FLOAT CAPITAL (Alpha Whitepaper)

Perpetual synthetic asset protocol

Jonjon Clark
Mphil. Data Science, BSc (Hons) Computer Science

Jason Smythe
BSc (Hons) Computer Science and Mathematics

Denham Preen
BSc (Hons) Computer Science

Paul Freund
BSc (Hons) Computer Science, Hons Philosophy

Guy Paterson-Jones
MSc Mathematics

WooSung Dong
Hons. Actuarial Science

Michael Young
BEng (Hons) Computer and Electrical Engineering

Jono VDM
Hons. Finance and Investments. Passed CFA III exam.

Stent
Physics and Mathematics

FLOAT CAPITAL

ABSTRACT

Float is a synthetic asset protocol providing decentralized, trustless, and efficient exposure to arbitrary asset classes. The Float alpha smart contracts are incentives based perpetual smart contracts that algorithmically react to market demand for long and short positions on a synthetic asset. Market demand triggers dynamic adjustments of additional protocol rewards (alphaFLT and raw yield) rewarded to users taking valuable market positions. Synthetic asset exposure to the underlying asset class may dynamically fluctuate based on imbalanced market demand for various positions.

The fundamental design of the protocol is highly scalable and abstracts all complexity from the end user allowing fast entry and exit with no minimum trade sizes. Synthetic asset positions can be minted without over-collateralization and without the threat of being liquidated.

```solidity
pragma solidity 0.8.9;
import "@float-capital/contracts/Degen.sol";
import "@float-capital/contracts/Chad.sol";
import "@float-capital/contracts/Ape.sol";

contract FloatCapital is Degen, Chad, Ape {
    // Your move ...
}
```

Contents

1 INTRODUCTION

Float Capital allows users to create **long or short tokenized positions referencing any arbitrary asset**.

To expand further:

- vanilla or leveraged
- long or short
- perpetual
- tokenized exposure (ERC20)
- to arbitrary asset classes (ETH, BTC, gas, NFTs, commodities ...)
- without liquidations

This DeFi lego is extremely useful for many different reasons to many different users.

1.1 Alpha release

Creating a highly complex and innovative financial protocol is difficult. It presents unique engineering, economic and financial challenges. The core philosophy of Float is rapid and iterative development allowing us to improve and scale at the fastest possible rate while providing safety. To meet the above goal an alpha version has been deployed affording the following advantages:

- Early user feedback.
- Test and validate incentive models.
- Quicker release cycle.
- Reduce legacy.
- Improve infrastructure.

The alpha version of Float Capital is deployed live on Polygon with the intention of releasing a new protocol version 4-6 months after the alpha launch based on the insights it affords. To use the live alpha (real money), visit Float Capital and mint a synthetic asset in under a minute. Please note while the protocol is audited, many risks exist.

1.2 Whitepaper structure

This paper will revolve around explaining the design of the **alpha system as depicted in Figure 1**. Sections will continually reference Figure 1 throughout the paper.

2 SYNTHETIC ASSETS

Please refer to label 1 in Figure 1 for this section.

2.1 Summary

A user begins by speculating on a certain asset class (long or short). The user provides DAI[1], a stable coin pegged to the dollar, and in return receives a synthetic asset representing their position. This receipt is in the form of an ERC20 token. The price of this ERC20 token is such that it tracks the underlying asset class. The user is able to do 3 things with their synthetic asset token.

- Sell the synth on the open market.

[1]The contracts are engineered such that any underlying asset can be used instead of DAI.

