Interest Protocol:
Fractional Reserve Banking in Decentralized Finance

Andy Choi* Getty Hill† Eddy Lee‡

May 2022
Preliminary - please do not cite or circulate

Abstract

Interest Protocol is the first fractional reserve banking protocol on the Ethereum blockchain that pays interest to all depositors. Interest Protocol issues a stablecoin, named USDi, that is both over-collateralized and highly scalable. USDi holders automatically earn yield without having to stake, which means Interest Protocol provides yield opportunities to gas-conscious users. Compared to lending protocols without fractional reserves, Interest Protocol can generate more loans from a given amount of capital while incurring less liquidity risk.

Keywords: decentralized finance; stablecoin; lending protocol; fractional reserve banking; gas efficiency

*University of Chicago, Department of Economics. Email: andybchoi@uchicago.edu.
†GFX Labs. Email: getty@gfxlabs.io.
‡GFX Labs. Email: eddy@gfxlabs.io.

1 Introduction

Interest Protocol is a decentralized banking protocol on the Ethereum blockchain that applies fractional reserve banking to decentralized finance. Interest Protocol issues a stablecoin called USDi, which is a liquidity provider (LP) token that represents a one-to-one claim on the protocol's USDC. USDi can be minted by either depositing USDC into the protocol or borrowing USDi from the protocol. Interest Protocol generates revenue from interest paid by borrowers, and this revenue is distributed to all USDi holders. Interest Protocol provides three major benefits to its users:

- *Issue a scalable stablecoin that is pegged to $1.* USDi is pegged to $1 because it can be exchanged with USDC one-to-one, and because it is over-collateralized by a combination of USDC reserves and borrowers' collateral. USDi scales flexibly in response to demand because anyone can mint USDi by either depositing USDC into the protocol or borrowing USDi from the protocol.

- *Create a capital-efficent lending market.* Compared to lending protocols without fractional reserves, Interest Protocol uses the same amount of capital to generate more loans while incurring less liquidity risk. This efficiency gain is shared by all users of the protocol. Borrowers can borrow more at lower rates, and each additional loan creates more USDi holders who earn yield.

- *Provide yield opportunities to gas-conscious users.* All circulating USDi continuously and automatically accrues interest. This allows users to simply hold USDi in their wallet and see their balance grow, without having to pay gas for staking and unstaking.

The rest of the paper proceeds as follows. Section 2 compares Interest Protocol to existing DeFi protocols that lend stablecoins. Section 3 introduces the USDi stablecoin. Section 4 defines the accounting variables used by the protocol, and Section 5 describes how protocol operations affect these accounting variables. Section 6 explains how the protocol makes loans and sets interest rates, and shows that the protocol's lending operation is more capital efficient than lenders that do not implement fractional reserve banking. Section 7 explains the implementation of automatic yield distribution. Section 8 lays out how the protocol manages liquidity and credit risk. Section 9 introduces the protocol's price oracles. Section 10 describes the protocol's strategies to guarantee USDi liquidity. Finally, Section 11 describes protocol governance.

2 Comparison to Existing DeFi Protocols

2.1 MakerDAO

The current iteration of MakerDAO (Maker Team, 2020) issues a stablecoin, DAI, and maintains a Peg Stability Module that consists of fiat-backed stablecoins. There are four major differences between MakerDAO and Interest Protocol. First, Interest Protocol automatically pays interest to all Ethereum addresses that hold USDi. In contrast, DAI holders do not receive interest payments from MakerDAO unless they stake their DAI in the DAI Savings Rate contract. Second, Interest Protocol offers cross-margin vaults, so that users can take out a single loan against a combination of multiple collateral assets. MakerDAO only offers single-collateral vaults. Third, Interest Protocol employs an interest rate function to algorithmically determine interest rates and manage liquidity risk. MakerDAO's interest rates are chosen manually through governance. Fourth, Interest Protocol has a simpler and more efficient liquidation mechanism that can quickly unwind under-collateralized loans, and Interest Protocol does not profit from liquidations.

2.2 Compound and Aave

Compound (Leshner and Hayes, 2019) and Aave (Aave Protocol, 2020a,b, 2022) are multi-asset lending protocols where users can borrow assets against other assets. The major advantage of Interest Protocol over Compound or Aave is the capital efficiency that comes from Interest Protocol's fractional reserve system. Issuing USDi against a reserve of USDC allows Interest Protocol to generate more loans and incur less liquidity risk. Section 6.3 provides a detailed description.

Compound and Aave utilize an aggressive liquidation mechanism that can over-liquidate user accounts. In contrast, Interest Protocol only liquidates the necessary amount of assets to bring the borrower back to a healthy state. Compound also collects a fee from liquidations, unlike Interest Protocol.

2.3 Terra Protocol

Terra Protocol (Kereiakes et al., 2019) issues a stablecoin called TerraUSD (UST). UST is under-collateralized because it is largely backed by LUNA, the staking and governance token of Terra Protocol, and because the value of Terra Protocol mostly comes from issuing UST. In stark contrast, USDi is over-collateralized because it is backed by external assets - USDC in reserves and collateralized loans - whose values do not directly correlate with the performance of Interest Protocol.

Like most other under-collateralized stablecoins, UST's peg is not robust to the market's demand for UST. Once no one wishes to hold UST, UST will lose most of its value because UST is mostly backed by LUNA, and the right to govern a stablecoin protocol is worth little if there is no demand for the stablecoin. On the other hand, USDi retains value regardless of its popularity because USDi can always be used to redeem for USDC or repay collateralized loans.

3 USDi

USDi is a stablecoin pegged to $1 that automatically earns interest. Anyone can mint USDi by depositing an equal amount of USDC into the protocol, and anyone can redeem their USDi for an equal amount of USDC held by the protocol. USDi is also minted when users borrow from the protocol. Borrowers can post collateral and take out loans denominated in USDi. The protocol mints USDi to generate loans and burns USDi when they are repaid.

3.1 Fractional Reserve

New USDi is created only when an equivalent amount of USDC is deposited into the protocol, a borrower takes out an over-collateralized loan from the protocol, or when interest accrues to existing loans. Hence USDi is fully backed by USDC and loans. At the same time, whenever there are outstanding USDi loans, the amount of USDC held by the protocol is smaller than the amount of circulating USDi. This is why we say that Interest Protocol implements *fractional reserves*. Section 6.3 explains how fractional reserve banking makes Interest Protocol's lending capital efficient.

