RAID

“In the beginning was The Word...”

John 1:1

Page of contents

Decentralized word-of-mouth marketing service

As of today internet personality bubble reaches absurd levels, a mere mention by an internet celebrity can a very significant amount of money. It is generally believed that buying internet personalities' time or creating an ad-trailer should be expensive and it’s worth it, however it appears than a new more effective alternative to this already exists and actively being used. And it appears that it is certainly possible to achieve this in a decentralized manner when the employer and the poster don't need to know anything about each other except the stake, eliminating the need of assembling a marketing team or trying to apply to real-life jobs.

What ad trailers and internet personalities attempt to do is not just to sell, but to start a *discussion* to increase awareness, to pump up the popularity of the product. Word-of-mouth marketing, proof-of-discussion, whether it's praising or sick burning, FUD or optimistic insights, blind cult following or elaborate arguments, - the discussion behind the product is what really promotes the product. Celebrities and ads are just a third-party and it is certainly possible to eliminate this third-party and pay directly for discussion and mentions instead. A mass discussion by nobodies for nobodies. RAID attempts to provide income to socially inept, traumatized, schizophrenic, mentally challenged and physically disabled people, so that they will never be kicked out of their homes for being unable to adapt to life.

The primary utility of the token is being a sovereign currency for exchange between employers and posters. Employers buy RAID token, setup and fund campaigns with that token in Raid Market contract with chosen key strings or topics on chosen websites, posters commit to these campaigns by discussing certain posts or anything eligible(matching keyword) for paid discussion. By default, posters are allowed to express any opinion on every topic, and, depending on the resource, posters by default have the right to completely derail the discussion, talk about weather, discuss investments, etc. Posters will get paid for unrelated discussion as long as it for example bumps the thread, or adds another comment to discussion making it look more heated and popular. RAID default marketing paradigm promotes critical thinking. Again, it mostly depends on the resource and a forum could simply ban/remove unrelated posts. Posts need to be witnessed by oracles, so a poster has to ensure that his post satisfies the rules of a website. Campaign settings are flexible:

1.Ppp – pay-per-post. Defines how much posters get paid for a post in USD value. As RAID token price decreases, the amount of RAID being paid to posters increases. USD index is monthly and is average price of RAID token last month. Therefore if a campaign lasts more

than one month, amount of RAID paid per post adjusted automatically to ppp. Ppp can only be increased by the employer if edited.

2.Array of key strings. A key string can be a word or a phrase, a sentence, a text of any length. If employer sets more than one key string, then these key strings become options to make the posting more natural, if a poster mentions just one of those key strings or posts below an op post with one of those key strings, he is eligible for a reward, and there is no point for him to mention all key strings.

3.Mandatory key string can be left blank. Requires to use mandatory key string in every post regardless whether the post is related to key string discussion or not.

4.Array of target urls. The campaign works only on the websites with these urls. If none set – it means everywhere, dapp interface will help to easily choose most popular resources.

5.minStaked. Minimum requirement of locked RAID tokens for a poster to have to join the campaign. It can potentially help with moderation or eliminate the need of moderation completely, depending on the chosen route of marketing campaign.

6.nonEditable, a boolean. Can be set to true in it's inception or at any point in time, to allow the employer to first set it up looking at results. If a campaign is non-editable, it can attract funding from other employers, basically allows to make it last longer potentially.

7.noFiring, a boolean. If set to true, posters can't be fired at all, so that posters will more likely join the campaign.

8.onlyManualApproval, a boolean. If set to true, posters can't join the campaign without employer' approval, when this is set, then minStaked is ignored completely, even posters with 0 RAID tokens locked can join as long as approved.

9.KeyStringPerWords. As an example, in a job which requires 1 key string per 1000 words, if a poster writes 4 posts 250 words each, he has to mention the key string in those 4 posts at least once, and he will get paid for those 4 posts.

10.MinPostLength. If the requirement isn't met by the poster in a post, he is not eligible for a reward for the post.

11.ModsPay. Needed if the employer does not feel confident and wants to moderate the campaign, but has no time for that. RAID mods are not obliged to follow the rules he wants to enforce, however they are assumed to follow the rules he wants to enforce, since mods salary(independent from ModsPay) is not fully tied up to RAID to USD index and because the governance can fire them prematurely.

12.rulesLink. A link to a post explaining the rules for posting in detail and ban rules. In the spirit of default RAID campaign can be left blank.

13.expirationDate. By default it’s 3 months from creation, can be set only longer. If the budget of the campaign is not exhausted, and non-editable set to false, the remaining budget is being refunded to the employer.

14.minCreativity. Decentralized moderation evaluates some posters’ creativity, emotions and sense of humor and increases it with time.

15.postRate. Currently in Polygon blocks. Determines how often a poster can post eligible for payout posts in this particular campaign. Default is ~1 minute, if poster posts more often he is not punished, just not getting paid for more.

16.maxPosters. A limit to make small budgets viable. If not set, might be computed automatically, considering that posters have to cover expenses of rewards claiming.

17.startTime-endTime. By default 0 to 0, which means 24 hours per day. Specifies time of day, when the campaign is active.

18.An array of campaign languages. Left blank if any language. RAID can probably support at least up to 256 languages. If not left blank, then only posters who can join are those who stated their language. A poster can’t alter his language, he can only choose to set one or not to set at all. From the start, posters of different languages will have same pay-per-post for default campaigns for fair distribution.

-Some of the most expensive commercials ever made cost around $30 millions. This amount of money in RAID could produce 300 millions of posts, if the compensation is 10 cents per post, and an absolute overkill campaign of 3 billions of posts if the compensation would be set to 1 cent. The most commented videos on Youtube have less than 15m posts, so a campaign of this scale could be used to promote brand commercials on Youtube and potentially create unprecedented so far public interest. This campaign could also build the community around official accounts and to set given keywords trending for a prolonged period of time on different social networks.

