Ventureum: A Milestone-driven Community-governed Crowdfunding Protocol for Blockchain Projects

Timothy Wang and Nathan Liu
dev@ventureum.io

Version 0.1.6

Abstract

Ventureum is an innovating crypto-crowdfunding protocol specifically designed for the Ethereum ecosystem to provide better control for investors over blockchain projects. This protocol implements a **Milestone-Driven Funds Management Module**. Milestone-Driven Funds Management Module acts as a Decentralized Autonomous Organization (DAO), which releases funds to project founders only if milestone deliverables have been delivered on time. It consists of a set of smart contracts, with which investors can vote to release or delay milestone payments. In the event that project promises are not met, investors can vote for a refund. The Milestone-Driven Funds Management Module is completely open source, and any blockchain projects can choose to integrate it into their own project by themselves without the Ventureum dev team's endorsement. If the founders of a blockchain project want the endorsement from the Ventureum dev team on their usage of the Milestone-Driven Funds Management Module, we will perform audits and security reviews of the source codes they deploy, and the use of Ventureum Network Token (VTH) for refund coverage is mandatory.

Contents

1 Introduction

1.1 Background

Initial coin offering (ICO) or Token Sale is a way of crowdfunding via use of cryptocurrency[1]. The process of a token sale involves a project founder issuing cryptographic tokens and distributing the tokens to investors and contributors of the project as a representation of a stake or interest in the project. A blockchain project usually involves technical innovations for blockchains (Ethereum, EOS) or various application scenarios of blockchain + X, such as blockchain + Instant Messaging (Status), or blockchain + Identity Verification and Protection (Civic), etc.

Token sales provide unique features and more flexibility to blockchain project founders and investors, compared to traditional capital funding mechanisms such as Venture Capital or Initial Public Offering. First, investors use cryptocurrencies, such as Bitcoin, Ether (ETH) instead of fiat currencies to participate in a token sale. Second, the exclusion of expensive intermediaries, such as investment banks, reduces the cost of capital funding and creates a fast and direct channel to deliver the committed capital from investors to project founders. Third, the tokens issued in a token sale can be traded immediately or shortly after the token sale period is over, thus creating high liquidity for investors to take profit and exit their positions. This is in sharp contrast with traditional venture financing in which early investors have almost no measure to exit shortly after they have committed their capital. These features combined with the breakthrough of blockchain technologies, especially the fast evolving Ethereum ecosystem, have made token sales grow abruptly from its inception in 2013 to a phenomenal capital market wonder in the first half of 2017 (Figure 1)[2].

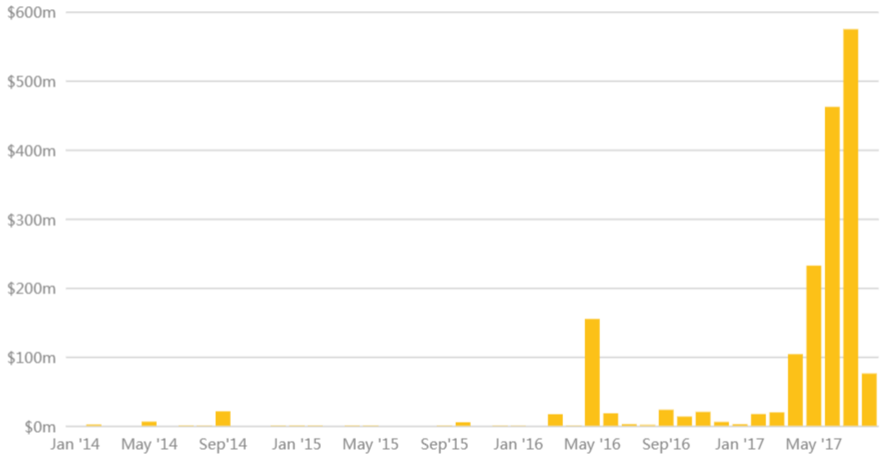

Figure 1: Monthly New Token Sale Funding (source: www.coindesk.com/ico-tracker/)

1.2 Problem Overview

However, the cryptocurrency world, including token sales, is still a largely unregulated territory that unfortunately provides rich soil for scam projects[3]. ICOrating.com lists a prolonged list of scam or Ponzi blockchain projects[4]. CHAINANALYSIS estimated that phishing, Ponzi schemes, and other scams account for about 10% of ICOs[5], and several cryptocurrencies have been announced as illegal by regulatory bodies across countries ([6, 7]). Another big issue is, even for blockchain projects with true motivation and goals, after receiving excessively large amounts of funding during crowdsales, the project founders have little to no incentive to deliver their products, as the investors have no voice or power to influence the progress of the projects[8].

The explosive growth of token sales has drawn close attention from the regulatory bodies of securities and finance across countries[9, 10, 11]. If the savage growth of token sales, especially the proliferation of scam token sales, continues and no self-regulatory mechanism emerges, it will be only a matter of time before token sales, and even the whole cryptocurrency world, are hit in a hard way. Regulatory bodies could unleash harsh control over token sales, or even ban them altogether in their jurisdictions.

It is a pressing call for a self-regulating mechanism or platform for token sales for anyone who wants the long-term existence and prosperity of crypto-token crowdfunding.

1.3 Ventureum: A Solution to Token Sale Self-regulation

To address the pressing issue of lack of self-regulation in token sales, the Ventureum team envisioned a mechanism of milestone-driven based funds management, implemented with the very spirit of Decentralized Autonomous Organization (DAO), to protect the interest of investors. The essential idea is that risk control over projects can only be realized by mobilizing the decision power of the whole community of investors with the help of open sourced smart contracts running on the Ethereum blockchain. To put it simply, investors won't betray themselves, and open sourced code does not lie.

The Ventureum team is fully aware of some crowdfunding platform competitors who claim to provide risk control over token sales. Yet, to our best knowledge, all of them rely on human effort (mostly the staff of their own crowdfunding platforms or a centralized governing body such as a board) to make decisions and filter out fraudulent projects, and most importantly, to manage funds. Again, this kind of crowdfunding platform works against the very spirit of DAO, and does not entitle investors to their deserved voice and power to monitor and influence the progress of projects and the release of funds. In the worst-case scenario, they could collude with project founders and harm investors' interests by pretending to have performed their due diligence and investigations.

The Ventureum project is the first ever proposed solution to self-regulating token sales, implemented in a decentralized way with total transparency. In the following sections, we present the technical details of the **Milestone-Driven Funds Management Module**. An elaborated description of the Ventureum token (VTH) allocation and the roadmap of development is also presented in Section 5.3 and 8.

2 Milestone-Driven Funds Management

One of the major problems of existing ICOs is that investors have no control over how funds are used. For instance, investors who participate in ICOs can be taken advantage of because usually they do not have voting rights over projects. Worst of all, after receiving excessively large amounts of funding during crowdsales, project founders have little to no incentive to deliver their products. Another critical issue is that it is difficult for inexperienced investors to distinguish legitimate projects from projects designed to be exit scams for their founders. These exit scam projects have caused investors to lose a large amount, or even all, of their investments. Without having a well-defined set of milestones, performance metrics, and independent auditors, an ICO is more of a gamble for investors.

