# The Yield Protocol: On-Chain Lending With Interest Rate Discovery

Dan Robinson dan@paradigm.xyz

Allan Niemerg allan@yield.is

April 2020
WORKING DRAFT, rev. 1

**Abstract**

This paper presents a sketch of a new building block for decentralized finance: yTokens. yTokens are like zero-coupon bonds: on-chain obligations that settle on a specific future date based on the price of some target asset, and are secured by collateral in another asset. By buying or selling yTokens, users can synthetically lend or borrow the target asset for a fixed term. yTokens are fungible and trade at a floating price, which means their "interest rates" are determined by the market. The prices of yTokens of varying maturities can be used to infer interest rates, and even to construct a yield curve. Depending on the target asset, yTokens can settle through "cash-settlement" using an on-chain price oracle, through "physical settlement" in the target ERC20 token, or by synthetically issuing or borrowing the target ERC20 token on another platform.

## 1 Introduction

The Yield Protocol is a standard for a token that settles based on the value of a target asset on a specified future date, and which is backed by some quantity of a collateral asset. We call these tokens "yTokens."[1]

You can create yTokens by depositing collateral, then sell them to effectively borrow (and short) the target asset. Buying yTokens is economically similar to lending the target asset. The effective "interest rate" received by yToken holders is determined by the discount at which yTokens currently trade, as well as the time to maturity.

yTokens can also be used as a building block for more sophisticated products. While each maturity of yTokens has a fixed expiration date, you could build a perpetual product on top of it, by implementing a pool that invests in short-term yTokens and automatically rolls them over upon expiry.

[1] This nomenclature is inspired by the "cTokens" used in the Compound Protocol [1].

Perhaps most interestingly, the market prices of yTokens can be used as an "interest rate oracle." The price of each maturity of yToken implies a particular yield. These implied yields could be used to settle on-chain interest rate derivatives, to inform Maker's choice of stability fee, or as an input to the interest rate formulas used by Compound or dYdX. By charting those yields for different maturities, we can even construct a yield curve, which could be a useful indicator of the expected path of interest rates or prices.

This paper will present three constructions of yTokens. The first ("cash settlement") assumes the existence of a price oracle that, when asked, can tell the contract the value of the target asset in terms of the collateral asset. The second ("physical settlement") does not require any price oracle, but assumes that the target asset is itself a token (rather than an off-chain asset like USD). The third (settlement to synthetics) assumes that the target asset is either a collateralized synthetic like Maker, or supported by a floating-rate lending platform like Compound.

# 2 Prior work

The design of the Yield Protocol is heavily influenced by other projects on the Ethereum blockchain that offer synthetics, borrowing, lending, and/or leverage. Yield primarily differs in that its interest rates are implicit and set by market prices, rather than being set by governance or a formula. Additionally, whereas most lending protocols use floating interest rates, Yield enables term loans with fixed interest rates, while still maintaining some degree of fungibility. Section 4.3 shows how, thanks to these properties, the Yield Protocol can provide unique insight into the term structure of interest rates for on-chain lending.

Synthetic asset systems, like Maker [2], Synthetix [3], UMA [4], and the Rainbow Network [5], create tokens that are pegged to a target asset but backed by a different collateral asset. These synthetics are supposed to trade at parity with the target asset (which often necessitates some kind of interest rate, usually set by governance or agreement of the parties involved). By contrast, yToken prices are expected to float, and the interest rate is implied by the discount at which they trade. Also, unlike most of these synthetics, yTokens have an expiration date.

Systems for on-chain lending or margin trading, such as Compound [1], Fulcrum [6], and dYdX [7], allow users to deposit assets to be lent out for interest, and/or as collateral for their own borrowing. These protocols offer floating "overnight" interest rates determined by a formula, and positions can be withdrawn at any time. By contrast, Yield's interest rates are determined by the market price of yTokens. Holders earn a predictable interest rate if they hold their yTokens to term, but "withdrawing" early requires selling yTokens on the market, and thus might incur a loss.

An early iteration of Dharma [8] was a secured lending platform that allowed users to enter into collateralized loans with a fixed term and interest rate. Lenders and borrowers were individually matched. Positions in Dharma were

tokenized, but were not fungible (even if they had the same expiration date), making price discovery and liquidity provision more difficult.