-Twitter influencers can mention default RAID campaign, so that the posters will more likely discuss that post, helping to promote that Twitter influencer profile.

-On websites which support nicknames posters will eventually be able to talk about anything with nearly anyone anywhere as long as they have keyword in their nicknames.

General technical principles

1.Accuracy is expensive.

- Stone Age math whenever worth it, minimum Solidity store loads and writes.

- Packed structs if more than one value of a struct is altered by a method.

- Store numbers with less decimals where possible and convenient. Restrict accuracy where possible and convenient

- Hardcoded addresses and values whenever possible to, again, reduce store calls.

- Queue Transfer Contract will be introduced later, briefly it's a bulk sender which allows to perform certain actions in a collective queue. The cost of Queue Transfer for the end user can be potentially reduced to 1 store write as trustless version and to 1 emitted event as a trusted/trustless version at the expense of waiting until the queue gets filled and executed.

2. Decentralization is the most valuable concept of our times.

- Bitcoin forks are more centralized than Bitcoin. If it will be required then top holders of Litecoin could be found and influenced in a way that allows the government of a nation where top holders reside to acquire a lot of control over Litecoin. BCH and BSV wealth distributions are considerably better than Litecoin, still worse than Bitcoin. A huge dump has the potential to weaken the security of the network, and after that it can be reorged in any way required. Today, cryptocurrency is potentially regulatable and mutable as is. There are ways to change this.

- RAID attempts to revive old concepts of decentralization, and aims to be decentralized in any possible way. There won't be a CEO, there won't be a Lead Developer, there won't be unilateral decisions, no wealth concentration hopefully.

- RAID solves blockchain wealth distribution problem by implementing old like the world CeFi feature - active income(salaries). With it, RAID can expect Gini coefficient closer to real world CeFi levels than any other blockchain project. This pushes decentralization security standards on another level.

3. No mandatory auto-updates. No hidden tracking.

- RAID dares not only to support this standard, but to promote it and make it the only acceptable standard. Auto-updates will be always disabled by default(if possible, Chrome browser does not allow that)

4. Free open source.

- If it wasn't for free open source, if no free libraries would exist, the development of minimum RAID functionality could take years. And RAID could be a puzzle for something greater.

Base contracts

Ethereum is the blockchain of choice as the most popular and robust censorship resistant chain, and with an assumption in mind that blockchain industry is fundamentally monopolistic in a sense, since the biggest network is always the most secure, which attracts even bigger capital and in turn makes it even more secure. And in case of Ethereum even more censorship resistant. Base contracts will not be owned by the deployer or governance and won't be upgradeable.

https://github.com/SamPorter1984/RAID/blob/main/contracts/VSRERC20.sol

1.First and main contract is "Very slow ERC-20" implementation(VSR ERC-20). The implementation utilizes standard ERC-20 function _beforeTokenTransfer() in such a way that prevents treasury fund from dumping on the market. The function checks how many blocks passed from rewards genesis block and allows to claim only a certain amount per every passed block. Basically developers and all other participants of RAID can only claim rewards within constant emission limits and this hard limit can't be avoided. Even if treasury happen to be upgradeable, no matter what kind of logic will be present, even if the contract will happen to have a bug, the bug in no way will be as devastating as it could be without _beforeTokenTransfer() guard.

Allowances in this ERC-20 implementation are made booleans instead of integers. As it seems so far, allowances computation is wasteful in very most cases, since nearly all protocols ask for infinity-1 allowance. RAID token has the cheapest transferFrom() of all ERC-20 at the time of writing.

Another feature is bulkTransfer() and bulkTransferFrom() methods. These methods require an array of addresses and amounts as arguments and compute balances of an array and only after that compute the balance of msg.sender, instead of how regular transfer would compute the balance of msg.sender after every transfer to an address, which makes bulk

transfer twice cheaper. BulkTransfer() can be used by treasury to distribute rewards and bulkTransferFrom() will be used by Queue Transfer contract.

https://github.com/SamPorter1984/RAID/blob/main/contracts/FoundingEvent.sol

2.Second base contract is Founding Event Contract. This is a liquidity generation event(LGE) proposed by CORE token team with certain differences which I believe most suitable for RAID. Founders Contract is a trust minimized LGE. It automatically creates liquidity on first transaction after last LGE block. I can't transfer Ether from it, nobody can. To ensure critically required in case of RAID decentralization, the LGE will last for 2 months. Liquidity is not locked at all, instead an incentive to keep liquidity is introduced. Rewards for Founders and liquidity providers in general depend not on the amount of liquidity shares they stake, but on the amount of RAID tokens present in their liquidity shares at the time of staking. And this number won't change as long as a given provider does not unstake the tokens. So, for liquidity providers the incentive to provide liquidity and stake increases if the price is going down. While RAID is not expected to give any returns or any kind of guarantees, if the price increases - founders have the least incentive to unstake. As soon as they unstake, they lose the advantage. Founders also lose Founder status as soon as they unstake their liquidity from Founders Contract and become a liquidity provider.

Founder status gives Founders ~150% higher rewards than a normal liquidity provider, which is done in an attempt to mitigate potential losses. Becoming a Founder is an even greater risk than participating in the ecosystem after Founding Event concludes, therefore their rewards are higher just in case. To learn more about risks, you absolutely need to read the last page. Founders, as well as liquidity providers, are able to switch addresses if they feel the need to, which allows them to claim their stake and rewards from a different address.