To mitigate these issues, we propose **Milestone-Driven Funds Management** to protect investors. One of the major contributions of this system is the ability to refund investors' initial investments (partially) if project founders failed to complete milestone objectives. Voting procedures are used to determine whether the objectives of a milestone have been met or not.

Funds raised during token sales is first transfered from token sale contracts to **Milestone-Driven Funds Management Contract**, which then either (1) releases funds to project founders, or (2) refunds investors.

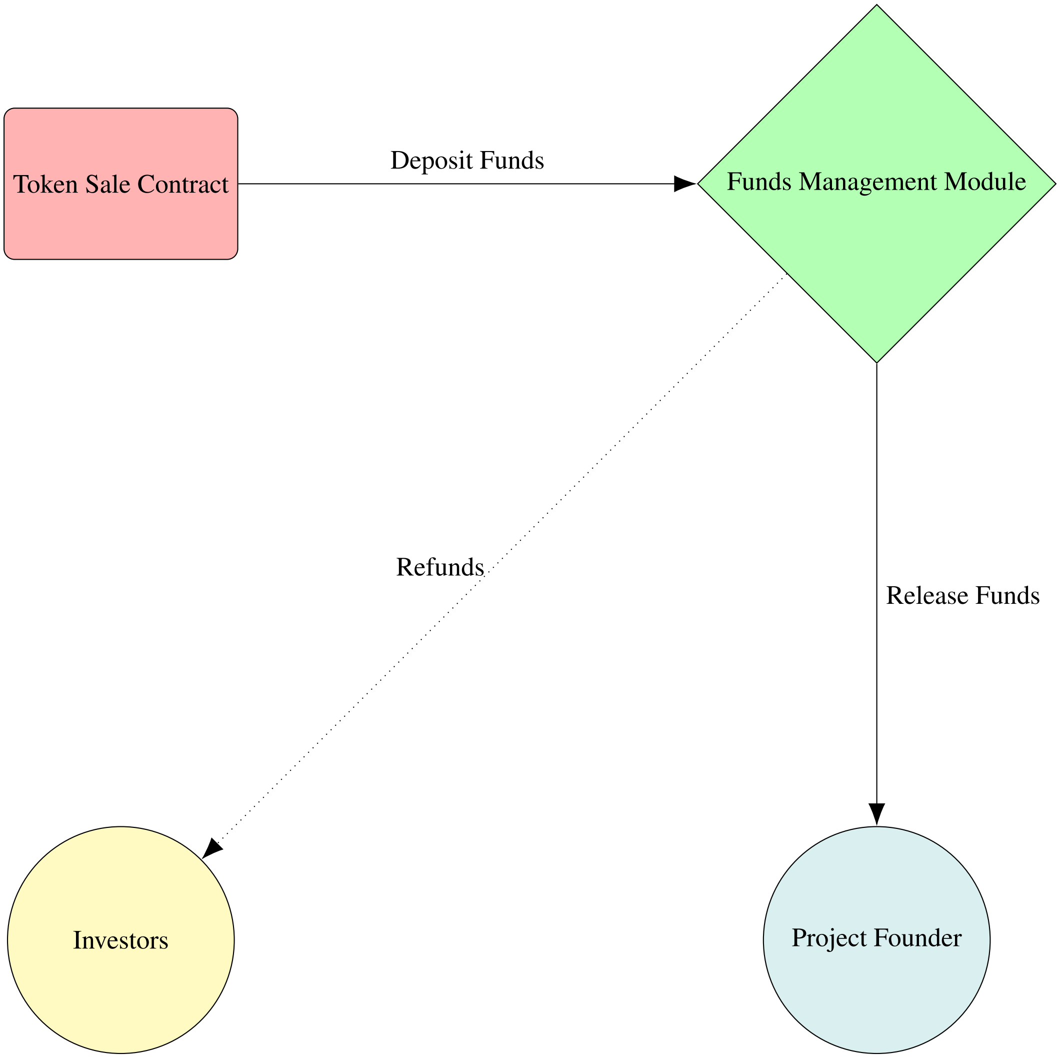

Figure 2: Funds Management Module Workflow

Usually, project founders have not only one, but multiple milestones inter-correlated with each other. Therefore, it is necessary to have a suitable data structure for milestones. We begin this section by looking at two types of data structures – linked lists and trees. Next, we will talk about how we model milestone states and state transitions with a finite-state machine in Section 2.3.

Before we start, it is necessary to have a few definitions:

Definition 2.1. *Milestone Node*
*A **milestone node** is represented by a tuple* $(\mathbf{d}, \mathbf{t_{ttc}}, \mathbf{p})$*, where* $\mathbf{d}$ *is a description of objectives,* $\mathbf{t_{ttc}}$ *(**Time-to-completion**) is the maximum amount of time required to complete theses objectives, and* $\mathbf{p}$ *is the percentage of total funds locked inside this milestone.*

Definition 2.2. *Investor*
*An investor is represented by an address that participated in a **Token Sale** of a blockchain project.*

2.1 Milestone Linked list

In the simplest form, each milestone node is composed of data and a reference (in other words, a link) to the next milestone node in the sequence in chronological order.

Example 2.1. *Example of a Milestone Linked List*

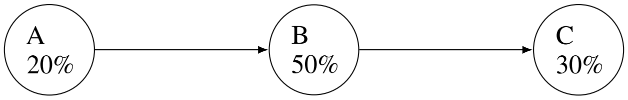

Figure 3: Example of a Milestone Linked List

Each milestone node points to the next milestone node. Percentage of total funds ($\mathbf{p}$) for a milestone is also shown below its label.

Suppose project founders failed to complete milestone B, then the funds of milestone B and all subsequent milestones will be returned to investors. In this example, 80% of total funds are refunded. The refund result is shown below:

Example 2.2. *Refunds of a Milestone Linked List*

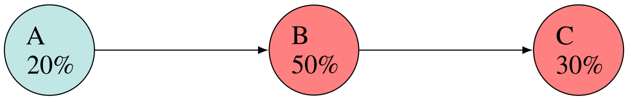

Figure 4: Refunds of a Milestone Linked List

Funds locked in nodes colored in red are returned to investors.

2.2 Milestone Tree

Instead of having a single timeline of milestones using a linked-list, with the help of tree structures, multiple parallel timelines can be easily constructed. To see the motivation of using tree structures, consider the following hypothetical blockchain project:

Example 2.3. *A Simple Project with 3 Milestones*

- *3 milestones: Smart Contract Implementation (A), Mobile Client (B), Desktop Client (C).*
- *Mobile Client interacts with the Smart Contract. B depends on A.*
- *Desktop Client interacts with the Smart Contract. C depends on A.*

Suppose developers failed to complete the objectives of milestone B, and milestone C is still in progress. This raises the question – "Should we stop funding this project?"

Suppose the completion of milestone A and C makes it a successful product in the market. Then we have two cases:

Case 1: The objectives of milestone C will be met.
Case 2: The objectives of milestone C will not be met.

