RT Voting Audit System

Implementation Phase of
Decentralized Voting Audit Solution
for Latin America
Contest

Submission by RadianceTeam

Introduction

GitHub:
https://github.com/radianceteam/voting-audit

Debots:
https://github.com/radianceteam/voting-audit/tree/master/debots

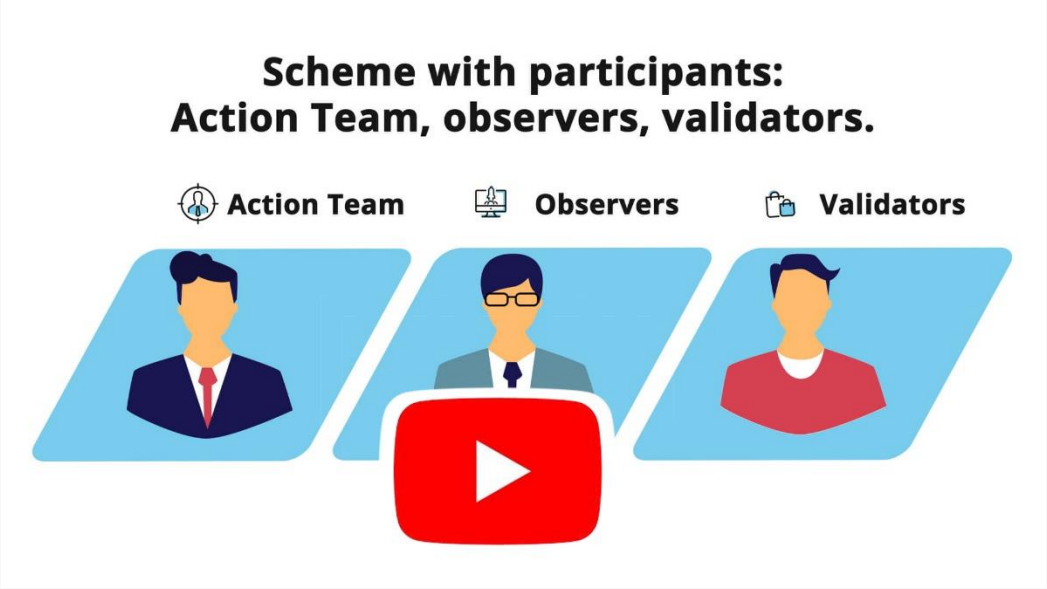
Voting Audits Explorer:
https://voting-audit.defispace.com

Tests:
https://github.com/radianceteam/voting-audit/tree/master/contracts#tests

Smart contracts documentation:
https://github.com/radianceteam/voting-audit/tree/master/contracts#rt-voting-audit-system-10-core-smartcontracts

License: Free software (GNU GPL 2.0)

Video Overview



[View on YouTube]

Team

General manager:	Dmitry Summit
Team manager:	Anzor Daurov
Consultant:	Nick R.
Lead developer:	Yaroslav A.
Developer:	Anzor B.
Developer:	Maxim Z.
Developer:	Dmitry S.

Concept

The demand for voting results audit can be met with a modern solution made using **FreeTON** blockchain.
The core functionality is the audit of **Act4** (Acta#4 form) protocols from the voting centers with the use of crowdsourcing to make the procedure democratic, accessible and transparent.

Make the voting transparent by crowdsourcing the results validation process.

Abstract

Innovative solution based on **FreeTON** blockchain to assist in voting results audit.

Vision

To create a universal framework for voting results audit and assist in making elections democratic and fair in different regions of the world.

Mission

To promote the use of blockchain-based solutions for fair and democratic elections.

Overview

A. Background

The auditing process of Guatemala's voting results has off-chain flaws that require a comprehensive on-chain solution. The authors of the current Guatemalan volunteer-based voting audit process have partners in other countries such as El Salvador, Ecuador, Honduras, and several other Latin American countries where voting audits face similar problems. Those problems are based on an old system where the paper is trusted Act4. Pictures of this Act4 are given to volunteers on a flash drive, which means potential manipulation before this information is provided to volunteers. To combat this issue, Carlos Toriello Herrerias, a Guatemalan activist and blockchain enthusiast and an avid proponent of blockchain technology, created an app to help mitigate this problem; however, it is still an imperfect system since the incoming information is potentially compromised.

In November-December 2020, the first phase of a "Decentralized Solution for Voting Audit for Latin America" contest took place, aiming at crowdsourcing solution ideas [1]. Base on the contest's first phase winner's submission 14, a more formal specification was developed to be implemented by contestants (see Addendum).

B. What is voting process?
Voting procedure is divided into several stages. Voters come to different voting centers based on location. Voting centers provide data for national elections. This data is crucial for the final results of voting as well as to uncover any violations in the voting process. Making this data public increase transparency and allows for easier auditing.

C. What is blockchain and it's advantages?
An idea of using automated software to count votes is obvious but it is prone to attacks from bad actors and requires solid information protection mechanisms. Today one of the examples of such technology is blockchain. Blockchain makes it difficult to tamper with the records and allows to record all the actions in a system to a chain of blocks. Modern blockchain is based on Proof-of-Stake consensus mechanism.

