

---

# AquaFi Protocol

*Release 0.0.1*

**Umar Salim**

**Oct 11, 2021**



# GETTING STARTED

<b>1</b>	<b>Getting Started</b>	<b>3</b>
<b>2</b>	<b>Developer links</b>	<b>5</b>
<b>3</b>	<b>Contents</b>	<b>7</b>
3.1	Overview . . . . .	7
3.1.1	What is AquaFi? . . . . .	7
3.1.2	How does it work? . . . . .	7
3.2	Frequently Asked Questions . . . . .	8
3.3	Overview . . . . .	8
3.4	Architecture . . . . .	8
3.5	<b>Aqua Primary Contract</b> . . . . .	9
3.5.1	<b>Stake</b> . . . . .	9
3.5.2	<b>Unstake</b> . . . . .	9
3.6	<b>Oracle Contract</b> . . . . .	10
3.6.1	<b>Fetch</b> . . . . .	10
3.6.2	<b>FetchAquaPrice</b> . . . . .	10
3.7	<b>Handler Contract</b> . . . . .	10
3.7.1	<b>V2 Handler Functions</b> . . . . .	10
3.7.2	<b>Update</b> . . . . .	10
3.7.3	<b>Withdraw</b> . . . . .	10
3.8	<b>V3 Handler Functions</b> . . . . .	11
3.8.1	<b>Update</b> . . . . .	11
3.8.2	<b>Withdraw</b> . . . . .	11
3.9	<b>Admin Functions</b> . . . . .	11
3.9.1	<b>AddPools</b> . . . . .	11
3.9.2	<b>UpdateIndexFundAddress</b> . . . . .	11
3.9.3	<b>UpdatePremiumOfpool</b> . . . . .	12
3.9.4	<b>UpdatePrimary</b> . . . . .	12
3.9.5	<b>UpdatePoolStatus</b> . . . . .	12
3.10	<b>Layer 2</b> . . . . .	12
3.10.1	<b>Background</b> . . . . .	12
3.10.2	<b>Finalised Approach</b> . . . . .	13
3.11	<b>Aqua Token</b> . . . . .	13
3.11.1	<b>Contract Addresses</b> . . . . .	13
3.11.2	<b>Contract Functionality</b> . . . . .	13
3.11.3	<b>Aqua Token Privileges</b> . . . . .	13
3.12	<b>Testnet Competition</b> . . . . .	14
3.13	<b>Bounty</b> . . . . .	14
3.14	<b>Audit</b> . . . . .	14

3.15	Distribution . . . . .	14
3.15.1	Eligibility . . . . .	14
3.15.2	Allocation . . . . .	15
3.15.3	Distribution . . . . .	15
3.16	KPI Options . . . . .	15
3.16.1	Background . . . . .	15
3.16.2	KPI Token Information . . . . .	16
3.16.3	Options Distribution . . . . .	16
3.16.4	Options Deadlines . . . . .	16
3.16.5	Governance Votes . . . . .	17
3.17	Water Charity . . . . .	17



Welcome! The purpose of the following pages are to provide a comprehensive document specifying all the different aspects of the AquaFi protocol and the Aqua token. If you are new to AquaFi you may want to check out the [Overview](#) or the [Frequently Asked Questions](#)

**This documentation is aimed at those who want to learn the basics of AquaFi to those who want to learn the more in-depth technical aspects of the protocol too.**



## GETTING STARTED

Explore the sidebar to find more specific documentation covering other aspects of the protocol.