The problem is that we cannot predict the future. If we made a wrong decision in any of these cases, investors may suffer a potentially large loss. In order to minimize investors' risks and maximize their profits, only the funds given to the failed-to-reach milestones and milestones depending on them should be returned to the investors, and we continue funding the remaining milestones. However, we cannot tell milestone dependencies in a milestone linked list. We argue that a tree structure is more appropriate. We call such a structure a ***Milestone Tree***. Figure 5 shows a milestone tree of the previous example:

In this milestone tree, funds of the subtree of milestone B (which includes B and all its descendants) are returned to the investors if the project founders fail to complete the objectives of milestone B, but we continue funding the subtree of milestone C.

Now, let us formally define a milestone tree:

Definition 2.3. *Milestone Tree*

A ***milestone tree*** *is a tree of milestone nodes. The root node $\mathbf{V_{root}}$ of the tree is a special milestone node where the* ***Time-to-completion*** *value is set to zero, the description of objectives is an empty string, and the percentage of total funds is set to be the amount of funds released to the project founders immediately after the project is funded.*

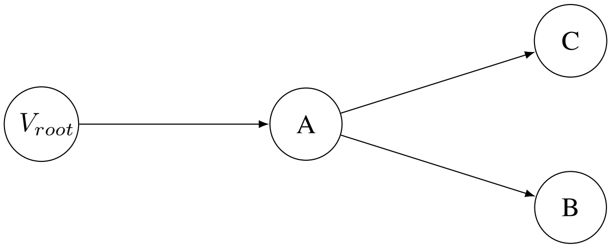

Figure 5: Milestone Tree

Definition 2.4. *Validity of a Milestone Tree*
A milestone tree is valid if

$$\forall_V \, \mathbf{V}.\mathbf{p} = \sum_{\mathbf{V}' \in descendants(\mathbf{V})} \mathbf{V}'.\mathbf{p} \tag{1}$$

where $\mathbf{V}.\mathbf{p}$ *denotes the* p *value of a milestone node,* $descendants(\mathbf{V})$ *is a set of descendants (i.e. A node reachable by repeated proceeding from parent to child) of node* $\mathbf{V}$*. It basically says that the funds given to the nodes of a subtree should sum to the funds given to that subtree.*

With the help of tree structures, we can easily create multiple parallel timelines of milestones that are independent of each other. A milestone tree with 6 milestones is given in the following example shown in Figure 6.

Example 2.4. *A More Complex Milestone Tree and Funds Distribution*

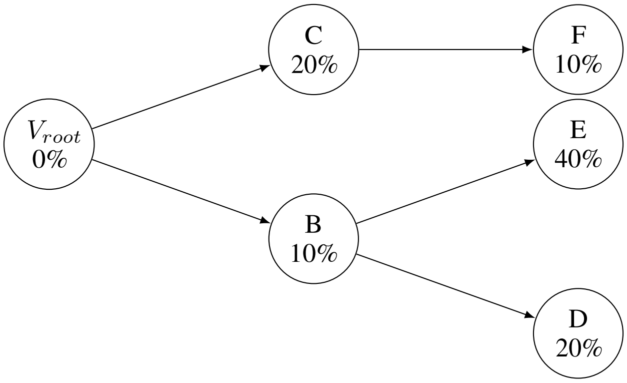

Figure 6: A More Complex Milestone Tree and Funds Distribution

Each milestone node is represented by a circle, labeled B,C,D,E,F. Percentage of total funds, $\mathbf{p}$ values are also shown for each node. Arrows represent dependencies.

Consider the milestone tree in Figure 7, suppose the objectives of milestones C and F are met, but project founders failed to complete objectives of milestone B. Then the funds of the subtree of B (i.e. node B, E and D) are returned to investors. In other words, 70% of total funds are refunded.

Example 2.5. *Example of Refunds*

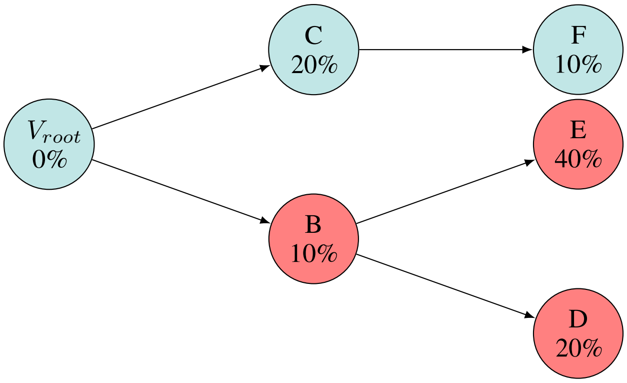

Figure 7: Example of Refunds

Funds locked in nodes colored in red are returned to investors.

2.3 Milestone States

Let t_{due} denote the due date of a milestone. W stands for 1 Week. A milestone node has the following states:

- **In Progress** (Starting state)

  Developers are working on this milestone. At $t_{due} - W$, **Voting Period 1** automatically starts and the milestone state changes to **Voting Period 1**.

- **Complete**

  This period begins immediately after **Voting Period 1** if investors have approved the decision to mark this milestone complete. Funds locked are (partially) released to the project owners (See more details in Section 4). Investors who voted 'Reject' are eligible for a refund.

- **Voting Period 1 (Voting for Completion)**

  Voting Period 1 automatically starts at $t_{due} - W$, and ends at t_{due}. Investors vote to decide if the milestone objectives are met. If not, project founders are provided with an option to initiate **Voting Period 2** and submit a milestone modification proposal (e.g. deadline extension, changes of milestone objectives). This voting period lasts for one week.

- **Voting Period 2 (Voting for Modifications)**

  Voting can be initiated by project founders at any time between $[t_{due}, t_{due} + W)$. In this voting period, the project owner proposes to modify the milestone tuple (d, t, p). If a majority of investors approve the modification proposal, the milestone's state is switched to **In Progress** and the milestone tuple is updated. Otherwise, the **Refund Period** automatically starts. This voting period lasts for one week and can only be *initiated once*.

- **Refund Period (Failure to Meet Milestone Objectives)**

  This period automatically starts at $t_{due} + W$ if a majority of investors vote for rejection in **Voting Period 1**, and the project founder forfeited the right to initiate **Voting Period 2**. Additionally, this period immediately begins if the modification proposal was rejected in **Voting Period 2**. Investors are able to withdraw funds locked inside the milestone before the period ends. The amount of funds refunded is proportional to the number of tokens (details can be found in Section 4) held by an investor. This period lasts for one week.

