flæx

Decentralized Margin Trading Exchange Powered by Flash Swap

Nguyen Tien Duy
November 28, 2022

Abstract

Decentralized Finance has introduced a grand new way for anyone with an internet connection to participate in the lividest financial world without the needs of an intermediary party.

Apart from the most common financial products, defi also gave birth to some of the most unique instruments that has no similar counter-peer in the traditional market. One of the most famous is probably Flash Swap.

For experienced traders in the defi world, this infamous term usually recalls some of the most shocking hacks that left its victims with hundreds of millions of dollars in never-recoverable loss.

In the next parts of this daring piece of paper, we attempt to redefine the term Flash Swap and by utilizing its unique power, we introduce flæx – a breakthrough decentralized margin trading exchange.

Table of Contents

1 Introduction

With the birth of defi, or decentralized finance, anyone with an internet connection and a crypto wallet such as Metamask, Trust Wallet… can now fully participate in the decentralized world without having to worry about the needs to trust an intermediary party.

With the facilitation of many existing crypto exchanges, assets now exist under digital forms can be freely traded against each other at certain price. Traders can speculate on the rise of prices of preferred assets by going for a long position. However, there still exists some difficulties to effectively speculate on the opposite movement of asset's prices, aka going for a short position.

flæx offers a grand new way for traders to speculate on either side of crypto assets' price movement. flæx also offers significant leverage should traders are confident on their decision and thus allows them to gain much more handsome rewards for their trades. The protocol is 100% transparent and permission-less, meaning users are in full control of their positions at any given point in time.

On a larger scale, it is estimated that the size of the derivatives market on existing financial infrastructure far outstrips the market size of any other type of financial asset at roughly over $1.2 quadrillion, or more than 10 times the total world GDP. With the undeniable advantages of the blockchain technology, more and more assets, including real-world ones will soon be available for trading in defi.

2 Existing Works

Most users are familiar with centralized exchanges when it comes to trading crypto assets. Although equipped with ease of uses and friendly interface, centralized exchanges' users are subject to users' fund manipulation. This fact materializes as recent shockingly incidents taken place in the crypto worlds, exposing harsh truths that user's fund is 100% in the hand of its custodial party.

A number of decentralized exchanges have been introduced in the past couple of years and they have successfully removed the mentioned risk. However, some drawbacks were also put forward.

In order for a margin trading exchange to efficiently operate, a certain number of mechanisms has to be implemented. These mechanisms will be responsible for deciding the price at which two (or more) assets are traded against each other, it also has to allow users to borrow funds that will be used to leverage their position as well as maintain an efficiently on-time liquidation process to ensure the protocol's solvency.

