

MemeCompetition Software Design Document

by: house _ 11/25/24

Contents

Objective

MemeCompetition Use Case & Features

Why memes?

What makes this solution unique?

What makes this solution unique (technical integration)?

Customer Base

How is the MemeCompetition revenue distributed?

What else can the tech. do?

Launch Schedule (TBD)

Why PulseChain?

Supporting metrics & research

Objective

Create a decentralized smart contract solution that facilitates the ability for people to compete in meme competitions w/ monetary reward based incentive for repeat business. The monetary incentive will be sourced from one of two user bases, creating a prize pool. The venture profits will be sourced as a small percent of these prize pools, similar to how DEXs (decentralized exchanges) source their profits.

MemeCompetition Use Case & Features

The use case is fairly straight forward. One set of users can use the smart contract (dapp) to create new meme competitions. Then another set of users simply create meme images to enter those competitions. Then a vote occurs & rewards are distributed to both voters and winners.

There are 2 types of user bases.. Competition “Voters” & “Meme Creators”

Consolidated work-flow process (the gist)

- 1) Any “Voter” may create a competition
 - they choose any topic
(ie. “best meme to support \$BEAR on PulseChain”)
 - they set a USD amount entry fee
(ie. \$10.00)
 - they set a submission time frame + voting time aloud
(ie. 1 week to submit memes + 24hrs for voting)
- 2) The public may freely view a list of open competitions on the dapp
 - “Meme Creators” choose one to participate in & submit a url link to their meme
 - the url may be from any social media or HTTP server in the world, etc.
 - they must also pay the entry fee along with their submission
 - can pay in any ERC20 token (amnt value must = the USD entry fee amnt)
- 3) “Voters” use the dapp to view all meme submissions for all ‘pending’ competitions
 - pending = submission time passed + voting time started
 - note: only voters are aloud to do this (not ‘meme creators’ or the public)
- 4) When any competition submission time has lapsed ...
 - “Voters” may then choose to vote for a winner
- 5) After voting is complete for any given competition (ie. the competition is now ‘closed’)
 - the full competition becomes available for the public to see
(including both winning & losing memes)
 - “Voters” then use the dapp to claim their rewards for voting
rewards: small percent of prize pool + minted voter tokens
 - “Winning creators” then use the dapp to claim their rewards for winning
rewards: large percent of prize pool + minted voter tokens
+ meme minted into an NFT

MemeCompetition Use Case & Features (cont'd)

Tokenomics & Value set offerings (EOA = externally own address; wallet)

Voter Status Description

- 1) your EOA must currently hold a 'certain amount' of voter tokens
 - this may indeed be freely purchased from the open market
- 2) your EOA must have won a previous competition in the past
 - note: this is a 'non-transferable' value set
- 3) EOAs DO NOT retain 'voter status' if ...
 - 1) they do not currently own that 'certain amount' of voter tokens
 - 2) they have not won a competition in the past

Reward Breakdowns (passive & active)

- 1) competition prize pools (ie. collective entry fees) ...
 - are held by the contract as USD stable
- 2) passive rewards (for winners)
 - any EOA that actively possess a minted NFT, will earn a small % of each competition prize pool
 - note: winning a competition is NOT required for this (ie. provides real world value & incentive for holding & trading our NFTs)
- 3) active rewards (for winning + voting)
 - winners get minted 'some amount' of voter token for each competition won
 - winners earn a large % of price pool won
 - winners get their meme minted into an NFT
 - voters get minted 'some amount' of voter token for each competition vote
 - voters earn a small % of each prize pool they voted in
 - note: the 'competition maker' earns an extra small % of that prize pool

Why memes?

Our market research shows us that memes are now a \$6B industry and growing by a billion dollars every year over the past 4 years.

75% of people between the ages of 13 to 36 engage with memes every day.

This is about 52 million Americans per day.

Memes are just the start. There are many industries that can be changed with the ability to combine DAO voting with 'hashed' data (discussed below). However, we feel memes represent the most efficient target industry with the highest potential for early monetization. We see it as the easiest way to get people to start using the smart contract, quickly, and in masses. (further details outlined in 'Customer Base' below)

What makes this solution unique?

The meme competitions will be fully automated and require little oversight aside from active marketing & manual contract parameter updates, in order to facilitate business logic in support of demand for participation (ie. raising/lowering profit & reward margins to cater towards user adoption and acceptance of the solution; especially when additional competitors to our solution arise on the open market)

NOTE: there will be NO administrative control (NO ADMIN KEYS) in regards to winner selection or reward distribution (including minted rewards)

Primary rewards are offered in direct USD stable, sourced from the user entry fees. This allows for a unique incentive to support initial user acquisition with direct monetary value, as opposed to speculative value like legacy solutions that merely mint worthless tokens.

However, secondary rewards are indeed offered in the form of tokenization.

There are 2 types

- 1) ERC20 token distribution
- 2) ERC721 non-fungible token (NFT) distribution.

Utilizing both ERC20 & ERC721 tokens will present an opportunity for all meme competition winners to access additional features (ie. voting), as well as passively earn additional income from all future competitions, w/o requiring additional work or entry fees.