Let **IP**, **VP1**, **VP2**, **RP**, **C** denote state **In Progress**, **Voting Period 1**, **Voting Period 2**, **Refund Period** and **Complete** respectively. We represent the above state transitions with a finite-state machine:

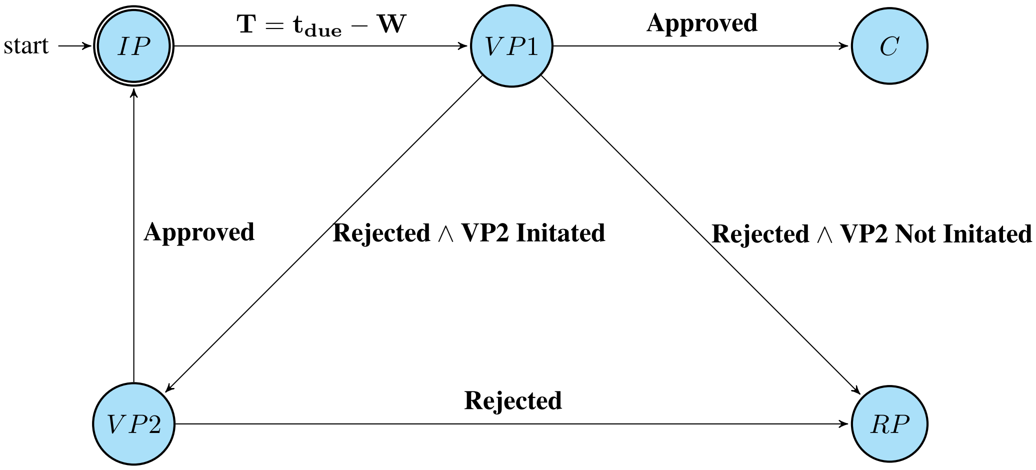

Figure 8: Workflow of Milestone-Driven Funds Management

3 Voting

The voting method is used to determine state transitions covered in Section 2.3. To prevent malicious manipulations by project founders, whales, and exchanges, project founders are required to issue two classes of tokens:

- Class A — Issued and sold to regular investors during a token sale. Held by regular investor with regular voting rights and eligibility for a refund.
- Class B — Not issued and sold to regular investors during a token sale, e.g., tokens held by project founders, or tokens unsold during a token sale. No voting rights and ineligibility for a refund.

There is *no difference* between Class A and Class B tokens in terms of their token functions. Class A tokens and Class B tokens are merged into **one type of tokens** when a project finishes.

Also, token holders are required to stake Class A tokens before **Voting Period 1** to be eligible to vote. The minimum staking length is set to 30 days. Specifically, in order to participate in a voting with X Class A tokens, token holders must send X Class A tokens to the staking contract controlled by the Milestone-driven Funds Management Module, and stake for at least 30 days before **Voting Period 1**. In the case of **Voting Period 2** voting, Class A tokens must be sent to the staking contract at least 30 days before its preceding **Voting Period 1** starts. The staked tokens are returned to investors after the voting (Specifically, tokens are returned when milestone state transits to either state **Complete**, state **Refund Period**, or state **In Progress**).

Specifically, a vote of an address **addr** is weighted by the number of tokens staked for at least 30 days. The voting weight W of an address for a voting is defined as

$$W(\text{addr}) = \frac{\text{Tokens Staked by the Address}}{\text{Total Tokens Staked for the Voting}} \tag{2}$$

The voting weight W is used to determine the final outcome of a vote. A decision is approved if a majority (>50%) of investors support it. **Investors who vote for a refund in a milestone will be provided an option to receive refund, regardless the final decision.** If more than one round of voting is held for a given milestone, the decision of an investor in the final round overwrites the previous round(s) to determine if the investor votes for a refund. The exact refund amount depends on the coverage investors purchased (see Section 5.1).

We are fully aware that 51% attacks and bribe attacks can still occur, thus, we follow the guidelines proposed by Vitalik Buterin ([12]). In the event of 51% attacks and bribe attacks to force a refund, project founders can simply organize another ICO, but with completed milestone nodes dropped. Furthermore, staked tokens will not be refunded after voting if investors choose to receive refund. Only Class A tokens are eligible for ETH refund.

4 Refund

Investors who wish to receive a refund must leave their staked project tokens in the **Milestone-Driven Funds Management Module**, during the **Refund Period** if a milestone is rejected, or **Complete** if a milestone is approved.

Both the **Refund Period** and **Complete** have two stages. The first stage lasts for 3 days and the second stage lasts for 4 days.

During **Stage 1** of the **Refund Period** and **Complete**, investors have the choice of retrieving their project tokens staked for the previous vote that led to this state, but they cannot deposit any more project tokens. The remaining unwithdrawn project tokens are taken as refund requests, and the maximum amount of the refund in ETH is the value of unwithdrawn project tokens in ETH, calculated using the project's average crowdsale price on Ventureum. This mechanism ensures that voting participation is a prerequisite for investors who wish to receive a refund.

Example 4.1. *Refund Calculation I*
Suppose the average crowdsale price of XYZ tokens is 1000 XYZ/ETH. Tom left 5000 XYZ tokens unwithdrawn during Stage 1. These XYZ tokens entitle Tom to a refund of maximum

$$5000\,\mathit{XYZ} \div 1000\,\mathit{XYZ/ETH} = 5\,\mathit{ETH} \tag{3}$$

During **Stage 2** of the **Refund Period** and **Complete**, the exact refund amount is determined for each address. Each address can request a cumulative refund up to its **Refund Coverage** defined in Section 5.1. Right now, we can simply regard it as a hard limit on the total amount of refund an address is able to retrieve.

Suppose an address has already received r_{pre} ETH worth of a refund, and its refund coverage is rc ETH. This implies it is entitled to receive up to $r_{max} = max(0, rc - r_{pre})$ ETH worth of a refund. Assuming this address has r' ETH worth of tokens left at the beginning of **Stage 2**, we only consider the **Effective Refund Amount** $r_{eff} = min(r_{max}, r')$ for refund calculations. Summing up r_{eff} for all addresses that participated in the previous vote gives us the **Total Effective Refund Amount** r_{tot}.

Example 4.2. *Refund Calculation II*
*Tom has a refund coverage of maximum 10 ETH. He has already received a refund of 6 ETH, then he is able to receive a refund up to $r_{max} = max(0, 10 - 6) = 4\,ETH$ this time. Now, at **Stage 2**, Tom has 5 ETH worth of XYZ tokens unwithdrawn. At most 4 ETH worth of XYZ tokens will be refunded since $r_{eff} = min(4, 5) = 4\,ETH$.*

The calculations of the exact refund amount in **Stage 2** of the **Refund Period** and **Complete** are slightly different. Assuming the amount of ETH locked in this milestone is X ETH, the amount of ETH locked in the milestone's subtree (which includes the milestone and all its descendants) is X_{sub}, and the approval rate is μ (e.g. $\mu = 0.6$ or 60%). We further assume that investors request a total effective refund of r_{tot} ETH. Now, we are ready to define our refund mechanism in each case.

4.1 Approval

In the event of approval, we transit to state **Complete**. At least μX ETH are released to project founders, and $(1 - \mu)X$ ETH are reserved for refunds. For each address, r_{eff} is automatically set to 0 if the address voted *Approve*. This ensures that only those who voted *Reject* will be provided with a refund.