## **DEVELOPER LINKS**

The AquaFi codebase is comprised of an ecosystem of open source components. AquaFi follows a highly modular design which allows functionality to be swapped out without a token migration - you can read more about this on the *[Architecture](#)*

- [Link 1](#)
- [Link 2](#)



## CONTENTS

[Keyword Index](#), [Search Page](#)



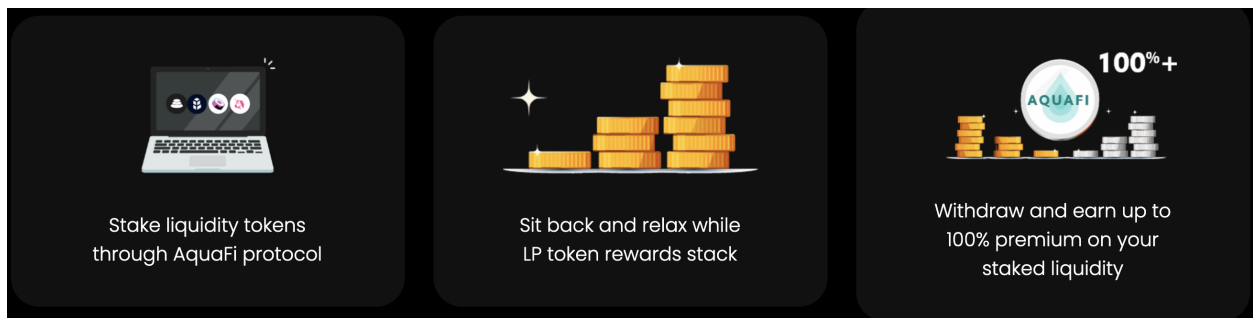
## 3.1 Overview

### 3.1.1 What is AquaFi?

Aqua Finance (AquaFi) allows liquidity providers to earn up to 100% additional yield on their liquidity.

### 3.1.2 How does it work?

AquaFi (in short) captures fees generated from staked liquidity and pays an additional premium upon unstaking. This additional premium and the returned liquidity fees will be paid in Aqua tokens.



The steps below show what this looks like from a users perspective:

- User stakes their decentralised exchange (DEX) liquidity tokens into AquaFi

- User allows some time to pass, during this time, the staked liquidity tokens are earning fees on their associated DEX
- User decides to unstake their initial liquidity from AquaFi at which point the initial liquidity tokens will be returned to the user, all fees generated will be returned to the user in Aqua tokens along with an additional premium which can be up to 100%

## 3.2 Frequently Asked Questions

There are currently no Frequently Asked Questions and this page will be updated as these questions become known.

