Betoken Whitepaper Draft

Zebang (Zefram) Liu

Introduction

Betoken is a decentralized hedge fund built on the Ethereum blockchain that invests in ERC20 tokens. It automatically redistributes control over investment decisions to managers who make the most profitable investment proposals, whose collected wisdom is compiled into good investment decisions, using a unique decision making system we call "Incentivized Meritocracy".

The core ideas behind Betoken's Incentivized Meritocracy is that control over decisions is tokenized, that the control tokens are valuable, that good decisions are rewarded with control tokens proportional to both the quality and the quantity of their benefits, and that bad decisions receive penalties in control tokens proportional to both how bad of a decision they were and how much damage they caused.

Betoken is for everyone: everyone can join, everyone can invest, everyone can make decisions for the fund and be rewarded for making good ones, and everyone can rise to the top if they have the merit. Betoken is unstoppable: it is a completely decentralized application built on the censorship-resistant Ethereum blockchain. Betoken is transparent: all statistics and decisions are publically available, and all fees and clauses are written in immutable open-source smart contracts.

1. The Betoken Model

1.1 Incentivized Meritocracy

An **Incentivized Meritocracy** is a system where

- The amount of control each actor has is proportional to their ability to make good decisions
- Actors are financially incentivized to maximize their control (therefore their decision-making ability)

The above definition is not rigorous, since "control" and "ability to make good decisions" are not clearly defined, but it provides a general idea of how an Incentivized Meritocracy should behave. To sum it up in one sentence: the best people are in charge, and everyone wants to be in charge. The first point is the desired result, and the second point is the means of achieving it.

Having the people with the most merit in charge is clearly good for an organization as a whole. In a hedge fund like Betoken, having the people who are the best at making investments handle the fund's investments means that the ROI of the fund is going to be pretty good.

Incentivized Meritocracies have never been successfully implemented before, since it is near impossible to have a centralized actor that can judge everyone impartially. However, newly-invented smart-contract-enabled blockchains such as Ethereum allow us to construct **decentralized** actors that can uphold unbreakable rules, making implementing an actual Incentivized Meritocracy possible. Betoken is the first decentralized application that incorporates an implementation of Incentivized Meritocracy.

1.2 Betoken's Solution

There are four central ideas behind Betoken's solution to Incentivized Meritocracy:

1. Control is denoted using a custom ERC20 token that must be staked when making investments for the fund, and the amount of the stake is proportional to the amount of investment.
2. The control tokens are valuable, in that holders of the token can expect income proportional to the amount of tokens they hold.
3. Good investment decisions are rewarded with control tokens proportional to both the quality (ROI) and the quantity (profit / prevented loss) of the investment decision.
4. Bad investment decisions receive penalties in control tokens proportional to both how far below 0 the ROIs were and how much money they lost.

We provide below a description of how Betoken functions and details of Betoken's Incentivized Meritocracy.

The Betoken fund runs in investment cycles, and at the start of each cycle there is a period of time where investors can deposit & withdraw their funds.

After that, users can propose investments into ERC20 tokens by staking some Kairo--the name we use for control tokens. You can stake Kairo into proposals other users made, which has the same effect as creating the proposal yourself. You can also stake Kairo on the opposing side of a proposal to bet on its failure, which doesn't have any effect on the investments being made but is important to keeping the Incentivized Meritocracy functional. There is a restriction where you must stake no less than a certain proportion of your Kairo balance when staking.

After a certain time has passed, any changes to proposals and stakes are no longer allowed, and existing proposals are turned into actual investments using the equation

- $investmentAmount = totalFunds \times \frac{proposalStake}{totalProposalStakeInThisCycle}$

Where $totalProposalStakeInThisCycle$ is the sum of all Kairo staked in support of all investment proposals during the current cycle. The reason $totalKairoSupply$ is not used as the denominator is that since it is unlikely that users would stake a large proportion of their Kairo, only a small fraction of the fund's assets would be invested every cycle, which will make the fund unprofitable. If you have Kairo and didn't stake anything during the staking period, a certain proportion of your Kairo will be taken away from you and be equally staked in opposition to all proposals.