# 3 Mechanism

This section describes the yToken mechanism at a high level. For simplicity, this description assumes the existence of a synchronous free on-chain price oracle for the underlying asset. Section 5 discusses other constructions of yTokens that do not depend on this assumption.

yTokens differ from each other in four dimensions: target asset (or oracle), collateral asset, expiration time, and collateralization requirement. Anyone can define a particular yToken by specifying those four parameters. For example, there would be one yToken for a given USD oracle, backed by ETH, settling at 11:59 PM on December 31, 2019, with a 150% collateralization requirement. Applications will be somewhat incentivized to converge on focal points (*i.e.*, by conventionally only buying or selling yTokens that expire at the end of each quarter), to ensure that liquidity is not too fragmented. yTokens that settle to synthetics or a floating-rate lending platform, as described in section 3.4, should mirror the parameters (such as collateralization requirement) used by the target platform.

One example of a yToken might be "yDAI/ETH (2021-01-01)". This refers to a yToken with DAI as the target asset, ETH as the collateral asset, expiring at 12:00 AM on January 1, 2020. If referring generally to any yToken that targets DAI, we just use the term "yDAI".

Each maturity of yToken has its own ERC20 token contract. These contracts are initialized and tracked using a global registry contract.

## 3.1 Minting yTokens

Once a token contract exists for a particular yToken, anyone can deposit collateral to create a *vault*. These vaults are analogous to (and named after) the vaults in the Maker system [9]. The owner of a vault can mint yTokens, which adds to the vault's *debt*. They can also burn yTokens to reduce their debt. The debt of a particular vault must not exceed the value of its collateral plus some required *margin*, or it will be *liquidated*, as described in section 3.5.

A yToken resembles a secured zero-coupon[2] bond. Upon expiration, it can be redeemed from the yToken contract for its face value. yTokens from different vaults in the same token contract are fungible.

[2]You can construct an instrument that includes coupon payments by creating a collection of different yTokens, with different face values and maturities. This process is essentially the inverse of "coupon stripping" from traditional finance.

## 3.2 Settlement

Each yToken system needs a settlement mechanism. The purpose of the settlement mechanism is to ensure that yTokens trade at the same price as the target assets at the moment of maturity.[3]

### 3.2.1 Cash settlement

"Cash settlement"—or, more precisely, settlement paid in the collateral asset—is a settlement mechanism that depends on the existence of a precise price oracle for the price of the target asset in terms of the collateral asset.

In this mechanism, yTokens can be redeemed for their face value as of the moment of maturity, with the redeemer receiving the appropriate amount of the collateral asset. Mechanically, as soon as a particular yToken expires, anyone can call a function on the token contract that triggers *settlement*. That function calls the oracle to look up the current price of the target asset in terms of the collateral asset and stores it as the settlement price. After that point, anyone can redeem yTokens for an equivalent amount of collateral, at the settlement price.

Additionally, after maturity, any vault owner can withdraw their collateral, minus the value of the outstanding debt as of the moment of maturity.

The advantage of this settlement mechanism is that it can support arbitrary target assets; the other settlement mechanisms described only support ERC20 assets. However, it requires a precise oracle for the price at the moment of settlement. Additionally, after the moment of settlement, yToken prices begin to track the price of the collateral asset (since they are redeemable for a fixed amount of that asset), rather than continuing to track the price of the target asset.

## 3.3 Physical settlement through auction

If the target asset is itself an ERC20 token on the Ethereum blockchain, the yToken could settle using physical settlement, meaning yToken holders receive the underlying token.

This could be implemented with an auction. For each vault that has outstanding debt, the collateral can be sold in a reverse Dutch auction to repay that debt. Suppose an expired vault has 1 ETH in collateral and 100 yDAI in debt outstanding. The protocol would offer anyone 0.01 ETH in exchange for 100 DAI, gradually raising that offer over time until someone accepts.[4] The remainder of the collateral would be returned to the vault creator, while the tokens collected in these auctions would be distributed to yToken holders.[5]

[3] It is also convenient for them to continue to trade close to the target asset after maturity, but this is not strictly required, and one of the mechanisms—cash settlement—does not achieve this goal.