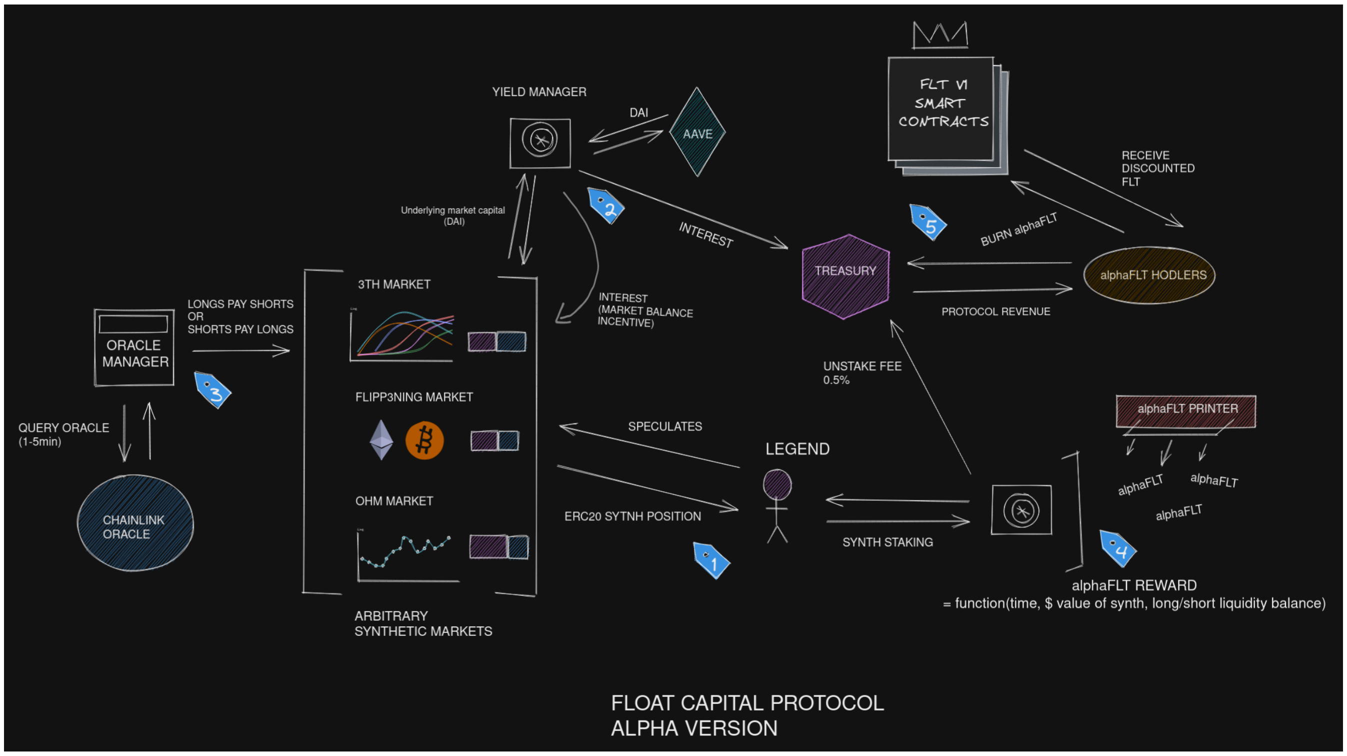

Figure 1: Float Capital Alpha System

- Stake the synth to earn alphaFLT.
- Burn the synth to redeem DAI equal to the value of the synth burned.

2.2 Motivation

Usually synthetic assets are based upon traditional financial instruments, allowing people to gain exposure to $AMZN or the $SP500, for example, on the blockchain. Yet, this is just one of the more common use cases of synthetic assets. Synthetic assets can be created from any quantifiable data feed allowing users to gain exposure to a far larger universe of assets.

In a traditional environment, it would allow users to go long or short on say, for example, *the average amount of rainfall in Bordeaux, France*, this would allow wine farmers to hedge against poor crop yields in a given season by going short while allowing guesthouses in the region to hedge against rainy weather and a lack in tourism by going long.

In a much more practical and relevant DeFi scenario, it gives users the power to perform powerful DeFi strategies through synthetic assets that previously have not existed. For example, an OHM synth can allow users to hedge and earn high yields on staking while maintaining a delta neutral position.

2.3 Mechanic

All long and short synthetic assets are ERC20 tokenized positions allowing composibiliy across the DeFi ecosystem. The tokens are engineered such that a minted position settles at the next price update received by the smart contract. This system is known as Next Price Execution and thwarts attempts at front running to extract value from the system. Custom hooks in the ERC20 tokens allow the tokens to be minted in a single transaction step and available for immediate use upon settlement (transfer, stake, burn) without any additional user interaction. This hook based lazy allocation method allows a superior user experience where the complexity of the front running prevention is entirely abstracted from the user.

All synthetic mints and redeems are batched into a single efficient order that is executed upon the next oracle price update. Coupled together with hook execution, this allows arbitrary scaling of entry and exits into the system, making the system suitable for extremely large trade volumes. Oracle price updates from Chainlink occur as fast as every 27 seconds on the Polygon network, allowing extreme speed for entry and exit to the system (sub 1 minute).