Every Ether deposit is being subtracted by 0,5%, and this 0,5% will be used for audits, bug bounties and development like oracles, servers, RPC, ddos protection(as it starts with centralized oracle and only after moves to completely trustless architecture) and any other expenses required by RAID during LGE.

https://github.com/SamPorter1984/RAID/blob/main/contracts/TrustMinimizedProxy.sol

3.Raid Market which was already explained.

4.Trust minimized proxy. It's an altered OpenZeppelin upgradeability contract with some features that allow to remove trust to developers and/or governance. New logic implementation is not being set suddenly, instead it is being stored in NEXT_LOGIC_SLOT up to NEXT_LOGIC_BLOCK_SLOT, or for a month or so. The period allows participants to identify if the deployer or the governance is malicious and therefore to exit safely. Next logic can be canceled in case of a bug discovered or upgraded to after month passes. It is

impossible to cancel next logic and immediately propose another next logic, because there is also PROPOSE_BLOCK_SLOT which disallows proposing next logic more often than once a month. It is also possible to add a value to PROPOSE_BLOCK_SLOT if for example a situation arises in which there are no plans to upgrade a particular contract for year maybe, so that it keeps participants piece of mind for that period, because no upgrades are possible during that period. This variable also can be set to infinity-1 to essentially.

Additionally there is DEADLINE_SLOT, the block after which it becomes impossible to upgrade the contract at all. Admin keys are burnable and just in case DEADLINE_SLOT stays. I would like to encourage DeFi developers community to use this proxy as soon as it gets an audit, so that the amount of scam could potentially decrease at least a little bit. However, there is no warranty even after audits.

Public temporary database

RAID will use fast centralized public blockchains as temporary databases or optimistic roll-ups which are incapable of any censorship, one of such roll-ups is Arbitrum. First, RAID can start from Polygon Network. The database should be blockchain agnostic, because fees on a particular network can become unacceptably high for posters.

Database contract is a simple event emitter with settings variables for oracles to act upon. Posters emit events in database contract with the information about their posts. Oracles then verify if those posts exist. Commit-reveal scheme nearly eliminates front-running as well as disallows oracles to alter the transactions and allows this system to scale to any number of posters and websites. Emitting an event ideally has to be a lot cheaper than 1 cent. After and if all fast blockchains become more expensive than RAID requires, there are at least 2 solutions to resolve it:

1. Add oracles between posters and the blockchain on which they emit events. This solution could also allow to use slow chains like Ethereum Classic.

2. Create super lightweight restricted to certain functionality second layer only for oracles to act upon. Could use Autistic roll-ups(an even lighter version of optimistic roll-ups, specifically dedicated to support oracles with events being wiped regularly).

Or both. Ideally, RAID will have to give a poster a choice: broadcast through an oracle aggregator, or broadcast on his own.

Raid Wallet

RAID wallet is a browser extension. As of yet, no ability to connect to any website, no ability to transact either, it's a brick. Extension was supposed to be a Metamask fork from the start, however, Metamask has committed heresy against free software license. RAID wallet does not connect to any website as of yet, so the probability of it being fished is reduced. It's very insecure nonetheless as of yet.

The extension fetches specified form data, stores the post and sends a hash of current post and previous post to the blockchain. It can operate on nearly every website on the internet, any social network or html-js chat, from Twitter to Twitch. Functionality of this extension is restricted during beta-test to certain imageboards and Twitter by centralized oracle. It will probably be restricted for a lot longer than that, and different relevant websites support will be added gradually over time.

Planned functionality:

1.Custom encoding for languages which could require that, as UTF-8 is inefficient by blockchain standards. Every community dedicated to specific language can submit an efficient encoding scheme if they feel like it's needed.

2.Everything that a modern wallet has, including encrypted keys being stored on the user device only.

3.Limit orders and any other orders useful for dex trading.

4.Queue Transfer contract interface for cheaper but slower transactions.

During beta-test posters will be able to specify any secure address which to receive beta-test salary, the extension itself is not supposed to hold any funds as of yet. After extension will be secure enough, the posters will lock a certain amount of tokens in poster wallet to be able to work. Different jobs could require different lock amounts. Posters can also participate in governance with their locked amounts.

Decentralized moderation

Sense of humor is the only way to prove that you are human. It's the last frontier of humanness in AI' world. Moderators of different language communities have to be not only Patriots, but also Comedians. Good sense of humor could be promoted and given a higher pay(but the difference should not be too high, so it won't discourage anybody). There are probably other ways to recognize genuine humanness at least as of today' AI development. Creativity, emotions. Promoted posters can help with moderation. Moderators can potentially be unfair so it's required that they will be reelected frequently, probably once a year, and have a high enough pay to care. To become a candidate defining language is required and it will be set in stone. To vote for or against candidates, it's also important to define the language, so voters have to not only lock their voting power, but also they have to set their language(or they can leave it blank, anonymous, except they won't be able to vote for mods and anything else related to specific languages).

How can we be sure that moderators of a language community we don't understand at all actually does the job correctly? Simply test that community. First we try to spam in our languages and get banned eventually, then we go to other languages community and try to spam random text after translator or fetched posts from all over the internet, maybe from previous threads, and see if we get banned. If we don't get banned – we just add more bots and get more tokens. Either way, bad moderators' job will be revealed regardless of language. All moderators' job should be public and shown on the website with history of addresses they have banned, so that independent reviewers could point out to censorship. No moderator should be able to ban anybody single-handedly but instead a group of moderators vote, and only most voted addresses will be banned. We can add a second layer of moderation to this to make it nearly trustless: most voted addresses by moderators cannot receive any pay for posting until DAO decision. The DAO can take from locked stake as much as the account posted or revoke punishment.