[4] The vault holder could short-circuit this process by either repaying their yToken debt or by accepting the auction's offer themselves.

[5] If the collateralization requirement was high enough that the vault does not become

One advantage this mechanism has over cash settlement is that after this auction, assuming it is successful, each yToken will be fully backed by the target asset, rather than being backed by some amount of the collateral asset. This means that yToken holders will maintain the same exposure to the target asset without having to take any action themselves (though they will no longer be earning a yield on it), and can redeem it at their leisure.

However, this mechanism does not allow *borrowers* to maintain their debt position after maturity; it leaves them only with unlevered exposure to the remainder of their collateral.

## 3.4 Settlement to synthetic

When the target asset for a yToken is a collateralized synthetic asset like DAI, the yToken can use the token's own issuance mechanism for settlement.

For example, when yDAI backed by ETH matures, the protocol can move its ETH collateral into a Maker vault. As yDAI holders show up to redeem their yDAI, the protocol can borrow DAI from Maker and pay it out to the holders. As borrowers show up to pay off their debt, the protocol pays down any debt to Maker, and releases the borrower's collateral back to them. Alternatively, the borrower may fork their collateral and debt off into their own Maker vault.

When a yToken matures, the protocol needs to charge borrowers a floating rate to keep the debt position open, to account for the stability fee that it may need to pay if it has to borrow DAI from Maker. Conversely, the protocol may begin to pay additional yield to yToken holders at a floating rate.

Therefore, another way of looking at this settlement mechanism is that users' fixed-rate yToken positions are "rolled" into floating-rate debt at maturity.

In the example of yDAI backed by ETH, at maturity the protocol could begin to charge borrowers the Maker stability fee on ETH, while paying lenders the Dai Savings Rate. This would guarantee that the protocol itself does not run a deficit (assuming the Dai Savings Rate does not exceed the stability fee).

A major advantage of this mechanism is that both borrowers and lenders can continue to keep the same positions open after maturity, with the only difference being that they are now paying floating-rate interest rather than fixed.

### 3.4.1 Settlement to borrowing platform

Most ERC20 tokens cannot be minted simply by posting collateral. Nevertheless, we can generalize from settling to a synthetic platform like Maker to settling to other platforms that allow users to borrow and lend the target asset against the same collateral asset.

For example, certain yTokens could settle to Compound. At settlement, the yToken protocol posts its own collateral to Compound in order to borrow the target asset for settlement. Borrowers would pay the Compound borrow

undercollateralized during this process, we would expect all of these auctions to complete successfully. If some of the collateral remains unsold when the auction is done, that collateral could be distributed to yToken holders along with the physical assets.

rate until they repay their debt, and lenders would receive the Compound lending rate until they redeem their yTokens for cTokens for that asset, which the protocol borrows from Compound as needed.

It follows that yTokens may be settled to any other protocol that permits synthetic exposure to a target asset for collateral (whether by minting or borrowing it). As long as that exposure closely tracks the market price of the target asset after maturity, yToken prices before maturity should reflect the true yield curve of the target/collateral asset pair.

### 3.5 Liquidation

If the price of the collateral asset falls relative to the target asset before a yToken expires, some of the vaults—and thus the yTokens they back—may become undercollateralized. To protect against this, anyone can liquidate a vault that is too close to being undercollateralized.

A vault can be liquidated if the value of its collateral (the current price of the target asset in terms of the collateral asset, $P$, times the quantity of collateral in that vault, $C$) is less than the value of its debt $D$ adjusted for the collateralization requirement $R$: $P \cdot C < D \cdot R$. The liquidator burns $D$ of the same yTokens, and receives $C$ collateral.

The extra margin in the contract serves as a penalty for the borrower and an incentive for the liquidator. When $D < P \cdot C < D \cdot R$ for a particular vault, someone can collect a profit by liquidating that vault. This also incentivizes borrowers to top up their vaults with additional collateral (or pay down debt) to avoid liquidation.

For example, suppose a vault for a yToken with a collateralization requirement of 150% has 1 ETH as collateral, and outstanding yToken debt with a face value of 100 USD. If the price of ETH drops to $149, someone can liquidate the vault by burning 100 yUSD and receiving 1 ETH, for a total profit of at least $49 (in fact a little more, assuming the yUSD was trading at a discount). Alternatively, the collateral could be auctioned to repay, with the vault owner receiving some or all of whatever is left. This is similar to the liquidation mechanism used in Maker and Compound.