After waiting for a certain time (ex. 30 days), the fund sells all tokens it invested in at the current market price. After the sell process is finished, the fund automatically determines how profitable each investment proposal was and redistributes Kairo based on the results. The amount of Kairos a user gets back for each proposal is $userStake \times (1 + ROIofProposal)$ if they supported it, and $userStake \times (1 - ROIofProposal)$ if they went against it, so if a proposal had a 20% ROI, everyone on the supporting side gets 20% more Kairos back, and everyone on the opposing side loses 20% of their stake.

At the end of every cycle, a certain proportion of total profits is set aside as commission and distributed among Kairo holders proportional to the amount they hold. A certain proportion of fund assets is also sent to Betoken's developers as a fee for using the platform.

1.3 Potential Problems

1.3.1 Cyclic Design

The reason that Betoken functions in rigid cycles rather than a more asynchronous manner is that it make the model much, much simpler. Asynchronicity will introduce many problems that we don't necessarily know good answers to, such as: How do we ensure that users can't just hold on to their Kairos without ever making investment decisions? How can we prevent users from canceling their stakes in a proposal that starts crashing right before its profitability is supposed to be evaluated? How do we evaluate the profitability of a proposal if anyone can stake in it at any moment before its evaluation? How do we handle investments if users can deposit and withdraw at any time? Each of the problems mentioned above has more than one potential solutions, thus many design choices will have to be made, often without a way of providing good justification. Further more, introducing additional complexity to a smart contract based system is often a bad idea, since computations and storage are expensive, and bugs are often deadly.

Due to the above reasons, Betoken employs a cyclic design. However, the lack of asynchronicity introduces a number of problems.

1.3.1.1 Short Term Decisions

Suppose each cycle is 30 days long. If some user knows that token A's price will rise greatly on the 10th day of the current cycle and drop soon afterwards, there's no way for the fund to utilize this information and sell at the peak. If another user knows that token B's price will drop greatly on the 10th day of the current cycle and rise back soon afterwards, there's no way for the fund to buy the dip either. This means that the fund misses out on opportunities shorter-term than the specified cycles.

In a cyclic model, there exists a dilemma between utilizing short-term opportunities and having consistent gains. To be able to bank on short-term price changes, the length of cycles needs to be short; to be able to have consistent gains resistant to temporary price extremities, the length of cycles needs to be long so that erratic changes are evened out over time. We think that relatively long cycles are good, because investors should focus on the long-term potential that a token and its related technology has, rather than only on the price fluctuations.

//TODO: Add in more problems

1.4 Reasons Why Betoken's Model May Work

Since Betoken is something unprecedented, we do not have evidence that its model will work as intended. A formal proof of Betoken's plausibility also seems unlikely, since it's quite difficult to accurately model actors' behaviors. Therefore, we can only give here several possible reasons for which Betoken will work as intended.

1.4.1 Better Than Direct Investment

To be able to attract people with flair in investing, we must make participating in Betoken's investment process more lucrative than directly investing in the tokens oneself. Fortunately, it is easy to prove that the model satisfies this requirement (discounting the fluctuation of Kairo's price):

- $ROI_{Betoken} = ROI_{DirectInvestment} + \frac{commission}{investmentAmount} \geqslant ROI_{DirectInvestment}$

Therefore, investors are incentivized to join Betoken and make investment decisions. If Betoken does work successfully, this would be one of the main reasons.

1.4.2 Analogous to Markets

Betoken's Incentivized Meritocracy shares many similarities to markets of investable assets, such as the stock market and the cryptocurrency market. In fact, staking for a proposal is almost exactly the same as directly investing the token, except that the ROI is better. Therefore, we can estimate Betoken's success as a meritocracy by looking at how meritocratic the stock market and other markets currently are.

To our knowledge, there is no evidence that they are not meritocratic: no one's heard of a dumb and inexperienced investor besting market growth, and smart people (like those at Renaissance Technologies) have achieved amazing ROIs (71.8% annual on average! [source]). Thus, we can expect that Betoken will also be meritocratic.