Moderators salaries have a base salary which is not dependent on monthly RAID to USD index, and also they will have a small bonus depending on the amount of active posters in their language. The bonus is small, because a big one can further the disparity of adoption between popular and less popular language communities. RAID moderators are also community managers and tech supports, and they decide between each other how and what and who will do, the governance will be able to fire lazy or unfair elected officials prematurely. First year officials could be elected by Telegram polls. Officials can define how they will run the community, they decide if they need to run a blog, a Twitter account, or whatever else. For second year elections governance contract will most probably be ready, so that the community decides upon if they want to reelect former officials or to introduce

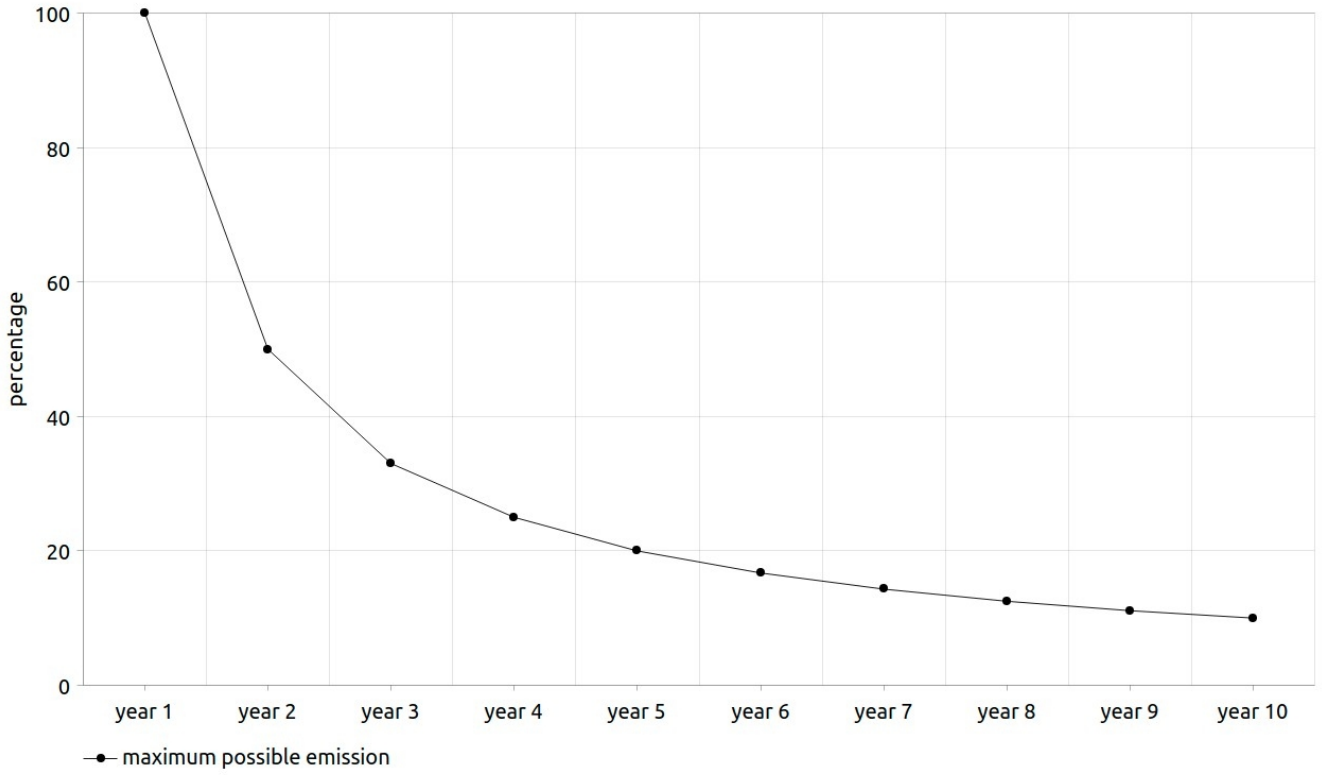
new ones in a trust minimized way. A Telegram bot, and probably more than just Telegram bot, will be required to grant mod status to elected mods.

Tokenomics and Treasury

Token utility and multi-year token locks require high inflation in order to support decentralization progress.

Starting supply: 1 million.

Total supply: 1 billion.



Emission: maximum emission is approximately 1 million each year. First year includes testnet rewards, 0.001 token per post. Note: while some funds' tokens are being unlocked on fixed schedule, it does not mean that all unlocked tokens are being claimed, so it's better