D. How it can be helpful for voting audit?
Theoretically the algorithm is developed to be used in a trustless environment. It creates blocks of data which are reliable enough to create detailed statistical analysis of possible violations of the voting process. This in turn allows to determine the result that is far more reliable than the one produced by the currently used procedure.

E. Analysis of possible violations in voting process.
The main problem of elections is the lack of trust and transparency of the results of voting. The algorithms described above allow to create trust based on the data that hypothetically has zero trust associated with it. In case errors or violations are uncovered the algorithm allows to still input correct data thus creating a complete dataset with absolute trust (in the context of algorithm). The nature of the process also allows to track the actions of observers and other participants to monitor errors and bad actors which further increases trust in the auditing team.

F. Which problems does voting audit solve?
Possible attacks on the system by external agents. Do not allow changing single data items without interfering with other blocks of information, which is provided by blockchain technology [2; p.1 4].
Based on the theory of probability, we can assume that the normal distribution of the turnout, votes, the absence of peaks and other markers may indicate violations of the voting procedure. This factor is a strong point of counterfeiting protection. Isolated facts cannot influence the election results, while significant interventions will leave traces and can be identified by interference markers [2; p.1 5].
The main problem of elections is the lack of trust and transparency of the results of voting. The algorithms described above allow to create trust based on the data that

hypothetically has zero trust associated with it. In case errors or violations are uncovered the algorithm allows to still input correct data thus creating a complete dataset with absolute trust (in the context of algorithm). The nature of the process also allows to track the actions of observers and other participants to monitor errors and bad actors which further increases trust in the auditing team.

G. Participants: action team, collators, validators.

Participant - anyone willing to interact with a Decentralized Voting Audit. To do that one must register as a participant using Voting Audit application. After that they can become one of the following:

Collator (Observer) – anyone who initiated new Act4 verification and locked a stake for it.

Validator – any randomly selected person deployed a validator's smart contract and registered it in DASC following the process described in this specification.

Action Team – a reputable group of people from a country that initiates an DeAudit.

H. Rewards and Slashing.

Rewards - for honest participation in DeAudit are paid out in Democracy Tokens. Validators and Collators should get 2 types of tokens a reward: transferable (DT1) and nontransferable (DT2). If society supports DeAudit then merchants can accept DT1 as payment to show off their civil position.

Slashing (Fines) - when a Validator or a Collator are not supported by the majority of Validators, their stakes will be slashed resulting in loss of the stake. This is mainly a spam protection mechanism which also ensures that both Validators and Collators do their jobs properly.

I. DeAudit Results.

The algorithm allows you to get primary data from polling stations (Act4) with a high level of confidence, confirmed by independent validators. Thus, the formation of general results has a high degree of transparency and public confidence, which significantly increases the democratic value of the voting procedure.

Checklist

Requirements

Hard criteria by contest rules

	Criteria	Status
A	DeAudit smart contract system	**DONE**
B	DeAudit web-based explorer	**DONE**
C	DeBots for all system user interfaces	**DONE**
D	Auto-tests designed as a smart contract OR a script to test Scenarios	**DONE**
E	Free Software license	**DONE**
F	Deployed and tested on the DevNet	**DONE**

Checklist

Evaluation criteria

by contest rules

	Criteria	Status
A	All actions inside a solution should be easily accessible via DeBots interfaces	**Done**
B	Pass the attached tests (a tests should cover all scenarios from requirements)	**Done**
C	The solution should be scalable to millions of participants	**Done ***

***** Every participant is a smart-contract, so the upper limit of participants in the DeAudit system corresponds to the upper limit of deployed smart-contracts in the FreeTON blockchain network.

System Requirements

- For debot interface in debot browsers:
  - iOS, Android, Desktop browser
    * TON.Surf, Web debot browser (preliminary support)

- For debot interface in command-line:
  - MacOS, Windows or UNIX-based OS
  - Tonos-cli of the latest version