//TODO: Add more reasons

2. Implementation Details

2.1 Kairo's Initial Distribution

As of writing, we have not decided on Kairo's initial distribution scheme. There are two options that we're considering:

1. Any user who deposits investment during the first investment cycle will receive Kairo proportional to the investment. Withdrawing investment will of course be disabled. This is the model currently implemented in the smart contracts.
2. We will have a traditional ICO for Kairo.

The reasons for choosing the first scheme are:

- It makes bootstrapping the Incentivized Meritocracy easier, since users aren't paying for the Kairo they get.
- It allows us to attract testers for the pre-release versions that will be released on Mainnet by making pre-release Kairo and after-release Kairo compatible, and providing a more favorable conversion rate for the test versions. For example, if you get 1 Kairo for every Ether you deposit in the final release, you can get 2 Kairo for every Ether you deposit in the test versions.

The reasons for choosing the second scheme are:

- It provides us with the funding that we desperately need.
- People are more familiar with ICOs, so an ICO may have more traction.

2.2 Cycle Phases

Each cycle is divided into 5 phases:

- Deposit & Withdraw: When investors deposit and withdraw their funds.
- Investing: When users stake Kairo to make investment decisions for the fund.
  - Transition (Investing => Waiting): Stakes are taken from users who had Kairo and didn't stake anything, buy orders are made.
- Waiting: When everyone waits and let the token prices change.
  - Transition (Waiting => Ended): Sell orders are made.
- Ended: When the invested tokens are sold and users wait for the sell orders to go through.
  - Transition (Ended => Finalized): Kairo is redistributed according to proposal results, commission, developer fee, and Oraclize fee are paid, and funds are returned to investors' balances.
- Finalized: A placeholder phase before the next cycle begins.

In a preliminary setup, the lengths of each phase are as follows:

- Deposit & Withdraw: 1 day
- Investing: 1 day
- Waiting: 27 days
- Ended: 1 day
- Finalized: no time

totaling 30 days.

2.3 Token Trading

As of writing, Betoken uses EtherDelta, a decentralized ERC20 token exchange, to handle its token trading. When making orders, Betoken uses Oraclize to fetch the current market price of each token and set that as the price used in the corresponding order. However, since EtherDelta has been purchased by a dubious party and suffers a lack of credibility, KyberNetwork is selected as a potential replacement.

2.3.1 Oraclize

Oraclize in an oracle service that provides off-chain data to smart contracts. In Betoken, it is used in for fetching the current price of tokens, so that orders can be correctly made with appropriate prices. The inclusion to Oraclize poses two problems:

- Currently, CryptoCompare is used as the source of token prices, and a problem inherent to the usage of off-chain price feeds is that the feeds can go down/be manipulated. This brings a new attack surface into existence.

- Queries to Oraclize cost a small fee. In the current model, this is solved by setting aside the maximum possible fee from the fund's balance at the end of each cycle (the fees in the first cycle has to be paid manually).

2.3.2 EtherDelta

EtherDelta is a decentralized token exchange that uses a traditional model, where users can make and take limit orders. Its overarching advantage is that it is currently the largest decentralized exchange with the most volume and trading pairs. However, apart from its declining credibility, its inclusion in Betoken introduces a host of other problems:

- It's always possible that orders made by Betoken won't be picked up by takers.
  - Maybe the token's price changed significantly after making the order.
  - Maybe the price obtained from CryptoCompare was faulty or quite different from the price in EtherDelta.
  - Or maybe it's because the market is very sensitive to even small differences in order prices and it's inherently difficult to consistently pinpoint the appropriate price window.

  This introduces a great deal of uncertainty to Betoken's normal operation, which is extremely undesirable.
- There's always a delay between making an order and the order being fulfilled, and after Betoken becomes a major market player it is possible for malicious actors to use this delay to their own benefit. For example, if the price of a token usually sees a small spike after Betoken invests in it, someone can easily frontrun Betoken's order and profit on the spike, since Betoken is completely transparent.

Therefore, we hope to transition to using KyberNetwork as soon as it becomes a viable option.