- **Case 1** $r_{tot} \leq (1 - \mu)X$
  In this case, all refund requests are fully fulfilled. In other words, each address receives the entirety of its effective refund r_{eff}. $\mu X + [(1 - \mu)X - r_{tot}]$ ETH are released to project founders, where $[(1 - \mu)X - r_{tot}]$ represents refunds unwithdrawn after **Stage 2** ends.
- **Case 2** $r_{tot} > (1 - \mu)X$
  There are not enough funds to fulfill all refund requests. Therefore, we scale down each

effective refund amount by a factor of $\frac{(1-\mu)X}{r_{tot}}$. The extra project tokens received by the refund smart contract are returned to investors. Formally, $\bar{p}[r_{tot} - (1-\mu)X]$ extra project tokens are returned, where $\bar{p}$ denotes the average crowdsale price in ETH. μX ETH are released to project founders.

Example 4.3. *Refund Calculation III*
*Following Example 4.2, we assume $r_{tot} = 30$ ETH, $X = 100$ ETH and $\mu = 0.6$. We further assume we are at **Stage 2** of state **Complete**. Tom voted* Reject *against the milestone. Since $r_{tot} < (1-\mu)X$, Tom's effective refund amount is the entirety of his effective refund. Therefore, Tom receives a refund of 4 ETH.*

Using the same average crowdsale price in Example 4.1, or $\bar{p} = 1000$ XYZ/ETH, Tom submits 5000 XYZ (worth 5 ETH), but only 4000 XYZ (worth 4 ETH) are kept due to the limit of his refund coverage (4 ETH), the remaining 1000 XYZ are returned to Tom. The effective refund amount is 4 ETH or 4000 XYZ. These 4000 XYZ are taken by the system. In the end, Tom receives a refund of 4 ETH and 1000 XYZ. $\mu X + [(1-\mu)X - r_{tot}] = 70$ ETH are released to project founders.

Example 4.4. *Refund Calculation IV*
*Following Example 4.2, we assume $r_{tot} = 50$ ETH, $X = 100$ ETH and $\mu = 0.6$. We further assume we are at **Stage 2** of state **Complete**. Tom voted* Reject *against the milestone. Since $r_{tot} > (1-\mu)X$, Tom's effective refund amount is scaled down to $r_{eff} \cdot \frac{(1-\mu)X}{r_{tot}} = 4 \times \frac{(1-0.6)\cdot 100}{50} = 3.2$ ETH. Therefore, Tom receives a refund of 3.2 ETH.*

Using the same average crowdsale price in Example 4.1, or $\bar{p} = 1000$ XYZ/ETH, the number of tokens returned is $\bar{p}(r_{eff} - 3.2) = 1000 \times (4 - 3.2) = 800$ XYZ. In short, Tom submits 5000 XYZ (worth 5 ETH), but only 4000 XYZ (worth 4 ETH) are kept due to the limit of his refund coverage (4 ETH), the remaining 1000 XYZ are returned to Tom. The effective refund amount is 4 ETH or 4000 XYZ. Of these 4000 XYZ, only 3200 XYZ (worth 3.2 ETH) are taken by the system due to the scaling factor, and 800 XYZ (worth 0.8 ETH) are returned. In the end, Tom receives a refund of 3.2 ETH and $1000 + 800 = 1800$ XYZ. $\mu X = 60$ ETH are released to project founders.

4.2 Rejection

In the event of rejection, we transit to state **Refund Period**, and X_{sub} ETH are reserved for refunds.

- **Case 1** $r_{tot} \leq X_{sub}$
  In this case, all refund requests are fully fulfilled. In other words, each address receives the entirety of its effective refund r_{eff}. The remaining funds (if there is any) are donated to the Ethereum Foundation.
- **Case 2** $r_{tot} > X_{sub}$
  There are not enough funds to fulfill all refund requests. Therefore, we scale down each effective refund amount by a factor of $\frac{X_{sub}}{r_{tot}}$. The extra project tokens received by the refund smart contract are returned to investors. Formally, $\bar{p}[r_{tot} - X_{sub}]$ extra project tokens are returned, where $\bar{p}$ denotes the average crowdsale price in ETH. The funds locked inside the milestone are fully refunded in this case.

Example 4.5. *Refund Calculation V*
*Following Example 4.2, we assume $r_{tot} = 60$ ETH and $X_{sub} = 80$ ETH. We further assume we are at **Stage 2** of state **Refund Period**. Since $r_{tot} < X_{sub}$, Tom's effective refund amount is the entirety of his effective refund. Therefore, Tom receives a refund of 4 ETH.*

Using the same average crowdsale price in Example 4.1, or $\bar{p} = 1000$ XYZ/ETH, Tom submits 5000 XYZ (worth 5 ETH), but only 4000 XYZ (worth 4 ETH) are kept due to the limit of his refund coverage (4 ETH), the remaining 1000 XYZ are returned to Tom. The effective refund amount is 4 ETH or 4000 XYZ. These 4000 XYZ are taken by the system. In the end, Tom receives a refund of 4 ETH and 1000 XYZ. $X_{sub} - r_{tot} = 20$ ETH are donated to the Ethereum Foundation.

Example 4.6. *Refund Calculation VI*
*Following Example 4.2, we assume $r_{tot} = 100$ ETH and $X_{sub} = 80$ ETH. We further assume we are at **Stage 2** of state **Refund Period**. Since $r_{tot} > X_{sub}$, Tom's effective refund amount is scaled down to $r_{eff} \cdot \frac{X_{sub}}{r_{tot}} = 4 \times \frac{80}{100} = 3.2$ ETH. Therefore, Tom receives a refund of 3.2 ETH.*

Using the same average crowdsale price in Example 4.1, or $\bar{p} = 1000\,XYZ/ETH$, the number of tokens returned is $\bar{p}(r_{eff} - 3.2) = 1000 \times (4 - 3.2) = 800\,XYZ$. In short, Tom submits 5000 XYZ (worth 5 ETH), but only 4000 XYZ (worth 4 ETH) are kept due to the limit of his refund coverage (4 ETH), the remaining 1000 XYZ is returned back to Tom. The effective refund amount is 4 ETH or 4000 XYZ. Of these 4000 XYZ, only 3200 XYZ (worth 3.2 ETH) are taken by the system due to the scaling factor, and 800 XYZ (worth 0.8 ETH) are returned. In the end, Tom receives a refund of 3.2 ETH and $1000 + 800 = 1800\,XYZ$.

Note that all project tokens left unwithdrawn after **Stage 2** are burned. All refunds (ETH) left unwithdrawn after **Stage 2** of state **Complete** are released to project founders. All refunds (ETH) left unwithdrawn after **Stage 2** of state **Refund Period** are donated to the Ethereum Foundation.

5 Ventureum Network Token (VTH) - An ERC20 Token to Enable Refund Coverage

Ventureum Network Token (VTH) gives its holders refund coverage in the Milestone-driven Funds Management module.