3 YIELD AND INCENTIVES

Please refer to label 2 in Figure 1 for this section.

3.1 Summary

All underlying capital that is deposited by users to mint synthetic assets is deposited into a smart contract known as the yield manager. The yield manager lends this underlying capital out in order to earn interest. Interest earned accrues to both synthetic assets (increasing the price) and the treasury. The split is based on the balance of liquidity in the long and short positions.

3.2 Mechanic

The yield manager is a flexible contract that can plug into any yield bearing protocol. Different yield protocols have varying levels of risk and associated returns. From day 1, Float Capital is using Aave in the alpha. Aave is the largest blue chip yield protocol and provides desired risk adjusted returns. Float has been implemented in a modular format to allow use of other yield protocols in the future. A yield manager is deployed per market, allowing scenarios where different synthetic markets may seek yield in differing venues, congruent with the synthetic markets appetite for risk. The decisions above will be conducted by the community (Float Capital DAO) in the near future.

3.3 Yield split

Yield is split between the Treasury and the underlying synthetic market. Yield sent to the treasury is protocol revenue. Yield sent to the underlying synthetic market, increases the collateral backing each token and therefore increases the synthetic token price.

The split of interest sent between the treasury and synthetic market is a simple linear function dependant on the balance of long and short liquidity in the market. All yield flows to the treasury if there is equal long and short liquidity (yield is not required to incentivize certain market positions). The amount of yield flowing to the underlying market increases as the market is more imbalanced. All yield allocated to the underlying synthetic market is allocated to the position (long or short) with less liquidity, as an incentive to attract more liquidity into that position. The purpose being to incentivize equal liquidity in long and short positions.

The yield generated by the whole market is more than would be generated by one side (e.g. if the users of one side put their capital into the lending platform instead of into a FLOAT market) so if all the yield is given to the underbalanced side then an incentive is created for users to add liquidity to that side since they would be earning more yield than if they went directly to the lending platform. In reality it is slightly more complicated because there is another actor, **the treasury** (a smart contract), that is given a portion of the yield.

A simple example is as follows. Imagine the capital in the market was imbalanced: the short side only had \$100 000 capital while the long side had \$500 000. FLOAT will invest the total capital into a lending platform, which we will suppose has an APY (projected interest rate per year) of 8%. If all the yield is given to the short side then the short side would receive a yield of \$600 000×8% per year, which is an effective APY of $\frac{\$600000}{\$100000} * 8\% = 48\%$.

4 SYNTHETIC ASSET PRICE BEHAVIOUR

Please refer to label 3 in Figure 1 for this section.

4.1 Summary

Every few minutes, the smart contract will receive a new price for the underlying asset class. If the price of the asset has increased, underlying short liquidity will transfer to underlying long liquidity. If the price of the asset has decreased, the converse is true. The amount of value transferred between long and shorts is based on the current liquidity held in long and short positions, which can vary based on market demand. The variance in liquidity of long and short positions, can cause the synthetic asset to have floating exposure to the underlying asset. This is expanded upon below.

4.2 Exposure basic example

Where long liquidity = short liquidity, (\$100 000 of long exposure and \$100 000 of short exposure) the following is true:

- in the case that the underlying asset price increases by 1%,
- value to shift $1\% * \$100000 = \$1000$
- the value of the long side will now be worth \$101 000,
- while the value of the short side will be worth \$99 000.

Note that the value locked in the system is still \$200 000. Changes in the price of the asset simply shift the value between the pools of collateral backing the long and short synthetic tokens. As the price increases, value flows from shorts to longs and visa versa.

Now that the value on the long and short side is different, there will be floating exposure. Exposure is always based on the notional value of the market position with less liquidity. In this case, that is now the short side worth \$99 000. This means that the long side will only have 99 000 / 101 000 = 98% exposure, while the short side will continue to have 99 000 / 99 000 = 100% exposure.

Given the above, a subsequent market movement would create the following scernario

- the underlying asset price again increases by 1%,
- value to shift is always based on the price movement multiplied by the notional value of the side with less liquidity. Therefore, $1\% * \$99000 = \$900$
- the value of the long side will now be worth \$101 900 ($101000 + 900$),
- while the value of the short side will be worth \$98 100 ($99000 - 900$).
- hence mathematically the long side has 98% exposure ($1\% * 98\% * \$100000 = \$900$) as explained.

