SMART CONTRACTS MARKED SAFER

July 9, 2018

Table of Contents

This document is for informational purposes only and does not constitute an offer or solicitation to sell shares or securities in Panvala, ConsenSys Diligence, or any related or associated company. Any such offer or solicitation will be made only by means of a confidential offering memorandum and in accordance with the terms of all applicable securities laws and other laws.

This document is a draft, and provided only as a courtesy and to begin gathering industry and community feedback. This document is not to be considered final, and all information contained herein is subject to change without notice. It should not be relied upon, and shall not confer rights or remedies upon, you or any of your employees, creditors, holders of securities or other equity holders or any other person. To the extent that Panvala, or any related or associated company, someday in the future offers for sale any products or services, including any tokens, you must refer to and review any terms, conditions and disclosures in effect at that time, including any updated versions of this white paper. By accessing this whitepaper, you agree to be bound by the foregoing limitations.

Introduction

The Problem

Software vulnerabilities have existed since the dawn of software. In 1988, the first self-propagating Internet worm used a buffer overflow in the Unix finger daemon to spread from machine to machine. Fast forward to 2018: Even though a multi-billion-dollar software security industry has emerged, the problem is far from solved. In fact, the number of published software vulnerabilities is steadily increasing according to industry participants[1].

Ethereum smart contracts have not been spared their share of coding vulnerabilities. In part, this is caused by design: Ethereum's most popular programming language, Solidity, is easy to pick up, but the semantics of code interacting with the Ethereum Virtual Machine (EVM) are difficult to understand, and there are quite a few compiler quirks developers need to be familiar with. In accordance with Murphy's law, this has resulted in serious smart contract vulnerabilities, along with monetary losses of hundreds of millions USD, such as the widely reported hack of TheDAO in 2016[2].

---

[1] https://resources.flexera.com/web/pdf/Research-SVM-Vulnerability-Review-2017.pdf
[2] http://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/

Centralized Trust

Certifications are used in the IT industry for various purposes, such as attesting the authenticity of cryptographic certificates and attesting code security. Traditionally, such certifications are issued by trusted authorities. For example, Common Criteria certifications are issued by testing labs that are approved by national certification bodies.

The shortcomings of centralized trust are most visible in the X.509 Public Key Infrastructure, which is used today to protect web users from man-in-the-middle attacks. The system is familiar to end users as the “security lock” symbol that is commonly shown next to the address bar in the web browser. In the X.509 scheme, the web server presents a certificate that binds the web server’s DNS name to its public key. To verify authenticity of the certificate, it must be possible to trace a valid certification path to a trusted Certificate Authority (CA). Current web browsers ship with a list of about 1,200 CAs, each of which is trusted to issue valid certificates.

The weak point in this design is the reliance on trusted third parties. The whole system is compromised if a single CA fails to protect its private signing key. This has occurred on multiple occasions [3] [4] [5].

Value Proposition

Smart contracts and decentralized applications on the Ethereum public blockchain enable a new level of security and trust without depending on any centralized institution. However, while smart contracts do not require anyone to be trusted in order to run, they do need to be certified as secure, safe and correctly engineered, just like any other computer program.

Currently, users base their choices of which contracts to use based on the security claims made by the contract authors and their security auditors. In our young ecosystem, there are few trusted voices when it comes to which contracts are safe to use. These recommendations and the reputation that backs them up are hard for many users to discover, and our decentralized ethos makes it difficult to choose a single source of truth. True to the ideals of a decentralized ecosystem, we believe that this certification can best be done by the community itself.