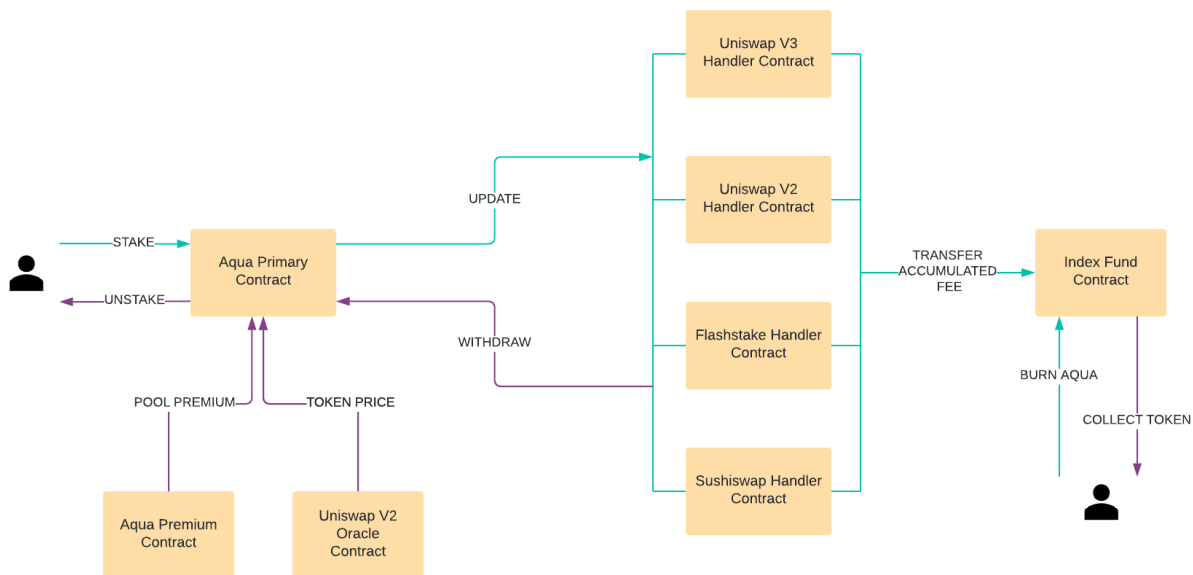
## 3.3 Overview

This page is potentially not needed and may be removed in the near future.

## 3.4 Architecture

Here at Blockzero Labs we have learnt a significant amount from the development process and architecture of the [Flash protocol](#). This is why we decided to ensure AquaFi was designed in the most upgradable way possible from the beginning. It is due to this reason we followed a highly modular design for the protocol.

These modules can be upgraded/modified after the initial launch to extend or change the functionality. This is possible via governance and will require the Blockzero multi-sig wallet and a time delay to approve such a change - more information can be found on the governance section of this documentation.



The AquaFi protocol can be broken down into the following discrete pieces:

- **Aqua Primary Contract:** This module is responsible for bringing together the other modules of AquaFi. This contract specifies what handlers are allowed to interact with the Aqua Primary as well as additional functionality such as Stake, Unstake and Unstake all.

- **Oracle Contract:** This module is responsible for obtaining the price of the Aqua token in a decentralised manner. We currently have an implementation of this using time-weighted-average-price (TWAP) oracle from Uniswap Version 2. All pairs must have their pairs created with WETH in order for oracle to work.
- **Handler Contracts:** These “handler” contracts are responsible for supporting a specific decentralised exchange such as Uniswap V2. The purpose of this feature is to integrate the exchanges with aqua protocol upon launch, we intend to launch with the following handler contracts: Uniswap V2, Uniswap V3 and Sushiswap.
- **Index Fund Contract:** This module is responsible for dispatching the tokens requested by the users against the aqua token they hold. It calculates the equivalent amount of tokens to be dispatched on the basis of the aqua token percentage they provide. After transferring the tokens, this contract burns the aqua token of that user thus decreasing the aqua total supply and increasing the value of the token.
- **Timelock Contract:** The sole purpose of this contract is to provide time-lock functionality on selected functions. This is the contract which “owns” or can execute the “admin” functionality. The Blockzero council multi-sig will be the proposer and executor
- **Premium Contract:** This contract is responsible for calculating the premium on fees earned by the user while unstaking. All of the logic for calculating premium resides in this contract.

## 3.5 Aqua Primary Contract

The primary contract consists of some generalized components that ensures the integration with different exchanges like Uniswap V2, Uniswap V3, Sushiswap and others.

### 3.5.1 Stake

Parameters: (uint256 tokenIdOrAmount, address handlerAddress, address contractAddress, bytes data)

Description: This function is responsible for staking the users lp token and transfer it to handler contract. It updates the stakes record and invokes the handlers update function where it checks for validation checks & stores the lp token data.

### 3.5.2 Unstake

Parameters: (bytes32[] calldata id, uint256[] calldata tokenValue)

Description: This function is responsible for unstaking the lp tokens. It iterates through the list of lp tokens and calculates the cumulative fees in aqua token from all the lp tokens. This function fetches the fees by calling the internal function `_unstake` in which which expects single id and tokenValue and returns the aqua amount in terms of fees accumulated for that token. Inside the function it fetches the token difference, aqua premium and token address by calling the `withdraw` function of the selected handler. Then on the basis of aqua price fetched from the oracle, it calculates the aqua amount which is then added to the fees calculated on the basis of aqua premium. It also mints additional 20% of Aqua tokens and sends it to the indexfund contract.