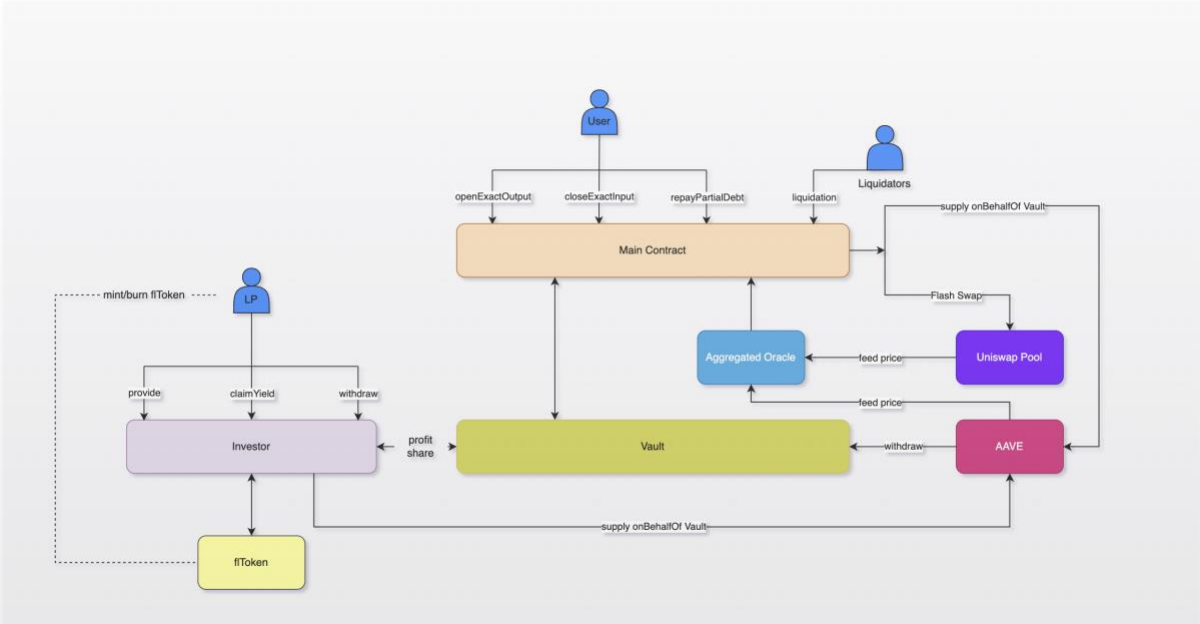
When it comes to borrowing funds for position leveraging, cost-wise, there are 2 issues that needs be solved. On traders' side, these funds need to be cheap and easily accessible. On liquidity providers' side, a sufficient rate of rewards needs to be ensured.

Existing protocols strive to maintain both trading and lending sections on their own, although some are significantly successful, it is clear to notice that bootstrapping enough liquidity on a new chain, or even on current chains, is extremely challenging. Any drops in liquidity provider's rewards that results in a less-than-accepted yield might lead to massive withdrawals and leave the protocol in great dangers of insolvency.

With this in mind, flæx sees that trying to maintain everything on its own is a tremendously challenging task that pays proportionally less rewards. Instead, flæx chose to rely on already successful and market-risk-tested partners to fuel its operation. By combining the rich and deep liquidity of traditional DEXes like Uniswap and the livid money market of lending protocols such as AAVE, flæx is capable of providing seamless margin trading experience to its users at an extremely low fee and low cost (which can go as low as 0.00009%/hour, or less than 0.8%/year).

3 Protocols

3.1 Architectural Implementation



There are 4 main contracts that are used to facilitate flæx's operation, in which 2 contracts will serve as users' main interaction point.

- Main Contract:

  Contain basic functions such as:

  - Open Orders
  - Close Orders
  - Repay Partial Debt
  - Liquidation

- Vault Contract:

  Vault contains all assets, including yield-bearing assets and debt assets that are generated by Main. This architectural choice eliminates the needs of Main to both act as an interaction point and assets storage at the same time. Instead, every time a user interacts with Main Contract, it will transfer all assets to Vault and only keeps an internal accounting figure for users.

  Since Vault most critical functions (such as withdrawals of eligible assets) are only callable by Main, it is much more difficult for an attacker to exploit some coding errors and directly interacts with Vault.

  On the other hand, Main Contract is set free from its storage role and can dedicate its resources for user interaction only.

- Investor Contract:

Because AAVE limits it maximum loan-to-value, Main contract alone cannot support users to enter a larger-than-4X leveraged position. By allowing other users to provide extra collateral (only one accepted asset allowed), flæx effectively decreases its loan-to-value and thus can support users to enter a much higher leverage position (which can go up to as high as 20X).

When a user decides to invest his asset to facilitate flæx's operation, he will be eligible to 2 sources of income:

- A default interest rate that comes from AAVE.
- A share of the protocol's sources of revenue, generated from other users' positions under the form of trading commission fee.

flæx also limits its Investor's volume at a proportional rate that only grows with its ongoing Open Interests. This secures the total yield accrued to its investors since the larger the Investor Pool is, the less yield each unit of capital will receive in return.

Investor Contract will keep track of each user's balance and rewards with the help of flToken Contract.

