Upgrading Aragon Voting

Jorge Izquierdo – Aragon One, CTO

EthCC – March 6th, 2019

CONSENSYS

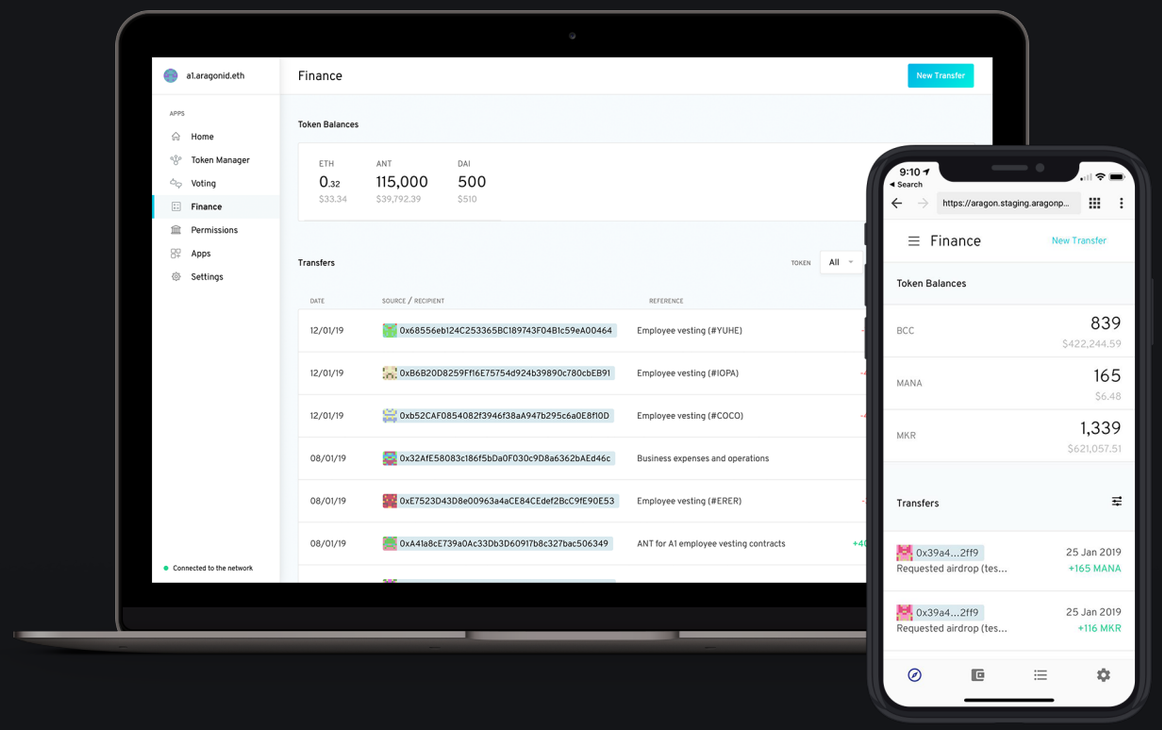
ASSETH

Ethereum **C**ommunity **C**onference

March 8 -10, 2018 | Paris, France

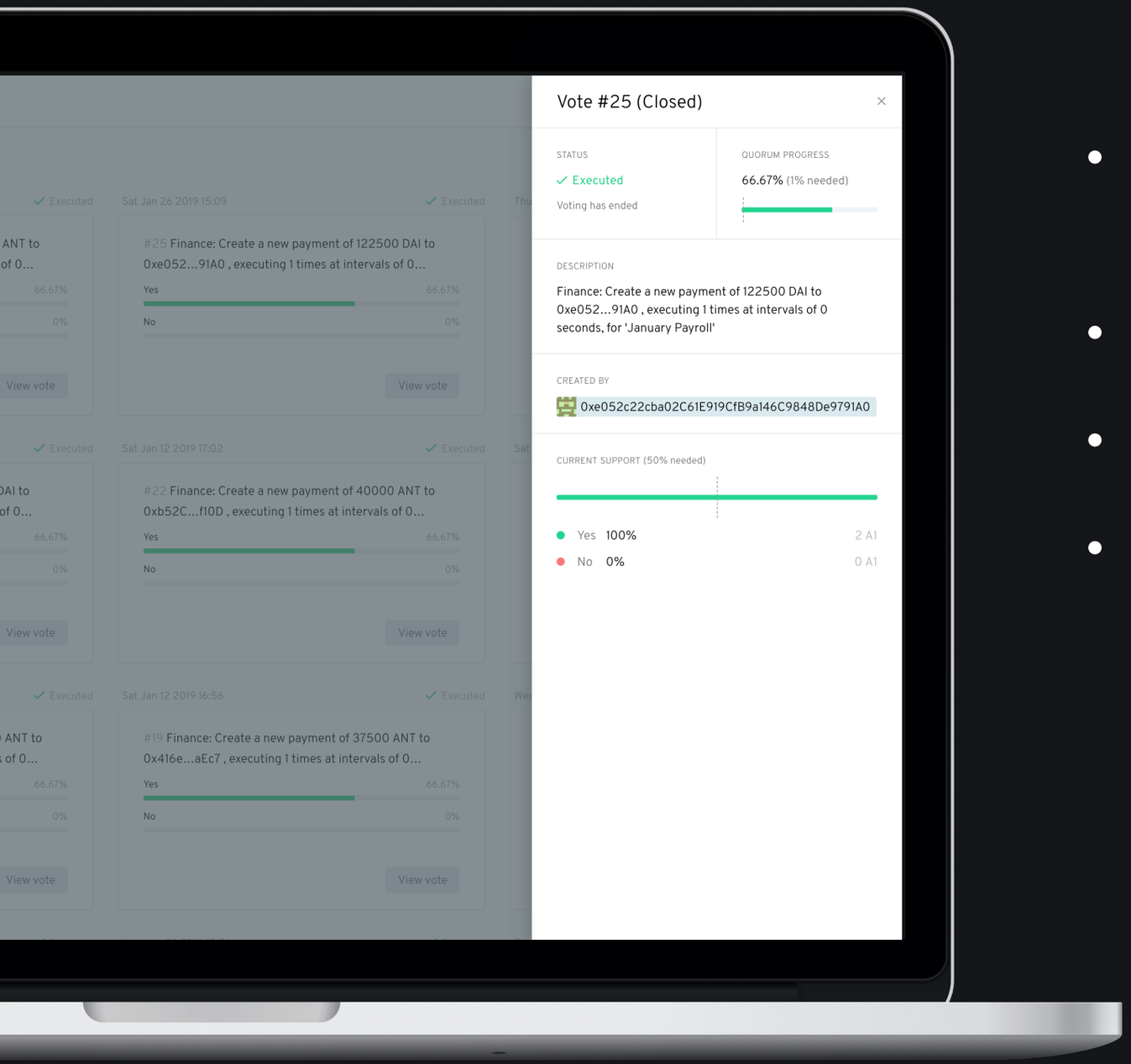
Aragon

- DApp to create and manage decentralized organizations
- Built on the Ethereum VM
- Available on the web, desktop and mobile
- 353 DAOs running on the Ethereum mainnet