Panvala is a system that produces a *decentralized security assurance registry* for Ethereum smart contracts. It is based on a Token-Curated Registry (TCR), curated by token holders who base their opinions on the findings of automated tools and human auditors. Panvala provides a process for contracts to earn “assurance marks” by convincing the token holders that the contracts are safer to use. Contract authors or their auditors may propose contracts for a

---

[3] https://arstechnica.com/information-technology/2017/01/already-on-probation-symantec-issues-more-illegit-https-certificates/

[4] https://threatpost.com/final-report-diginotar-hack-shows-total-compromise-ca-servers-103112/77170/

[5] https://www.cnet.com/news/behind-comodo-hack-an-insecure-web-roundup/

Panvala mark, the indicator that the contracts meet the standards of the Panvala participants. Panvala participants reach consensus on the approval of each application through processes they choose, backed by a self-regulating crypto-economic system that disincentivizes decision-making processes that hurt the value of the system. Client software, such as MetaMask, may leverage Panvala by displaying the assurance status in the form of an easily comprehensible badge, and displaying a warning if the user tries to access a contract that has not been marked safer by Panvala.

To promote the progress of security practices in the Ethereum ecosystem, the Panvala token holders can increase the number of Panvala marks that a smart contract system can earn. Once enough contracts have earned one Panvala mark, allowing contracts to earn additional Panvala marks creates an opportunity for contract authors and auditors to demonstrate exceptional security practices among their peers.

Panvala incentivizes high security standards for smart contracts deployed on the Ethereum blockchain, with the following benefits:

- Improves trust in the Ethereum blockchain, which is a requirement for wide adoption;
- Creates a healthy market for security tools and services;
- Helps end users protect themselves from hacks and financial loss;
- Creates an incentive for responsible disclosure of security flaws;
- Facilitates collaboration within the Ethereum security ecosystem and community;
- Provides contract owners a minimally subjective means of demonstrating that a smart contract has undergone proper verification.
- Enables applications, such as MetaMask, to add value-add features around security signalling for end users, and transaction safety.
- Encourages the development of standards and best practices within the Ethereum ecosystem

The Registry

A token-curated registry (TCR) is a smart contract for mapping contract addresses or bytecode to attestation status. In Panvala, the registry entries are collections of related smart contracts, just like the set of smart contracts that are included in a typical audit of a smart contract system. Contracts within a system can be referenced by individual contract addresses, factories that generate multiple contracts, or contracts deployed with the same transaction data.

The contents of the TCR are *curated* by token holders. In principle, tokens can be held by anyone, but we expect tokens to be held mostly by smart contract developers, smart contract auditors, and Ethereum enthusiasts who are confident in their ability to critically analyze claims teams make about their smart contracts. Token holders who uncritically vote with the crowd would make it easy for Panvala to be out-competed by more discerning registries.

Achievement Levels

Panvala maintains a ladder of achievement levels for contract systems to work their way up. Like Michelin stars for restaurants, a contract system has to earn one mark before it can try to earn a second mark. Applicants to the registry must deposit the required amount of Panvala tokens (PNV), and demonstrate that their contract system fulfills the requirements for the desired level.

Panvala will be initialized with a single registry for issuing one Panvala mark. Below, we suggest TCR parameters for a second Panvala mark as well. The levels can differ by the size of the deposit and the duration of the apply and commit stages. The technical requirements to reach a particular level are determined by the voters to maximize the influence of the registry.

It is important to note that none of the parameters and requirements are set in stone. Instead, they will evolve over time through the Panvala governance process described further below.

Assurance Level	**Requirements**	**Initial Parameters**
1. One mark safer	Adheres to baseline security standards that satisfy the Panvala participants.	MIN_DEPOSIT = 50,000 PNV APPLY_STAGE_LEN = 1 month COMMIT_PERIOD_LEN = 1 month REVEAL_PERIOD_LEN = 1 week DISPENSATION_PCT = 50 VOTE_QUORUM = 50

2. Two marks safer	Exceeds the security practices of most contracts with one Panvala mark.	MIN_DEPOSIT = 100,000 PNV APPLY_STAGE_LEN = 1 month COMMIT_PERIOD_LEN = 2 month REVEAL_PERIOD_LEN = 1 week DISPENSATION_PCT = 50 VOTE_QUORUM = 50

Multi-Identifier Applications

The token-curated registry smart contracts in typical TCRs use self-describing identifiers. For instance, AdChain uses the domain name of the applying site as the entire application. A domain name can be enough information for AdToken holders to determine if the site meets their criteria, and the name itself can be used as the key to look up entries in the registry.