- flToken:

  flToken has 2 basic functions that inherit standard ERC20 tokens that are:

  - mint
  - burn

  Whenever users supply or withdraw a certain amount of asset, flToken will be minted or burnt accordingly. It is worth notice that the amount minted/burnt is not exactly the same as the amount of token supplied/withdrawn. Inside the Investor Contract, a set of calculation is carried out in order to return a precise number. The math is inferred from the interest rate provided by AAVE.

  flToken also acts as a base to calculate each user's eligible yield.

3.2 Margin Trading Module

3.2.1 Description

In a margin trade, a trader borrows an asset and sells it immediately on the market for another asset, which is supposedly the asset that said trader would like to trade against.

It is crucial to understand that the definition of Long/Short is merely a simplified interpretation for end users. Let's take a look at below example:

Supposed a user would like to open a Long position on the trading pair of:

ETH/USDC

In this case:

- ETH is called a base token.
- USDC is called a quote token, which basically describes how much "USDC" it takes to be equal to 1 "ETH".

Longing with margin on ETH/USDC simply means that a trader is borrowing more quote token (USDC) than he has to buy a certain amount of base token (ETH). If the price of ETH rises against

USDC, he would be able to sell all his ETH for a larger than previous amount of USDC and repays all his debt. The remaining USDC would be his profit. In the event that ETH's value drops relatively to USDC, a trader would suffer a loss.

Now consider the case that a user would like to go Short on the same trading pair, ETH/USDC.

He would then need to borrow a larger amount of base token (ETH) than he has and trades it for quote token (USDC). If the price of ETH drops relatively to USDC, he would only have to spend a less amount of USDC in order to buy back the amount of the ETH owned previously and thus effectively pockets a difference as profit.

It is easy to notice that going for a Short position of ETH/USDC simply means going Long on the "opposite" pair, which is USDC/ETH.

If the logic for going Long is applied, one will clearly understand the mechanism behind and the quote "the definition of Long/Short is merely a simplified interpretation for end users" is justified.

Moving forward in this paper, the term "Long/Short" will no longer be used and instead, the term "base token" and "quote token" will imply the direction of the trade, along with "open order" and "close order".

3.2.1.1 What is a Flash Swap?

Flash Swap is an integral function of Uniswap V2 and V3. Since most current DEXes on the Ethereum blockchain as well as many other EVM-based-blockchains are forked from Uniswap, they all inherit the same function.

It is quite difficult to have an analogy of Flash Swap in real world examples. Instead, let's take a look at how an ordinary swap is carried out inside of Uniswap.

Suppose John wants to buy some ETH with his USDC, the technical term suggests that John is swapping his USDC for ETH.

John then proceeds to interact with Uniswap Pool of ETH/USDC. This pool contains both assets (ETH and USDC) and the logic to decide the price of each asset relatively to the other. In an ordinary swap, the first thing John would do is send his USDC to the Pool, after the Pool receives John's USDC, it will conduct its preset logic and calculates how much ETH is to be worth and then sends out ETH to John.

However, in a Flash Swap, the Pool doesn't necessarily need to check if it has received any USDC at the beginning of the swap. Instead, it optimistically transfers out the requested ETH to John. At this point in time, John owns that ETH and can do whatever he wants with it. At the end of the transaction, the Pool will enforce that enough USDC is returned to its reserve.

One might wonder that what if John doesn't pay back his USDC in a Flash Swap?

Since Ethereum transactions are atomic, the entire transaction can be rolled back to its initial state and John's ETH would actually never be transferred out.

It is also worth notice that everything must occur within a single block and John can decide to either return ETH or USDC to Uniswap Pool.

3.2.1.2 General Idea

Having grasped the basic concepts of Margin Trading and Flash Swap, let's dive into how flæx utilizes these features to power its own trading platform.

For margin trades, it is common that a trader has access to a pool of assets available for borrowing.

This is not the case for flæx!

Roll back to the example in 3.2.1, supposed John would like to go for a long position on ETH/USDC.