Architecture Overview

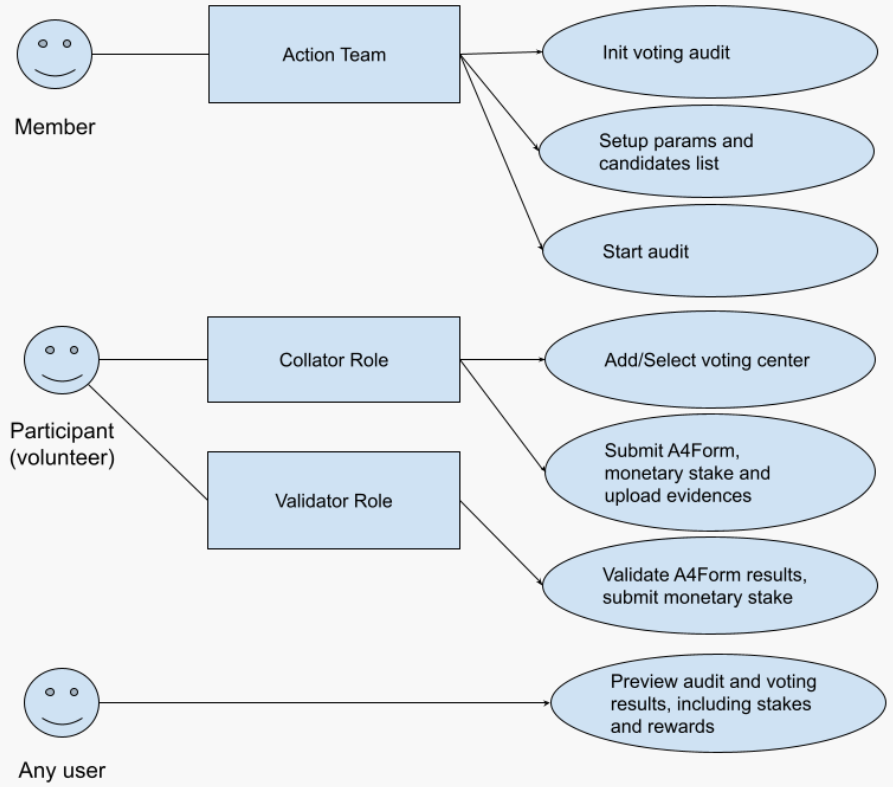


Figure 1. Voting Audits Roles breakdown

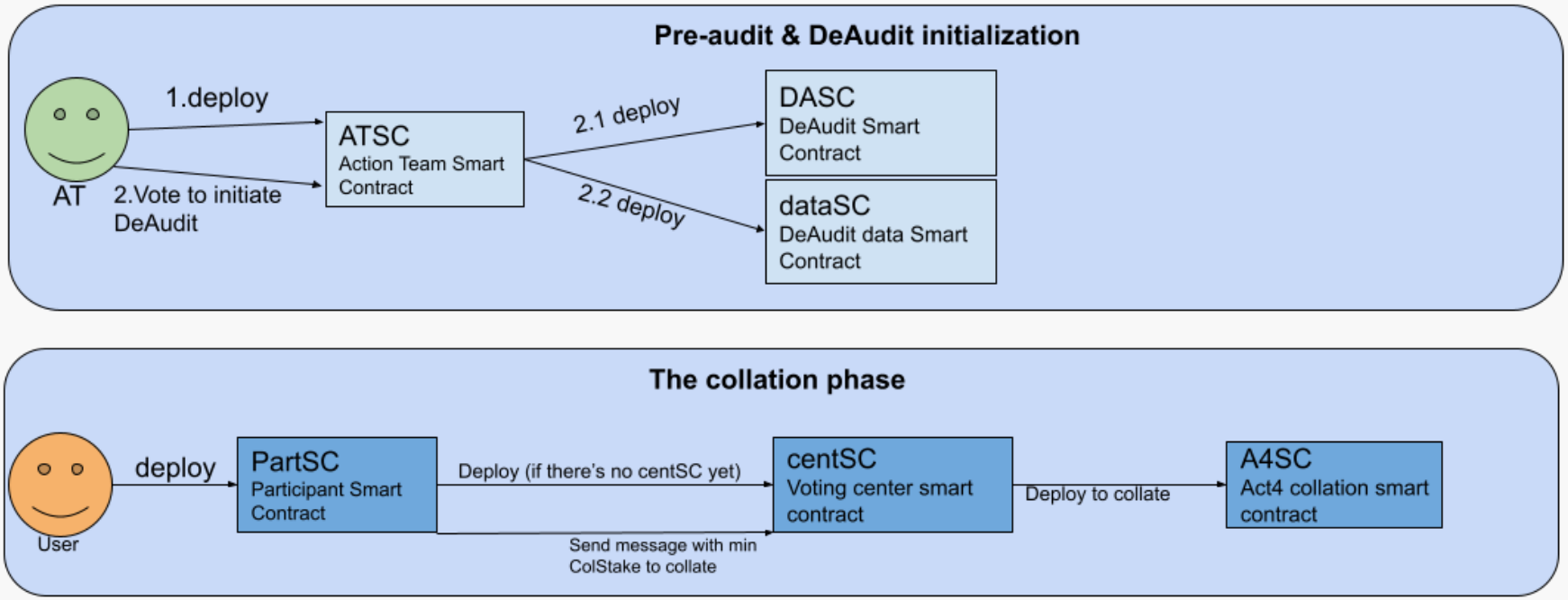


Figure 2. User journey in different roles (part 1)