The reduction in exposure to the underlying asset for the overbalanced side is true for both upward and downward price movements.

4.3 Mathematical Formulation of Floating Point Exposure

4.3.1 Definition of exposure. It is common in traditional financial markets to talk about 'having exposure' to an asset. To have exposure to an asset is to be in a position where changes in properties of that asset affect one's financial investment in that asset. Example: one can be positively exposed to the price of gold, which means that when the price of gold increases then the value of one's investment will also increase. If the percentage by which the investment changes, p_I, is equal to the percentage by which the value of the underlying asset's property changes, p_a, then the exposure is said to be 100%. The exposure can be seen as the factor that p_I must be multiplied by in order to get p_a. This can be formalized as follows:

$$Ep_a = p_I$$
$$\implies E = p_I \left(p_a\right)^{-1}$$

where E is the exposure. We can expand the percentage changes in order to get the following definition of exposure:

$$E = \frac{\Delta I}{I}\frac{a}{\Delta a} \tag{1}$$

where I is the value of the investment and a is the value of the property of the underlying asset. In this definition we assume that both values are elements of the real numbers.

4.3.2 Market rebalance equation. Suppose we have a market whose long & short synthetic tokens track the value of some property of some underlying asset, a. Suppose also that the total value that has been invested in the long side of the market is l, and the total value in the short side is s. Consider the case where there is a change in the price of the underlying asset, δa. If the change is positive then value needs to shift from the long side to the short side, and vice versa when the change is negative. The value that is shifted between the long and short sides needs to be such that the maximum exposure for both sides is 100%. The following update equation is the one that Float uses in order to achieve this:

$$\Delta l = \min\left(l, s\right)\frac{\Delta a}{a} \tag{2}$$
$$\Delta s = -\min\left(l, s\right)\frac{\Delta a}{a} \tag{3}$$

You can see that the exposure is always less than 100% by plugging the above values into the exposure equation (1):

$$E_l = \frac{\Delta l}{l}\frac{a}{\Delta a}$$
$$E_l = \frac{\min\left(l, s\right)}{l} \leq 1$$
$$E_s = \frac{\Delta s}{s}\frac{a}{\Delta a}$$
$$E_s = -\frac{\min\left(l, s\right)}{l} \leq 1$$

4.3.3 Differential equations. Consider the case where m remains fixed and a varies. We can talk about l & s as functions of a. So (2) & (3) become

$$\frac{dl}{da} = \frac{\min\left(l, s\right)}{a} \tag{4}$$
$$\frac{ds}{da} = -\frac{\min\left(l, s\right)}{a} \tag{5}$$

Solving these differential equations results in the following functions for l & s. The initial conditions are a_1, l_1, s_1.

When $l_1 \leq s_1$:

$$l(a) = \begin{cases} \frac{l_1}{a_1}a & : l \leq s \\ -\frac{m\alpha_1}{2a} + m & : l > s \end{cases} \tag{6}$$
$$s(a) = \begin{cases} -\frac{l_1}{a_1}a + m & : l \leq s \\ \frac{m\alpha_1}{2a} & : l > s \end{cases} \tag{7}$$

where

$$\alpha_1 = \frac{a_1 m}{2l_1}$$

When $l_1 > s_1$:

$$l(a) = \begin{cases} \frac{m}{2\alpha_2}a & : l \leq s \\ -\frac{m\alpha_2}{2a} + m & : l > s \end{cases} \tag{8}$$
$$s(a) = \begin{cases} -\frac{m}{2\alpha_2}a + m & : l \leq s \\ \frac{m\alpha_2}{2a} & : l > s \end{cases} \tag{9}$$
$$\tag{10}$$

where

$$\alpha_2 = \frac{2a_1 s_1}{m}$$

The above equations show the value of the long and short positions as a function of the underlying asset price in the scenarios where long liquidity is greater than short liquidity and vice versa.

4.4 Keeper

The mechanic is such that value shifting and interest accrual should take place every time a price update is made available by an oracle. This means that in order for the synthetic to accurately track the underlying assets, a keeper bot is necessary to continually perform this upkeep.