If the price drops so quickly that a vault goes from $P \cdot C > D \cdot R$ to $P \cdot C < D$ before anyone has a chance to liquidate it, it is possible for the system to become undercollateralized. In the above example, this would happen if the price of ETH fell from $150 to $99 before someone was able to profitably liquidate it. This "gap risk" is one reason to set high collateralization requirements, especially when the backed asset or target asset is particularly volatile.

## 4 Applications of yTokens

yTokens are a low-level primitive, and they are not designed to be particularly friendly to end users. However, they could be used to construct interesting

products by composing them with other protocols or building human interfaces on top of them.

## 4.1 Borrowing, lending, and leverage

yTokens enable a fungible market for fixed-term secured lending on-chain. By minting, holding, and/or trading yTokens, users can synthetically borrow and lend the target asset. Users are guaranteed a particular interest rate if they hold the position to maturity.

### 4.1.1 Borrowing

Opening a vault, taking out yTokens, and selling them is economically similar to *borrowing* the target asset and selling it. If the price of the target asset rises, so does the value of your debt, so the value of your vault decreases. You might want to do this if you expect the target asset to fall in price ("shorting"), or if you expect the collateral asset to rise in price and want to increase your exposure to it ("leverage").

Traders can get additional leverage by trading the yTokens for more of the collateral asset, depositing it into the vault, and taking out more yTokens. This gives them greater long exposure to the collateral asset (and greater short exposure to the target asset), but increases their risk of getting liquidated. An application could abstract away this process for the user.

The protocol could also be designed to make leveraged trading more efficient, by having a function that atomically mints yTokens, sells them on Uniswap [10] for the collateral asset, and only *then* checks that the vault is sufficiently collateralized (reverting the entire transaction if it is not).

### 4.1.2 Lending

Buying yTokens is economically similar to *lending* the target asset. Because yTokens are not redeemable until expiration, they are likely to trade at a discount until maturity, particularly if there is demand to borrow the target asset. This means the value of yTokens (denominated in the *target* asset) will tend to appreciate over time as they approach maturity. This is analogous to the interest earned by lenders in other protocols.

In order to make this more familiar for users, an application could present an interface that emphasizes the *market value* of their yTokens, rather than the face value, as well as showing them the expected annualized yield if they held those tokens to maturity. For example, if a user spent \$100 to buy 103 yUSD, the interface would display their current value of \$100, with that number gradually tending to rise until it hits \$103 upon expiration.

However, note that it is possible for yTokens to temporarily *decline* in value relative to the target asset. This poses something of a user interface challenge—a user earning "interest" on their DAI might be surprised to see their portfolio value tick down (even though it would necessarily correspond to the interest

rate going up, and the lender could receive the full value simply by holding the yToken to maturity).

#### 4.1.3 Example

Suppose that on October 1, 2019, Alice owns 1 ETH, currently worth $100, and she wants to enter a 1.5x leveraged long position on ETH. Alice looks up the USD/ETH yToken contract that expires at the end of the quarter (on December 31, 2019) with a collateralization requirement of 150%.

Alice creates a vault in that yToken contract, deposits 1 ETH as collateral, and takes out 50 yUSD as debt. Alice then sells those yUSD on Uniswap (see section 4.4). yUSD of that maturity is currently trading at a discount—say, $0.97—so she only receives 0.485 ETH. She deposits that ETH into her vault as additional collateral. She now has exposure to 1.485 ETH.

If Alice wanted, she could continue to take out yUSD, trade it for ETH, and redeposit the ETH as collateral, repeating until she reaches her desired amount of leverage or she bumps up against the collateralization requirement.

Since Alice's vault has $50 of debt outstanding and a collateralization requirement of 150%, it gets liquidated if the value of her collateral falls below $75, which corresponds to the ETH price falling to about $50.50. If someone liquidates her vault at that exact price, they will have to pay in 50 yUSD, and will receive $75 worth of ETH collateral, giving them a profit of $25, plus a little more based on the discount at which the yUSD are currently trading. If the price of ETH gets too close to her liquidation price, Alice should try to close her vault or deposit more collateral to avoid paying this penalty.