3.2 Yield Without Staking

The interest accrued to USDi loans, net of a protocol fee, is distributed pro rata to all USDi holders in the form of additional USDi. The protocol implements the distribution by scaling up the balance of USDi tokens in all Ethereum addresses so that the value of each USDi remains pegged to $1.[1] This means that USDi is a *liquidity provider (LP) token that is liquid;* USDi earns yield generated from protocol operations, but remains liquid because it is pegged to $1.

As a result, USDi holders are able to *earn compounding yield without staking*. This allows them to use their USDi in various trading activities while still earning yield. For

[1]Section 7 explains the scaling process in detail.

example, those providing ETH-USDi liquidity to a Uniswap pool continue to earn interest on the USDi portion of their liquidity.

USDi also brings gas efficiency; there is no need to switch back and forth between the LP token and the stablecoin because the LP token *is* the stablecoin. This is particularly beneficial to retail users who spend a higher percentage of their trade size on gas fees. For example, because of the gas cost of staking and unstaking, it typically makes sense for a DeFi user to stake 1000 USDC for yield only if they are quite certain that they will not wish to use the money elsewhere for at least a few months. In other words, the gas cost forces the user to choose between liquidity and yield. In contrast, a user with 1000 USDi can simply hold on to it and earn yield until they wish to use it. Gas-sensitive users no longer have to choose between liquidity and yield because USDi provides both.

3.3 Peg Mechanism

When USDi trades above $1, users can arbitrage by depositing USDC into the protocol, minting USDi, and selling USDi in the market. When USDi is priced below $1, users can buy USDi and redeem it for USDC against the protocol reserve.

It is possible that the protocol's reserve of USDC is at times insufficient to immediately meet a large redemption demand. However, as described in Section 6, the interest rate functions dynamically adjust the borrow and deposit rates to respond to the market's demand for liquidity. Whenever the protocol has insufficient reserves, the borrowing and deposit rates become prohibitively high. This incentivizes users to repay their loans or deposit more USDC, both of which increase the protocol reserve. The interest rate functions have been calibrated to facilitate quick redemption even in adverse market conditions.

3.4 Scalability

USDi can easily scale in response to higher demand due to two factors. First, USDi is minted from both depositing USDC and borrowing USDi. When the interest rate is low, those who wish to use USDi can borrow it at a low cost. Under higher rates, users can deposit USDC to mint USDi and benefit from the yield. Second, as explained in Section 6.3, the protocol's USDi lending operations are capital efficient. This means that given the same amount of capital, the protocol can generate more loans and thereby supply more stablecoins to the market compared to traditional lending protocols.

4 Accounting

Interest Protocol's operations are based on the following accounting identity:

$$\text{Reserve} + \text{Loan} + \text{Treasury} = \text{Deposit} + \text{Equity}$$

The variables on the left-hand side represent the protocol's assets, and those on the right-hand side are the protocol's liabilities. Each variable is defined as follows:

- **Reserve** is the pool of USDC that is available for redemption by holders of USDi. Whenever a user deposits USDC to mint USDi, the USDC is added to Reserve. USDC in Reserve can only be withdrawn by burning an equivalent amount of USDi.
- **Loan** is the balance of all outstanding USDi loans that have been generated by the protocol.
- **Treasury** is the pool of assets controlled directly by the protocol. It will mostly consist of USDi but can hold any other asset. The protocol's revenue from fees accrues to the Treasury. The Treasury is used to fund community-approved protocol operations, such as audits or grants.
- **Deposit** is the total supply of USDi.
- **Equity** equals the total value of the protocol's assets minus Deposit. Equity grows when the protocol earns fees from lending operations.

Since USDi is minted if and only if USDC is deposited into Reserve or a borrower takes out a loan, Deposit always equals the sum of Reserve and Loan. Section 5 provides a detailed description of how the accounting variables change from various protocol operations.

5 Protocol Operations and Accounting Variables

If we denote Treasury balance and Equity balance by T and E, the accounting identity in Section 4 can be written as

$$R + L + T = D + E.$$

Table 1 describes how the five accounting variables change as a result of common protocol operations.

Operation	Changes	Description
User borrows x USDi	$L \leftarrow L + x$ $D \leftarrow D + x$	Mint x USDi for user
User repays x USDi	$L \leftarrow L - x$ $D \leftarrow D - x$	Burn x USDi from user
User deposits x USDC	$D \leftarrow D + x$ $R \leftarrow R + x$	Mint x USDi for user
User withdraws x USDC	$D \leftarrow D - x$ $R \leftarrow R - x$	Burn x USDi for user
Protocol pays x USDi in grant	$E \leftarrow E - x$ $T \leftarrow T - x$	Transfer x USDi from T
Interest accrues for t blocks at rate r_{block}	See Section 7.2 for details	

Table 1: How Protocol Operations Change Accounting Variables

All user deposits of USDC go to Reserve and remain there until they are withdrawn. USDC in Reserve can be withdrawn only by burning an equal amount of USDi tokens, and there are no exceptions, including for the protocol itself. The protocol is only able to spend USDi or USDC that it holds in the Treasury.

6 Lending USDi

Interest Protocol generates revenue from lending USDi. This section describes how the loans are made and interest rates are determined. Table 2 provides a list of variables that we introduce in this section.

Loans are always denominated in USDi, have a variable interest rate, and never mature. Any Ethereum address can borrow USDi from the protocol against registered collateral assets. The protocol maintains a whitelist of assets that can be used as collateral, and each collateral asset has a maximum loan-to-value ratio (LTV). To borrow USDi, a user creates a vault owned by his address and deposits collateral assets from their address into the vault. They can then borrow USDi up to their `borrowing_power`, which is calculated by multiplying the market value of each collateral asset by its maximum LTV, and summing across all assets. Formally, let collateral assets be indexed by $i \in I$, let $\bar{v}_i$ denote the LTV for asset i, and let p_i be the price of asset i. Suppose a borrower has

Notation	Variable in Smart Contract	Description
$\bar{v}_i$	`_tokenId_tokenLTV[i]`	LTV of asset i
$\bar{l}$	`borrowing_power`	Maximum amount a vault can borrow
R	USDC balance of `USDi` contract	Amount of USDC in Reserve
D	total supply of `USDi` contract	Total supply of USDi
s	`reserve_ratio`	R/D
r	`int_curve_val`	Borrow APR
L	`_totalBaseLiability` × `_interestFactor`	Balance of all outstanding loans
f	`_protocolFee`	Protocol fee rate
l	`usdi_liability`	Balance of a wallet's outstanding loan
k_i	`_tokenAddress_liquidationIncentive[i]`	Liquidation penalty of asset i

Table 2: Lending Variables

posted q_i of asset i as collateral for each $i \in I$. Then his borrowing power is

$$\bar{l} := \sum_{i \in I} p_i q_i \bar{v}_i.$$

6.1 Interest Rate

All USDi borrowers pay interest according to a single borrowing interest rate that varies over time. The interest paid by borrowers, net of a protocol fee, is distributed pro rata to all USDi depositors. Anyone who holds USDi is a depositor, regardless of how they obtained USDi - by depositing USDC, borrowing USDi, or trading in the market.