Since John is going on long on ETH, which is the base token, John has to already own some ETH and needs to borrow some USDC in order to buy more ETH.

John can go to any existed Lending Protocols such as AAVE, Compound… and deposits his ETH as collateral, he then is eligible to borrow any other allowed assets against his collateral. In this case, he can borrow some USDC, which is the quote token, to buy more ETH.

However, Lending Protocols limit his borrowing power due to their "over-collateralized" model, which means he cannot borrow more than what he has put in as collateral. In fact, he can only borrow at a maximum cap of around 80-85% depending on which kind of assets being supplied.

Let's say John put in 1 ETH at a value of 2000 USD per ETH, he decides to go for an 80% maximum cap of borrowable, which is equal to 1600 USDC and buys another 0.8 ETH.

At this point, John's position would be:

- Collateral: 1 ETH
- Borrow: 1600 USDC
- Available asset: 0.8 ETH

This would mean that John has entered a Long position in ETH/USDC at 0.8X Leverage.

What John can do now is re-supply his 0.8 ETH to the Lending Protocol to "rinse and repeat", effectively keeps increasing his leverage.

However, one can clearly see that this approach poses a great deal of risks and requires many repetitive transactions. John can be instantly liquidated right after he hits his first borrow at 80% as the interest is accrued to his debt.

This is where flæx comes into play and provide an extremely clever solution.

In order to increase leverage, the most crucial task is to be able to keep on borrowing, buying assets and re-supplying those assets back as collateral.

After the first borrow, John is left with 0.8 ETH which he would definitely continue to re-supply into the Lending Protocol as collateral. What if John has access this 0.8 ETH even before he has to supply the first 1 ETH? If John can achieve this, he can actually supply 1.8 ETH and thus has effectively increased his borrowable amount.

It turns out that this is entirely possible thanks to the power of Flash Swap. John can first Flash Swap 0.8 ETH, plus his initial 1 ETH, he can now supply 1.8 ETH into the Lending Protocol as collateral and proceed to borrow the same 1600 USDC as previously explained to repay the Flash Swap amount.

In this case, John only has to borrow a much less amount of USDC which is reflected in terms of loan-to-value because:

- Collateral: 1.8 ETH
- Borrow: 1600 USDC
- Loan to Value: {1600 / (2000 USDC per ETH)} / 1.8 = 44.44%

John can easily increase his leverage by increasing the Flash Swap amount, as long as his loan-to-value doesn't exceed the maximum cap, he will not be liquidated.

At 80% maximum cap, John can effectively go as high as 4X!

3.2.1.3 flæx's role

From the above example, one might wonder what the role of flæx is supposed John can carry out this entire maneuver on his own.

Firstly, it is worth notice that a Flash Swap is only available via smart contract interactions and requires deep understanding of not only defi but also program development expertise.

Secondly, when an individual user is attempting to carry out this entire transaction on his own, he is subject to maximum 4X leverage. At the same time, he would be liquidated if asset's price moves to the unfavorable direction much sooner than he expects. Because at any given point in time, if the loan-to-value reaches the maximum cap, which is only at about 80%, he would be instantly liquidated. This can be simplified by the below example:

A trader entered a long position of 2X in ETH/USDC, the entered price of 1 ETH is equal to 2000 USDC. The amount is 10 ETH.

His position would be:

- Collateral: 10 ETH
- Borrow: 10,000 USDC
- Loan to Value at entered point in time: 50%

If the price of ETH drops to a point where his loan-to-value increases to 80%, he would be liquidated. The implied price could be calculated to be at 1250 USDC per ETH.

This is rather unfavorable for traders as commonly speaking, when a trader goes for a 2X leverage position, he wouldn't be liquidated if assets price won't drop by nearly half!

flæx, with its robust and well-designed architectural implementation, can allow other users to act as "collateral" provider, effectively decreases the whole protocol's loan-to-value and thus increases trader's leverage level.

The suggested ideal leverage would be somewhere around 10X.