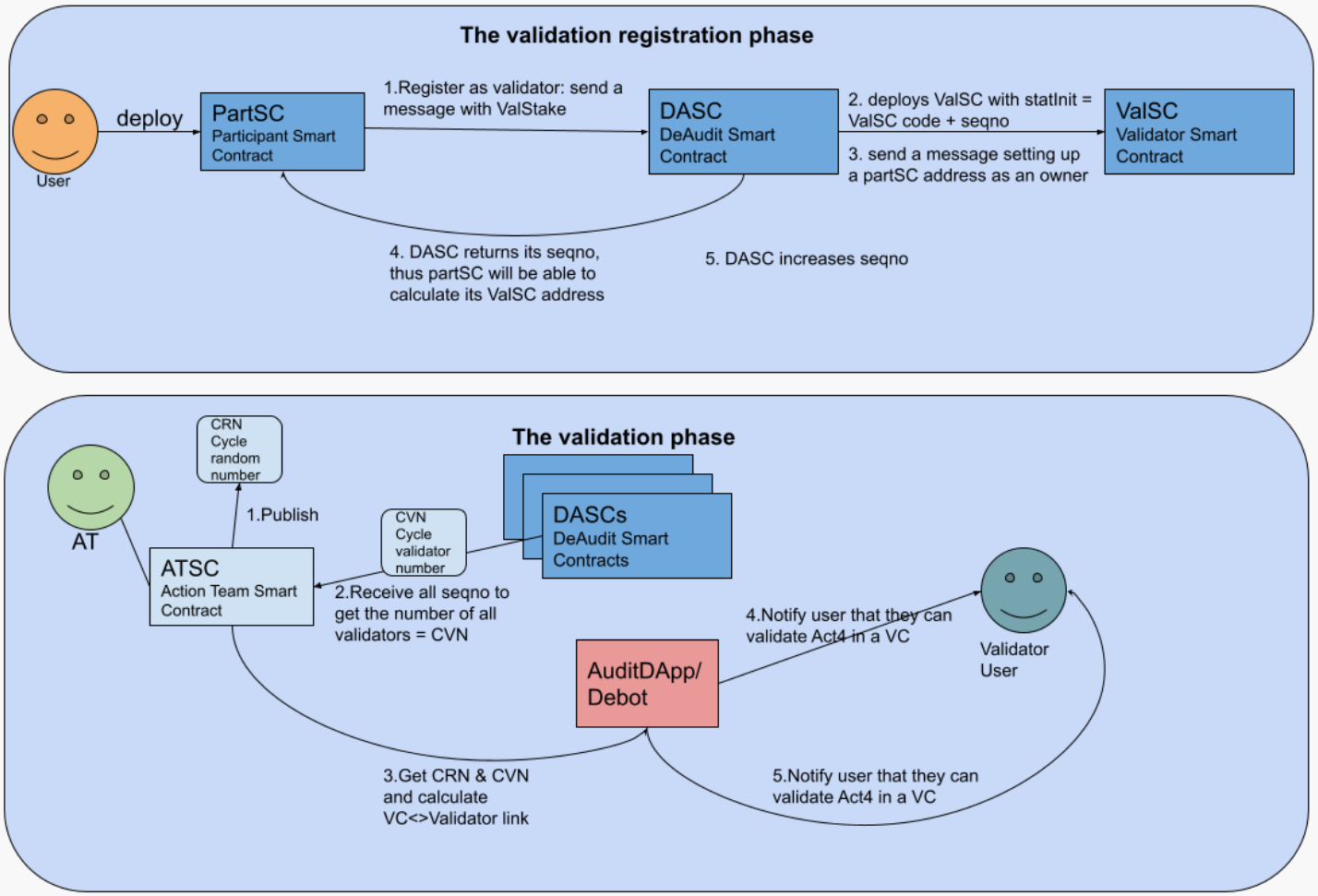


Figure 3. User journey in different roles (part 2)