5.1 Coverage for Refund

Specifically, to receive a refund up to X ETH, project token owners needs to stake (lock in) at least 0.25X ETH worth of VTH on the **Milestone-driven Funds Management Module** until the project finishes (or fails). One can send VTH to our staking smart contract at any time during a project's lifecycle, and the extra coverage immediately applies from that time onwards. The only exception is that after **Stage 2** of the **Refund Period** or after **Stage 2** of **Complete** of a milestone begins, the VTH tokens sent to our staking smart contract are not used to obtain refund coverage for the given milestone. We take 1% of staked VTH tokens out as fees, which helps us run and improve our platform. The rest of the staked VTH tokens are returned to investors at the end of a project's lifecycle, see Section 5.2 for details.

We determine the number of VTH required to cover X ETH of investment by using an **Approximate VTH Price** $\hat{p}$, which is the 30-day Exponential Moving Average price of VTH in ETH. Initially, it is set to the average price during the VTH crowdsale. Later, $\bar{p}$ will be updated daily with the mean value of VTH's price across multiple exchanges. The details of calculation and data points used will be published on our website.

Formally, covering up to **X** ETH requires staking

$$0.25\hat{p}X \ \text{VTH} \tag{4}$$

Example 5.1. *Coverage Calculation I*
*For example, Alice wants to cover up to 100 ETH of her investment in a project and the current **Approximate VTH price** $\hat{p}$ is 50 VTH/ETH. She will need to stake*

$$0.25 \times 50\,\mathit{VTH/ETH} \times 100\,\mathit{ETH} = 1250\,\mathit{VTH} \tag{5}$$

We need to be aware that investors hold project tokens instead of ETH. We calculate the value (in ETH) of a project's tokens based on the average crowdsale price.

Example 5.2. *Coverage Calculation II*
Suppose, Alice would like to cover up to 1000 XYZ tokens, and the average XYZ token crowdsale price is 10 XYZ/ETH, which implies that 1000 XYZ tokens worth

$$\frac{1000\,\mathit{XYZ}}{10\,\mathit{XYZ/ETH}} = 100\,\mathit{ETH} \tag{6}$$

*Assuming the same **Approximate VTH Price** as in Example 5.1, from Equation (4), we get that 1250 VTH must be staked to cover 1000 XYZ. In the even that the XYZ project promises are not met, Alice will receive a refund of 100 ETH.*

5.2 VTH Refund Schedule

Staked VTH tokens are released back to investors in a continuous manner. This approach mitigates the impact and risk brought by the potential intense fluctuations in the VTH market price at the end of a project's lifecycle.

Initially, 10% of VTH tokens are released at one time, after which VTH tokens are released continuously with a uniform distribution across 30 days.

Suppose Bob staked X VTH tokens, he receives $0.1X$ from his first withdraw request. Also the timestamp of the most recent withdraw is recorded. In the subsequent withdraws, bob receives

$$min(\frac{\text{Current Time} - \text{Last Withdraw Time}}{\text{30 Days}} \cdot 0.9X, \text{Number of VTH Unwithdrawn}) \tag{7}$$

Example 5.3. *VTH Withdraw Example*
Bob stakes 100 VTH in the XYZ project. At the end of the project's life cycle, Bob first withdraws $100\,\mathit{VTH} \cdot 10\% = 10\,\mathit{VTH}$ *on* ***Day 1****. Then Bob sends another request to withdraw funds on* ***Day 16****. Bob receives*

$$\min(\frac{16-1}{30} \times 0.9 \times 100\,\mathit{VTH}, 90\,\mathit{VTH}) = \min(45\,\mathit{VTH}, 90\,\mathit{VTH}) = 45\,\mathit{VTH} \tag{8}$$

Next, Bob withdraw his remaining funds on ***Day 31***

$$\min(\frac{31-16}{30} \times 0.9 \times 100\,\mathit{VTH}, 45\,\mathit{VTH}) = \min(45\,\mathit{VTH}, 45\,\mathit{VTH}) = 45\,\mathit{VTH} \tag{9}$$

In summary, Bob receives 10 VTH on Day 1, 45 VTH on Day 16, and another 45 VTH on Day 31.

Investors need to send at least two withdraw requests in order to get their VTH fully refunded. It is beneficial to send the first withdraw request as soon as possible, since it reduces the total time to get a full VTH refund. This extra complexity is due to the limitation of the Ethereum Virtual Machine.

If investors wish to transfer the staked VTH tokens to another project for refund coverage, they can transfer the entire balance of their staked VTH tokens in one request at the end of the current project's lifecycle. The other project must have an address of funds management module verified by the Ventureum team.

5.3 VTH Token Allocation

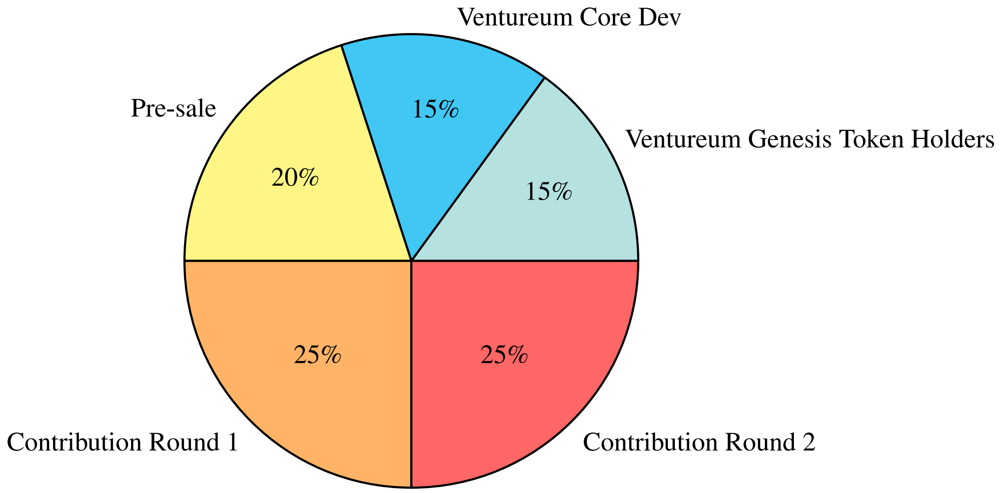

Figure 9: VTH Token Allocation

- 1,000,000,000 (1 Billion) VTH will be created. The total supply of VTH is fixed.
- 70% of VTH tokens are created during the Contribution Period (Pre-sale, Contribution Round 1 and 2) and allocated to public contributors. Tokens will be available for trading 24

hours after the conclusion of Contribution Round 1. They are Class A tokens with voting rights as well as refund coverage for the Ventureum project (see more details in Contribution Round 2 in Section 5.4).

  The Ventureum team promises to use a reasonable portion of the funds raised during Pre-sale and Contribution Round 1 for development and operation. The unspent portion (if exist) will be deposited into the **Milestone-driven Funds Management Module** together with the funds raised during Contribution Round 2 to provide extra refund for investors if they decide to pull out their investment.

- 15% VTH is contributed to founders and developers fund to compensate the current and future efforts on Ventureum and to attract new talent on board, locked in a smart contract with a 6-month vesting period which starts at the end of Contribution Round 2. They are Class B tokens that can be used to obtain refund coverage for other projects.
- 15% VTH is reserved for Ventureum Genesis Token (VGTH) holders.