The first keeper is simply users. All smart contract interactions call a hook that performs this update functionality to ensure an up to date system state. To further improve the accuracy of synth tracking, we have also developed a custom keeper smart contract. When called this smart contract reads whether price updates are available and if so performs upkeep action. The bot uses smart logic to automatically bump gas prices and monitor deviation thresholds of underlying assets ensuring timely upkeep is performed. A redundancy bot further brute force calls the contract periodically if the main bot goes offline.

The system will be moved to a more decentralized chainlink keepers mechanism when this becomes available on the Polygon network.

5 STAKING

Please refer to label 4 in Figure 1 for this section.

5.1 Summary

Synths can be staked (locked up in a smart contract) in order to earn alphaFLT tokens. The amount of alphaFLT earned is function of the dollar value of the synth staked, the length of time the synth is staked, and the balance of long and short liquidity in the underlying market. Synths may be staked and unstaked at point in time, with users incurring a 0.5% fee when unstaking synthetic assets.

5.2 Basics

Equal liquidity in long and short positions is desirable as then both long and short synthetic tokens will have perfect exposure (100%) to the underlying asset. Section 2 already described how yield is used in order to incentivize equal liquidity in long and short positions.

The second mechanism used for this incentivization is the rate at which alphaFLT tokens accrue to long and short liquidity staked. The basic mechanic is that liquidity staked from an underbalanced position will accrue at a much faster rate than the overbalanced position. This obvious incentive here is for users to mint and stake liquidity of the underbalanced side of the market to take advantage of the much higher rate at which alphaFLT is earned.

Every couple minutes, a new oracle price update is received and users new positions in the system are processed, leading to continued change in the liquidity balance of longs and shorts. There is a significant engineering challenge in providing a dynamic and accurate rate of alphaFLT accrual for all stakers given the liquidity balance is changing so frequently in markets. Fortunately, we are mathematicians with extensive solidity experience. The following sections outline mathematically how we relied on the technique of memoisation plus incentive curves to achieve this desired result.

Short explanation of the math in `_calculateFloatPerSecond` function in *Staker.sol* (link to code) works.

5.3 Theory

Note that the goal here is to find a way to split up the alphaFLT rewards between the long and short sides of the market. An imbalanced market is undesirable because the exposure on one side will be less than 100%. And so we want to have the alphaFLT rewards adjust in such a way that users are incentivized to rebalance imbalanced markets i.e. give more alphaFLT to the underbalanced side.

5.3.1 Key.

L	value of the **L**ong market side
S	value of the **S**hort market side
$M = L + S$	total value in **M**arket
E	a positive **E**xponent used to modify the slope of the curve
λ	percentage offset for market (adjustable variable)
$\Lambda = \lambda M$	offset scaled to market size
$d = L - S$	signed imbalance in market

5.3.2 Equations. Here are the unscaled rewards (alphaFLT per second) for the long & short side as functions of d:

$$R_L(d) = \begin{cases} 1 & : d + \Lambda \le -M \\ R_-(d) & : 0 < d + \Lambda < M \\ 1 - R_+(d) & : -M \le d + \Lambda \le 0 \\ 0 & : M < d + \Lambda \end{cases} \tag{11}$$

$$R_S(d) = 1 - R_L(d) \tag{12}$$

where

$$R_-(d) = \frac{1}{2}\left(1 - \frac{d + \Lambda}{M}\right)^E \tag{13}$$

$$R_+(d) = \frac{1}{2}\left(1 + \frac{d + \Lambda}{M}\right)^E \tag{14}$$

Note that the domains of R_L and R_S are both $D = \{d \in \mathbb{R} : -M \le d < M\}$ since $d = L - S = M - 2S$ and S has a range of $(0, M]$ (the reason 0 is not included is outside the scope of this derivation).

