text{iTKN}}}$$

is a function of the current vault's state and total LP token supply (and not on the particular deposit/withdraw amount), and we call it the *share price* for token `TKN`. A higher share price means that more tokens can be redeemed by burning LP tokens, so we can say that the share price captures the amount of fees generated by the platform.

2.1.2 Absence of arbitrage

Notice that this system is arbitrage-free: assume (we omit the indices not to make the notation too heavy) another investor deposits D' after the first one, thus getting M' wrapped tokens according to Equation (1), and that no fees have been generated in the meantime. This will increase the vault's balance by D' and the total wrapped token supply by M'. According to Equation (1), the

first investor can now redeem

$$D_1 = \frac{B + D' + L - T - I}{S + M'} M = \frac{B + D' + L - T - I}{S + \frac{S}{B+L-T-I} D'} M = \frac{B + L - T - I}{S} M \tag{2}$$

which is exactly the amount of tokens the first investor could redeem before the second one deposited, i.e. $D_1 = D$. Therefore, deposits or withdrawals from other investors do not affect the share price, and in particular, if an investors deposits and withdraws immediately some funds, he or she will get exactly the same amount that has been deposited. This means that the fees generated before one's own deposit cannot be claimed, and arbitrage is impossible.

2.2 Interest Rate

The interest rate is investment-specific and represents the risk beyond the investment as seen from the lender's perspective. Both the interest rate and the fixed fee applied are returned as a parameter from the dedicate vault's functions, while the repayment and the logic for the calculation of risk is included in the implementation of the investment strategies. The riskiness is captured in an integer called *risk factor* β (see following section) which is passed to the vault's `borrow` function as a parameter. The yearly interest rate is then computed as

$$r = \left(1 + \frac{\max(L - I; 0)}{B}\right) \frac{(r_{\text{base}} + \beta)}{\kappa} \tag{3}$$

Here r_{base} is the basic commonly-shared interest rate applied to all tokens and decided by the governance, κ is the *collateralization* of the loan (i.e. the ratio between the collateral posted by the trader, and the amount borrowed from the vault), B is the vault's balance, L is the net loan amount, and I is the insurance reserve balance. Notice that if the position is entirely covered by its margin, i.e. $\kappa \to \infty$, then the interest rate becomes null (but the vault's fixed fee is always applied). This is consistent with the fact that no value has been borrowed from the vault. Notice also that the fees increase with the net loans and decrease with the insurance reserve balance: when too many loans have been granted, it will become expensive to open a new position, thus decreasing the value at risk of the vault.

3 Strategies

We can think of strategies as configurable actions executed across several other protocols in a composable way. They span from basic swaps to convoluted lending and farming of liquidity pools' tokens. In general, the community can develop a strategy and whitelist it into Ithil; conversely, strategies can be delisted from Ithil by a community decision.

In order to be able to open a position in any strategy, a trader has to post a collateral to cover for potential losses caused by unfavorable market movements. The collateral can be posted in any token, and potentially gives the right to use any token type of the ones within the vault. The maximum loan size is determined by imposing that the interest rate, as of Equation (3), should not

exceed a maximum interest rate $r_{\max}$ fixed by the governance; this is the same of saying that

$$\kappa \geq \left(1 + \frac{\max\left(L - I; 0\right)}{B}\right) \frac{\left(r_{\text{base}} + \beta\right)}{r_{\max}}$$

For example, in the scenario of a very risky investment (high risk factor), or when most of the vault is uncovered (insurance pool amount too low), the allowed leverage is lower.

3.1 The base strategy

Although Ithil's strategies can greatly vary in type, they all must inherit by a *base strategy*, which is part of Ithil's core and cannot be listed or delisted. This, besides allowing for a better management of the infrastructure and fewer entry points for breaches, makes the implementation of the single strategies much easier, and enforces a "golden standard" for payout and interest calculations.

The base strategy is made of three functions:

- `openPosition`: this takes the margin payment from the trader, calls the `borrow` function to take the necessary funds from the vault, calls the strategy-specific opening function, and register the data and outcome on-chain in a `positions` array. Collaterals are allowed both in source and destination token.
- `closePosition`: this deletes the position, computes the due fees, calls the strategy-specific closing function, and repays both the vault via the `repay` function, and the trader with a direct transfer.
- `editPosition`: this takes the margin modification (which can be either withdrawals or deposits) from the trader, repays or borrows the necessary funds from the vault, and modifies the on-chain position array to reflect the new setting.