Suppose Alice holds the position until expiration. When it expires, $50 worth of her ETH collateral will go to yToken holders, and the rest will go back to her. So if the price of ETH rose to $200, yToken holders would receive 0.25 ETH of her collateral, and she would receive 1.235 ETH, worth $247.

Alternatively, Alice can exit the position early at any time by purchasing yTokens (likely, though not necessarily, at less of a discount than she initially sold them for), and burning them to repay her debt.

Now suppose Bob wants to lend $100 and earn interest on it until December 31, 2019. Bob can simply go to Uniswap and purchase yUSD. At the current discounted price of $0.97, Bob's $100 can buy about 103.09 of these yUSD. At maturity, Bob's yTokens will be redeemable for $103.09 worth of ETH, so he will have earned $3.09 in interest. If Bob wants to exit his position before that, he could do so by selling his yTokens (likely, though not necessarily, at a higher price than he initially paid for them).

#### 4.1.4 Applications of lending and borrowing

Most on-chain borrowing today involves borrowing "stable" assets, especially dollar-pegged assets like DAI or USDC [11], using ETH as collateral, and selling it for ETH. This would likely also be the most popular usage of yTokens. Indeed, creating a vault with ETH, minting yDAI, and selling it for ETH is quite similar

to depositing ETH as collateral into Maker or Compound, taking out DAI, and selling it for ETH.

Another possible use case is shorting an asset that you think will decline in value. To short an asset ABC, you can deposit a stable asset (say, DAI) in a vault, mint yABC, and sell it.

For greater capital efficiency (though potentially greater risk), you can get a kind of "rehypothecation" by using interest-bearing assets like Chai (a wrapper for DAI deposited into Maker's Dai Savings Rate), Compound's cTokens [1], Fulcrum's iTokens [6], or even other yTokens as a collateral asset. For example, you might prefer using cDAI—which represents an interest-earning claim on DAI from the Compound system—as collateral for your ABC short, rather than DAI.

Finally, certain yTokens allow you to speculate on interest rates. For example, yDAI whose collateral is Chai resembles, for the borrower, a pay-fixed receive-floating interest-rate swap. By depositing Chai, minting yDAI, and trading it for more Chai (and repeating to reach the desired level of leverage), you are effectively betting that the floating rate you earn on the Chai will be higher than the fixed rate you pay on the yDAI.[6] Conversely, yCHAI (which settles to a target amount of Chai, which will be worth some yet-to-be-determined quantity of DAI based on the cumulative Dai Savings Rate over that period) whose collateral is yDAI resembles, for the borrower, a pay-floating receive-fixed interest-rate swap.

## 4.2 Interest rate oracle

The price of a particular yToken floats freely, and is set by supply and demand. As mentioned in section 4.1.2, this means that yTokens will usually trade at a discount to their face value until expiration. This discount, along with the time to maturity, can be used to infer a *yield*—the annualized interest rate that would be earned by purchasing the yToken and holding it to maturity—in much the same way that you would infer an interest rate from the price of a zero-coupon bond.

Suppose yUSD that expire one year from today have a face value of $1 but are currently trading at $0.50. If you invested $1 in those yUSD, you would be able to buy 2, and you would receive $2 worth of ETH one year from today. Therefore, the yield on those yUSD (and the implied interest rate on the 1-year "loan") is 100%.

There is a simple formula for computing the annualized yield $Y$ on any yToken (where $F$ is face value, $P$ is present value, and $T$ is number of years to maturity):

$$Y = (\frac{F}{P})^{\frac{1}{T}} - 1 \tag{1}$$

[6] One difference from a traditional interest-rate swap is that the fixed leg is effectively paid upfront, when the user sells their yDAI at a discount. This does limit the amount of leverage that they can achieve.