In Panvala, an application can contain multiple contract identifiers that represent the bytecode, contract address, or contract factory used to deploy each contract in the application. While we can generate a single identifier for each application to represent all of its contents, those identifiers aren't self-describing—they won't be known when users are interacting with contracts themselves, so they can't be used as the lookup key.

As a result, Panvala's token-curated registry contracts expand the application data to include identifiers for each contract, as well as an identifier for the whole application (typically an IPFS hash of a document that includes both the individual contract identifiers and the content used to justify the application). As a result, the data that is available when evaluating a given contract is enough to generate an identifier that can query the Panvala mark registry.

Registry Schema

The following is a draft schema for Panvala applications.

Application

- Assurance defense (e.g. an audit report, source code, compiler versions, and optimizer settings)
- Proposal champion address
- One or more "Individual Contract," "Contract Factory," or "Contract Creation Code" entries

Individual Contract

- Contract address
- Network identifier

- Contract name
- Contract ABI

Contract Factory

- Factory address
- Network identifier
- Stored code hash of created contracts
- Creation event definition
- Factory name
- Factory ABI
- Created contract name
- Created contract ABI

Contract Creation Code

- Stored code hash of created contracts
- Transaction data prefix
- Contract name
- Contract ABI

The content for each registry application should be published as a JSON document on a distributed file system like IPFS.

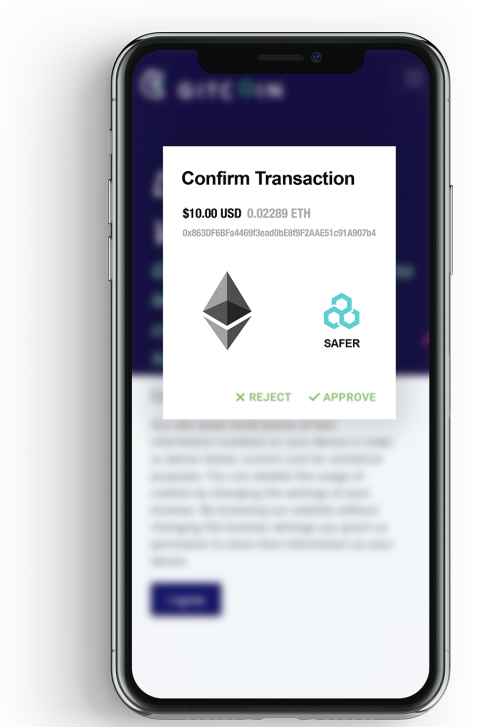
Evaluating Transactions

Determining whether a single contract has been marked safer by Panvala is straightforward, but most users send transactions that produce a call stack involving multiple contracts. Panvala aims to provide actionable information for users seeking information about transaction safety when multiple contracts are involved.

To evaluate a transaction's safety using Panvala, clients should simulate the transaction on the latest available block using the light client protocol and debugging features in a local EVM, like `ethereumjs-vm`. Each contract executed during the transaction should be queried in the Panvala Registry, and the lowest common mark status for the set of contracts should be used. That is, if any executed contract is not present in the registry, the transaction should be treated as unverified by Panvala. Information about each contract called by a contract can be presented in detail for advanced users.

Since the call stack depends on the unknowable state of the blockchain at execution time, it isn't possible to guarantee that a specific set of contracts will be executed. However, providing the user with an approximation of what will occur is valuable information to decide whether to execute the transaction.

User Interfaces



While Panvala token holders determine the content of the Panvala Registry, few users will directly query the registry on their own. Instead, the applications they are already using to browse and transact can integrate data from the Panvala Registry to help their users make safer decisions.

To make safer choices, users need to be able to tell if the contracts they're interacting with have been reviewed by someone they trust, not just the destination of the transaction and how much ether it will move. Our initial design goal has been to make it as easy to make decisions about whether a transaction is safer as it is to tell if a website asking for your credit card is encrypting that data. However, each application developer will make their own decisions about how to integrate Panvala.