Notice that both the fees calculations and the payout are agnostic of the particular strategy: each strategy must deliver the investment's risk factor, which is then used to compute the due fees.

3.2 Strategy-specific functions

The base strategy contains virtual internal functions which correspond to external contract calls effectively implementing the investment: adding a strategy amounts to implementing these functions. Such functions are `_openPosition`, `_closePosition` and `_quote`. The particular implementation depends on the particular strategy, although the common concept boils down to calling an *external contract* and registering the outcome. In particular, `_quote` is a view function used to monitor the status of the position on-chain: besides being useful for display reaason, it is fundamental for liquidators, which can liquidate positions which have too low a (see Section [SEZIONE]).

3.3 Liquidation

The base strategy also come equipped with the `liquidate` function, which ascertains whether a position's quoted value is below the minimum threshold. In practice, it calls the function `computeLiquidationScore`, which returns a signed integer representing the "health" of the position: calling σ the liquidation score, c the collateral, β the risk factor, and p the payout of the position (which is in turn computed via a call to `_quote`), then we have

$$\sigma = \beta c - p.$$

The position is *liquidatabe* if $\sigma > 0$. If a position is liquidatable, the liquidator can close it and get rewarded for this to compensate for the gas cost and service provided. Liquidators can also aggregate positions to liquidate in batch, to save in gas and overcome other liquidators in speed.

3.4 Examples of strategies

3.4.1 Margin trading

The first strategy we consider and support is Margin Trading. This is a classical investment strategy in which a trader borrows some tokens in an undercollateralized way, and swaps them on a dex to speculate on a price change of a particular token pair. When such swap is performed, a *position* is opened, which is stored onchain on Ithil's contract. The amount of tokens borrowed is called the *principal* of the position, while the amount obtained from the swap is the *allowance*.

Since the loan is undercollateralized, the allowance does not actually goes into the trader's hand. It is more convenient to say that the trader has bought the "right" to move the balance from the dealer's account to and from an external dex (which we will consider fixed).

We define the *ratio* r as the exchange value of `TKB`/`TKA` as published by the dex. This can be seen as the number of tokens `TKA` one can obtain by exchanging one unit of `TKB`. Notice that things in reality are much more complicated, since the very action of exchanging tokens in the dex will move this ratio; we will not discuss these subtleties now and simply assume that if we exchange x `TKB` on the dex, we will get xr `TKA`. Ithil has quoters implemented to get precise estimates of an amount obtained by the dex, which are used to perform liquidation calculations (see later).

Long position Assume the trader believes that the value of the ratio $r = 10$ is about to *increase* in the future, and that his belief is strong enough that he is willing to post a 100`TKA` worth of margin and going into a *long position* on `TKB` with a leverage of 10. At this point, the vault checks if there are 1000`TKA` available. If it is the case, it will exchange them to buy 100 `TKB` from the dex. We assume a fee of 1% on the vault relative to `TKA`, and a risk factor of 50% for liquidation. The vault then registers 900`TKA` in the net loans: they must be repayed when the position is closed.

Assume the day after we observe $r = 11$, i.e. an increase of 10% with respect to the previous day. At this point, the trader can choose to close his position: the vault will sell back the 100`TKB` to the dex, but now 1100`TKA` will be obtained

for this amount. The debt of 900TKA is repayed and the vault will refund the trader with the extra 200TKA, minus 10TKA representing 1% of the swapped amount, which represent the fee of the vault. Therefore the trader will go home with 190TKA, that is a gain of 90% of the original investment when the market has only shown a 10% increase. This happens because of the leverage of 10 the trader has chosen to undertake. To summarize:

- Trader's initial balance: 100TKA, vault's initial balance: 900TKA.
- Trader's final balance: 190TKA, vault's final balance: 910TKA.

Assume instead the day after we observe $r = 9.8$, i.e. a decrease of 2% with respect to the previous day. At this point, the trader can choose to close his position: the vault will sell back the 100TKB to the dex, but now only 980TKA will be obtained for this amount. Similarly to what happened before, the vault's debt of 900TKA is repayed and the remaining amount of 80TKA, minus 10TKA of fees, are transferred to the trader, which will go home with 70TKA, realising a loss of 30% of the original investment.

- Trader's initial balance: 100TKA, vault's initial balance: 900TKA.
- Trader's final balance: 70TKA, vault's final balance: 910TKA.