3.2.2 Use Cases

Margin trading in general is a method to multiply gains from traders' critical decisions. The rise or fall of asset price directly affects their profit/loss.

Margin trading can also be used for hedging another existent position.

As long as traders can well-manage their positions, their gains would be multiplied by many times.

Investors can decide to become “collateral” provider to earn real yield generated from both interest rate and sharable profit from other trader’s commission fees.

3.2.3 Protocol Walkthrough

3.2.3.1 Entering a position

As described in the Designed Architectural Implementation section, users interact directly with the Main Contract to open orders.

Behind the scene, Main Contract will make external calls to other relevant contracts to execute user’s orders.

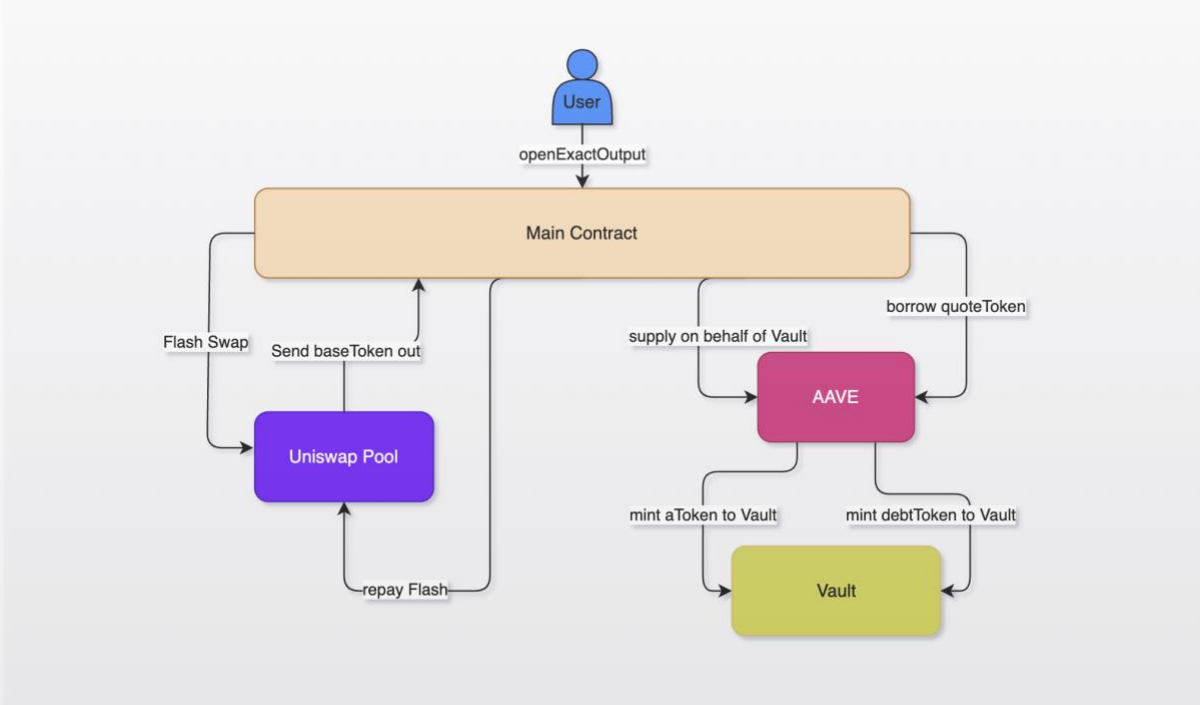
In order to enter a position, user will send the following transaction to *openExactOutput* function, detailed as below:

Name	Type	Description
baseToken	address	Address of the base token in a pair
quoteToken	address	Address of the quote token in a pair
baseMarginAmount	uint256	Initial margin to be put in, denominated in base token currency
maxQuoteTokenAmount	uint256	Maximum amount of quote token in acceptable, denominated in quote token currency
uniFee	uint24	The fee tier of Uniswap, responsible for determining which specific pool to interact with; in percentage, scaled-up by 100
marginLevel	uint256	Position leverage, in percentage, scaled-up by 100

