

PLATFORM KITA

Profitshare From Ethereum Unique Token & Projects
v0.0.1

The Problem

- Too many unused coin/token without value added function
- Too expensive & complicated to release security token
- For investor: developer miss roadmap target & resulted as scam after initial coin offering
- For developer: too many pressure from investor using the project for speculative gain
- For non-technical user: too expensive to release a token & sell them to crypto world

The Solution: Platform KITA

is a decentralized application built on top ethereum blockchain with two main products:

Token Factory & Project Factory

Token Factory

Decentralized application to create ERC20 token with unique features: burnable & profit sharing

why this token better than ordinary coin/token:

1. Burnable as utility token
When supply decreased, demand will automatically increase
2. On chain profit sharing token
Token holder will receive profit share proportional to total token
3. Combination of burn & profit sharing effect
when a token get burned, its total supply will be reduced. indirectly reduce number token to receive profit. Its mean increased profit share for every token.
4. Integrated to profit share smart contract
A decentralized application can send its profit/revenue to profit share manager smart contract to ensure trust-less profit sharing
5. Integrated to rest API for payment gateway.
By connected to a payment gateway, a non blockchain application/website will be guaranteed to share its profit to token holder

cost: 1 ETH & 1% token

Project Factory

Decentralized application to create crowd funding or ICO, that fair & trust-less for every stake holder (developer & investor)

Project factory will create ICO smart contract with these features:

- automatically refund on unreachable softcap

- on successful ICO, token will be created & distributed automatically
- token created will have features of token factory token: burnable & profit sharing
- ICO fund will be locked & managed by smart contract
- ICO fund will be released step by step over time
- Roadmap review system & automatic buyback

Roadmap review system is a system to ensure developer to finish their roadmap & investor to support/abandon the project elegantly. It will work like these:

- for every 3 months after ICO ended, there will be one week window for development review.
- on this week ICO fund will not be released to developer. Investor/token holder will have chance to review development progress.
- if an investor give negative review. he will have chance to cut loss his investment, by refund his token. Just sell the token back to smart contract
- if an investor give positive review & smart contract have enough token. He can buy more token on lower price to support the project.
- remaining token in smart contract after review week will be burned. these will make good effects: increase demand for token already in investor hand & increase profit share value for every token.

All features will be run automatically by ethereum smart contract without human intervention/manipulation.

Project Factory cost:

- 10 ETH & 5% token created by project factory
- refundable if ICO failed to reach softcap

MVP (Minimum Viable Products)

Platform KITA will consist of three parts:

- Ethereum + ERC20 token wallet
 1. user friendly because no need use metamask plugin or mist browser to execute smart contract
 2. connected to ethereum blockchain via infura. no need to download entire eth blockchain
- Projects sections. all main features of platform KITA can be found here:
 1. token factory
 2. project factory
 3. pyramid prize: demonstrate profit sharing function
 4. king of tokens: demonstrate utility / burned function
 5. faucet of demo token
- Profitshare manager. Interface to profitshare smart contract, will have 2 functions:
 1. check profit from any token created by token factory
 2. withdraw profitshare in eth

About Me

Platform KITA initialized by Tono. 10 years exp in software development + 1 year exp in blockchain.

Visit <https://ethdevid.github.io/> and join our telegram group