Application developers will have easy access to the current state of the Panvala Registry, and can query for individual contracts to see if they have a Panvala mark. For some applications, historical data will be helpful to display, which will require indexing the history of the Panvala Registry contract to note when contracts were added or removed from the registry.

The Token Capacitor

The Panvala Registry will be functional from day one, but like all systems, ongoing work will be required to maintain and improve the system. Some systems rely on volunteers to do this work, like Bitcoin. Other systems fund a company or a foundation to perform ongoing work that improves the system. Instead, Panvala uses its own tokens to incentivize people to cooperate to improve the system.

Panvala uses a *token capacitor* to coordinate rewards for ongoing work that improves the system. The overall Panvala token supply is fixed, so the token supply can't be inflated to fund ongoing work. The token capacitor is initialized with a portion of the fixed supply that it releases at a decreasing rate over time to destinations controlled by the token holders. The release rate is determined by the current balance of the capacitor: the higher the balance, the higher the release rate.

In systems without a token capacitor, increases in demand for a token are distributed as gains for token holders. Token capacitors have effects similar to property taxes: the effects of demand changes are divided between token holders and the effective budget for funding improvements to the system.

Releasing Tokens

When tokens are released, any token holder can place a deposit to propose a destination for the tokens. Like the registry itself, any token holder can challenge the destination by matching the deposit. The remaining tokens vote to approve or reject the release of tokens.

Each batch of released tokens can only have one destination. A smart contract can be used as the destination to divide the tokens among several recipients. Proposers are encouraged to seek a champion when they want to direct tokens from the capacitor so stable decision-making processes for allocating tokens can develop over time.

The token capacitor is configured with a half-life that determines how quickly tokens are released, and a release frequency that determines how often tokens are released. For example, with a half-life of two years, one half of the tokens will remain after two years, and one fourth will remain after four years. A release frequency of three months will release half of the tokens in eight batches over two years.

Depositing Tokens

Token holders can donate tokens to the capacitor to increase and extend its token releases. The release rate of the capacitor is based on the current balance of the capacitor, so depositing tokens can be viewed as recharging the capacitor back to where it was at an earlier time.

By default, the flow of tokens out of the capacitor will dwindle, along with the system's ability to incentivize cooperation. If the Panvala participants can find the funders of the system's common needs and encourage them to make deposits that keep the capacitor charged, they can provide a signal that cooperation will continue to be rewarded.

Making Decisions

Panvala is a decentralized system with *subjective* goals. The objective goals of decentralized blockchains allow anyone to fire up a miner and contribute in proportion to their resources, but subjective goals give decentralized systems a more democratic character: your influence is determined not just by your contributions, but by the confidence the rest of the participants have in your actions. The most important benefit Panvala gains from decentralization is that it is an agent-free system. Rather than appointing people to act on behalf of the system, anyone can act on behalf of the system to change parameters, add registry entries, or direct tokens from the

capacitor. Token holders can challenge actions to trigger a vote where they can reject any action.

Token-curated registries and token capacitors incentivize participants to approve proposals that will make the system more useful, and to reject proposals that will make it less useful. However, the more subjective the goals of the system are, the less certainty applicants will have about their proposals. Uncertainty will likely lead to fewer proposals and a less useful system. Using the challenge process as the primary mechanism for achieving consensus is expensive and contentious.

For this reason, we don't view the Panvala smart contracts as the way decisions are made. The smart contracts are just the mechanism token holders use to reject decision-making processes they no longer support. These decision-making processes are represented by the *champion* for each proposal. Since token holders can support multiple champions at a time, the processes for making decisions can evolve through competition during times of uncertainty rather than enduring the discord of forks or revolutions as the cost of progress.

Proposal Champions

In Panvala, applicants are encouraged to seek a champion for their proposals who have earned the confidence of the community. This allows deliberation to begin far before any tokens have been put at risk for the application's deposit. Champions evaluate applications and demonstrate to the token holders how they arrived at a decision about whether to recommend accepting or rejecting the proposal. Evaluating an application to recommend a decision requires a significant amount of work. Champions will likely seek compensation from applicants to cover these costs.