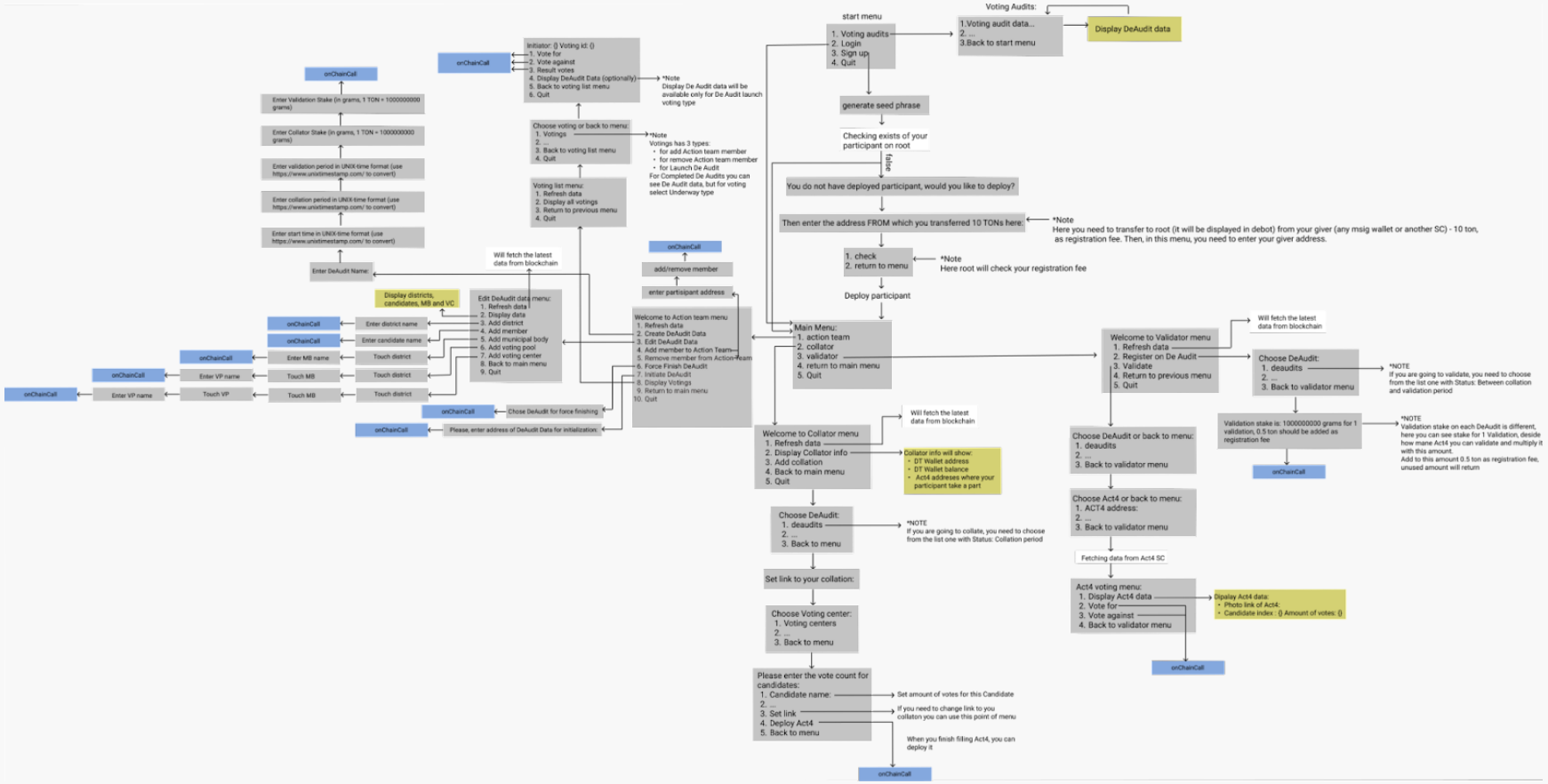


Figure 4. DeBot menu structure

[Click to open in high resolution]

How to use

User guide as test scenarios

The current version of Voting Audit has several interfaces for different tasks: **Web UI** to display stats for all DeAudits and a DeBot available both through DeBot browsers like **TON SURF** and command-line interfaces like **TONOS-CLI**.

To access the **DEBOT** please use this **ADDRESS** on the test network (net.ton.dev):

0:fca49c6cf57f1654988f944f2e9874cb3c86a33dabbc7c3948326fb912dedefa

Debot QR-code for TON.Surf:

You can use the following seed phrases to access this debot:

Keys for testing

Action Team Member:

taxi once course van differ only clutch gossip text gaze giant egg

Participant:

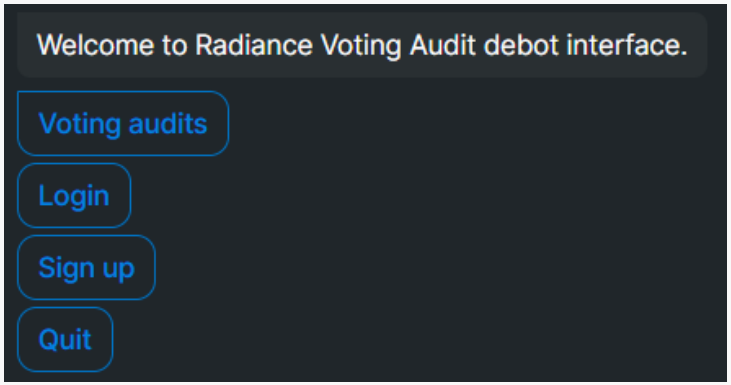
genius glance surge document blur object flash shine tourist disorder kiwi immense