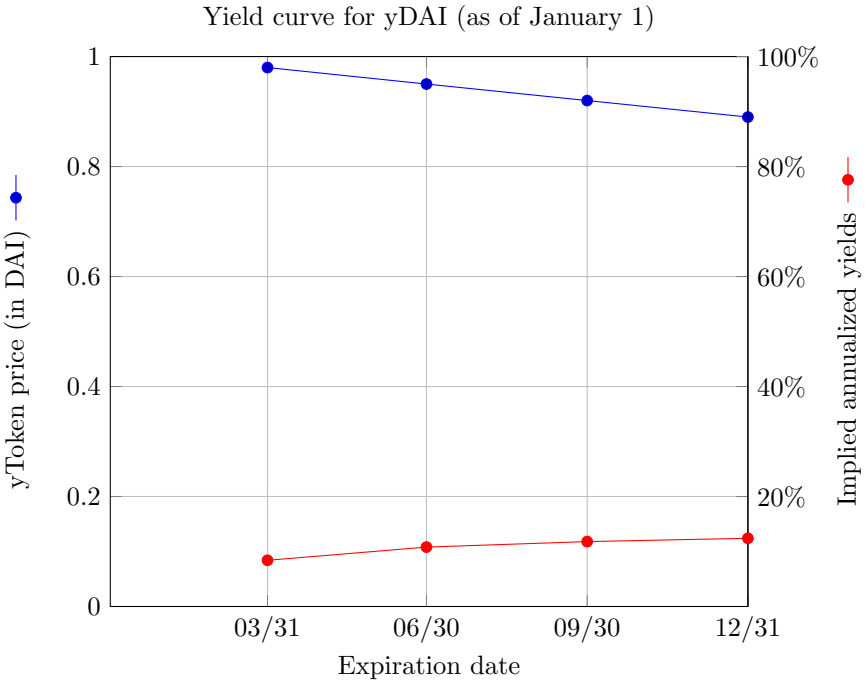
For example, in the above example, a $1 yToken that expired in 3 months was trading at $0.97. This price implied an annualized yield of $(\frac{1}{0.97})^{\frac{1}{0.25}} - 1$, or about 13%.

The yield on short-dated yTokens could serve as an indicator of the "spot rate" for borrowing that asset. This could be used as an input to inform the choice of interest rate paid by protocols like Maker, Compound, and dYdX. If the price could be determined on-chain, it could be used to settle on-chain interest rate derivatives (such as swaps).

## 4.3 Yield curve construction

yTokens for the same underlying asset but with different maturities will likely trade at prices that imply different annualized yields. This should reflect the market's expectations about how short-term interest rates will change over time—that is, the *term structure* of interest rates. (In practice, it will also be affected by other factors, including liquidity or the perceived risk of a bug in one of the smart contracts.)

You can graph the implied yields of different maturities of yToken for the same underlying asset, to create a *yield curve*:




The yield curve can be used to inform the governance of protocols like Maker and Compound, as well as providing potentially useful economic information to traders and analysts.

### 4.4 Market making

The above applications depend on the existence of a liquid market in some yToken. Traders who want to profit from such a market could deposit liquidity into Uniswap [10], an automated market maker protocol.

Uniswap v2 supports arbitrary ERC20-ERC20 pairs [12]. These make it possible to market-make directly betweeen yTokens and the underlying tokens (when those are ERC-20s), such as yDAI/DAI. Providing liquidity to such pairs is relatively less risky, because you know what the relative prices of those assets will be upon maturity, so your losses as a liquidity provider are capped.

Because 1 yDAI is guaranteed to be worth 1 DAI at maturity, the future value of a yDAI/DAI liquidity token is predictable (as long as the parameterization is safe). This means the system can allow these liquidity tokens to be used as collateral to mint yDAI (without even needing to support liquidation). This allows users to lever up their liquidity provision, by putting DAI and yDAI into Uniswap to mint liquidity tokens, using those liquidity tokens to borrow more yDAI, trading half of it for DAI, depositing the yDAI and DAI to create more liquidity tokens, and repeating until sufficiently leveraged.

More configurable automated market makers might expand from merely trading yTokens to minting and burning them, or trading between yTokens of different maturities to arbitrage different yields. A Balancer pool [13] could hold multiple maturities of yDai, and could set a lower fee.

### 4.5 yToken portfolios

One drawback of yTokens, compared to other on-chain systems for borrowing or lending a target asset, is that they are not perpetual. To maintain a particular position indefinitely, a borrower or lender would have to regularly roll over their position, selling expiring yTokens and buying longer-dated ones.