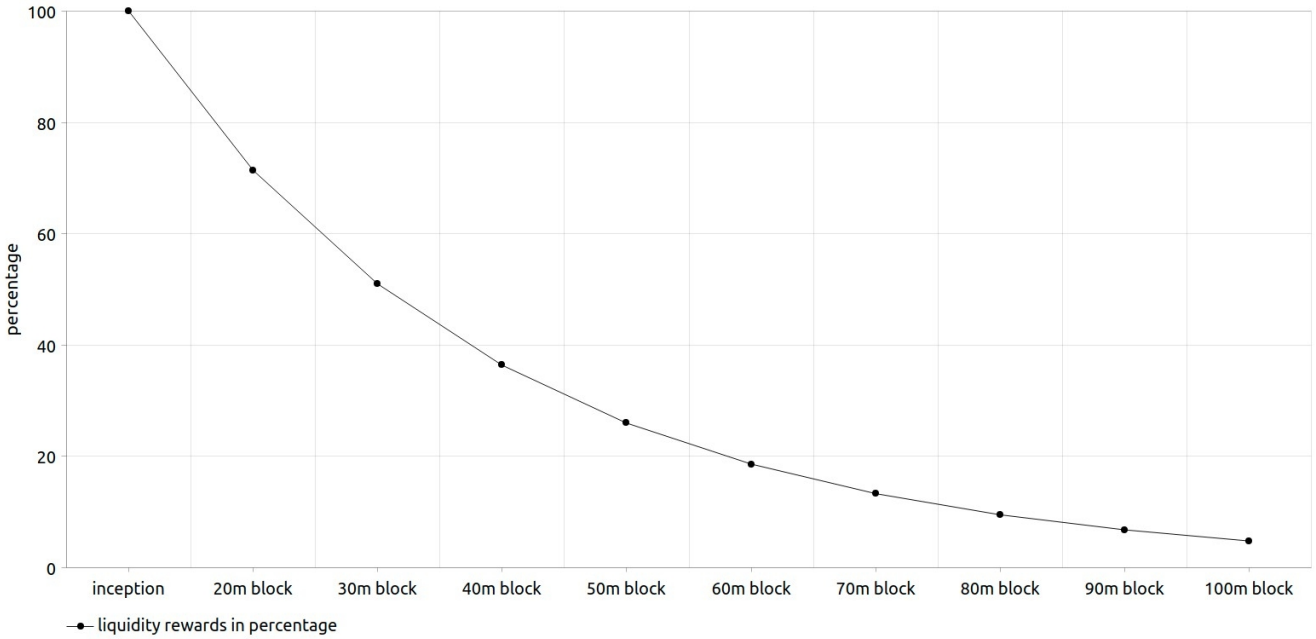
perceive these numbers as maximum possible inflation, not actual inflation. The reason behind these numbers is mainly an attempt to make 1 token always affordable to the residents of poorest countries, assuming that Pareto distribution in cryptocurrency is extreme, as the main income in cryptocurrency space is passive. It's important for numbers to be comprehensible to not only economists. Another reason is that if RAID uses mint function with upgradeable contracts which could potentially have a critical bug, it could destroy the project completely, so there is no minting at all and _beforeTokenTransfer() used instead.

Treasury is an upgradeable contract until finalized, the reason for this is that it's still a challenge on how to implement trust minimized management of the treasury. A deadline of 2 years for finalization is hardcoded in upgradeable proxy, if the contract won't be finalized at that point in time, it will be set in stone as is.

Treasury will support:

1. Default RAID promotion campaigns. Posters will receive salary for default campaigns from treasury. _beforeTokenTransfer() fixed emission ensures that too many posters will be unable to consistently claim salaries. It does not mean however, that a poster who was unable to claim loses his salary, he will be able to claim *eventually*. This will allow to properly setup and scale oracle network to prepare the network for more posters. This could also create an incentive to FUD RAID in case when there will be too many posters, which will result in mostly smart posters getting on board(while Founders would more likely rather shill RAID after Founding Event is over). Default jobs will have least requirement for tokensLocked, and probably least pays.

2. Founders and generic liquidity providers. Founders have starting rewards equal to 5% yearly for their ether contribution assuming that the price of ether and raid token stays constant. Founders will have 150% higher rewards for same tokenAmount than generic liquidity providers. Rewards for both groups of beneficiaries decrease by ~28.571428571% every 10 million blocks. TokenAmount is a variable that represents the amount of RAID tokens in liquidity shares at the time of staking these liquidity shares. Founders' liquidity shares are being staked from the very start, and their total tokenAmount equals to starting supply, 1 million. As a founder unstakes, his share of rewards is being redistributed among all remaining founders. Liquidity providers collectively get 150% lower rewards than founder rewards, and share their rewards together without accounting for founders rewards at all. If a liquidity provider stakes the minimum amount of liquidity shares to get at least 1 wei of tokenAmount, as long as he is the only one staking, he receives all rewards from generic liquidity providers pool. This solution could potentially eliminate third parties like Unicrypt.



3. Monthly meme contests. Governance will vote for best memes monthly and a top third of winning memes each month will receive rewards from treasury. To prevent too much spam in meme contest while at the same time not causing stagnation, only addresses with at least 1 RAID token will be able to submit memes to the community as NFT. Winning memes are being resubmitted for the next month, therefore best memes can receive rewards for several months.

4. Oracles. RAID is heavily reliant on oracles. The oracle network around RAID might become massive. Especially if we attempt to fulfill the idea of pseudo-anonymous oracles network. You can read more on this in the next chapter.

5. Decentralized development of RAID network and software, as well as support of established free open-source software.

It’s important to emphasize that RAID development starts decentralized from it’s inception. I have invited Odilitime as a second developer. Two first developers chosen by the community which will join RAID decentralized development each will have fixed 30k of RAID tokens from the treasury, since the very-very inception situation is rather hard to decide upon in truly decentralized way, even RAID strives for it the most. Absolutely anybody can optimistically contribute to the development even before the governance contract is finalized, assuming that after the governance goes live it has a strong incentive not to forget great contributors. Therefore after those two developers with fixed allocations any other developer can join and contribute so much, that the governance can decide to give that developer a bigger allocation than two first developers. All other developers can be hired by the governance when the governance contract will be ready, and they will not have

a specific fixed allocation but a salary instead based on the monthly index of RAID to USD or one time grants for contributions. Their salary can vary from $30k to $500k per year, depending on the governance decision evaluating the amount of work and responsibility for a particular developer, and the grants can be of a lesser amount. In it's final form the governance will create verifiable by oracles tasks and find developers to execute them. Emission for developers funds release is very slow, assuming that there are only two developers and they claim their grants immediately as they are released

Many will probably agree that small but dedicated teams can easily turn out to be more efficient, than bloated corporations. The treasury can enable developing free open-source alternatives of any closed source proprietary software. So it will be a lot of small teams up to 5 people maximum maybe. They can create an alternative to Photoshop, Sony Vegas, Google search engine, car driving AI, etc. Including Windows, GPU drivers, any sort of software, including even game engines, we can develop free open source games with all assets being completely free to reuse, which will decrease the cost of game development. If Linux Mint becomes compromised, and it is highly likely as soon as it gets really popular, we can build on top of their legacy and call it Linux Tall Hat, I guess. There are plenty of things not only to bring open-source but also to develop. The fund will also pay (hopefully the most generous) bug bounty and for audits.