What makes this solution unique (technical integration)?

The secret sauce lies in how the images are stored and referenced within smart contracts

Traditional solutions: they store web links inside the smart contract, which actually creates a 'centralized' solution. This means, that if/when those web links are down, everyone loses the ability to access their NFTs (they can't see their images). This flaw can easily be seen on every NFT marketplace as their images randomly fail to display for various NFT smart contracts throughout the day.

Our solution: we store image 'hashes' and integrate a comparison algorithm. This allows images to be stored anywhere on the internet, and simply be 'validated' during sales or transfers of ownership. This creates a 'decentralized' solution that can only fail if the entire internet goes down.

Customer Base

early adopters: Meme addicts & hobbyists.

Our market research shows us that creating and sharing memes is a top rising industry, currently valued at over \$6B, with 75% of early to middle-aged Americans engaging every day. Most of these people are not getting paid or earning recognition for their work.

Considering these numbers, we see a neglected market that is waiting to take off. We see meme addicts & hobbyists as an excellent opportunity for 'quick & mass adoption' in utilizing our smart contract to actually get paid in USD stable for being creative.

expanding adoption:

Successfully acquiring this 'quick & mass adoption' will allow our smart contract to easily & quickly claim a trusted 'DAO voting' like status on blockchains. After demonstrating and proving this status, we can then move on to additional voting & comparison solutions w/ NFT ownership, for any medium that is capable of using 'hashes'

Examples: anonymously validating professional opinions w/o risking manipulation, comparing & mapping food product testing w/o risking exposure, audio & video feedback services w/o censorship, digital safety deposit boxes, shared wallets & assets, shared passwords, etc. (additional info in 'what else can the tech. do' below)

Hence, in considering the possibility that meme hobbyist may indeed fail to adopt our technology, our solution offers easy pivots into these industry examples above.

late adopters: NFT market place and block explorer industry leaders (a low level platform) Our smart contract can easily be used as a utility. This is the basis of our expanding B2B model. When these industry leaders take notice of how we are attracting new users and providing unique value within NFTs, we see them building (wrappers) on top of our integration. We are striving to encourage this behavior (within our source code), as we will naturally be collecting fees underneath every single transaction.

This is analogous to a bunch of existing DEXs using a single DEX under the hood, thats collecting a small piece of a bunch larger pies.

How is the MemeCompetition revenue distributed?

The revenue model is fairly straight forward. In the section above, labeled 'MemeCompetition Use Case', you will see how competition 'entry fees' are paid in USD stable in order to generate a competition prize pool. Hence, the contract will simply take a small percent of each prize pool as a fee (similar to most DEXs)

Revenue Distribution

Priority will be of course set for expenses first.

Then the remaining should be distributed evenly between the engineering and marketing teams, 50/50.

Both teams will require active work efforts, and if we succeed in acquiring users, the work load will only increase.

As the work load increases, we may even need to 'hire' additional help. This would be added to running 'expenses'. As well, we would likely decide to setup payrolls for ourselves in the future, to add to running 'expenses'. And then the remaining can be distributed as revenue (50/50) from the contract vault once a week or once a month, depending on how the venture progresses.

The contract 'vault' will support a payroll like model for adding payment addresses for expenses. As well for adding 2 addresses for distributing the 50/50 profits (1 for engineering team and 1 for marketing team; each to then take responsibility for further distribution)

What else can the tech. do?

By storing 'hashes', we can effectively validate the creators of anything posted on the internet. For example, we can validate audio & video files...

- this will allow people to claim & build followings around music or audio tracks

- this could perhaps offer adoption within the indy-film industry

- this could effectively bridge or potentially take over the legacy copyright market

By storing 'hashes', the contract can act as a digital 'safety deposit box', effectively locking any data within the contract itself...

- scenarios can be created where 'data' can only be accessed with a password or key.

 - This opens up a wide array of adoption to all kinds of security industries.

 - You could lock and send private keys to each other

 - You could sell source code or web links anonymously

 - You could timestamp and store legal documents, etc.

The notion of using 'hashes' can also be applied to any kind of 'text' that you may want to 'hide', out-in-the-open, assuring that it can never be lost. (including images)

These examples, among many others, can all be executed within the same smart contract without the need to launch a new project or product (but is indeed an option). These different execution strategies can be utilized as potential expansion opportunities or pivot points depending on market adoption.

Launch Schedule?

TBD

Why PulseChain?

PulseChain is cheap, efficient and easy to onboard new blockchain users. The meme addicts and hobbyist that we are initially targeting, have a higher probability of jumping into PulseChain due to the lowered fees (as opposed to Ethereum).

Supporting metrics & research:

ref: <https://marketsplash.com/blockchain-statistics/>

ref: <https://contentdetector.ai/articles/meme-statistics>

ref: <https://contentdetector.ai/articles/wp-content/uploads/2023/11/Meme-Statistics-9.jpg>

ref: <https://contentdetector.ai/articles/meme-statistics>

ref: <https://www.theblock.co/data/nft-non-fungible-tokens/marketplaces>

END