Since many users might be interested in such a strategy, especially on the lending side, they might be able to pool their assets together and execute it on-chain. The full description of such a system is beyond the scope of this paper, but it could potentially be implemented as a Set token [14] or Balancer pool [13] that maintains a rebalancing basket of yTokens. Some strategies the portfolio might use include:

- Holding only yTokens that expire at the end of the current quarter, and rolling them over the day they expire.
- Targeting some *duration* (average time to maturity), and regularly rebalancing its portfolio, changing the weights of each asset, to maintain that duration.
- Seeking to hold the highest-yielding maturities available, without regard to duration.

These approaches have the beneficial side effect of market-making for those yTokens.

# 5 Alternatives to synchronous price oracle

As described above, the protocol sometimes needs to know certain facts about the price of the target asset. This is simple if the contract has continuous access to a synchronous cheap price oracle that provides the price of the target asset in terms of the collateral asset. For some asset pairs, like ETHUSD, the contract may be able to piggyback on an existing oracle, like the one operated by Maker.[7]

If no such oracle exists, there are alternative ways of constructing the contract, as described below.

### 5.0.1 What we need from our "price oracle"

The contract needs to learn something about the price of the target asset (in terms of the collateral asset) in the following cases:

- When yTokens are minted, the protocol needs to know that the vault is fully collateralized.[8]
- When someone attempts to liquidate a vault, the protocol needs to know that the position is undercollateralized.
- If using the cash-settlement mechanism 3.2.1, when the yToken is settled after expiration, the protocol needs to know the price of the target asset (in terms of the collateral).

Note that for the first two categories, (*i.e.*, when determining whether someone should be allowed to withdraw yTokens, or determining whether a vault should be liquidated), the security of the system does not depend on an extremely precise measurement, because you can trade off capital-efficiency for precision.

For example, suppose you are only able to determine the relative price within a factor of 2. If you would otherwise have set the collateralization requirement to 150% (so a vault with $100 in debt would be liquidated when the collateral was worth $150), you can set the requirement to 300% instead, and give yToken holders the same level of protection against gap risk. (vault creators would likely want to maintain at least *600%* collateral, to avoid the risk of getting unfairly liquidated.)

[7] If the yToken is already using Maker as a settlement mechanism, as described in section 3.4, then also using it as an oracle may not add significant additional trust assumptions.

[8] Such a check also needs to be done when removing collateral from a vault. However, removing collateral from a vault is economically equivalent to repaying the debt, closing the vault, opening a new vault with the reduced collateral amount, and taking out the same amount of debt. The same rules can therefore be applied to both cases, so we only describe the rules in terms of taking out additional debt.

#### 5.0.2 Price oracles for on-chain assets

If the target asset is an ERC20 token, that could make it easier to create an oracle for it. There are many plausible ways to construct an imprecise relative price oracle for ERC20 tokens. One possibility is to use Uniswap v2's TWAP accumulator as a price oracle [12].

Another possibility is to use an on-chain exchange that tracks the length of time that a particular offer is open. A long-lived offer to trade the target asset for the collateral asset can be used to infer a lower bound on the target asset's price, and can therefore be used to liquidate a vault.

#### 5.0.3 Asynchronous or expensive price oracles

If the price oracle is asynchronous (like UMA's Data Verification Mechanism [15]), or expensive to use, it may be impractical to use it every time a vault is created or liquidated.

There are some optimizations that could reduce the number of calls to the oracle (although they may require setting larger collateralization requirements to maintain the same level of safety). To sketch out some of them:

- When a vault is liquidated, instead of calling the price oracle, the liquidator could put up a security deposit. The vault's owner can then cancel the liquidation and claim the security deposit by calling the oracle and proving that the vault was fully collateralized. If the liquidation has not been cancelled within 24 hours, the liquidation completes.
- When yTokens are minted, the creator could either call the price oracle, or choose one of the following alternatives:
  - Point to an existing vault for the same yToken with an equal or lower liquidation price, that is not in the process of being liquidated. The existence of such a vault implies that the liquidation price has not been reached.
  - Initiate a withdrawal of new yTokens, then wait 24 hours. During that waiting period, anyone can initiate a liquidation of the position if it is undercollateralized (based on the *post-withdrawal* quantity of collateral). Once the waiting period is over, if no such liquidation is in progress, the withdrawal completes.