RAID software needs special license so that the developers will be unable to move away from free license, a license that can't be changed. It's either the owner of the software is RAID as decentralized entity, or it's a license that can't be changed as defined in the license, but there could be holes in the law.

6. Non-profit social networks like Mastodon and imageboards. Imageboards could need next level popularity, next level mainstream adoption, so that people will ask more and more about what is it about being anonymous? Why is it important? So that they will more likely to value privacy, lean towards spyware-free programs and OS.

7. Anything else that governance will be interested in supporting, as long as functionality for grants and financial support for a particular idea is possible to fulfill in trust minimized way and as long as it is not illegal.

Pseudo-anonymous oracle network

While Raid oracles could provide KYC for simplicity of the design, with Chainlink verifiable random numbers it is possible to allow anonymous and pseudo-anonymous oracles to deliver true results. Oracles shouldn't know what role they are performing in a given iteration of publishing results. There have to be two roles: witnesses and supervisors. Chosen supervisors have to be a small uneven amount of all oracles. Supervisors' results are considered to be true, and witnesses results have to match it. If supervisors' results don't match, majority of identical results of supervisors and witnesses are considered true, and minority results are punished, if there is no clear majority, another attempt of choosing supervisors occurs, until supervisors results match, while published results stay, no republishing occurs during that. If in case of lies we will redistribute a liar stake between "honest" oracles, it creates an incentive to attempt and publish fake results by majority of oracles. Therefore in case of lies, at least 75% of the stake should probably be burned or goes to the funding of default campaigns. An honest oracle can easily miss, say, 3 posts out of 10k posts, depending on the resource, therefore there have to be safe limits for inaccuracy which is not considered a lie. To increase the probability of that the several oracles are definitely not one person, we can use these facts about an anonymous wallet:

1.Balance. Allow anonymous oracles only with considerably high balance, we can even start from whales, top 100 addresses, and if nobody joins, decrease the requirements to top 200 addresses, etc. Higher Raid balances have the least incentive to lie and ruin posters' trust.

2.Transaction history. Democracy Earth Foundation is building tech which attempts to measure unique humanness. Measuring DAO choices could be the best and the only way without specifically asking to provide any other information. We could use their framework or build our own which evaluates the differences in views in different DAO choices, and not just membership of different DAOs.

Raid specifically also can use these variables:

3.Language. Language communities can elect oracles, and the probability of them being one person or collaborating decreases even more. An oracle cluster dedicated to specific job could consist of 20 oracles from different language communities with 3 of them being chosen randomly as supervisors every iteration.

4.Poster history activity and uniqueness. Raid can elect only most active unique non-bot posters as oracles, to decrease the probability of oracles being one person even more.

There will probably be more than one poster database contract and oracles for each of them, oracles cannot include any addresses not registered in their given database as

eligible for payout, so oracles can at best censor some addresses collectively or approve all existing addresses transactions even if those transactions are fake. If a poster finds out that oracle cluster he is working with censors him, then he moves to a different oracle cluster, so that censoring oracles lose money and will probably lose reputation in the eyes of posters, since the censorship could be verifiable with most resources. In case of fake rewards, the governance will be able to punish oracles, but only within certain limits, depending on the lies occurred. An independent observer software is required for this. This however is still a pending issue to resolve in a trust minimized way but not from the side of posters and governance, but oracles' point of view, because this solution requires oracles to trust governance to be fair with them. If in case of finding a trust minimized solution to this specific issue will be an impossible challenge, like if adopting a supercheap second layer, Arbitrum for example, somehow becomes not possible, the worst what can happen is that the oracles will need to provide Chainlink KYC, so that the punishment can be reduced. Autistic roll-ups could be required.

Oracles constantly verify posts and keep the data on the amount of verified posts for every address in their databases and publish the rewards data every month to Polygon chain, so they won't need to keep the data longer than a month. Oracles then use privacy oracle solution like Deco, to generate random disposable keys to move the rewards data from Polygon to Ethereum mainnet through trust minimized bridge, so that it will be impossible to alter the data.

Adding oracles between posters and the blockchain is also an option in case of fees being to high for posters to stay cover posting expenses. And there are numerous ways of implementing that, we could make it the same way, oracles publish posts in bulk, and a poster sends the data to several oracles(relayers), and if these oracles censor this poster, then he moves to a different oracle cluster, essentially censoring oracles lose money again. We optimistically assume that anonymous oracles will have the least incentive ever to censor anybody.

Trust minimized cross-chain bridge

Commit-reveal scheme disallows oracles to alter transactions, the worst they can do is to censor the transaction, which they are less likely to do, if we use the same oracle system with roles assigned by verifiable random numbers. For the user it could be a bit

cumbersome but worth it, since it's trust minimized. First what he need to do is to announceHash(), Hash has to be generated by the off-chain by the user maybe through a web ui and has to correspond with:

keccak256(abi.encodePacked(userAddress,arg1,arg2,arg3,arg4,anyDisposableKey))

The user keeps all arguments and disposable key to himself, until oracles relay the hash to the other chain, he then must verify if the hash is indeed his, and if it is, then he sends the actual transaction with all argument and used disposable key. The contract on the other chain will only accept address, arguments and a key that matches previously posted hash. If the contract indeed receives correct arguments – oracles are rewarded. This bridge allows not just to cross() or simply relay tokens value, but it also allows to callAcross() - to relay data which enables trustless cross-chain contracts communication. This function will be used by the oracles to relay rewards information from Polygon to Ethereum.