A. Pre-audit process.

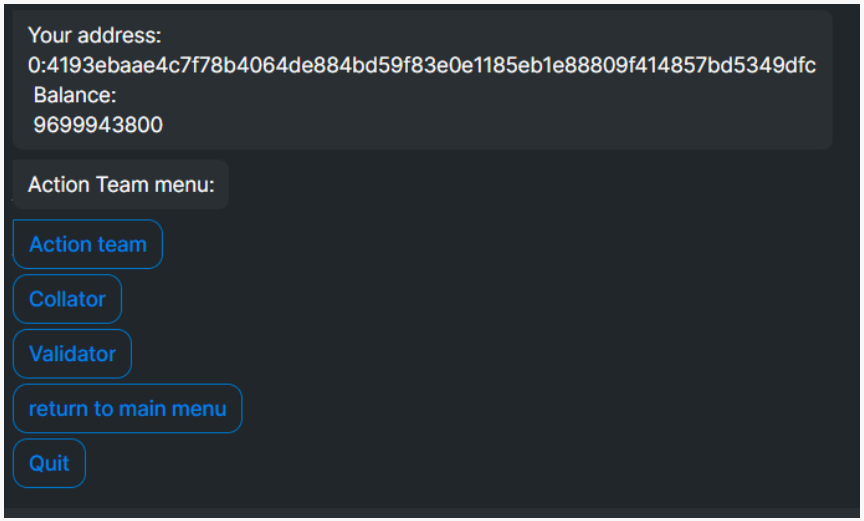
Action Team (AT) should appear and deploy an AT smart contract (ATSC).
The first member of an Action Team deploys a DeAudit Root contract.
After that the initial member can add other Action Team members using the Action Team menu.
We have already deployed a test DeAudit instance to the **net.ton.dev** network that you can access using the debot address above.

B. New Participants.

To fill any role in a DeAudit, be it Action Team Member, Collator, Validator you need to register as a Participant first.

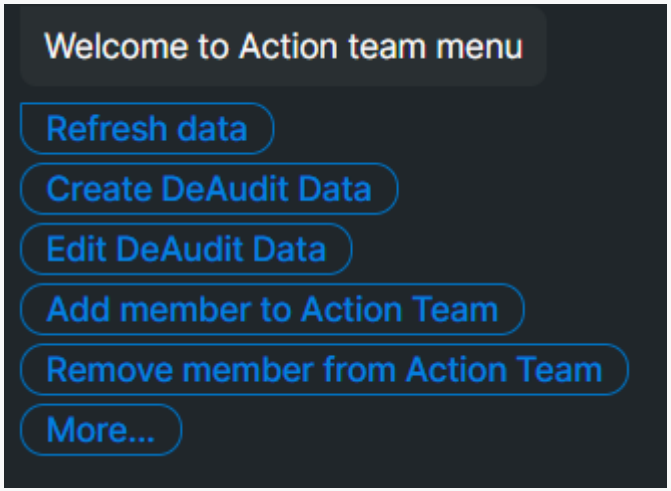


In the main menu choose “Sign up” if this is the first time you use the Radiance Voting Audit interface.
A debot will generate a seed-phrase for you. Write it down in a safe place and then enter it in a debot.
After that Choose “Yes” to deploy a participant smart contract using the seed phrase you provided.

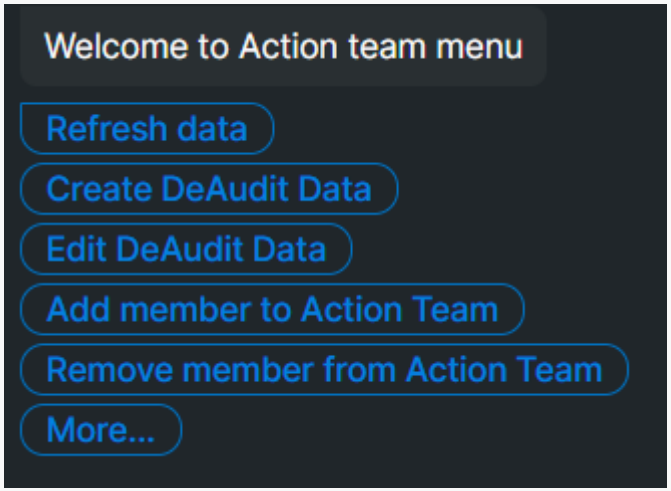


The debot will provide you the address of your newly created participant contract and a root address.
You need to fund root address using any account in the network with 10 TONs and insert the address of the account you used to fund it in the debot.
After you do that your participant address will be deployed and ready for work.

C. DeAudit initialization. DeAudit process could be launched by ATs voting.



If you are an Action Team member (either you deployed the DeAudit or were added to the Action Team by the person who did that), you can access the Action Team menu and perform several administrative functions.



If you choose to Create DeAudit Data you will need to enter the DeAudit name first

Then enter DeAudit Start Time in UNIX-time format. You can use https://www.unixtimestamp.com to convert your time into UNIX-time.
Enter Collation Period of DeAudit in seconds. 3600 = 1 hour.
Enter Validation Period of DeAudit in seconds.
Enter Collator stake in nanoTONs. 1 TON = 1 000 000 000 nanoTONs. This defines the amount each Collator has to stake every time he/she creates a collation.

Enter Validator stake in nanoTONs. This defines the amount each Validator has to stake for every collation he/she is willing to validate.
Sign the transaction with your seed phrase.

D. The collation phase.

During this phase Collations can be created. This is also the phase during which Validators can register for DeAudit to be later selected by the algorithm.
To become a collator you can go to the Collator menu from the Main Menu and choose “Add collation”.
Choose a DeAudit you want to add collation to.
After that you can provide link to online storage of collation materials (photo).
Next step is to choose a voting center and fill the vote counts for each of the candidates according to data in the Act4, photo of which you provided in the previous step.
After that you can Deploy Act4 and sign transaction to finish the collation.
Your collation will be available for validation in the next phase.