Class A and Class B VTH tokens are merged into one type of tokens once investors approve the completion of the beta release of the Ventureum project in the **Milestone-driven Funds Management Module** (see more details in Section 8).

5.3.1 Ventureum Genesis Token

We introduce Ventureum Genesis Token (VGTH), an ERC20 token that will be issued before the Contribution Period begins, and that will be redeemable for Class B VTH tokens after the conclusion of Contribution Round 1.

VGTH will be issued to our community members and contributors who support us with feedback, critiques, and contributions to our source code and marketing.

5.4 Contribution Period Details

Three crowdsales will be organized around smart contracts running on Ethereum.

- **Pre-sale (TBD)**
  The pre-sale allows investors to contribute early, before the actual opening date of the official token sale. Maximum of 20% of VTH (or 200,000,000 VTH) are offered during the pre-sale, the remaining unsold VTH will be fully distributed among Contribution Round 1 and 2. Price per token is set to 0.0001 ETH and is guaranteed to be 50% or less of Contribution Round 1 and 2's token price.
- **Contribution Round 1 (TBD)**
  This is the first round of the official ICO. 25% of VTH (250,000,000 VTH) will be offered.
- **Contribution Round 2 (TBD)** This is the second round of the official token sale. The Milestone-driven Funds Management Module is responsible for managing funds via voting described in Section 2. 25% of VTH (250,000,000 VTH) will be offered.

5.5 Specific Issues Related to Voting and Refund for the Ventureum Project

All Class A VTH token owners will have voting rights and refund coverage in the **Milestone-driven Funds Management Module** for the Ventureum project as long as they stake VTH for at least 30 days before a voting period. Due to the special circumstances of using VTH tokens to steer the direction of the Ventureum project development, VTH tokens are treated as regular tokens in the **Milestone-driven Funds Management Module**, except that refund coverage is always 100% with no staking required. Furthermore, the VTH refund schedule (Section 5.2) does not apply.

6 Official Endorsement from the Ventureum Team

The Milestone-Driven Funds Management Module is completely open source, and any blockchain projects can choose to integrate it into their own project by themselves without the Ventureum dev team's endorsement. In this case, the following rules apply:

- The use of the word "Ventureum", the Ventureum logo, and any mention of the Ventureum project or team are strictly prohibited. The Ventureum team reserves all the rights to take legal actions against anyone who violates this rule.
- The use of VTH tokens is optional.
- The Milestone-Driven Funds Management Module will provide an option to turn off the use of VTH tokens and set the refund coverage for all investors automatically to infinity.

If the founders of a blockchain project want the endorsement from the Ventureum team on their usage of the Milestone-Driven Funds Management Module, the following rules apply:

- The Ventureum team will perform audits and security reviews of the source codes of the Milestone-Driven Funds Management Module they deploy.
- The founders are required to issue and strictly distinguish between Class A and Class B project tokens defined in Section 3 to comply with the requirements of the Milestone-Driven Funds Management Module. The Ventureum team will oversee their compliance with this requirement.
- The use of VTH tokens for refund coverage is mandatory.
- Results of audits and security reviews are published and stored in blockchain. A DApp will be developed for viewing these results.

7 Extra Funding via Staking

In this section, we introduce a staking mechanism to provide extra funds to project founders, which also implies extra refunds for investors in the event of project failure. The decision to activate this part lies with project founders. **Casper** is a planned future change in the way the Ethereum network forms distributed consensus and is primarily aimed at reducing energy waste. Casper achieves this goal by using a consensus mechanism called **Proof of Stake**. Extra funds comes from "staking" – earning interest by locking up funds for a predetermined amount of time. Specifically, the Milestone-Driven Funds Management Module's smart contract deposits funds into the **Casper Contract**, and withdraws funds when a milestone state is in either the **Refund Period** or **Complete**. Meanwhile, interest earned by staking funds is transferred back to the smart contract.

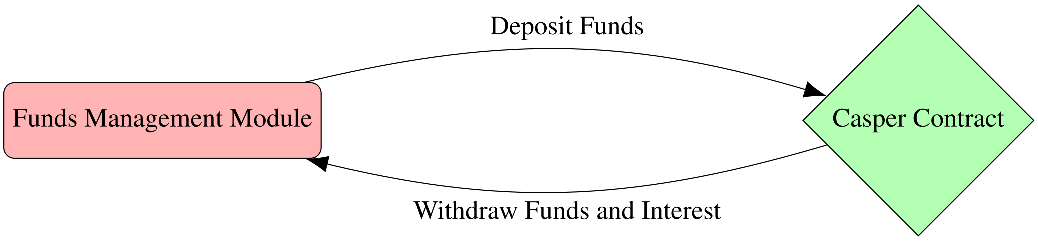

Figure 10: Interaction Between the Funds Management Module and the Casper Contract

7.1 Funds Staking Schedule

We will briefly introduce how the Funds Management Module interacts with the Casper Contract. It is important to note that Casper is not a finished protocol and is still under heavy development. The Ventureum development team will keep a very close eye on the Casper development process and react accordingly.

Assuming the minimum time for staking allowed by Casper is M months. To simplify the problem, we require $\mathtt{t_{ttc}}$ (**Time-to-completion**) of a milestone to be larger than $M + 3W$, where W represents 1 week.

Note that the above restriction for $\mathtt{t_{ttc}}$ does not apply if project founders wish not to receive extra funds.

Suppose we are at milestone node V, and assume that the milestone node enters the **Refund Period** or **Complete** at time T, which can be between $[t_{due}, t_{due} + 3W)$. For a children milestone node V' of V, we denote its due date as t'_{due} and let the T' be the time when V' enters the **Refund Period** or **Complete**, which can be between $[t'_{due}, t'_{due} + 3W)$. The amount of funds $F^{V'}_{deposit}$ deposited for the subtree of V' (which includes the node V') at time T is the total amount of funds given to the subtree, in other words

$$F^{V'}_{deposit} = \sum_{\tilde{V} \in descendants(V')} \tilde{V}.p \tag{10}$$

The staking duration for the funds $F^{V'}_{deposit}$ is set to $t'_{due} - T$. Now, we claim that we will able to withdraw $F^{V'}_{deposit}$ at the time T'.

Proof. To prove it, we argue that $t'_{due} - T$ (The staking duration) is always larger or equal to M (Minimum staking duration required). Also, we will show that the duration between T and T' is larger than the staking duration.

$$\begin{aligned} t'_{due} - T &\geq t'_{due} - (t_{due} + 3W) \\ &= t'_{due} - t_{due} - 3W \\ &= t'_{ttc} - 3W \end{aligned} \tag{11}$$

Note that $t'_{ttc} \geq M + 3W$ by our assumption. Thus

$$t'_{ttc} - 3W \geq M + 3W - 3W = M \tag{12}$$

We know that $T' \geq t'_{due}$, therefore, $T' - T \geq t'_{due} - T$ which completes the second part of the proof. □

If the milestone node V' passed **Voting Period 2**, and project founders postponed the due date to $t^{\dagger}_{due}$, then it is obvious that $t^{\dagger}_{due} - T \geq t_{due} - T$. Thus the above stacking schedule still applies in this situation.