When Main Contract receives the user’s sent transaction, the following will happen:

- *transferFrom baseMarginAmount* from user to Main.
- Main calls Flash Swap on Uniswap Pool to request *baseMarginAmount * marginLevel* to be sent to Main.
- Main calls Supply on AAVE Pool to supply *baseMarginAmount + baseMarginAmount * marginLevel* on behalf of Vault.
- Main calls borrow on AAVE Pool to borrow a sufficient amount of *quoteToken*, the *amountToBorrow* plus *tradingFee* must be less than or equal to *maxQuoteTokenAmount*.
- Main repays the previously Flash Swap with the *amountToBorrow*.

- Main completes the transaction by recording all relevant information containing user's position in a public mapping from user's wallet address to *abi.encode(baseToken, quoteToken)* to *orderInfo*.



All steps happen atomically, meaning that they will all succeed or fail together, ensuring that no malicious or errors can happen in between.

If at the end of the transaction, if the condition *amountToBorrow + tradingFee <= maxQuoteTokenAmount* is not met, transaction will fail.

3.2.3.2 Closing a position

Closing a position applies the same Flash Swap mechanism, along with some twerks.

User sends a transaction containing below information to Main, specifically calling *closeExactInput* function.

Name	Type	Description
baseToken	address	Address of the base token in a pair
quoteToken	address	Address of the quote token in a pair
baseTokenAmount	uint256	Amount of base token to be closed, use *type(uint256).max* for 100% close
minQuoteTokenAmount	uint256	Minimum amount of quote token out acceptable, denominated in quote token currency

uniFee	uint24	The fee tier of Uniswap, responsible for determining which specific pool to interact with; in percentage, scaled-up by 100

Upon receiving user's sent transaction, Main will conduct some sanity check on the *baseToken, quoteToken, baseTokenAmount* to enforce user's legitimacy.

After, the following steps will happen:

- Main calls Flash Swap on Uniswap Pool to request an *AmountToFlash*, this amount is derived from the *baseTokenAmount* input to Uniswap Pool and is denominated in *quoteToken* currency. Since this is technically an exactInput type of swap, later on the condition *AmountToFlash >= minQuoteTokenAmount* must hold true, otherwise transaction will fail.
- Main calls Repay on AAVE Pool to repay user's debt using the *AmountToFlash*. Depending on the current price, *AmounToFlash* could be greater than user's current Debt. If this is the case, a full withdrawal will be made.
- Main calls *withdrawFromVault* on Vault to withdraw *baseTokenAmount*. Main contract is responsible for calculating the correct *baseTokenAmount* in case user inputs *type(uint256).max*. If *AmounToFlash > user's Debt,* withdraw 100% of user's collateral.
- Main transfers a certain amount of both *baseToken* and *quoteToken* to user depending on the current price. The amounts that are transferred out reflect user's profit and loss.

Again, all steps happen atomically, rendering them successful or failed all together.

3.2.3.3 Repay Partial Debt

At any given point in time, users can choose to repay parts of their current debt. This is usually because users' position is on the brink of being liquidated or it could simply because they do not wish to continue incurring extra interest on the borrow amount.

By repaying parts of their debt, users effectively increase their Margin Ratio and could avoid a potential liquidation.

The transaction sent to Main's function *repayPartialDebt* contains below information:

Name	Type	Description
baseToken	address	Address of the base token in a pair
quoteToken	address	Address of the quote token in a pair
quoteTokenAmount	uint256	Amount of base token to be repaid

It is worth notice that users cannot repay 100% of their debt because that would simply revert to closing a position.