6.1.1 Borrow Rate

Let R denote the balance of USDC held in Reserve, and D the balance of Deposit, which equals the amount of all circulating USDi. We first define the *reserve ratio*, denoted s, by

$$s = \frac{R}{D} \in [0, 1].$$

The reserve ratio measures how much USDC the protocol holds to meet immediate redemption demand, relative to the total supply of USDi. If the reserve ratio is 10%, for every 10 USDi that exists in the market, there is 1 USDC that can be immediately redeemed. The higher the reserve ratio, the more likely that the protocol can immediately honor unexpected redemption demands.

Since making more loans from a given amount of Reserve lowers the reserve ratio, the protocol wishes to encourage borrowing when the reserve ratio is high and discourage borrowing when the ratio is low. Thus the interest rate for borrowers is decreasing in

the reserve ratio. Specifically, the borrow annual percentage rate (APR), denoted r_B, is a continuous, decreasing, and piecewise-linear function of the reserve ratio s, given by

$$r_B(s) = \begin{cases} r_3 + \frac{r_2 - r_3}{s_2} s & \text{if } 0 \leq s \leq s_2 \\ r_2 + \frac{r_1 - r_2}{s_1 - s_2}(s - s_2) & \text{if } s_2 < s \leq s_1 \\ r_1 & \text{if } s_1 \leq s \leq 1, \end{cases}$$

where r_3 is the upper bound on the rate, r_1 is the lower bound on the rate, $s_1, s_2 \in (0, 1)$ are the thresholds at which the kinks occur, and r_2 is the rate when $s = s_2$. Figure 1 plots $r_B(s)$ for the initial parameter values that will be used at launch: $r_1 = 0.005$, $r_2 = 0.1$, $r_3 = 2$, $s_1 = 0.6$, and $s_2 = 0.4$.

The upper bound r_3 is intended to be high enough that it is prohibitively expensive for borrowers to borrow at rates close enough to r_3. As long as the borrow rate is strictly below r_3, the reserve ratio s remains strictly positive, and the protocol has USDC in Reserve to meet additional redemption demand. We expect the protocol to operate between the two kinks, (s_1, r_1) and (s_2, r_2). That is, we expect the interest rate to stay between r_1 and r_2, and the reserve ratio between s_1 and s_2.

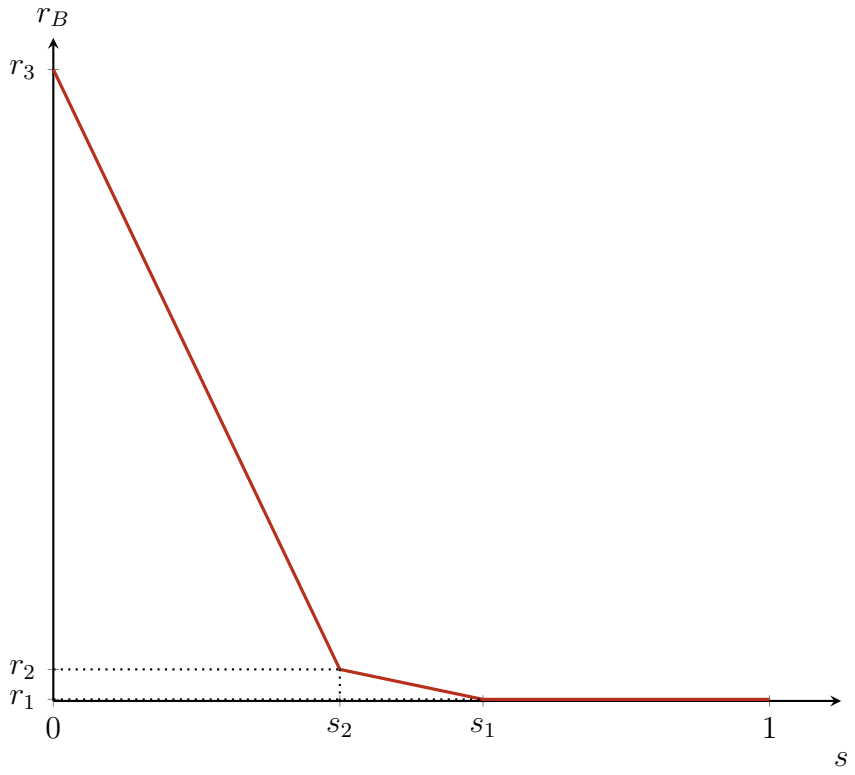

Figure 1: Reserve ratio (s) and Borrow APR (r_B)

6.1.2 Deposit Rate

Suppose that the current balance of outstanding loans is L, and that the current reserve ratio is s. Then the borrow APR is $r_B(s)$, and the annualized interest accrued to all outstanding loans is given by $r_B(s)L$. A percentage of this interest is withheld by the protocol as fees, and the rest is distributed pro rata to all depositors. Therefore, if the current circulating supply of USDi is D, and the protocol fee rate is $f \in [0, 1]$, the current deposit APR, denoted r_D, is given as a function of s by

$$r_D(s) = \frac{r_B(s)L(1-f)}{D}.$$

The total supply of USDi equals the balance of all outstanding loans plus Reserve, that is, $D = R + L$. The deposit APR can thus be rewritten as follows:

$$r_D(s) = r_B(s)(1-s)(1-f).$$

The functions $r_B(s)$ and $r_D(s)$ are called interest rate functions. Like the borrow APR, the deposit APR is decreasing in the reserve ratio. When the reserve ratio is low, the protocol encourages more deposits by setting a higher deposit APR. Figure 2 plots $r_D(s)$ along with $r_B(s)$ for $f = 0.15$ and otherwise using the same initial parameter values as in Figure 1.