2.3.3 KyberNetwork

KyberNetwork is "an on-chain protocol which allows instant exchange and conversion of digital assets (e.g. crypto tokens) and cryptocurrencies (e.g. Ether, Bitcoin, ZCash) with high liquidity."[source] It is a far superior option than EtherDelta, since it allows for instant token exchange which makes it able to avoid the problems mentioned in 2.3.2. Its inclusion would also eliminate the need for the "Ended" cycle phase, since we don't have to wait for orders to go through anymore, making the cycles more concise. Oraclize would also not be needed, since KyberNetwork provides its own on-chain price feed. Therefore, the inclusion of KyberNetwork will minimize the number of moving parts in Betoken's operation and significantly reduce Betoken's attack surface, reducing Betoken's running and maintenance costs and increasing its security.

However, KyberNetwork is still an in-progress product, unlike EtherDelta which has been online for a long time, so it's still unclear whether we'd want to use KyberNetwork in our final release. Furthermore, it's likely that the types of tokens that KyberNetwork supports won't be able to match EtherDelta for quite some time, so using KyberNetwork would limit the types of tokens that Betoken can invest in.

We will decide on this matter based on how much progress KyberNetwork will make before we roll out Betoken's Mainnet Alpha.

2.4 Rules In The Fund

Partly to ensure scalability and partly to prevent spamming attacks, there are certain rules in the fund's smart contract, which will be listed below. Note: most of the exact numbers used are only placeholders, and will likely be different in future releases.

- Each member can **create** at most 2 investment proposals during each cycle. (No restrictions on staking in proposals.)
- At most 20 investment proposals can be created in each cycle.
- Each proposal can invest in only one token, and two proposals cannot invest in the same token.
- When staking into a proposal, the size of the stake must be no less than 25% of one's Kairo balance.
- If a user has Kairo and didn't stake anything in the current cycle, 25% of their Kairo balance will be staked equally into the opposing side of all proposals.
- A user cannot stake into both sides of the same proposal.
- If the number of supporters of a proposal reaches zero, the proposal will be removed from the list. No similar rule for the number of opposers.

2.5 Smart Contract Maintenance

2.5.1 Upgrading Contracts

To upgrade the **BetokenFund** smart contract, the following steps will be taken:

1. The old contract is paused before the Waiting phase of the current cycle, and all users withdraws their investments.
2. The new contract is deployed.
3. The list of participants is read from the old contract and transferred to the new contract.
4. The addresses of subcontracts (**OraclizeHandler**, **ControlToken**) are sent to the new contract.
5. The owner of the subcontracts is set to the new contract.
6. The upgrade is now complete.

The script to do this can be found in Betoken's GitHub repository.

To upgrade subcontracts other than **ControlToken**, simply set the relevant address variable in **BetokenFund** to be the address of the new contract.

The **ControlToken** contract is not expected to be upgraded, since it is just a simple ERC20 token with minor additions.

2.5.2 Handling Emergencies

The **BetokenFund** contract inherits the **Pausable** contract from OpenZeppelin, so that it is possible to pause the normal operation of the fund when an emergency occurs, such as an attack or a market black swan event. When paused, an emergency withdraw function will be able to be called by users to withdraw all funds.

One thing to note about the emergency withdraw function is that if the fund is paused when the fund is invested in tokens, things would get more complicated, since the tokens have to be sold before users can withdraw, and the amount they can withdraw would likely be different from the amount at the start of the cycle.

2.5.3 Contract Administrator

The **BetokenFund** contract inherits the **Ownable** contract from OpenZeppilin, and the owner is given administrator rights. Calling the emergency functions, pausing and unpausing, calling the functions related to upgrading, and changing the fund's fee rates all require administrator rights. Initially, the owner is set to be an account owned by the Betoken team, so that Betoken may be smoothly bootstrapped; after Betoken has enough community support, it is

possible to set up a DAO (Decentralized Autonomous Organization) contract as the owner of Betoken, so that Betoken is completely decentralized.

3. Market Analysis

3.1 Competitors

3.2 The Cryptocurrency Market