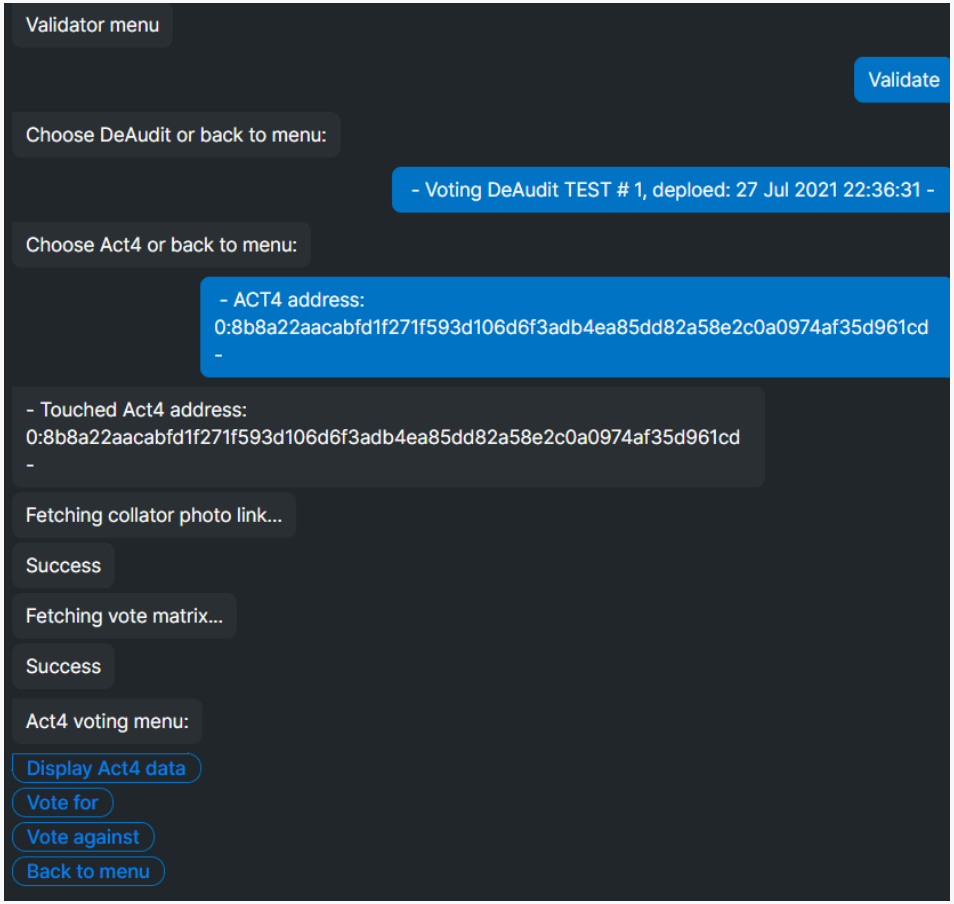
E. The validation phase. Registration. AuditApp and/or DeBot.

You can register as a validator during the collation phase of a DeAudit. First choose “register on DeAudit”, select the amount of coins you’re willing to stake plus 0.5 TON as a registration fee. Based on this amount, you will be registered for a certain amount of validations: the more coins - the more validations available. So if a required Validator Stake parameter was set to 1 000 000 000 nanoTONs and you select that you want to stake 5 500 000 000 nanoTONs, you will be eligible for 5 validations.

F. The validation phase. Validation.

A user can select one valid collated Act4 and confirm that data in A4SC corresponds to a photo or select to reject all collated Act4.

When the validation phase starts every validator is assigned voting centers based on how many coins were staked. To validate, choose the “Validator” from the main menu, and send “Validate”, then choose a voting audit and corresponding Act4 protocol collation for validation. You will be provided with data from the collation to review and validate.



This data includes the list of candidates along with their vote counts and a link to photo materials of the chosen Act4. Choose “Display Act4 data” to see the materials and vote counts, review them and have the ability to vote for or against a collation. After you’ve voted you will have to sign the transaction with your seed phrase. This has to be done for every validation you perform.

- Voting Audit System internally supports 3 types of logic (see DeAuditRoot.sol, Act4.sol):

  - Simple majority voting
  - Soft majority voting
  - Super majority voting

Selector for this logic is in DeAudit Root contract: https://github.com/radianceteam/voting-audit/blob/master/contracts/DeAuditRoot.sol