6.2 Liquidation

Recall that a user can borrow USDi up to his debt limit $\bar{l}$. A borrower is *under-collateralized* if the balance of his outstanding loan, denoted l, is greater than his debt limit: $l > \bar{l}$. This can occur when l increases because of accrued interest or when $\bar{l}$ decreases due to a fall in the value of collateral assets.

Once a borrower becomes under-collateralized, anyone can liquidate the borrower's collateral by calling `liquidateAccount()`. An entity performing the liquidation is called a liquidator. A liquidator purchases the borrower's collateral from the protocol by paying USDi to the protocol. If a collateral asset i has an oracle price of p_i, a liquidator purchases the asset at the price of $(1 - k_i)p_i$, where k_i is the liquidation penalty for asset i. The liquidation penalty serves as an incentive for liquidators to purchase the asset and an incentive for borrowers to keep their loan balance below debt limit. The liquidator can liquidate up to the minimum amount that makes the borrower no longer under-collateralized.

Formally, suppose a borrower has posted q_i of collateral asset i for each $i \in I$, and

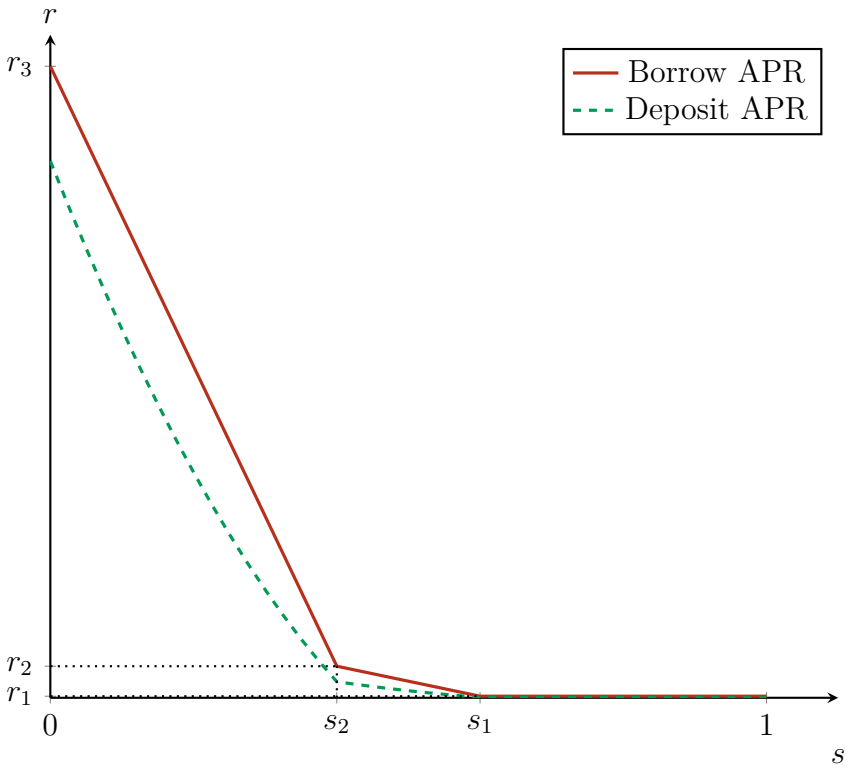

Figure 2: Reserve ratio (s), Borrow APR (r_B), and Deposit APR(r_D)

that the price of asset i is p_i. If $l > \bar{l} = \sum_{i \in I} p_i q_i \bar{v}_i$, a liquidator who wishes to liquidate this borrower can purchase from the protocol an asset y_i at price $(1 - k_i)p_i$, subject to the following constraints:

1. The liquidator can only purchase up to the amount of collateral posted by the borrower: $y_i \in [0, q_i]$.

2. The liquidator can only purchase up to the amount necessary to make the borrower over-collateralized:

$$y_i \leq \bar{y} := \frac{l - \bar{l}}{p_i(1 - k_i - \bar{v}_i)}.$$

The upper bound $\bar{y}$ in the second constraint is the value of the minimization problem:

$$\begin{aligned} \bar{y} &= \min y \\ \text{subject to} \quad & \bar{l} - p_i y \bar{v}_i \geq l - p_i y (1 - k_i). \end{aligned}$$

The liquidator should of course only be able to purchase up to the amount of the borrower's outstanding loan: $(1 - k_i)p_i y_i \leq l$. However, it is easy to see that this is

implied by the two constraints above. In words, the liquidator is allowed to liquidate up to the minimum amount of asset i such that, if liquidated, would bring the borrower's outstanding liability, l, below $\bar{l}$.

As a result of the liquidation, the protocol seizes the borrower's collateral and transfers the collateral to the liquidator. The protocol reduces the borrower's loan balance by the amount of USDi paid by the liquidator. That is, the borrower now has $q_i - y_i$ of asset i posted as collateral, and his outstanding loan balance is $l - (1 - k_i)p_i y_i$.

It is worth emphasizing that the borrower can never be liquidated more than is necessary to make the borrower over-collateralized. At the same time, only one liquidation is always sufficient to liquidate a given collateral asset of a vault, regardless of how under-collateralized the vault is. These are advantages over other lending protocols with arbitrary caps on liquidation. For example, a borrower at Compound (Leshner and Hayes, 2019) can be liquidated up to 50% of a collateral asset for being marginally under-collateralized. On the other hand, when more than 50% of a collateral needs to be sold to make the borrower over-collateralized, multiple liquidations are required. Interest Protocol customizes the liquidation amount to protect borrowers from excessive liquidation while allowing liquidators to efficiently offload risk for the protocol.

6.3 Capital Efficiency

Many existing lending protocols directly lend out the stablecoins that are supplied to them, such as USDC or USDT. Interest Protocol implements fractional reserve banking so that the USDC deposited into the protocol is kept in Reserve, and loans are only made in USDi. This innovation allows Interest Protocol to be capital efficient. Specifically, we make the following claim:

Claim 1. *Compared to Compound or Aave, Interest Protocol can generate more loans from a given amount of USDC while incurring less liquidity risk.*

To prove this, we will define two concepts, utilization rate and liquidity risk, and compare how the former affects the latter in each protocol. The *utilization rate* of a lending protocol is defined as the amount of loans generated per each unit of stablecoin supplied to the protocol.[2] If R USDC has been supplied to each of Aave, Compound, and Interest Protocol, and each protocol's total outstanding loan is worth L USD, then the utilization rate at each protocol is $\frac{L}{R}$.