Challenging an application in Panvala doesn't just call into question the safety of the contracts in the application. Voters are also voicing their opinion about their confidence in the application's champion and their process for determining that contracts are safer. Champions have no power in the protocol itself: if they recommend applications or use a decision-making process that the voters disagree with, token holders can challenge the application to trigger a confidence vote on the champion's recommendations. A champion whose proposals do not predictably survive confidence votes serves no purpose, so a failed confidence vote is a strong signal to revamp the processes by which a champion makes their recommendations. Each proposal can only have one champion, but multiple trusted voices in the community can define their own process for deciding which proposals to champion together.

By judging champions along with the contracts themselves, Panvala creates a second incentive layer. Instead of just seeking quality registry entries, Panvala token holders also seek quality recommendation processes that can earn and maintain their confidence through difficult decisions. Good proposals can and should be rejected if the token holders do not have confidence in their champion.

Here are some examples of how people might take on roles as champions. Token holders are free to support the proposals of any champions they choose as individuals.

Independent Champions

A security researcher, for instance, can act as an champion by publishing independent recommendations of whether to accept or reject applications. Building up trust from scratch without a widely recognized name in the community would probably be difficult, but in the absence of better ways of reaching consensus on applications, token holders might put their trust in the loudest person that presents a path forward.

Auditing Firms as Champions

Since most developers get their smart contracts reviewed by independent auditors, they can just ask their auditor to recommend their application to the token holders. However, it's already difficult to know which auditing firms are worth trusting and which will maintain their quality over time. Rather than evaluate every new auditing firm, token holders might just put their trust in one firm with an established track record, which could steer the industry toward monopoly. Token holder should seek to maintain a healthy competitive landscape when they choose which champions to support.

Representative Champions

The most straightforward way for a champion to gain the trust of token holders is for the token holders to elect a body to champion proposals. It's still easy for an elected body to lose the trust of token holders with poor decisions or an opaque decision-making process, and the token holders can always challenge applications even though they were championed by people they elected.

The process for electing a representative body has important effects on the incentives of token holders. The token holders might want an explicitly partisan council that represents many views proportionally, or it might want a unified council that is more responsive to outside competition than internal disagreements.

Application and Curation Process

Champion Stage

Before interacting with the registry contracts, potential applicants can seek a champion for their application. Anyone can champion an application, but to be successful, applicants should seek a champion whose decision-making processes have earned them the confidence of the community. If an applicant makes a proposal without a champion, the applicant is considered the champion.

Application Stage

Once a development team feels that their contracts have been thoroughly reviewed and audited, they can decide to apply for a Panvala mark. The applicant will stake the required number of Panvala tokens for the number of marks they are applying for to the registry contract, where the tokens will be held for the lifetime of the registry entry. These tokens will be lost if the application is successfully challenged at any time. If the application succeeds, the applicant can remove their entry from the registry to reclaim the tokens.

During the specified application stage, any token holder can challenge the application by matching the applicant's deposit of Panvala tokens. This triggers a confidence vote of all token holders to accept or reject the proposal.

Commit Stage

To vote, a token holder locks their tokens and commits to a hash of their vote while keeping their vote secret. This encourages token holders to come to independent decisions without being able to prove how other people already voted.

Reveal Stage

Once the commit stage is over, no more voting decisions can be made. Users who have committed to a vote can reveal their choice, which unlocks their tokens and tallies their votes. When the reveal stage ends, the application succeeds if it exceeds the vote threshold. The losing deposit is forfeited to the winning side and their voters, who divide the deposit as configured.

Acceptance

If no challenge occurs within the application period or the challenge fails, then the application is granted a Panvala mark.

Confidence Votes

A confidence vote can and should be triggered by a token holder who believes that a proposal's champion no longer has the confidence of the token holders. If the token holders reject a proposal, future proposers will consider the champion to be unreliable, and will likely seek other champions. Confidence votes can come at a cost to the system: if it's unclear how the token holders intend to make decisions, proposals will become riskier, so people will be reluctant to make proposals.