8 Ventureum Development Roadmap

8.1 A Brief Review of the Ventureum Team's Work

The core dev team of Ventureum is a group of software engineers and data scientists who have worked at companies such as Amazon, Bank of China (Toronto) and GE. We are also University of Waterloo alumni, blockchain tech enthusiasts and active cryptocurrency investors. We have followed the evolution of Ethereum since its very beginning, when Vitalik Buterin released the whitepaper for the Ethereum crowdsale. When ICOs entered a state of frenzy in the beginning of 2017, we started to notice two serious phenomena that plague Ethereum, and even the whole cryptocurrency universe, in both the short term and the long term. The first one is the network congestion on the Ethereum blockchain when a hot ICO opens to the public, which causes both frustration for investors and price fluctuations of ETH. The second one is the proliferation of scam ICOs. The true blockchain tech enthusiasts and proponents of Ethereum have been looking for a solution to both issues since then.

The core team of Ventureum started to form between late April and early May 2017. Since then, we have been intensively doing research, forming ideas, and collecting feedback to devise a solution to both aforementioned issues. In June 2017, our team envisioned a milestone-driven funds management mechanism, implemented with the very spirit of Decentralized Autonomous Organization (DAO), that would be essential for protecting the interest of blockchain project investors in the long run. We started the drafting of the whitepaper in late June 2017.

8.2 Alpha Release (Minimum Variable Product, Q1 of 2018)

The Ventureum development team endeavors to achieve the following core features to enable the basic functionality of the Ventureum protocol.

Development on the Ethereum blockchain:

Smart contracts that implement milestone-driven funds management workflow, including key features:

- The investment in the form of ETH committed by the investors during a crowdsale will be distributed and locked in the milestones defined and published by the blockchain project founders by smart contracts.
- Smart contracts that implement the workflow of milestone-driven funds management as depicted in Figure 8. A simple rephrase of the workflow is presented: "The investment locked in milestones can only be released to project founders if milestone deliverables have been delivered on time. Otherwise, investment is returned to investors." Specifically, if the majority of investors (weighted by their proportion of investments) voted **Approve** for a milestone, the investment locked in the given milestone will be released to the project founder. If the voting was unfavorable for a milestone, the investment locked in the given milestone, and all the milestones that require the given milestone as a prerequisite, will be refunded to the investors. The workflow also allows the project founder to initiate **Voting Period 2** if the due date of a milestone has passed with the milestone objectives unrealized, and the project founder wants the consent from the investors to postpone the due date and/or modify milestone objectives.
- Smart contracts that implement Class A token staking.
- Smart contracts that implement VTH token staking.

Frontend development:

Two web user interfaces will be developed.

- A web user interface for project founders with the following functionalities:
  - Define project milestones' objectives and allocate (a percentage of) funds to them.
  - Deposit the funds(ETH) raised in a token sale into the Milestone-driven Funds Management module.
  - Initiate **Voting Period 2** for a milestone.
  - Access to voting results.
  - Withdraw funds (ETH) if milestone deliverables have been delivered on time. That is, investors have approved the decision to release funds to project founders in **Voting Period 1**, in which case the milestone is in state **Complete**.
- A web user interface for investors with the following functionalities:
  - Send Class A tokens for staking to obtain voting rights or for a refund.
  - Send VTH tokens to obtain refund coverage.
  - Vote (**Approve** or **Reject**) during **Voting Period 1** or **Voting Period 2** for a milestone.
  - Access to voting results.
  - View the balance of funds (ETH) they can withdraw.
  - Withdraw funds when refund criteria are met (such as a milestone reaches **Refund Period** state or the investor voted **Reject** with VTH tokens staked).
  - Withdraw Class A project tokens after **Voting Period 1**, **Voting Period 2** or during **Stage 1** of **Complete** or **Refund Period**.
  - Withdraw VTH tokens at the end of a project's lifecycle.

8.3 Beta Release (Q2 of 2018)

Development on the Ethereum blockchain:

- Smart contracts that implement the fund staking schedule.

- A mock **Casper contract**. The Ventureum team is fully aware that the exact time of the release of **Serenity** (the 4th release of Ethereum), on which the **Casper contract** runs, is highly unpredictable. To ensure a smooth development of the smart contracts that implement the fund staking schedule, a mock **Casper contract** will be developed to provide the underlying infrastructure. The mock **Casper contract** will only mimic the functionality and features of the **Casper contract** that are essential for running the fund staking module.

Frontend development:

- A web service to send notification emails will be developed to implement the following features:
  - Notify project founders that the due date for a milestone is approaching.
  - Notify project founders if they can initiate a **Voting Period** in the near future.
  - Notify project founders of the result of a vote.
  - Notify project founders of the balance of funds they can withdraw.
  - Notify investors of a pending **Voting Period 2**.
  - Notify investors of the result of a vote.
  - Notify investors of the balance of funds (ETH) they can withdraw.
  - Notify investors of the balance of Class A tokens they can withdraw.
  - Notify investors of the balance of VTH tokens they can withdraw.

References

[1] Wikipedia. Initial Coin Offering.
https://en.wikipedia.org/wiki/Initial_coin_offering, .

[2] Smith and Crown. Smith and Crown Research.
https://www.smithandcrown.com/research-search/.

[3] John Koetsier. ICO Startups.
https://www.inc.com/john-koetsier/ico-bubble-startups-are-raising-hundreds-of-millio.html.

[4] ICOrating. ICOrating.
http://icorating.com/.

[5] CHAINANALYSIS. The Rise of Cybercrime on Ethereum.
https://blog.chainalysis.com/the-rise-of-cybercrime-on-ethereum/.

[6] Wikipedia. OneCoin.
https://en.wikipedia.org/wiki/OneCoin, .

[7] Garrett Keirns. GemCoin.
https://www.coindesk.com/gemcoin-ponzi-scheme-operator-hit-74-million-judgment/.

[8] Matthew Di Ferrante. Towards Responsible Token Sales (ICOs).
https://medium.com/@matthewdif/towards-responsible-token-sales-icos-291e69cc9ccf.

[9] SEC. SEC Rule on ICO.
https://www.sec.gov/oiea/investor-alerts-and-bulletins/ib_coinofferings.

[10] MAS. MAS Rule on ICO.
http://www.mas.gov.sg/News-and-Publications/Media-Releases/2017/MAS-clarifies-regulatory-position-on-the-offer-of-digital-tokens-in-Singapore.aspx.

[11] Cointelegraph. Chinese Government Eyes ICO Crackdown Under New "Illegal Financing" Rules.
https://cointelegraph.com/news/chinese-government-eyes-ico-crackdown-under-new-illegal-financing-rules.

[12] Vitalik Buterin. 51% Attacks and Bribe Attacks.
https://medium.com/@VitalikButerin/when-i-see-voting-games-i-usually-analyze-i-51-attacks-and-ii-bribe-attacks-looking-here-da7412a4a217.