## 3.6 Oracle Contract

This contract is responsible for fetching the price of a provided token address with respect to aqua token. It only works for pair who has pair with WETH.

### 3.6.1 Fetch

Parameters: (address token, bytes32 data)

Description: This function is responsible for fetching the amount of aqua per token provided.

### 3.6.2 FetchAquaPrice

Parameters: (no params)

Description: This function retrieves aqua price in terms of eth i.e one Eth is equal to X Aqua.

## 3.7 Handler Contract

There are different handler contracts specific to the exchanges and their respective logic, each handler contract makes sure the interaction with aqua primary contract and allow only aqua primary contract to access its functions to perform the aqua protocol specific operations.

- Handlers with ERC20 lp tokens (Uniswap v2, Sushiswap, Flashstake etc)
- Handlers with ERC721 lp tokens “nfts” (Uniswap V3)

### 3.7.1 V2 Handler Functions

The major v2 handler functions are as follows.

#### 3.7.2 Update

Parameters: (bytes32 stakeId,uint256 lpAmount,address lpToken,bytes calldata data)

Description: It stores the rootk ( $K = \text{root}(\text{token0} * \text{token1})$ ) against the lpToken that has been staked. It is only callable by the primary contract. It decodes the staker and pool address from the encoded data param. If the pool is whitelisted and the stake does not already exists then it finds the rootK value of the lpToken and saves the required stake data in the mapping.

#### 3.7.3 Withdraw

Parameters: (bytes32 id, uint256 tokenIdOrAmount, address contractAddress)

Description: Calculates the rootK difference to see if any fees has been accumulated. If so, it only pays the lp amount back to the user and returns fees in token0 & token1 back to aqua primary contract from where aqua gets paid out. Indexfund receives lp token. It is only callable by the primary contract, it returns tokenAddress, premium, tokenDifference and encoded data of premium, tokenAddress and tokenFess accumulated. Once it is done then it deletes the stake entry from the storage.

## 3.8 V3 Handler Functions

The major v3 handler functions are as follows.

### 3.8.1 Update

Parameters: (bytes32 id, uint256 tokenValue, address contractAddress, bytes32 data)

Description: It checks the fees against the nft, if the fees exists then it updates the fees in the user stakes record. It does so by first decoding the pool address from the encoded data received from the params. It checks if the pool is whitelisted if yes then it updates the token0AtDeposit and token1AtDeposit by fetching it from nftPositionManager positions on the basis of tokenValue received in the params. This function is only callable from Aqua Primary Contract.

### 3.8.2 Withdraw

Parameters: (bytes32 id, uint256 tokenIdOrAmount, address staker, address contractAddress)

Description: This function performs validation checks & then sends the lp tokens back to the staker. It also transfers the fees accumulated in token0 and token1 to indexfund. It returns the amount of fees in token0 & token1 back to aqua primary contract from where aqua token gets paid out with premium.

## 3.9 Admin Functions

The functions specified below are only callable by the owner address. The owner address for these functions is a timelock contract controlled by the Blockzero council. These functions are available in all of the handler contracts.

The timelock contract enforces a waiting period after a proposed change has been suggested.

### 3.9.1 AddPools

Parameters: (address[] calldata tokenA, address[] calldata tokenB, uint256[] calldata aquaPremium, uint24[] calldata fee)

Description: This is an owner modified function (only callable by owner). It expects the lists of token addressees, list premium of the pairs and their corresponding fee values. It's purpose is to whitelist the pools in the handlers.

### 3.9.2 UpdateIndexFundAddress

Parameters: (address newAddr)

Description: This is a owner modified function (only callable by owner). It expects the new index fund address. It is used to update/change the index fund contract address to be used in the handler.

### 3.9.3 UpdatePremiumOfpool

Parameters: (address pool, uint256 newAquaPremium)

Description: This is a owner modified function (only callable by owner). It expects the pool address and its corresponding new aqua premium value. It is used to update the aqua premium value of the whitelisted pools in the handler.