Confidence votes initially require at least half of the votes to succeed. If fewer than half of the votes support the proposal, it will be rejected. Token holders control the support threshold and can change it if they believe a different threshold will lead to better results. Voters on the

majority side of a confidence vote are rewarded with a portion of the proposer or the challenger's deposit depending on which side lost.

Token holders should be ready and willing to participate in every confidence vote. Low voter turnout can make decisions more unpredictable, and can make the system vulnerable to takeover by a small bloc of active voters. Token holders should consider improvements to the system that make voting easier, and adjust voter rewards to achieve the level of voter turnout they desire.

Responsible Disclosure

If the contracts within a successful Panvala mark application are later found to have security flaws, they can and should be challenged by any token holder who stakes the required amount of tokens. The application deposit gives security researchers an incentive to disclose security flaws instead of exploiting them. In addition, token holders can use their control over the release of the deposit to encourage vulnerabilities to be disclosed responsibly by rewarding challengers who disclose vulnerabilities responsibly and withholding rewards from those who disclose irresponsibly. Token holders should maintain voting periods that are long enough for security researchers to privately disclose flaws so teams can coordinate responses.

Coordinating vulnerability response is more difficult than evaluating new applications. New applicants can hold off on putting their deposit at risk until they earn the support of a champion. But vulnerabilities are time-sensitive: if you wait too long, someone else might challenge the entry before you do and claim the bounty for themselves.

When someone is confident that there is a flaw in a system that has a Panvala mark, they should immediately post a deposit to challenge the entry in the Panvala Registry smart contract. Once the challenge has been initiated, they should work with active champions in the community to responsibly disclose the issue. Token holders set the standards for responsible disclosure through their confidence in their champions. If champions believe the challenger has behaved irresponsibly, they should encourage the community to reject the challenge. The irresponsible reporter will lose their deposit, and the next challenger to place a deposit will be able to easily claim the original application deposit if the community has already been convinced that the flaw exists.

If the challenger is correct that the new vulnerability makes the contract system unworthy of a Panvala mark *and* the challenger responsibly disclosed the vulnerability, the token holders should support the challenge so the challenger can collect their reward from the application deposit. Consistent voter turnout is essential to making responsible disclosure a predictable and attractive process to participate in.

Replacing Contracts

The canonical smart contracts to refer to for the status of the Panvala system cannot be enforced by smart contracts themselves. Instead, the contracts people refer to are the result of consensus among participants, application developers, and individuals querying the registry. To help establish consensus for replacement contracts, proposals can be made with the address of a new contract to refer to. These proposals are subject to the same challenges and confidence votes as any other proposal. Accepted proposals for contract changes aren't guaranteed to be respected by all participants, but they are a clear signal of consensus that participants should voluntarily adhere to.

Proposal Statuses

Proposals can change system parameters, add registry entries, or direct tokens from the token capacitor.

- **Pending**: A proposal has been made, but has not been challenged. The proposal will succeed if challenge period expires without a challenge.
- **Confidence Vote Underway**: The proposal has been challenged. Token holders must vote to express or deny confidence in the champion of the proposal and their decision-making process. If the confidence vote fails, the proposers deposit will be split between the challenger and the votes in the majority. Otherwise, the challenger's deposit will be split between the proposer and the votes in the majority.
- **Accepted**: The proposal has been accepted and it has taken effect.
- **Rejected**: The proposal has been rejected.

Panvala Mark Statuses

- **Unverified**: The contract has not been evaluated by Panvala token holders.
- **Panvala Mark(s):** The contract has earned one or more Panvala marks.
- **Vulnerability Disclosed:** The contract has earned Panvala marks, but a challenge is underway.
- **Not Secure:** The contract once had Panvala marks, but lost them due to a challenge.

Community Roadmap

The initiating team will build the smart contracts and user interface for token holders to make proposals, challenge proposals, and cast their votes. This software will be ready for token holders to use soon after the tokens are distributed.