3.2.3.4 Liquidation

Liquidation on flæx happens quite similarly to liquidation on Lending Protocols such as AAVE, Compound… Liquidators send an amount of asset to repay parts of the liquidated user's debt and as a liquidation incentive, liquidators receive a percentage of the liquidated user's collateral, apart from the liquidated amount.

flæx as a whole is subject to its lending partner, which is AAVE, in terms of liquidation; however, its users should also rely on Uniswap price as a source of valid liquidation price. The solution is an oracle aggregator which aggregates prices from both AAVE's native oracle and Uniswap Oracle. Each party accounts for an equal percentage of 50% in the final liquidation price.

flæx embraces the common definition of "Margin Ratio" as liquidation condition. The formula to calculating "Margin Ratio" is:

$$MarginRatio = \frac{Value\ of\ Collateral}{Value\ of\ Borrow}$$

Currently, the threshold for liquidation to happen is at Margin Ratio <= 1.1.

At Margin Ratio = 1.25, users will receive Margin Call. Upon receiving such notifications, users can decide to repay parts of their debt, or close their position to avoid liquidation should the movement of the price keeps steering towards unfavorable direction.

Using Uniswap Oracle is quite risky, especially on Layer 2 chains such as Optimism, Arbitrum… because the cost for manipulating prices is much less than on Layer 1. On the other hand, using AAVE's native Oracle is indeed safer, but subject to delays in price broadcasting and insensitivity to small price movements.

flæx's architectural mechanism allows for a unified aggregation of both Oracles and only inherits the most critical traits of both while eliminating the risks of oracle manipulation to carry out malicious liquidations by attackers.

In order for a valid liquidation to happen, the condition below must hold true:

$$AAVEMarginRatio * 99\% \leq BWAPMarginRatio < AAVEMarginRatio$$

Where:

- *BWAPMarginRatio:* Balance-Weighted Average Price Margin Ratio of all Uniswap Pools containing the same pair of assets.

$$BWAPMarginRatio\ = \frac{\sum_i (Margin\ Ratio\ of\ Pool\ i * balanceOf\ Pool\ i)}{Total\ balanceOf\ all\ Pools}$$

- *AAVEMarginRatio:* AAVE Margin Ratio

This allows for a more precise source of liquidation price to step into play. It also removes the risk of Uniswap Oracle manipulation because it requires Uniswap Margin Ratio to be less than AAVE Margin Ratio, but no less than AAVE Margin Ratio * 99%.

Liquidators call Main's function *liquidation* with below parameters:

Name	Type	Description
baseToken	address	Address of the base token in a pair
quoteToken	address	Address of the quote token in a pair
liquidatedUser	address	Address of the user to be liquidated
debtToCover	uint256	Amount of debt to repay for liquidated user

Upon receiving the transaction, Main will perform sanity checks and conduct its logic. At the end of the transaction, the liquidator will receive his eligible collateral and an extra amount of collateral under the form of liquidation incentive.

3.3 Fees

3.3.1 Commission Fees

Commission fees only apply for *openExactOutput* and *closeExactInput*. flæx waives fee for *liquidation* and *repayPartialDebt*.

However, a proportion of the liquidation incentive is sharable between the Protocol and the liquidator as well as a proportion of trading commission fee would be kept inside Vault as a source of protocol development fund.

The standard trading commission fee is currently at 0.1%.

The trading fee sharable for the Protocol is currently at 20%.

The liquidation incentive is currently at 2%.

The liquidation incentive sharable for the Protocol is currently at 20%.

All fees are transferred to Vault and is re-distributed to Investors later.

3.3.2 Funding Rate

Funding Rate is derived directly from flæx's lending protocol partners. Due to their market-proved efficient model, funding rate (or more precisely referred to as borrow rate) is extremely low on carrying an open interest on flæx. This is most significantly reflected on large positions that are opened for a rather longer period of time.

The funding rate on flæx depends on the interest rate (or APY) of base token and quote token. The formula is as follow:

$$fundingRateAPY = baseLendingAPY - \frac{marginLevel}{1 + marginLevel} * quoteBorrowingAPY$$

Supposed users opened a position on the pair of ETH/USDC with *marginLevel* of 200%, where:

- ETH Lending APY is: 0.55%
- USDC Borrowing APY is: 1.71%

The fundingRateAPY would be: 0.59%
This converts to roughly: ~0.000067%/hour

Which is quite insane compare to other existing trading platforms!