### 3.9.4 UpdatePrimary

Parameters: (address newAddress)

Description: This is a owner modified function (only callable by owner). It expects the new address of the primary contract. It simply updates the aquaPrimary contract address in the handler.

### 3.9.5 UpdatePoolStatus

Parameters: (address pool)

Description: This is a owner modified function (only callable by owner). It expects the address of a pool for which the status in the pool whitelist has to be changed.

## 3.10 Layer 2

### 3.10.1 Background

During the research and development phases of AquaFi we constantly analysed the layer 2 landscape to ensure maximum success for the protocol. We recognise that transactions fees on the Ethereum main-net may continue to rise over time thus making the protocol unusable for smaller stakes.

We analysed many different solutions to get AquaFi onto layer 2:

- Force the user to migrate their liquidity from L1 to L2 and then accept the stake (all within a single transaction and transparently via the UI)
- Launch AquaFi with support for pre-determined L2 solutions - eg Arbitrum and/or Optimism
- Create a token bridge when needed in the future upon L2 deployment
- Add support to the Aqua token to allow updating of the minter addresses

Each of these solutions has its advantages and disadvantages. The main disadvantages being:

- additional development work
- having to know and permanently finalise the L2 networks to support upon Aquafi launch
- enabling additional minters
- traditional token bridges only work for tokens that do not have a minting functionality



### 3.10.2 Finalised Approach

The solution we eventually decided on was to add support to the Aqua token which would allow the Blockzero council to update the minter addresses. This will allow us to have the flexibility of creating a token bridge and whitelisting the token bridge address on the Aqua token for a true 1 to 1 Aqua exchange between layer 1 and layer 2.

This was proposed to the community and approved by nearly 90% of all XIO holders agreeing with the solution, additional information about this can be [found here](#)

## 3.11 Aqua Token

The Aqua token is the utility token that will be used within AquaFi to pay premiums to users. The AquaFi protocol will capture liquidity from staked tokens and upon unstaking, the user will be paid their fees with newly minted Aqua tokens.

### 3.11.1 Contract Addresses

This token has been deployed on both test-net and the Ethereum main-net.

- [Mainnet] 0xd34a24006b862f4e9936c506691539d6433ad297
- [Rinkeby] 0xB2dc7059D7c9c2b93E924dc7C938FcbF986aC5C2

### 3.11.2 Contract Functionality

The Aqua token conforms to the ERC20 token standard which means the following functionality is supported:

- Mint
- Burn
- Transfer
- TransferFrom
- BurnFrom

### 3.11.3 Aqua Token Privileges

The Aqua token will allow users to:

- Have governance rights over almost all AquaFi related proposals and almost all AquaFi related decisions - e.g. change pool premiums - the exhaustive list is as follows:
  - Governance will be executed through timelock contract.
  - Base premium can be decided by the aqua community.
  - Updating oracle address, handler addresses, locking and unlocking of handler contracts.
  - Grant/Revoke Minting roles to addresses.
  - Allows users to burn aqua tokens to redeem fees from indexfund contract. Eg, If users burns 10% of the aqua tokens then users can withdraw 10% off underlying assests in the indexfund.

## 3.12 Testnet Competition

Information here will be updated as the testnet competition begins and results have been finalised.

## 3.13 Bounty

Information here will be updated when/if the community bounty takes place along with any findings.

## 3.14 Audit

Information here will be updated as soon as the audit has been completed.