We measure the *liquidity risk* of a lending protocol as the percentage of deposits that are not immediately redeemable for the underlying asset. Suppose once more that each

[2]In Compound and Aave, the interest rate curve determines the borrow rate as a function of the utilization rate.

protocol has been supplied with R USDC and lent out L dollars' worth of loans, where $L < R$. Both Compound and Aave have a deposit of R.[3] Among this R, L has been lent out, and only $R - L$ is still held by the protocol in the form of USDC. Hence, according to our definition, the liquidity risks of Aave and Compound both equal $\frac{L}{R}$. However, Interest Protocol lends out USDi rather than USDC and thereby *creates new deposits* with each loan. Thus Interest Protocol has $R + L$ in deposits, R of which is redeemable for USDC and L of which is not. This implies that Interest Protocol's liquidity risk equals $\frac{L}{R+L}$.

Figure 3 plots the relationship between utilization rate and liquidity risk for Compound, Aave, and Interest Protocol.

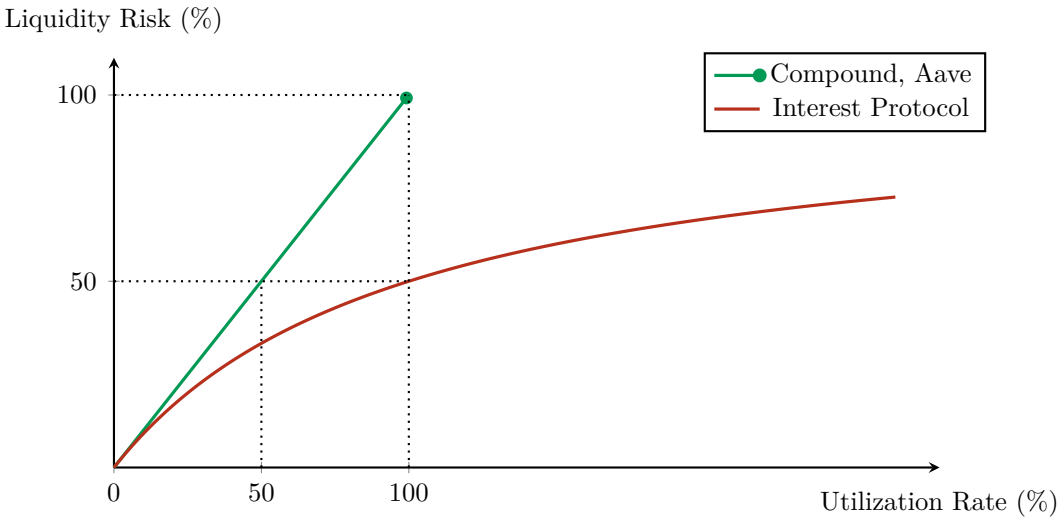

Figure 3: Utilization Rate and Liquidity Risk

We can see from the figure that for any given level of liquidity risk, Interest Protocol is able to achieve a higher utilization rate. For example, to maintain a liquidity risk of 50% or lower, Compound and Aave must keep their utilization rate below 50%. However, Interest Protocol can utilize up to 100% of its deposits while keeping liquidity risk below 50%. In fact, while Compound and Aave's utilizaton rates are capped at 100%, Interest Protocol can achieve an arbitrarily high utilization rate. This is because as utilization rate grows, Interest Protocol's liquidity risk approaches but is always strictly below 100%.

Algebraically, suppose each protocol aims to keep liquidity risk below a fixed level, $a \in [0, 1]$. Compound or Aave must maintain $\frac{L}{R} \leq a$, which means the maximum amount of loans they can generate is aR. For Interest Protocol, the constraint is $\frac{L}{R+L} \leq a$, which is equivalent to $L \leq \frac{a}{1-a}R$. Therefore, compared to Compound or Aave, Interest Protocol can lend $\frac{1}{1-a}$ times more from a given amount of USDC while maintaining the same level

[3] Aave's deposit would be represented by R aUSDC, while Compound's deposit would be represented by a corresponding amount of cUSDC scaled down by the cUSDC-USDC exchange rate.

of liquidity risk. Finally, since liquidity risk varies continuously with utilization rate, whenever Compound or Aave has a non-zero utilization rate, Interest Protocol can use the same amount of USDC to generate a strictly larger amount of loans while taking strictly less liquidity risk. This proves Claim 1.

Compound, Aave, and Interest Protocol do not impose a fixed upper bound on liquidity risk. Instead, all three protocols set their interest rate functions so that the liquidity risk remains low enough at practical levels of interest rates. It is therefore useful to compare each protocol's borrow rate as a function of the utilization rate. For Interest Protocol, a utilization rate of u implies a reserve ratio of $\frac{1}{1+u}$, and thus a borrow APR of $r_B(\frac{1}{1+u})$. At Compound and Aave, the interest rate curves directly pins down the borrow rate at each utilization rate. Figure 4 plots these relationships. Interest Protocol's graph is drawn with the parameters to be used at launch as described in Section 6.1.1, and Compound's and Aave's are based on the parameters of their USDC markets at the time of writing. It can be seen that *for any given utilization rate, Interest Protocol offers a lower borrow rate than Compound or Aave.*