Assume the day after we observe $r = 9.5$, i.e. a decrease of 5% with respect to the previous day. At this point, we notice that selling back the 100TKB to the dex, one can only obtain 950TKA. Since the margin is of 100TKA, the fees are 10TKA, and the risk factor has been set to 50%, the position will be *liquidated*, i.e. closed by a liquidator, who then repays 900TKA plus fees to the vault, and gets the remaining margin of 40TKA.

- Trader's initial balance: 100TKA, vault's initial balance: 900TKA.
- Trader's final balance: 0TKA, vault's final balance: 910TKA.

Short position Assume the trader believes that the value of the ratio $r = 10$ is about to *decrease* in the future, and that his belief is strong enough that he is willing to post a 100TKA worth of margin and going into a *short position* on TKB with a leverage of 10. At this point, the vault with native token TKB checks if it has 1000TKA worth of TKB in its pool as of today's market condition: i.e. the vault checks if it has 100TKB. If it does, it exchanges them to obtain 1000TKA from the dex and registers 100TKB net loans. Since the trader already posted 100TKA of margin, the allowance for this position will be 1100TKA. We assume a fee of 1% on the vault relative to TKA, and a risk factor of 50% for liquidation.

Assume the day after we observe $r = 9$, i.e. a decrease of 10% with respect to the previous day. At this point, the trader can choose to close his position selling back a quantity of TKA necessary to repay the debt 100TKB, plus 1TKB representing the fee. As of today, only 909TKA are necessary. The extra 191TKA will go to the trader, that is a gain of 91% of the original investment when the market has only shown a 10% decrease.

- Trader's initial balance: 100TKA, vault's initial balance: 100TKB.
- Trader's final balance: 191TKA, vault's final balance: 101TKB.

(Notice that the trader has 1TKA more than the analogous scenario we discussed for a long position: this is caused by the fact that the margin posted is in a token, which is different from the one borrowed from the vault. Since the fees are always in the vault's borrowed token, the trader will profit or lose also on the fees for the exchange ratio movements).

Assume the day after we observe $r = 10.2$, i.e. an increase of 2% with respect to the previous day. At this point, the trader can choose to close his position: the vault will sell back a quantity of TKA necessary to obtain back 101TKB to restore the liquidity and get the fees, but now 1030.2TKA are necessary. The trader will then obtain 69.8TKA by closing this position:

- Trader's initial balance: 100TKA, vault's initial balance: 100TKB.
- Trader's final balance: 69.8TKA, vault's final balance: 101TKB.

Assume the day after we observe $r = 10.5$, i.e. an increase of 5% with respect to the previous day. At this point, the vault notices that 1060.5TKA are now necessary to obtain 101TKB to have back the liquidity plus fees. Since the margin is of 100TKA, and the risk factor is set to 50%, the position will be *liquidated*. As in the long case, a liquidator will close the position and grab the remaining 39.5TKA.

- Trader's initial balance: 100TKA, vault's initial balance: 100TKB.
- Trader's final balance: 0TKA, vault's final balance: 101TKB.

3.4.2 Leveraged staking

Much less risky than margin trading, *leveraged staking* allows trading to stake the vault's liquidity on staking protocols such as Yearn or Aave, and get a boosted APY thanks to the leverage. Margin is needed to cover for occasional losses and to give an implicit deadline on the position: if the interest accrued is so high, that the external protocol's APY cannot cover it, the position could be liquidated.

3.4.3 Pool Together

The leveraged staking can be made more elaborate by calling external contracts with stochastic payouts, such as Pool Together. In this way, the quoter needs to always assure that the margin posted and the eventual payout can repay the loan and the accrued interest, and in case to liquidate the position.

4 Cross-chain support

Ithil will support provisioning liquidity from multiple evm-compatible chains as well as running strategies on other chains too. For instance, LPs can provide liquidity from Ethereum mainnet, Polygon, Avalanche, Harmony, BSC and strategies can be run on all those L2 and sidechains thanks to Composable finance *Persona*s.

5 Tokenomics

Ithil's tokenomics is the study of the management and usage of the governance token `ITH`, and it is conceived to reach some goals for the protocol:

1. Bootstrap the protocol at launch
2. Give particular advantages to holders of the token
3. Back the token value to stable assets
4. Avoid detrimental arbitrage

5.1 Launch

We start from the assumption that, when a protocol is launched, speculative views are more frequent than in the following period, and trust is not high. In particular, speculators will want to buy `ITH` at launch, to profit from market enthusiasm, and `ITH` price will probably show a typical "pump-and-dump" pattern few days after launch.