It is also worth notice that the funding rate is implied within the position of users and is directly cleared when closing a position.

3.4 Investor Module

3.4.1 Description

The Investor Contract plays a vital role in facilitating flæx's operation as well as ensuring the protocol's solvency. One can think of the Investor Contract as a kind of a Pool Contract in other protocols such as Uniswap, GMX, Open Leverage, dYdX… where the Pool provides sufficient liquidity for traders to borrow funds against their collateral. At the same time, it enables its liquidity providers to earn real yield generated from the protocol's revenue. As crypto market evolves, this model of real yield sharing is becoming more and more preferable to retail investors as it creates true and measurable values to its token/stake holders.

One unique trait of flæx's Investor Module lies in the previously described Flash Swap mechanism. Since traders alone have already been able to fuel their position up to 4X leverage, they only require a relatively smaller amount of funds inside the Vault to fuel their larger leverage orders. This enables Investor to capture much more handsome rewards on their investment and thus creates extremely efficient capital usage. flæx also applies a supply cap on the total amount of investment eligible, meaning two things:

- Only early investors are able to join the protocol's Investor Module to earn real yield.
- Existed investors are guaranteed of their yield returns based on the protocol's current business revenue.

Without the supply cap, as the protocol grows in public reputation, more and more investors will be able to join freely in order to earn yields while its revenue cannot keep up with the growth of capital supplied, leading to a significantly less yield generated per unit of capital.

Some estimations could be made as below:

For most of other protocols, the pool must be responsible for the total of their amount of Open Interest, let us presume this type of model yields a 100% efficiency in capital usage.

For flæx, suppose all open interest amount is X, at average leverage of 4X, this renders the whole protocol's margin ratio at 1.25. The total amount of capital eligible for supply that is capped at margin ratio <= 2 is:

$$totalSupply \leq 0.6 * X$$

Compare to other protocols where every unit of open interest amount is fueled by 100% of the liquidity available inside the Pool, flæx only needs 0.6. Which means at its worst state where the supply cap is reached, the capital efficiency is at least **66.67%** more efficient.

This is all thanks to novel implementation of the Flash Swap function!

3.4.2 Use Cases

Investors can diversify their portfolio and expect a high accuracy for rate of return.

Smart contract architectural designs ensure the safety of users' fund.

Compare to other options such as supplying directly to Lending Protocols, investors of flæx earns an extra source of yield generated from trading commission fees on top of the already accrued interest.

3.4.3 Profit Sharing Architecture

The profit sharing model is quite straight-forward.

Apart from the default interest accrued from flæx's lending protocol partners, investors are eligible to another source of income, which is the trading fees generated from other users' positions.

Inside the Investor Module, a global index is kept track of under the name of *flIndex*. This index is responsible for highly accurate calculation of each user's eligible yield and embraces the "ever-increasing" model that resembles industry-standard lending protocol architecture.

3.4.4 Function parameters

The Investor Module exposes some simple interfaces for users to interact with:

Function Name	Name	Type	Description
provide	*amount*	*uint256*	Amount of assets to provide
provideLock	*amount*	*uint256*	Amount of assets to provide
	daysToLock	*uint256*	Amount in unit of days to lock
claimYield			
withdraw	*amount*	*uint256*	Amount of flToken to withdraw

4 Summary

- flæx
  - A complete defi suite platform.
  - Open-sources, free to use.
  - Non-upgradeable smart contracts.
- flæx Margin Trading Module
  - Peer-to-Peer decentralized margin trading protocol.
  - Margin leverage supported up to 20X.
  - Can be used to speculate on assets' prices to earn multiplied profits.
  - Can be used to hedge other opened positions.
  - Extremely low funding rate.
  - Competitive trading commission fee.
- flæx Investor Module

- Profit sharing, real yield distributors.
- Highly efficient in terms of capital usage.
- Double sources of income for investors.
- Creates true and measurable values for token/stake holders.