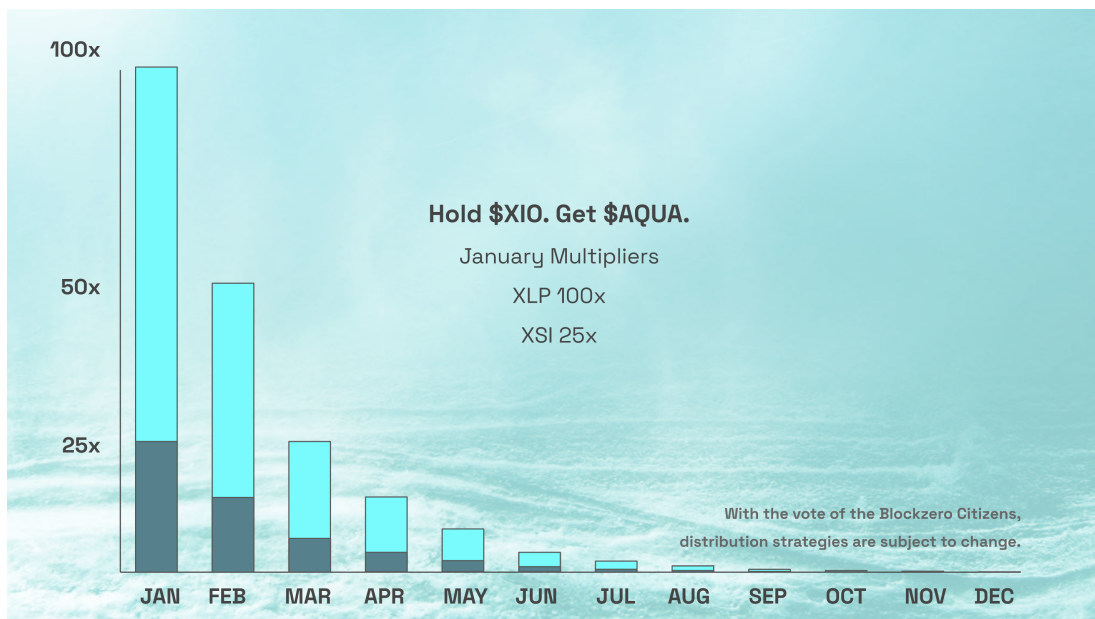
## 3.15 Distribution

This page will briefly discuss the initial token allocation and distribution of the Aqua token.

### 3.15.1 Eligibility

The AquaFi protocol and the corresponding Aqua token was developed by Blockzero Labs as part of the token studio model.

It was stated in December of 2020 that all XIO holders would receive free Aqua tokens upon launch. Since this date, XIO holders have been earning the rights to Aqua tokens on a daily basis as per the image circulated below:



XIO holders were eligible to Aqua tokens so long as they held XIO tokens from December 2020 to December 2021.

This has since changed as per [this governance vote](#) due to complications introduced during development which you can find out about [here](#).

**The eligibility of Aqua tokens will run up until the launch of AquaFi. Once the protocol launches, XIO holders will no longer be earning Aqua tokens over time (as per the model illustrated above)**

As mentioned in the governance vote, we will determine a rough estimate of Aqua based on current projections (if eligibility continued until December). These tokens will be minted and placed into the Vortex where citizens will be able to earn them by staking XIO. This effectively stopped all future eligibility and allows earning of Aqua based on staked XIO in the Vortex.

### 3.15.2 Allocation

Initially it was decided that all the Aqua tokens in existence would be allocated to XIO holders. This was announced in December of 2020 as part of the Christmas announcement.

The token allocation has since changed through governance votes which were all approved by the community:

- What percentage of Aqua tokens would you want to see in the Blockzero Vortex? (result: 41%)
- How many % of the Vortex Aqua tokens should be kept in orbit 1 (treasury) for Aqua KPI Options and other marketing purposes? (result was overturned by below vote)
- Should 20% of the total Aqua supply be allocated to KPI Options as mentioned below and 21% be allocated to the Vortex Treasury? (result: yes)

Due to the above governance votes, the token allocation will be as follows:

- 21% to Vortex Treasury
- 20% to KPI Options (you can read more in the appropriate section of this documentation)
- 59% to XIO Token holders who earned Aqua

We estimate the total token supply to be somewhere around 700-900 million.