From there, the community will continue to improve the system over time. Anyone can take the initiative to do work that makes Panvala better. After completing that work, contributors are encouraged to seek tokens from the capacitor as a reward for their work.

Token holders should lean towards rewarding work after contributors take risks that they think they will be rewarded for. It's more predictable to fund work in advance, but allocating all or most of the tokens in advance makes it harder to build an ecosystem where anyone who sees a way to add value can profitably act without permission to make the system better.

Below are some examples of clear opportunities for improvement that the community can take on.

Integration Libraries

The Panvala Registry UI queries the registry contracts for the state of the registry, but applications that intend to integrate with Panvala have different needs than the UI. Support for multiple platforms could be valuable, particular for mobile applications. Additional functionality that assists applications in evaluating entire transactions rather than individual contracts will help application developers display Panvala Registry information in the ways they desire without needing to implement much of the functionality on their own.

Registry Analyzer

Automated contract analysis tools, like Mythril, Manticore, and Oyente, are improving at a steady rate. As more potential vulnerabilities can be detected by automated tools, it will be easier to find flaws in deployed contracts that have already entered the Panvala Registry. Systems that scan the registry with the latest versions of multiple tools can send alerts to Panvala participants who can evaluate the output of the tools and decide whether the report is valid.

The easier it is for publicly available tools to find vulnerabilities, the more work vulnerability researchers will have to do to stay ahead of public tools and claim deposits.

Periodic Voting

Since proposals and confidence votes can occur at any time, token holders must always be ready to participate in decisions. Challenges for existing registry entries must always be possible at any time, but restricting proposals and confidence votes to specific windows can increase voter turnout by limiting the effort necessary to pay attention when decisions are being made.

As the community finds a cadence for decision-making that works for them, they should consider enforcing that cadence at the contract level.

Turnout Incentives

Voters who vote in the majority are currently rewarded with part of the losing deposit. This incentivizes consensus, but it's unclear whether it is more effective to incentivize consensus or

to just incentivize turnout. The community can consider modifications to the registry contracts to reward all voters if they believe that will have more desirable results.

Delegated Voting

Rather than vote in each confidence vote or challenge, many voters would rather choose a delegate to vote on their behalf. Using delegates to organize votes should make voter turnout higher and more stable over time. However, delegated voting potentially has the negative side effect of encouraging factions to form: the easiest way for a delegate to attract more votes is to create discord for them to be on the right side of.

Another use case for delegated voting is to delegate voting rights to a private key that cannot control the tokens themselves. This makes it safer to be an active voter since compromised voting keys will not lead to a loss of tokens.

Panvala Participants

Smart Contract Authors

Active smart contract developers are one of the best sources of information about the current state of smart contract security, and are valuable participants in the Panvala system. Their votes and proposals are informed by their day-to-day experience building smart contract systems for people to use. Smart contract authors will likely propose their own contracts for the Panvala Registry, so many might hold tokens to use as deposits in the future.

Security Auditors and Researchers

Auditors and researchers use their security expertise to help the community define what makes a smart contract system safer. Auditors evaluate a wide range of contract systems and make recommendations for how to make them safer. They are natural champions for the contract systems they review, and together they can set and raise the standards for what it takes for a contract system to be considered safer. Beyond voting, auditors can use their tokens to help their customers apply for the Panvala registry. Putting their own assets at stake to vouch for a contract system makes careful consideration of each system's potential flaws even more important.

User Agent Developers

Developers of Ethereum wallets and block explorers have made it their mission to help users transact on Ethereum. Helping those users make safer decisions is the next step in improving the Ethereum experience for users. These developers aren't just passive consumers of a contract registry controlled by someone else—holding tokens allows them to challenge contract systems that they don't feel are safe enough for their users to interact with.

Ethereum Enthusiasts

Enthusiasts with advanced understanding of smart contract security can put their knowledge to use by evaluating applications and voting accordingly to help keep the ecosystem secure. Keeping up with the latest smart contract systems and their quality doesn't have to just be a hobby. It can be the work you do to make Ethereum safer.