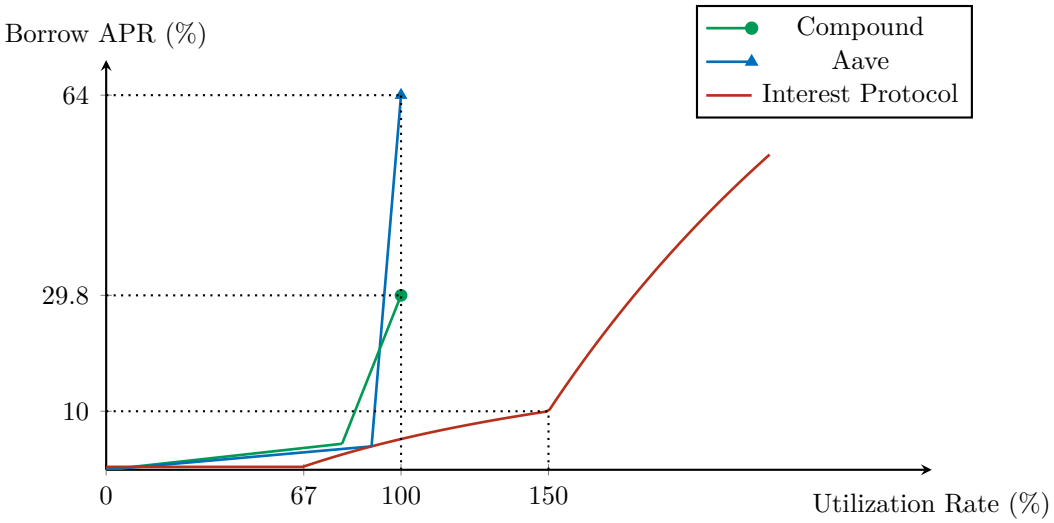

Figure 4: Utilization Rate and Borrow APR

Figure 5 combines Figures 3 and 4 for Compound and Interest Protocol and allows us to perform an interesting comparison exercise. Suppose a common borrow APR of 3% is given by market. At this rate, the utilization rate is 54.0% for Compound and 82.7% for Interest Protocol. Thus Interest Protocol can lend out 53% more than Compound using the same amount of USDC. Also, at these utilization rates, the liquidity risk is 54.0% for Compound and 45.3% for Interest Protocol, which means Interest Protocol is taking 16% less risk. In this scenario, Interest Protocol would be able to use the same amount of USDC to extend a larger amount of loans while incurring less liquidity risk.

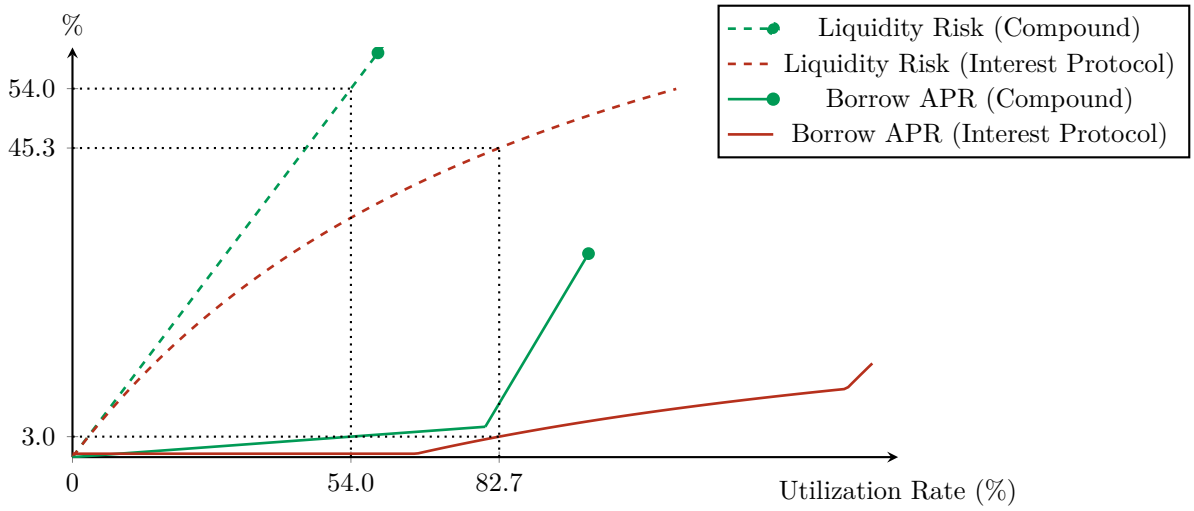

Figure 5: Utilization Rate, Borrow APR, and Liquidity Risk

Capital efficiency of Interest Protocol's lending mechanism implies that USDi is highly scalable. Because Interest Protocol can supply a large amount of loans while minimizing liquidity risk, USDi can scale flexibly in response to demand for leverage while maintaining a robust peg to the dollar.

6.4 Voting with Collateral

Active governance participants in decentralized finance often face a dilemma - if they want to borrow against governance tokens that they hold, they can no longer vote with them. When depositing governance tokens as collateral on their Interest Protocol vault, borrowers have the option to delegate the voting power of their tokens to an address of their choice. This means users can borrow against their governance tokens while retaining voting power.

7 Implementing Automatic Yield

Interest Protocol automatically and continuously distributes yield to all USDi holders. To implement this, the protocol builds upon the contract `UFragments.sol`, which was developed by Ampleforth (Kuo et al., 2019) and allows the balance of an ERC20 token in every address to scale automatically in proportion to a global index called `gonsPerFragment`. Interest Protocol adds minting and burning functionalities so that users can mint USDi by depositing USDC or taking out a loan.

Intuitively, fragments represent the USDi balance of an address, and gons represent the balance of the "internal token" that is actually used for transactions. The ratio between

the two variables is the global index `gonsPerFragment`, denoted by g. When yield is distributed to USDi holders, the balance of gons in each address remains unchanged, but `gonsPerFragment` decreases so that each gon corresponds to more fragments. This allows the protocol to distribute yield in a gas-efficient way by updating a single global index.

7.1 Using `gonsPerFragment` to Scale USDi Balances

At time t, the USDi balance of an address is given by

$$\frac{\texttt{gonBalance}}{\texttt{gonsPerFragment}},$$

where `gonBalance` is the amount of gons in the address.

For example, suppose that an address with 0 USDi balance receives 100 USDi at block t. If `gonsPerFragment` at t is $g_t = 0.5$, this means that the address has received $100 \times 0.5 = 50$ gons. If `gonsPerFragment` decreases to $g_{t'} = 0.25$ at block $t' > t$, then the USDi balance of the address at t' is scaled up to $50/0.25 = 200$. This represents the fact that all USDi holders have gained a yield of $(0.5 - 0.25)/0.25 = 100\%$ from block t to block t'.

7.2 Updating `gonsPerFragment` by calling `pay_interest()`

Any transaction that changes the Reserve or Deposit (and therefore the reserve ratio) updates `gonsPerFragment` and `interestFactor`. Suppose that we are at time t, and the last time before t in which Reserve or Deposit changed was t_0. This means that the borrowing rate, which is determined by the reserve ratio, must have remained constant between t_0 and t. Let r denote this constant borrow rate. Let R_τ, L_τ, T_τ, D_τ, g_τ, and h_τ denote the values of Reserve, Loan, Treasury, Deposit, `gonsPerFragment`, and `interestFactor` at time τ, respectively. Recall that f denotes the protocol fee. Now, suppose that a protocol operation that changes either Reserve or Deposit occurs at time t. This affects the variables in two steps:

1. Call `pay_interest()`, which makes the following updates:

$$
\begin{aligned}
h_t &= h_{t_0}(1 + (t - t_0)r) \\
L_t &= \text{totalBaseLiability} \times h_t \\
g_t &= g_{t_0} \frac{D_{t_0}}{D_{t_0} + (1 - f)(L_t - L_{t_0})} \\
T_t &= T_{t_0} + f(L_t - L_{t_0}) \\
D_t &= D_{t_0} + (L_t - L_{t_0}).
\end{aligned}
$$

2. Run the intended operation. This will once again update R_t or D_t (and hence update the borrow rate).

Any function that changes the reserve ratio is first required to update `interest factor`, Loan, `gonsPerFragment`, Treasury, and Deposit by adding the interest that has accrued since the last time at which such a function was called. Then, the actual contents of the function are run.