The bridge can support any chain or token. For a particular token oracles will create a wrapper contract, if the token lacks liquidity they have no incentive to deploy another ERC-20, but anybody can just request a bridge and pay money for the deployment. For example wrappers on Ethereum Classic, could use a prefix of lower or upper case "c".

If the oracles are anonymous or pseudo-anonymous, the risk for value transactions is a possible event when all oracles are being paid by a third party and all of them agree to censor a transaction. The possibility is really low, personally I am optimistically assuming that the reputation costs more than censorship. However, for the case specifically like this, a user can allocate value as a tip to oracles. Which decreases the probability of censorship even more. Without KYC however it could still be a leap of faith for transactions exceeding oracle rewards, so the design could be improved, or the bridge just has to operate with KYC oracles.

Fundamentally pure governance

Malicious governance is common. The way is to simply disallow being malicious as much as possible. In case with RAID it's not just potential riches of treasury, it is also about being compromised by scammers.

1.First problem is updating contracts maliciously. It seems impossible to me to implement any restrictions on upgrades that the governance can approve. The DAO can override nearly

any hardcoded limits. Checking a part of bytecode of next logic implementation in assembly is futile, any variables can be reassigned and overriden.

As far as I went, there is no way to implement any limits in proxy contract except of which we currently have in trust minimized proxy, basically we can only set time limits and locks. If the governance is compromised, it could potentially completely ruin the idea behind the project. So very minimum quorum and a long period of voting is definitely required.

2.Second problem is about managing treasury and is a lot harder to solve. This functionality requires absolutely next level DAO' purity of intentions. What I propose to resolve it is to only allow certain options for governance to decide upon fetched by trustless oracle network. For example, to sponsor open source project and to ensure that it is not a scam to get money from RAID governance, oracles fetch only established projects with certain minimum measurable limits like time since inception. Same could go any other grants governance can approve, oracles can propose only existing established companies or individuals as beneficiaries with verifiable profiles or anyhow transparent and convenient. It can be possible by fetching right information from right resources with right filtering.

So the governance can choose options that oracles propose. And in this particular case all oracles will be capable of reaching 100% accuracy matches in results. At least with proxies they probably will be, or we will have to exclude certain websites. Any grant is not being transferred in one big transaction. Claiming of the grant starts from 0, not truly final, and the DAO can shut it down, if something is wrong. With this governance model it’s also possible for the DAO to create verifiable tasks in many fields. Receiving grants address should not be a contract, so it will be much harder for oracles to transparently cooperate for successful lie.

Creating a balanced voting system is a challenge. Passive income for voters is what creates imbalance, a fair amount of voters just do not vote, they only need passive income, and therefore the outcome of voting participation is impossible to predict.

So what I personally believe in is to lock tokens for at the very least 3 years for voting. And, half of last year of token lock voting is forbidden for a voter, unless he prolongs the lock. I believe currently, that even if Founders and liquidity providers have passive income, it should be accessible with their staked liquidity, however only after a lock, if they choose to lock it(or to elect themselves if you will) – they can use it as a voting power, but if they choose to only have passive income, they don’t need to lock it. A lock should be a sort of sacrifice I believe, it has to be long enough, so nobody will lock and disrupt voting balance just because he maybe could be voting, and so that it won’t be way too long and exhausting for a voter to lose interest in voting. The functionality requires only dedicated voters and the numbers are still being collectively worked on by the community.

With this model we can have a high percentage for minimum quorum. We can have 40-60% instead of 4-10% for a proposal to be executed. The governance model still remains the biggest challenge of RAID architecture and probably requires way more brain power than I will ever have. We need to discuss it and decide upon the best possible model.

ERC-20 allowances can be used to trade locked voting power transparently, and while there is a workaround to solve it rather transparently for RAID token, it is not possible for Uniswap LP token as it is a standard ERC-20 created by the factory. So the system really requires truly decentralized distribution, so one party could not be able to maliciously vote and get away with that easily while creating an imbalance in a given proposal vote. Important to mention that the system does not really critically break if there will be ways to trade votes transparently and even trustlessly.

The most critical voting has to be recorded on Ethereum chain. Voting like meme contest and moderators elections could be run on Ethereum Classic.

Where it is required, voting based on fetched by oracles data can be split in stages:

1.Voting by active human posters without taking into account their stake, we could also use KYC but only for small balances without sense of humor.

2.Proposals approved in the first stage go through voting by stake.

For some proposals there could be even more than 2 stages, more sophisticated, but basically it will be a lot harder to compromise such a system. For example, oracles could just fake the data about nonexistent medical facility which has a lot of wallets, enough for all oracles, so active posters will probably reject that.

Beta-test

Beta-test will launch together with Founding Event and will last for at least 2 months. During the beta-test, depending on Mumbai testnet capability and if RAID and Polygon will be able to reach an agreement, transacting will not require any Matic tokens, in other words will be free. Only one campaign will be available during beta-test: RAID campaign. The compensation for one post will be fixed to 0.001 token per post, instead of index of RAID to USD. Supported websites will be imageboards and Twitter. The posters will be able to claim their beta-test rewards after the Founding Event concludes and trading of RAID token goes

live. After the platform hits mainnet, the salary will probably start from 1 cent per post using starting token price as reference as there is no price history yet, and will increase if only the price of RAID token has the room for that and current network scale is ready for more posters. While decentralized moderation might not be ready in time, blatant spam-botting and anything illegal will be still banned and no rewards will be received by that poster address. An extensive review will be conducted before beta-test rewards claiming, so attempting to bot it could be pretty much a waste of time. Since there is no requirement at all to join the beta-test, an option to limit beta-test to only Founders exists.