Initiating Team

Panvala is built by ConsenSys Diligence. We actively audit smart contracts and publish smart contract security best practices to advance the level of care we take as a community of smart contract developers. Our mission is to solve smart contract security so people can safely use this groundbreaking technology.

For users to be able to use smart contracts safely, they need trusted, human opinions about which contracts are safer to use. But we don't want ConsenSys Diligence to become a single source of truth about what's safe and what isn't—rebuilding a centralized world is a waste of time. Panvala lets us build a decentralized source of truth so the entire community can cooperate to make Ethereum safer.

Token Supply and Distribution

PNV are used to make proposals and drive voting behavior that determines whether a proposal is accepted or denied. Ownership of PNV does not bestow fractional ownership of any collective digital assets.

PNV are designed for people who intend to write or audit smart contract systems and apply to the Panvala Registry. They are also intended for enthusiasts who are opinionated about smart contract security to cast informed votes that influence the Panvala system.

The total supply of PNV will be 100 million tokens. Half of the token supply will be distributed initially. The other half will be distributed over time by the token capacitor.

The tokens will be distributed among four domains:

- ten million tokens to **security partners**,
    - "Security partners" are active participants in securing the Ethereum ecosystem by auditing contracts, developing security tools, and researching security
- ten million tokens to **user agent partners**,
    - "User agent partners" build apps that people use to interact with arbitrary smart contracts, like MetaMask, Etherscan, Status, Toshi and MyEtherWallet.
- ten million tokens to the **initiating team**, and
- twenty million tokens to **the public**.

Partner Distribution

We will begin accepting applications for security and user agent partners on July 16. Applicants will explain the role they play in the Ethereum ecosystem, how they plan on using the tokens, and the maximum amount they are willing to pay for tokens. Our primary goal is to distribute tokens based on relative impact in the Ethereum ecosystem, not relative willingness to pay. To do this, we will treat each partner's specified maximum amount as a contribution cap, then lower the caps for each partner to more closely approximate our view of their relative impact.

Security partners will participate in a sale for ten million tokens, and user agent partners will participate in a separate sale for ten million tokens. Each participant will be limited to a contribution cap that is equal to or lower than the maximum amount they specified on their application. These sales are designed to achieve a broad distribution among teams that have an impact in the Ethereum ecosystem, but some applicants will not be accepted. We encourage those applicants to participate in the public sale, earn tokens from the token capacitor, or acquire tokens after the sales have completed.

The initiating team will not receive any of the security partner tokens, but user agent projects within ConsenSys, our parent company, can participate in the user agent partner sale. We will publicly communicate how much of the user agent partner sale is allocated to ConsenSys-affiliated user agent projects.

Public Sale

Following the partner sales, we intend to sell twenty million tokens to the public (40% of the initial circulating token supply, 20% of the total supply). The public sale will have the same price cap per token as the partner sales.

Risks

Adoption

Panvala's success depends on many actors using and integrating the system. Without consensus among these actors, the registry will still be functional, but it won't be influential.

Third-Party Custody

Panvala is designed for active participants who hold their own tokens in a wallet they can easily access for voting. However, some token holders might put their tokens on exchanges or allow a service to hold tokens on their behalf for convenience.

When token holders vote, their tokens are locked, so they're exposed to any changes in price that their decisions cause. Letting third parties hold tokens allows them to vote without price

exposure. As a result, third parties can make money by betting against the price of the tokens they hold and voting to damage the system.

Participants are encouraged to hold their own tokens. People who don't intend to hold or use their own tokens should not acquire Panvala tokens.

Game Theoretical Flaws

Token-curated registries incentivize participants to be good curators of a registry, but these systems have not been heavily tested in real world situations. These incentives might break down, or incentives to attack the registry instead of improving it might become clearer over time.

Security Flaws

Panvala runs on Ethereum, a platform for immutable smart contracts. When contracts have bugs or security flaws, they often cannot be mitigated or resolved after deployment. This can lead to the loss of assets or the failure of the system.