After covering various expenses, necessary for the platform activation and to compensate the newborn Ithil community, the liquidity obtained should be used to bootstrap the protocol. Since we assume the buyers of `ITH` to be rational, a launch strategy which attracts many funds and stakers will cause the value of the token to be higher.

5.1.1 First phase: partial decentralization

In a first phase, the governance will be concentrated in the developers group and few other chosen individuals, to avoid predatory governance to damage the protocol at an early stage.

Recall that Ithil's vault is two-fold: the *liquidity* is the part of the vault's balance which is free to use, or already used, for investments, while the *reserve* cannot be used for risky investments and is intended to cover the liquidity from failed liquidations. The *reserve ratio* ρ is the percentage of reserve with respect to the total:

$$\rho := \frac{\text{reserve}}{\text{reserve} + \text{liquidity}}.$$

Notice that since Ithil's vault is multi-token, such number is token-specific. To simplify the discussion, we will ignore this in what follows and we will fix one stablecoin, say `DAI`, and refer always to that token's vault. Cross-token issues will be addressed in a separate discussion.

Bootstrap Assume we have raised an amount worth of M `DAI` through the token's launch. We choose an *optimal ratio* ρ_0 to distribute M in the vault such that $M\rho_0$ goes to the reserve and $M(1-\rho_0)$ to liquidity. In this way, the protocol has some liquidity to start operating and generating fees (thus, it will have a positive APY), even if no staker has yet arrived: this should attract the first stakers. Since no wrapped token are minted to the governance (see next paragraph), the governance *must* have a way to sweep away this part of liquidity, otherwise nobody could claim these tokens and they would be lost forever.

Pumping APY When $M(1-\rho_0)$ is routed to the vault's liquidity, no wrapped token is minted to the governance. This means that the *stakers' APY*, meaning the amount of fees per year and per unit staked, is virtually infinite: if a single staker comes to stake 1`DAI`, he is entitled to the entire amount of fees generated by the liquidity in the vault. (Disclaimer: of course this is not infinite but it's very high. In this example if $\rho_0 = 0.25$ and $M = 1000000$, and assuming average fees of 0.1% per day, then the APY seen by that lonely staker would be approximately of $75000\% \times 365 = 27375000\%$ ignoring compounding effects: not a bad return for a stablecoin! If the staker comes to stake 0.01`DAI`, the previous amount is multiplied by 100, and so on). This very high APY will attract stakers after this early stage; when staking, the APY will go down gradually eventually reaching an equilibrium depending on the protocol's usage. After that, people will not stake anymore and will look for more profitable staking opportunities.

Formula to avoid arbitrage. In order to avoid arbitrage, some care must be exercised. We use the same notations as in the whitepaper: B is the vault's balance, T is the part of the liquidity owned by the governance, i.e. the one whose fees must be distributed to the stakers, and we assume that such balance is constant. I is the insurance reserve balance and L are the total loans. In particular, we will have $T + I \leq B + L$. We consider a scenario in which a staker has W wrapped tokens: we need to find how many tokens D he can withdraw. We call S the total supply of wrapped tokens

In order to treat the mathematical problem, we consider a "virtual" supply of T wrapped tokens belonging to the governance, and apply the formula in the whitepaper. The part of balance he can withdraw simply having W wrapped token is

$$D_1 = \frac{B + L - I}{S + T} W,$$

and since he is entitled also to the governance's fees, proportionately to how many wrapped tokens he has, he will be able to withdraw also

$$D_2 = \frac{B + L - I}{S + T} \frac{W}{S} T$$

but he is *not* entitled to the treasury's capital, which is proportionately

$$D_3 = \frac{W}{S} T.$$

Putting things together and simplifying, we obtain that the amount the staker can withdraw is

$$D := D_1 + D_2 - D_3 = \frac{B + L - T - I}{S} W. \tag{4}$$

We thus define, by solving Equation (4) with respect to W,

$$W := \frac{S}{B + L - T - I} D. \tag{5}$$

Notice that these are the same formulas as in the whitepaper but with $B - T$ instead of B: the proof of absence of arbitrage follows in the same way as in the whitepaper.