Also note that R_L and R_S are percentages i.e. $R_L(D) \subseteq [0, 1]$

5.3.3 Derivation. Requirements for the 2 curves R_L and R_S:

(1) Both must have a value of $\frac{1}{2}$ when $L = S$ (equal issuance for a balanced market)
(2) Both must be continuous for all values of L & S
(3) $R_L + R_S = 1 \ \forall \ L, S$ (we don't want any alphaFLT to be wasted)
(4) R_L increases (with) when L decreases and vice-versa for R_S
(5) Both must have an adjustable intensity; higher intensity meaning that, around the point where $L = S$, R_L increases faster as L decreases, and vice-versa for R_S i.e. alphaFLT rewards quickly shift toward the underbalanced side as a market becomes imbalanced
(6) R_L & R_S are both ≥ 0 (we can't have negative alphaFLT issuance)

One way of finding curves that fit is by starting with $a = \frac{U}{M}$ where U is the value of the underbalanced side i.e.

$$U = \begin{cases} L & : L \le S \\ S & : S < L \end{cases}$$

Consider the case where $U = L$

If we make $R_O \propto a$ (where O is the value of the overbalanced side, in this case $O = S$) then #4 of our requirements is half-satisfied i.e. satisfied just for L. In fact, if we make $R_O \propto a^E$ then #4 of our requirements is still half-satisfied as long as $E \in \mathbb{N}$, and we now have #5 half-satisfied. Let

$$R_O(a) = a^E$$

Obviously #2 & #6 are half-satisfied for all $L \leq S$, and we can half-satisfy #1 by adjusting the equation to

$$R_O(a) = \frac{1}{2}(2a)^E$$

since

$$a = \frac{1}{2} \iff U = \frac{M}{2} \iff L = S$$

and

$$1^E = 1 \; \forall E \implies \frac{1}{2}1^E = \frac{1}{2}$$

Finally, if we set $R_U(a) = 1 - R_O(a)$ then #3 is satisfied for all $L \leq S$. Also, it follows that #1, #2, #4 & #5 are now wholly satisfied for all $L \leq S$.

Consider the case where $U = S$

Similarly, we have R_O & R_U that satisfy #1-6 for all $S < L$. So now all conditions are fully satisfied.

Turn R_O & R_U into R_L & R_S

Start with:

$$R_O = \begin{cases} \frac{1}{2}\left(\frac{2L}{M}\right)^E & : L \leq S \\ \frac{1}{2}\left(\frac{2S}{M}\right)^E & : L > S \end{cases}$$

and use definitions of d & M to get

$$2L = M + d \quad \text{and} \quad 2S = M - d$$

$$\implies R_O(d) = \begin{cases} \frac{1}{2}\left(\frac{M+d}{M}\right)^E & : d \leq 0 \\ \frac{1}{2}\left(\frac{M-d}{M}\right)^E & : d > 0 \end{cases}$$

$$= \begin{cases} R'_+(d) & : d \leq 0 \\ R'_-(d) & : d > 0 \end{cases}$$

where

$$R'_-(d) = \frac{1}{2}\left(1 - \frac{d}{M}\right)^E$$

$$R'_+(d) = \frac{1}{2}\left(1 + \frac{d}{M}\right)^E$$

Also,

$$R_U(d) = \begin{cases} 1 - R'_+(d) & : d \leq 0 \\ 1 - R'_-(d) & : d > 0 \end{cases}$$

Now,

$$R_L = \begin{cases} R_U & : L \leq S \\ R_O & : L > S \end{cases}$$

$$\implies R_L(d) = \begin{cases} 1 - R_+(d) & : d \leq 0 \\ R_-(d) & : d > 0 \end{cases}$$

and

$$R_S = \begin{cases} R_O & : L \leq S \\ R_U & : L > S \end{cases}$$

$$\implies R_S(d) = \begin{cases} R'_+(d) & : d \leq 0 \\ 1 - R'_-(d) & : d > 0 \end{cases}$$

$$\implies R_S(d) = 1 - R_L(d)$$

All the conditions are satisfied for R_L & R_S since they are satisfied for R_U & R_O, as long as $-M \leq d \leq M$. There are now only 2 differences between these final equations and the original ones (1) & (2): the variable Λ and the edge cases that it creates.

The percentage offset λ was added so that if a market tends to always be skewed in a particular direction (e.g. market is stable when long side is 10% more than short side) then we can adjust the alphaFLT rewards to increase incentive for the market to stabilize around the 50/50 (long/short) mark instead. (50/50 markets are preferred because the exposure for both sides is 1.) The percentage offset has to be scaled to the market size in order to give the desired impact, so Λ is added to the equations. Due to the offset 2 new cases were added to the piecewise function to keep condition #4 & #6 satisfied.