## 6 Future work

As mentioned in section 4.2, there are likely some interest rate derivatives that could use yToken prices as an oracle. These might be useful for expressing a view on the future path of interest rates, or for hedging duration while holding yTokens or owning a vault.

The physical settlement system described in section 3.3 could likely be adapted to create Bitcoin futures on Ethereum, adapting ideas from Summa

[16] and tBTC [17] (such as using an SPV-proof-based orderbook for a price oracle for proof of undercollateralization, and using SPV proofs to enforce physical settlement).

# 7 Acknowledgments

This paper is heavily indebted to discussions with Hayden Adams, Arthur Breitman, Jill Carlson, Karl Floersch, Alex Herrmann, Ben Jones, Martin Köppelmann, Zubin Koticha, Aparna Krishnan, Hart Lambur, Teo Leibowitz, Robert Leshner, Lev Livnev, Allison Lu, Matt Luongo, James Prestwich, Cyrus Younessi, and Noah Zinsmeister. The authors are particularly grateful to Paradigm for supporting this research.

# References

[1] Robert Leshner and Geoffrey Hayes. *Compound: The Money Market Protocol*. Feb. 2019. URL: https://compound.finance/documents/Compound.Whitepaper.pdf.

[2] MakerDAO. *The Dai Stablecoin System*. URL: https://makerdao.com/en/whitepaper/.

[3] *Synthetix Litepaper v1.1*. Mar. 2019. URL: https://www.synthetix.io/uploads/synthetix_litepaper.pdf.

[4] Hart Lambur. *UMA – Universal Market Access*. Dec. 2018. URL: https://medium.com/uma-project/uma-enabling-universal-market-access-266eb9e5fd90.

[5] Dan Robinson. *Rainbow Network: An Off-Chain Decentralized Synthetics Exchange*. Mar. 2019. URL: http://research.paradigm.xyz/RainbowNetwork.pdf.

[6] Kyle J. Kistner. *Introducing Fulcrum: Tokenized Margin Made Dead Simple*. Mar. 2019. URL: https://medium.com/bzxnetwork/introducing-fulcrum-tokenized-margin-made-dead-simple-e65ccc82393f.

[7] Antonio Juliano. *Introducing the new dYdX*. URL: https://medium.com/dydxderivatives/introducing-the-new-dydx-9675719bacb6.

[8] Nadav Hollander. *Dharma: An Open Protocol for Generic Tokenized Debt Agreements*. URL: https://blog.dharma.io/dharma-an-open-protocol-for-generic-tokenized-debt-agreements-9a4e6a4e6fc0.

[9] MakerDAO. *Vaults*. URL: https://community-development.makerdao.com/makerdao-mcd-faqs/faqs/vault.

[10] Hayden Adams. *Uniswap*. URL: https://uniswap.org/docs/.

[11] Coinbase. *USD Coin (USDC)*. URL: https://www.coinbase.com/usdc.

[12] Noah Zinsmeister Hayden Adams and Dan Robinson. *Uniswap v2 Core*. URL: https://uniswap.org/whitepaper.pdf/.

[13] Fernando Martinelli and Nikolai Mushegian. *A non-custodial portfolio manager, liquidity provider, and price sensor*. URL: https://balancer.finance/whitepaper/.

[14] *Set: A Protocol for Baskets of Tokenized Assets*. URL: https://www.setprotocol.com/pdf/set_protocol_whitepaper.pdf.

[15] *UMA Data Verification Mechanism: Adding Economic Guarantees to Blockchain Oracles*. July 2019. URL: https://github.com/UMAprotocol/whitepaper/blob/master/UMA-DVM-oracle-whitepaper.pdf.

[16] Summa. *Welcome to Summa Auctions*. URL: https://summa.one/auction.

[17] *tBTC: A Decentralized Redeemable BTC-backed ERC-20 Token*. Aug. 2019. URL: http://docs.keep.network/tbtc/index.pdf.

# 8 Disclaimer

This paper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal or tax advice or investment recommendations. This paper reflects current opinions of the authors and is not made on behalf of Paradigm or its affiliates and does not necessarily reflect the opinions of Paradigm, its affiliates or individuals associated with Paradigm. The opinions reflected herein are subject to change without being updated.