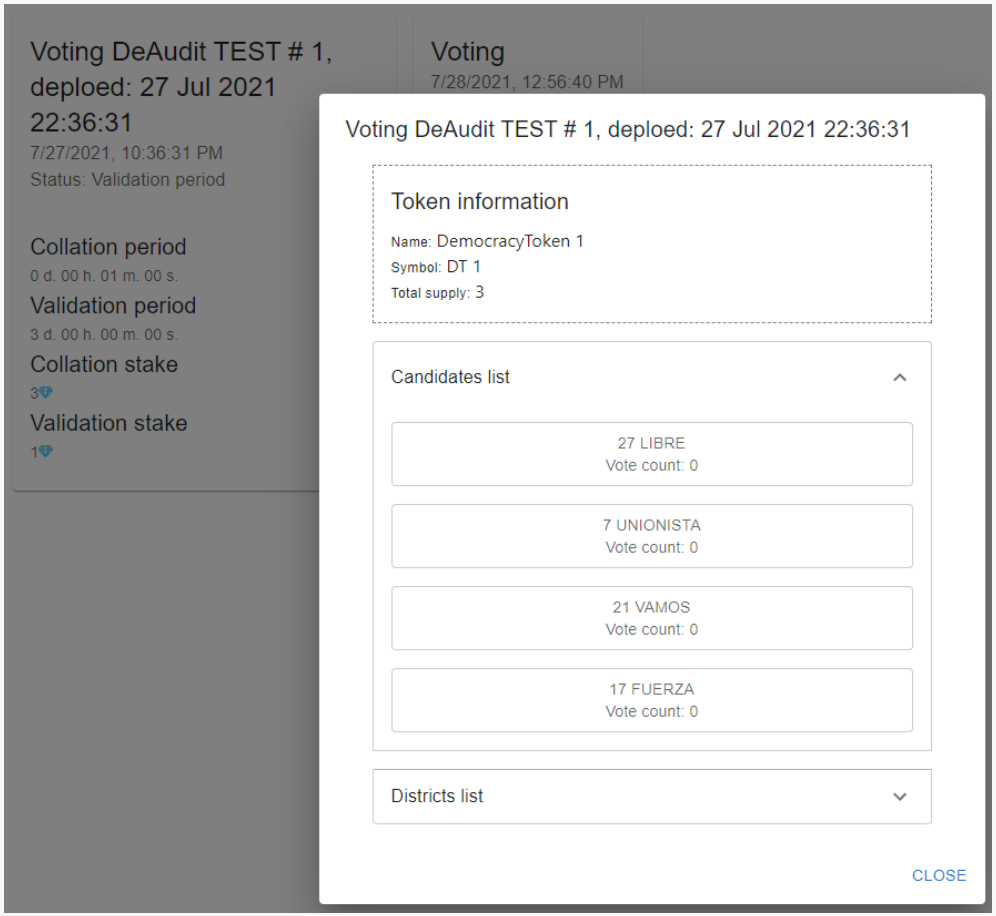
G. Collators and validators interim slashing.
Slashing happens to stakes that belong to collators and validators that were voted against by the majority during the validation phase. So if a majority of validators decide that a collation is not done well, both the collator and validators that voted for this collation will be slashed and vice versa.

H. Collation-validation cycles. Selection of one Act4 for each Voting Center.
Validators are randomly assigned collations for voting. There is a parameter in DeAudit Root contract (https://github.com/radianceteam/voting-audit/blob/master/contracts/DeAuditRoot.sol) that can also regulate the upper limit of validations per 1 validator – function setLimitVFC(uint128 settingLimitVFC) public checkOwnerAndAccept

I. Reward phase.
Validators and Collators that weren't slashed will receive special tokens issued by the DeAudit root as well as their deposits. Action Team is free to change the parameters of this token minting and assign value to the token in any way they see fit.

J. Audit slashing. AT can vote to initiate a DeAudit slashing.
The concept of Audit Slashing requires further development since the current design proposed in the contest description can lead to abuse of the system. For example, if one political party that started the DeAudit process decides to stop DeAudit for its own reasons (whatever they are), many participants, which invested their time and possibly money, will be deceived.

K. Audit explorer.



Minimal visualization capabilities of AuditDapp and/or DeBot.
We have developed a web interface to allow anyone to easily review existing Voting Audits. You can access it here:
https://voting-audit.defispace.com

Roadmap for the next stages

1	Improve integration with TON Surf and other Debot Browsers
2	Complete Web UI for all roles and functions
3	Statistical analysis of results
4	Implementation of anonymous (zk-SNARK based) voting support
5	Self-sovereign identity integration

References

1. The first phase of a “Decentralized Solution for Voting Audit for Latin America” contest.

   https://gov.freeton.org/proposal?proposalAddress=0%3Ae4cdeb29d95d940ead30fd7ce93db4c6f6397c4ae1bd6ee6814b5c07612839ec

2. “Challenge MIT/Harvard paper on Blockchain Faults in Election Systems”. The contest’s first phase winner’s submission 41.

   https://gov.freeton.org/submission?proposalAddress=0%3A06edf25391f9bf003546f2332fa22f3aa525c5ae88e3736980c6a1a9c96705ec&submissionId=41

Contacts

Contact:

https://t.me/UltraNihilist

https://t.me/dnugget

Wallet:
0:dda08e34a2ea623b2b7b3d448c10c7b1ab3a3df6b1525561214d4c8f907b5ce7