8 Risk Management

To operate safely in adverse market environments, Interest Protocol manages two major financial risks - liquidity risk and credit risk. This section defines these concepts in the context of Interest Protocol and describes how the protocol manages the risks.

8.1 Liquidity Risk

Liquidity is defined as the protocol's ability to meet redemption demand at all times. The protocol has an obligation to USDi holders to redeem 1 USDi for 1 USDC on demand. On the other hand, the protocol allows borrowers to maintain a outstanding loan balances as long as they wish to, provided that they maintain enough collateral to stay over-collateralized. This causes a maturity mismatch, and it is possible at times for the protocol to face redemption demand in excess of the USDC held in Reserve. The protocol uses two mechanisms to minimize this possibility.

First, Interest Protocol aims to maintain a sufficiently high reserve ratio at all times by using a variable interest rate. As explained in Sections 3.3 and 6.1.1, whenever the reserve ratio is low, a high interest rate incentivizes users to replenish the reserve by repaying their debt or depositing USDC. This ensures that any shortage in reserve is short-lived.

Second, as described in Section 10, the protocol aims to guarantee liquidity in various USDi markets. As long as USDi can be swapped for other assets with little slippage, users can hold USDi to earn interest until they wish to purchase other assets, at which point they can buy the assets by paying with USDi. Thus liquidity in USDi markets lowers the demand to redeem USDi for USDC.

8.2 Credit Risk

Credit risk is the risk of borrowers defaulting on their loans. To manage credit risk, Interest Protocol only makes over-collateralized loans and liquidates under-collateralized vaults. As described in Section 6.2, the protocol employs an efficient liquidation mechanism that quickly and accurately unwinds under-collateralized vaults. This means that, compared to lending protocols with less efficient liquidation mechanisms, Interest Protocol is able to extend loans at favorable terms (such as with higher LTVs) while incurring less credit risk.

Timely liquidation also depends on characteristics of collateral assets, such as volatility, DEX liquidity, and smart contract or governance risk. We expect that managing the risk of collateral assets will be one of the main tasks of protocol governance, as explained in Section 11.

9 Oracles

Price oracles play a crucial role in every protocol that lends assets against collateral. An accurate, reliable, and safe oracle for asset prices allows loans and liquidations to occur quickly without errors. Interest Protocol uses a read-based oracle system that calls for live prices at each instance a price is needed. While this method could be inefficient at times, it provides the highest level of accuracy and safety for borrowers and lenders.

All asset prices are aggregated into the Master Oracle, which maintains a list of asset-specific oracle contracts, called Anchor View Relays. Each asset's Anchor View Relay has two sub-oracles and a configurable deviation. One sub-oracle is specified as the main oracle and the other as the anchor. Whenever a price is utilized in a transaction, the protocol reads prices from the main oracle and the anchor oracle and certifies that the main oracle price is within the specified deviation of the anchor oracle price.

If the main oracle price falls outside the deviation, the price will not update, and the underlying transaction will fail. This acts as an immediate safety feature and prevents usage of the protocol until governance can react or the market rectifies itself. Additionally, governance can pause an oracle to effectively pause all operations that require the price

of the asset.

This system allows Interest Protocol to use Chainlink oracles as the main oracle for markets without it becoming a single point of failure. The anchor oracle for the initial markets will be Uniswap V3, but governance can utilize other sources as necessary.

10 USDi Liquidity

In this section, liquidity means the ease with which USDi can be converted into other assets, such as ETH or BTC, without having a large impact on the market price.[4] USDi liquidity allows USDi holders to purchase other assets without having to redeem USDi for USDC in Reserve. This saves gas costs and helps to increase the usage of USDi, expanding Interest Protocol's balance sheet.

To encourage market makers to provide USDi liquidity, the protocol operates a standard liquidity mining program that incentivizes ETH-USDi liquidity providers with IPT tokens (introduced in Section 11).

11 Governance

While many operations of Interest Protocol happen automatically on-chain, the protocol relies on governance actions for any maintenance or upgrades. For example, the protocol automatically sets the interest rate as a function of the reserve ratio, but this function itself may need to be updated from time to time in response to secular market trends. Similarly, while users can permissionlessly borrow USDi against registered collateral assets, the protocol may wish to register a new asset as collateral.

Interest Protocol's governance is controlled by a decentralized autonomous organization, which we call Interest Protocol DAO (IP DAO). IP DAO manages the smart contracts associated with the protocol by updating parameters or upgrading contracts.

Interest Protocol Token (IPT) is the governance token of the protocol. When the DAO votes on a proposal, one IPT represents one vote. IPT will be distributed to the community through a public sale, liquidity mining programs, and other methods.

11.1 Responsibilities of Governance

Interest Protocol governance has three main responsibilities:

[4]Note that this definition of liquidity is related, but distinct from, the concept of the protocol's liquidity as the ability to meet USDC redemption demand.

1. Governance is responsible for managing control of the protocol and preventing centralization of power and information asymmetry. A protocol cannot be safe for use without a strong foundation, and the base of that foundation is the community. The community needs to share control of the protocol, so that no one entity can control the protocol, but the protocol can still be controlled.
2. Governance is responsible for the maintenance of the existing protocol. This includes adjusting parameters such as LTVs and the interest rate function.
3. Governance is responsible for the future of the protocol. Decentralized finance has a rapidly evolving landscape, and good governance will guide the protocol to adapt to the change, remain competitive, and grow further.

11.2 Actions Controlled by Governance

IP DAO can make the following parameter changes:

- register new collateral assets
- update an asset's LTV
- update an asset's liquidation incentive
- update an asset's oracle
- change the USDi interest rate function
- set the protocol fee
- manage the protocol treasury.

The following contracts are upgradable by the DAO:

- USDi.sol
- VaultController.sol.