Consequences At inception, there is a part of fees which need to be assigned to future stakers. The system is such that early stakers take the biggest part of the governance fees, bearing a very high APY for small deposits. In order to overcome such early stakers and get the governance fees, subsequent stakers need to stake more funds, to cover the fees accumulated by the earlier ones.

Example 1. Assume we have accumulated 1000000 `DAI` at inception (from now on, all amounts will be in `DAI` unless otherwise specified) and that we choose the optimal ratio to be $\rho_0 = 0.25$, so that we have $T = 750000$ and $I = 250000$, and a total balance of $B = 1000000$. We assume also no open loans, so $L = 0$, and no stakers, so $S = 0$. Staker A arrives and stakes 1 `DAI`, thus obtaining 1 `iDAI` (notice that in the formula there is a $0/0$ expression at this point, so the actual amount now is irrelevant). At this moment, he is the only staker so he is entitled to the entire fee amount generated by the vault. Assume no staker comes and 1000 `DAI` are generated as fees: staker A can now withdraw 1001 `DAI`, or we can say that 1 `iDAI` will be worth 1001 `DAI`.

Assuming now staker B comes to stake, staking 1 `DAI` only entitles him of $1/1001$ `iDAI`, thus allowing him to grab a tiny portion of the governance fees: the "price" of the governance fees is now much higher. In particular, staker B has to stake 1001 `DAI` to get approximately 1 `iDAI` and therefore being entitled to half the governance fees (the other half being for staker A, who also has 1 `iDAI`).

The above example shows that it is convenient to come first (before fees have been generated) and to stake more than the others do, to obtain a larger portion of the governance fees. This ideally would cause a "run for stake" to get the greatest amount of the cake, with stakers fighting each other to stake more and before all the others.

5.1.2 Second phase: DAO

When the community is solid enough, we will gradually converge to full decentralization. With respect to the previous discussion, the APY pumping and sweep of the treasury must be re-discussed.

Redistribution of governance APY After some point, the amount staked will be much greater than the governance treasury (if it were not, the APY will be still very high and very attractive, thus bringing more stakers). Therefore, the governance fees will only account for a small portion of the protocol fees and will therefore be mostly neglected by the stakers. However, since we assume the governance treasury as *constant*, if many stakers withdraw their funds, the relative percentage of the governance fees will be significantly higher, thus bringing the APY back upwards and attracting funds: we foresee these two contrasting forces to balance each other, eventually bringing to a dynamic equilibrium. The initial parameters to get the best possible equilibrium are yet to be computed.

Sweep of the treasury Via governance vote, the DAO could also decide to sweep away all or part of the treasury or, more aggressively, introduce more funds to raise the APY and produce a new "staking battle" for a new equilibrium.

5.2 Advantage for token holders

We plan to give value to our Ithil token `ITH` through two main systems: liquidations, and insurance reserve. With the former, holders will have the possibility of grabbing the margin of deteriorated positions thus earning the respective tokens. With the latter, holders will be entitled to rebalance the insurance reserve by withdrawing the part above the optimal ratio ρ_0.

5.2.1 Liquidation

As discussed in the whitepaper, liquidation is essential to cover the vault from losses caused by undercollateralized positions. We say that a liquidation is *good* if it closes a position and successfully repays the loan caused by that position, and *bad* if it closes a position but fails to repay the loan, thus attacking the vault's liquidity. A liquidator must stake `ITH` into the protocol in order to be able to liquidate. The amount of loans he can repay through a single liquidation call will directly depend on how much he has staked.

Good liquidation If a liquidation call successfully repays the loan linked to one or more positions, then the remaining margin of the positions is transferred to the liquidator, which therefore earns risk-free. This has no effect on the `ITH` he has staked, and gives a clear advantage of holders of Ithil: the more they have, the more they can liquidate, the more they can profit from this.

Bad liquidation Let us assume that the reserve ratio at the moment the liquidation is called is ρ. Assuming token holders will always try to profit from excess of reserve (see following section), we can assume that the ratio is the optimal one $\rho = \rho_0$. Liquidators should be compensated anyway, otherwise they could simply revert bad liquidations leaving us with dangerous open positions. Since if a liquidation fails to repay the loan, the liquidity amount and the liquidator's compensation will be covered by the insurance reserve, *a bad liquidation will decrease* ρ (assuming no withdrawals are made in the meantime). Therefore, some countermeasures should be taken in order to reduce the probability of such an event.