RAID chain

A wizard proposes the chain to be a fork of Oxen. Oxen by default is an XMR POS fork, therefore RAID chain will have privacy of data by default and ETH daemon for smart contracts. Since it's POS it's possible to significantly increase block size for higher transaction throughput without hurting decentralization. Currently 2 options of rewards incentive mechanism for POS nodes are being explored by the community:

1.Default rewards mechanism. With possible slightly modified EIP-1559 without fee burn.

2. Fixed gas price and fixed rewards for smart contract execution regardless of transaction throughput. The nodes will most likely reject contracts which require high transaction throughput, like dex contracts. And, we optimistically assume that the nodes will approve all RAID contracts because those contracts will allow posters to expand awareness of the chain. With RAID token main utility being marketing, nodes could potentially receive higher rewards as the awareness of the chain increases, so default mechanism might not be required. Basically, the nodes will always disapprove contracts that do not provide any value to them, and transaction prices will always stay low, affordable to posters. With this rewards mechanism or similar the chain could also easily restrict transparent trading of locked voting power, which is could be important for the purity of the governance.

In both cases described above, if a situation of insufficient decentralization occurs, posters can be allowed to run nodes with virtual stake which can eventually be filled with their salaries to make the chain decentralized in no time. Virtual stake nodes after accumulating at least some balance could help validate plenty of 0 value transactions.

Secondary projects

Secondary projects will yield no rewards to RAID participants, these projects will aim to increase overall blockchain community awareness of RAID network. If some of these projects require governance it could be RAID governance in some cases, and in most cases autonomous governance specifically dedicated to that project.

Risks

You acknowledge and agree that there are numerous risks associated with purchasing RAID Token, holding RAID Token, and using RAID Token for participation in the RAID Network. In the worst scenario, this could lead to the loss of all or part of the RAID Token which had been purchased. IF YOU DECIDE TO PURCHASE RAID Token, YOU EXPRESSLY ACKNOWLEDGE, ACCEPT AND ASSUME THE FOLLOWING RISKS:

Uncertain Regulations and Enforcement Actions : The regulatory status of RAID Token and distributed ledger technology is unclear or unsettled in many jurisdictions. The regulation of virtual currencies has become a primary target of regulation in all major countries in the world. It is impossible to predict how, when or whether regulatory agencies may apply existing regulations or create new regulations with respect to such technology and its applications, including RAID Token and/or the RAID Network. Regulatory actions could negatively impact RAID Token and/or the RAID Network in various ways. RAID may cease functioning in a jurisdiction in the event that regulatory actions, or changes to law or regulation, make it illegal to operate in such jurisdiction. For the token sale, the sale strategy may be constantly adjusted in order to avoid relevant legal risks as much as possible.

Inadequate disclosure of information : As at the date hereof, the RAID Network is still under development and its design concepts, consensus mechanisms, algorithms, codes, and other technical details and parameters may be constantly and frequently updated and changed. Although this white paper contains the most current information relating to the RAID Network, it is not absolutely complete and may still be adjusted and updated by the RAID Network Development team from time to time. The RAID Development team has no ability and obligation to keep holders of RAID Token informed of every detail (including development progress and expected milestones) regarding the project to develop the RAID Network, hence insufficient information disclosure is inevitable and reasonable.

Competitors : Various types of decentralised applications are emerging at a rapid rate, and the industry is increasingly competitive. It is possible that alternative networks could be established that utilise the same or similar code and protocol underlying RAID Token and/or the RAID Network and attempt to re-create similar facilities. The RAID Network may be required to compete with these alternative networks, which could negatively impact RAID Token and/or the RAID Network.

Failure to develop : There is the risk that the development of the RAID Network will not be executed or implemented as planned, for a variety of reasons, including without limitation the event of a decline in the prices of any digital asset, virtual currency or RAID Token, unforeseen technical difficulties, and shortage of development funds for activities.

Security weaknesses : Hackers or other malicious groups or organisations may attempt to interfere with RAID Token and/or the RAID Network in a variety of ways, including, but not limited to, malware attacks, denial of service attacks, consensus-based attacks, Sybil attacks, smurfing and spoofing. Furthermore, there is a risk that a third party or a member of RAID development team may intentionally or unintentionally introduce weaknesses into the core infrastructure of RAID Token and/or the RAID Network, which could negatively affect RAID Token and/or the RAID Network. Further, the future of cryptography and security innovations are highly unpredictable and advances in cryptography, or technical advances (including without limitation development of quantum computing), could present unknown risks to RAID Token and/or the RAID Network by rendering ineffective the cryptographic consensus mechanism that underpins that blockchain protocol.

Other risks : In addition, the potential risks briefly mentioned above are not exhaustive and there are many other risks associated with your purchase, holding and use of RAID Token, including those that the risks that RAID development team cannot anticipate. Such risks may further materialise as unanticipated variations or combinations of the aforementioned risks. You should conduct full due diligence on the RAID Development team, as well as understand the overall framework, mission and vision for the RAID Network prior to purchasing RAID Token.

Notice: RAID does not tolerate any illegal activity. Personal attacks is the biggest problem currently legally. And we have find a way to conveniently and legally resolve it.

RAID is aims not to be a security as much as possible in every way.