11.3 Governance Process

Interest Protocol introduces Governor Charlie, which extends Compound's popular governance contract, Governor Bravo. Charlie maintains the core system of Bravo, in which proposals are reviewed, voted on, queued on timelock, and then executed. As in Bravo, votes can be cast with or without a message, and votes can be cast via signature. Proposals can be authored by addresses that meet the proposal threshold or have been added to the proposer whitelist controlled by governance.

Unlike Bravo, Charlie has an emergency governance proposal process that bypasses review and has a separate configurable quorum, voting period, and timelock period. There are times when protocols need to quickly form consensus and rapidly respond to critical events. The emergency proposal process offers an additional tool for governance to manage the protocol in such times of need. In addition, Charlie merges the timelock and Bravo contracts into one to simplify the system.

The following governance parameters are managed by the DAO:

- proposal threshold
- proposal review period
- proposal voting period
- proposal quorum threshold
- proposal timelock period
- emergency proposal voting period
- emergency proposal quorum threshold
- emergency proposal timelock period
- contract delegate.

11.4 Emergency Pause

To protect the protocol and its users against unforeseen risks, Governance is given the ability to designate an address as a Pauser. The Pauser can disable and re-enable certain protocol functions, such as deposits, withdraws, liquidations, borrowing, and repaying.

Initially, the Pauser will be a multisig controlled by GFX Labs. However, that multisig, and any thereafter, can only pause the protocol in one of the following circumstances:

- The protocol has been exploited, and funds belonging to the protocol or its users have been lost.
- A vulnerability has been found such that, if exploited, would lead to a loss of protocol or user funds.
- Any other critical scenario comparable to the above two cases, in which pausing the protocol is in the best interest of the protocol and its users.

Governance can change the Pauser address through a governance proposal. GFX Labs intends to transfer the Pauser role to the community within three months after launch.

11.5 Programs

In order to grow, Interest Protocol needs contributors with a wide range of expertise. Interest Protocol governance uses an off-chain framework called *programs* to organize the efforts of various contributors and to allocate resources toward them. Programs are designed by governance in conjunction with the leader of the program to meet the needs of the protocol, and are funded through IPT or the protocol's USDi holdings.

Programs will not be lead by GFX Labs and instead leaders are only empowered through a governance proposal. The program leader is held responsible for achieving the objectives set by governance, but is given the autonomy in running their program to achieve the objective.

11.5.1 Recognized Delegates

If approved by the DAO, the Recognized Delegate program could be the first program. The goal of the Recognized Delegate program is to decentralize protocol governance and encourage active maintenance and innovation of the protocol by attracting a wide range of participants to the protocol.

Any IPT delegate that maintains a minimum threshold of voting activity, forum participation, and communication can become a Recognized Delegate. At launch, Recognized Delegates must meet the following criteria:

- Unambiguously and publicly post their intention to become a Recognized Delegate in the appropriate section of the Interest Protocol forum.
- Provide a delegate platform in the Interest Protocol forum.
- Provide an address for vote delegation.
- Disclose any conflict of interest and their policy when such conflict is relevant to a vote.
- Participate in at least 90% of all votes that have occurred in the last 90 days (voting Abstain counts towards participation).
- Communicate in the Interest Protocol forum how and why they voted within 7 days of the end of the vote. If a delegate has not voted, they should communicate why. At least 90% of all votes should be followed by this timely communication.
- Attend at least one protocol development call a month.

Recognized Delegates will receive an annualized payment of IPT according to Figure 6 below. The payment to a delegate is a function of the number of votes delegated to them,

and both the payment and the vote requirement are specified in terms of basis points of the total supply of IPT. Recognized Delegates with more votes will earn more, but the maximum compensation is capped at 10 bps of total supply. A Recognized Delegate must be delegated at least 1 bp of the total supply to receive compensation.

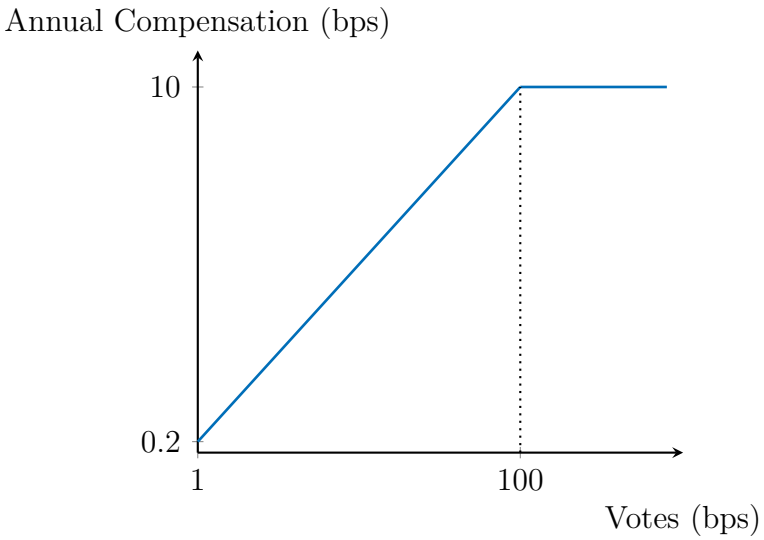

Figure 6: Delegate compensation as a function of votes

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 GFX Labs or their affiliates and does not necessarily reflect the opinions of GFX Labs, their affiliates, or individuals associated with them. The opinions reflected herein are subject to change without being updated.

References

Maker Team. The maker protocol: Makerdao's multi-collateral dai (mcd) system. 2020. URL https://makerdao.com/en/whitepaper/.

Robert Leshner and Geoffrey Hayes. Compound: The money market protocol. 2019. URL https://compound.finance/documents/Compound.Whitepaper.pdf.

Aave Protocol. Aave protocol whitepaper v1.0. 2020a. URL https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf.

Aave Protocol. Aave protocol whitepaper v2.0. 2020b. URL https://github.com/aave/protocol-v2/blob/master/aave-v2-whitepaper.pdf.

Aave Protocol. Aave v3 technical paper. 2022. URL https://github.com/aave/aave-v3-core/blob/master/techpaper/Aave_V3_Technical_Paper.pdf.

Evan Kereiakes, Do Kwon, Marco Di Maggio, and Nicholas Platias. Terra money: Stability and adoption. 2019.

Evan Kuo, Brandon Iles, and Manny Rincon Cruz. Ampleforth: A new synthetic commodity. 2019. URL https://www.ampleforth.org/papers/.