5.2.2 Insurance reserve

The optimal ratio ρ_0 of Paragraph 5.1.1 is used to determine if profit is generated by Ithil. When fees are generated, a fraction bigger than ρ_0 goes to the insurance reserve (the calculation of such portion is dynamic such that the lower the insurance reserve balance, the higher ρ. The precise calculation is illustrated in a dedicated section of the documentation). This means that, if few bad liquidations occur, the reserve ratio will grow with fees.

After some time, the vault will be in a state such that $\rho > \rho_0$: after a certain threshold calculated to not waste gas, a bot will swap the extra amount to restore the optimal ratio ρ_0, obtaining stablecoins in the process. These stablecoins are transferred to the backing contract (see Section 5.3) and will raise the bid price of `ITH`.

5.3 Token backing

Dumps can be very detrimental to holders at early stage, therefore we will use a system of incremental backing, which never decrease its value with time. This assures that `ITH` cannot ever go below a certain *backed price*, and that such price is always non-decreasing.

5.3.1 The bid price

This is a contract with one or more stablecoins in its balance. For simplicity, we assume there is only one kind of stablecoin, say `USDC`, in the backing contract. The *bid price* of `ITH` will then be

$$\beta = \frac{B}{S} \tag{6}$$

where B is the `USDC` balance of the backing contract, and S is the total `ITH` circulating supply (that is, excluding the supply locked in the backing contract).

Ithil always guarantees that any `ITH` can be redeemed on the platform at the bid price, therefore the market cannot price `ITH` less than β, otherwise this will lead to arbitrage (and subsequent increase of the market price). Notice that redeeming does not keeping the bid price constant: redeeming n `ITH` we will have

$$\beta' = \frac{B - n\pi}{S - n} = \frac{B - n\frac{B}{S}}{S - n} = \frac{B}{S} = \beta.$$

The excess insurance reserve is swapped into stablecoin by a bot, and transferred to the backing contract, thus increasing B and ultimately increasing β. Since there is no event which causes a decrease of β, we see that *the bid price is a non-decreasing function of time*, and it strictly increases every time fees are distributed to the governance.

5.3.2 Initial backing

When the token is emitted and publicly sold the first time, we can transfer part of the funds obtained to the backing contract, so to guarantee the value of the token cannot go below it. Since no minting is allowed at the first place, there is no direct arbitrage possibility if the token is traded *above* the bid price, so the backing implemented is not a pegging. See the following section for further details.

5.4 Avoiding detrimental arbitrage

As already discussed, a market price below the bid price will open the way to an arbitrage opportunity (buy on market, sell on Ithil), thus bringing the `ITH` market price up. If we also allowed people to *buy* `ITH` on Ithil at the bid price, and if the market price is above the bid price, an arbitrage opportunity will arise from buying (mint from Ithil, sell on market). While the former arbitrage can be considered "good", in that increases the token market price, the latter is detrimental to token holders, since it decreases the token market price.

5.4.1 The ask price

We define the *ask price* α be the price such that `ITH` can be bought on Ithil at α. To avoid a trivial arbitrage on Ithil, we clearly must have

$$\alpha \geq \beta.$$

We then define the *spread* σ as the percentage difference of the bid and the ask price:

$$\sigma := \frac{\alpha - \beta}{\beta} \times 100\%.$$

Notice that letting π be the market price of `ITH` on a separate dex, then no arbitrage opportunity arises as long as $\beta \leq \pi \leq \alpha$. In particular, a higher spread means more freedom on the market price, while a zero spread completely locks the price of `ITH` to one single price (the bid/ask one). From now on, we will always assume $\sigma > 0$.

Liquidity mining Following our philosophy that `ITH` should directly represent the protocol income, the stablecoins extra resulting from a nonzero spread are transfered to the vaults, in the form of treasury-owned liquidity (see Section 5.1). This will have the result of pumping again the APY for that vault, increasing the insurance reserve, and ultimately constructing a positive feedback mechanism which is beneficial to the platform and, in turn, to the holders.

The spread can be dynamically adjusted based on the vault's insurance reserve, so that investors and arbitrageurs can profit from providing liquidity to the right stablecoin vault. A higher insurance reserve means in turn lower interest rates, thus more trades and ultimately more fees and profit for the protocol.

6 Risks

7 Competitors

8 Acknowledgments

We would like to thank the incredible Ethereum community for its support and welcoming atmosphere as well as ETHGlobal for running hackathons and on boarding new developers while creating connections with key projects in the DeFi space like Uniswap or Yearn finance.