5.4 Theory adjusted to Solidity code

R_- & R_+ are both adjusted like so:

$$R_-(d) = \frac{1}{2}\left(1 + \frac{d + \Lambda}{M}\right)^E = \frac{1}{2}\frac{\left[2\left(S - \frac{\Lambda}{2}\right)\right]^E}{M^E}$$

$$R_+(d) = \frac{1}{2}\left(1 - \frac{d + \Lambda}{M}\right)^E = \frac{1}{2}\frac{\left[2\left(L + \frac{\Lambda}{2}\right)\right]^E}{M^E}$$

Problem - in the EVM the maximum integer size is 2^{256} - so if $M^E > 2^{256}$ we have an overflow. We can prevent this issue by dividing both the numerator and the denominator in the above equations by the same amount before raising them to the power E. Note we loose precision due to integer division. Let

$$\alpha = \text{divisorToPreventIntegerOverflowOnExponentiation}$$

then

$$R_{-}(d) = \frac{1}{2}\frac{\left[\frac{2\left(S-\frac{\Delta}{2}\right)}{\alpha}\right]^{E}}{\left(\frac{M}{\alpha}\right)^{E}}$$

$$R_{+}(d) = \frac{1}{2}\frac{\left[\frac{2\left(L+\frac{\Delta}{2}\right)}{\alpha}\right]^{E}}{\left(\frac{M}{\alpha}\right)^{E}}$$

How can we optimise this for solidity? We can use bitshifting rather than division, as long as we keep α a power of 2. Bitshifting left by 5 is the same as dividing by 2^5.

6 ALPHAFLT TOKENS

Please refer to label 5 in Figure 1 for this section.

6.1 Summary

alphaFLT tokens accrued can be burnt in exchange for either the alpha protocol revenues or FLT tokens upon the release of the Float v1 system. The strategic move allows not only the core protocol mechanics to iterate without legacy, but the tokenomics of the system to also improve in the coming release.

6.2 Motivation

The motivation for releasing alphaFLT is the power a two token model affords rapid iteration. When engineering such a revolutionary and complex protocol such as Float, tokenomics are another massive component that warrants meticulous research to ensure long term success of the protocol. Given our interest in deploying the alpha for incentive testing and user insight, the alphaFLT is an opportunity to not be pigeon holed into a legacy tokenomic design that may not be suited for the protocol.

alphaFLT is well thought out v0 tokenomic system that stands to greatly benefit Float early adopters though alpha treasury revenue and favourable conversion into FLT tokens upon the next system release.

7 OTHER

7.1 Float Capital Alpha Audit

Float Capital ran a \$50 000 smart contract audit competition where ethereum security experts competed to find vulnerabilities in the float alpha system. The results of the competition can be found here.

It is very important that you understand an audit does not equate to risk free. Many risks are present in the Float Capital alpha system and users should do their own due diligence before minting a synthetic asset. This blog post describes the risks in more detail.

7.2 GEMS

Users who interact with the float capital protocol can receive 250 GEMS per day. These are currently non-transferable ERC20 tokens. The tokens entitle users to special discord permissions, special NFTs, and early release access to name just a few.

7.3 Discounts

Float Capital is pioneering cross protocol partnerships and utility. A flexible fee discount model allows users with certain NFTs to have protocol fee discounts. Initial partnerships with NFT protocols Rumble Kong League (\$40m total value) and Wildcards (\$200k raised for animal conservation) are just the tip of the iceberg in this innovation.

7.4 Shifting

Float Capital is first synthetic asset protocol (that we are aware of) providing the ability to natively shift synth positions (staked or not) between long and short.

This allows efficient building of delta neutral liquidity vaults, key to scaling to mass market liquidity.

7.5 Governance

Float Capital is a community governed protocol that currently relies on pseudo discord role base governance, where roles are awarded to users based on GEMS collected. In the future, Float Capital will move toward becoming a DAO.

7.6 Alpha Live performance

Since the live alpha has commenced, more than \$250 000 has been organically deposited into float to mint synthetic assets, making cheap and fast synth exposure a reality. The goal of the alpha will be to continue adding more synthetic assets and testing current incentive parameters. These learning's will be integral for future protocol iterations.

7.7 Float v1

The subsequent release following the float alpha is going be informed by the insights from the alpha. The key to the design will be ensuring sustained and efficient protocol liquidity.