### 3.15.3 Distribution

The distribution of tokens to the XIO token holders who earned Aqua will take place through Dropzero. This means upon launch, we will upload a CSV file with all the earned tokens and mint these number of tokens before placing them directly into Dropzero. Once these tokens are placed into Dropzero, these tokens will be claimable by all participants.

## 3.16 KPI Options

### 3.16.1 Background

There was much discussion in the community over the duration of development regarding a proposal originally submitted by AllStuffCrypto. This proposal consisted of airdropping Aqua tokens to the target market of AquaFi - liquidity providers in the hope this would serve as a good marketing campaign to draw attention to the protocol.

This idea eventually morphed into designing KPI Option tokens which allow recipients to either immediately liquidate their option tokens for Aqua or help AquaFi by adding value which increases the value of said options.

### 3.16.2 KPI Token Information

We have decided to utilise 20% of the total supply of Aqua towards these KPI Option tokens. We will deposit this into a smart contract and mint 100,000 Option tokens. The value/exchange rate of these options tokens will be based on the Total Fees Captured KPI Metric. This means as the total fees captured by the protocol increases, these KPI Option tokens will also increase in value since they will be redeemable for more Aqua tokens.

The format of these options tokens will be as follows:

- Total Fees Captured:  $\geq \$0$ , 1 KPI Option = 0 Aqua
- Total Fees Captured:  $\geq \$250,000$ , 1 KPI Option = 0.25 Aqua
- Total Fees Captured:  $\geq \$500,000$ , 1 KPI Option = 0.5 Aqua
- Total Fees Captured:  $\geq \$750,000$ , 1 KPI Option = 0.75 Aqua
- Total Fees Captured:  $\geq \$1,000,000$ , 1 KPI Option = 1 Aqua

These options tokens will be using a linear scale rather than unlocking at each fee level. This means if the total fees captured is \$100,000, the options tokens can be exchanged for 0.1 Aqua tokens. We decided to use a linear scale to help subdue the downward pressure if recipients decide to liquidate their option tokens.

The KPI Metric “Total Fees Captured” will be updated by the Blockzero council to avoid any tampering or exploits which is something that may be present with on-chain calculations. The option to switch to a on-chain calculation will exist through governance but additional development work will need to be undertaken for this functionality.

The estimated number of Aqua tokens allocated to KPI Options is 150 million.

### 3.16.3 Options Distribution

We will initially distribute 50% of all KPI Options upon launch (or shortly after) of AquaFi. The Blockzero council will retain control of the remaining 50% and release these as new DEX’s and/or projects are added to AquaFi. These remaining 50% of KPI option tokens will be kept in the Vortex until they are voted on to be used to help incentivise additional DEXs.

Example: If AquaFi supports Curve Finance LP tokens in the coming months after the initial deployment then these reserved KPI Option tokens could be used to help incentivise Curve LPs to add their Curve LP tokens into AquaFi.

### 3.16.4 Options Deadlines

The initial distribution will be valid for upto three months after they are initially uploaded into Dropzero (available for claiming). This means the original recipients will have three months to claim their Aqua KPI Option tokens from Dropzero before they are returned to the Vortex.

These KPI Option tokens will be redeemable until the end of the ethereum blockchain. However, if the total fees captured does not hit \$1,000,000 USD by **XXXXXXXXXX** date then the value of the options at this date will be the permanent value. This means if the total fees generated is \$900,000 by the aforementioned date, then regardless of the KPI increasing (total fees generated), these option tokens will still be exchangeable for 0.9 Aqua tokens. **Please note, this is subject to how the KPI Option token from UMA works and is therefore subject to the same limitations.**

### **3.16.5 Governance Votes**

- Original Vote for token allocation: [here](#)
- Latest Vote for token allocation: [here](#)

## **3.17 Water Charity**