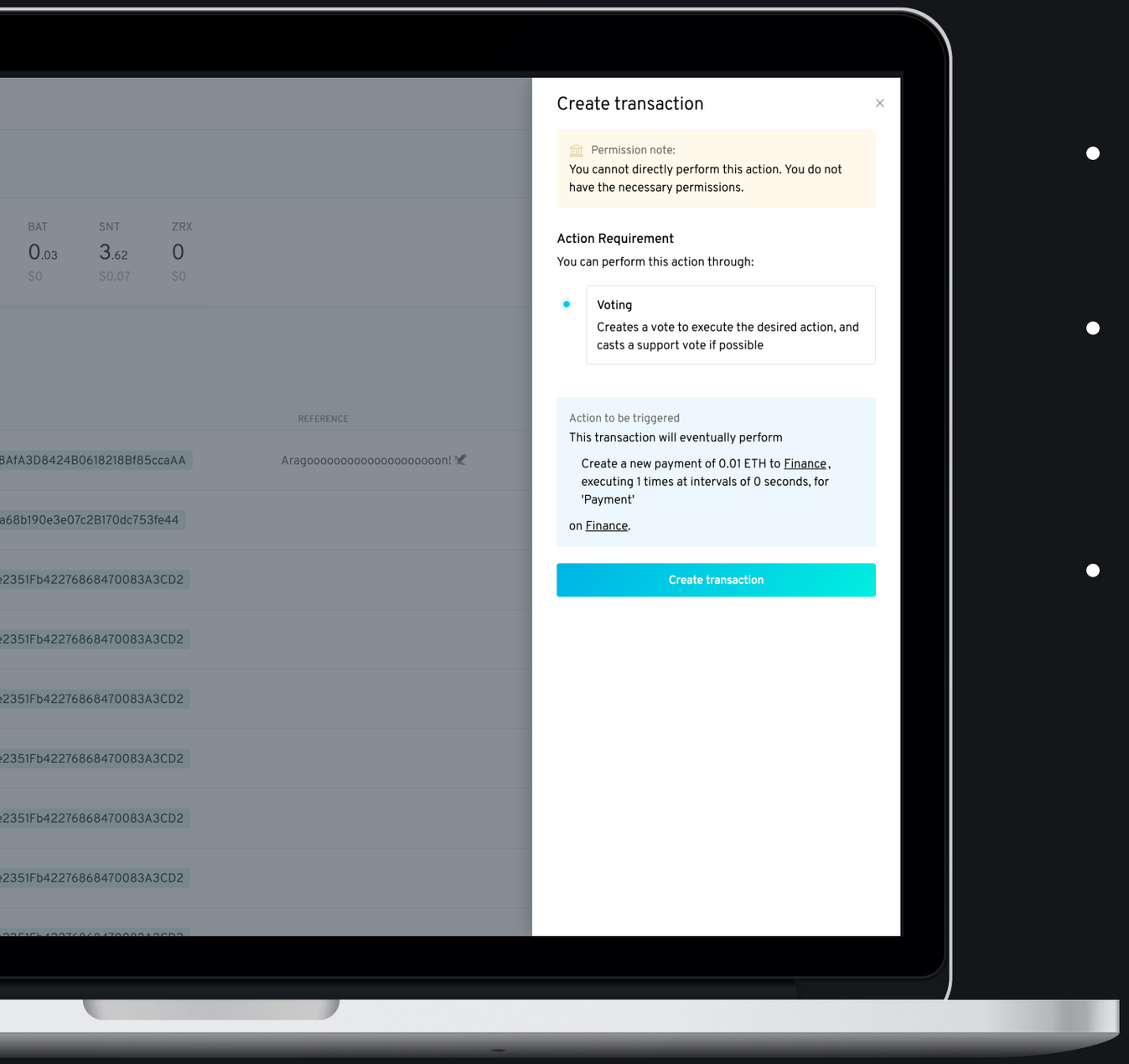
app.aragon.org

Aragon Voting v1



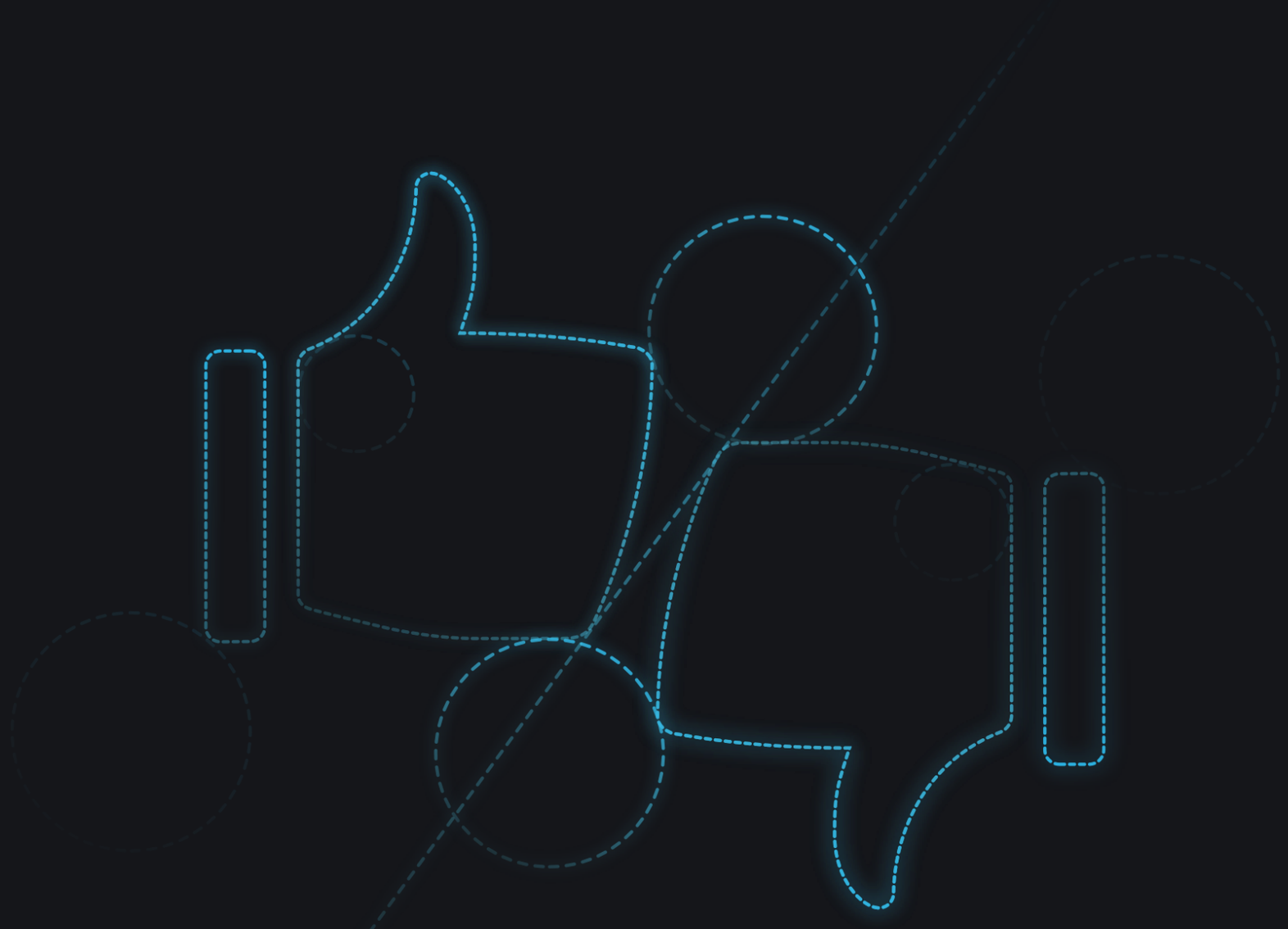
- Uses MiniMeToken as the only supported governance token
- Simple majority voting
- +$1m secured with Voting
- Voting is the default root authority for Aragon DAOs

Forwarding



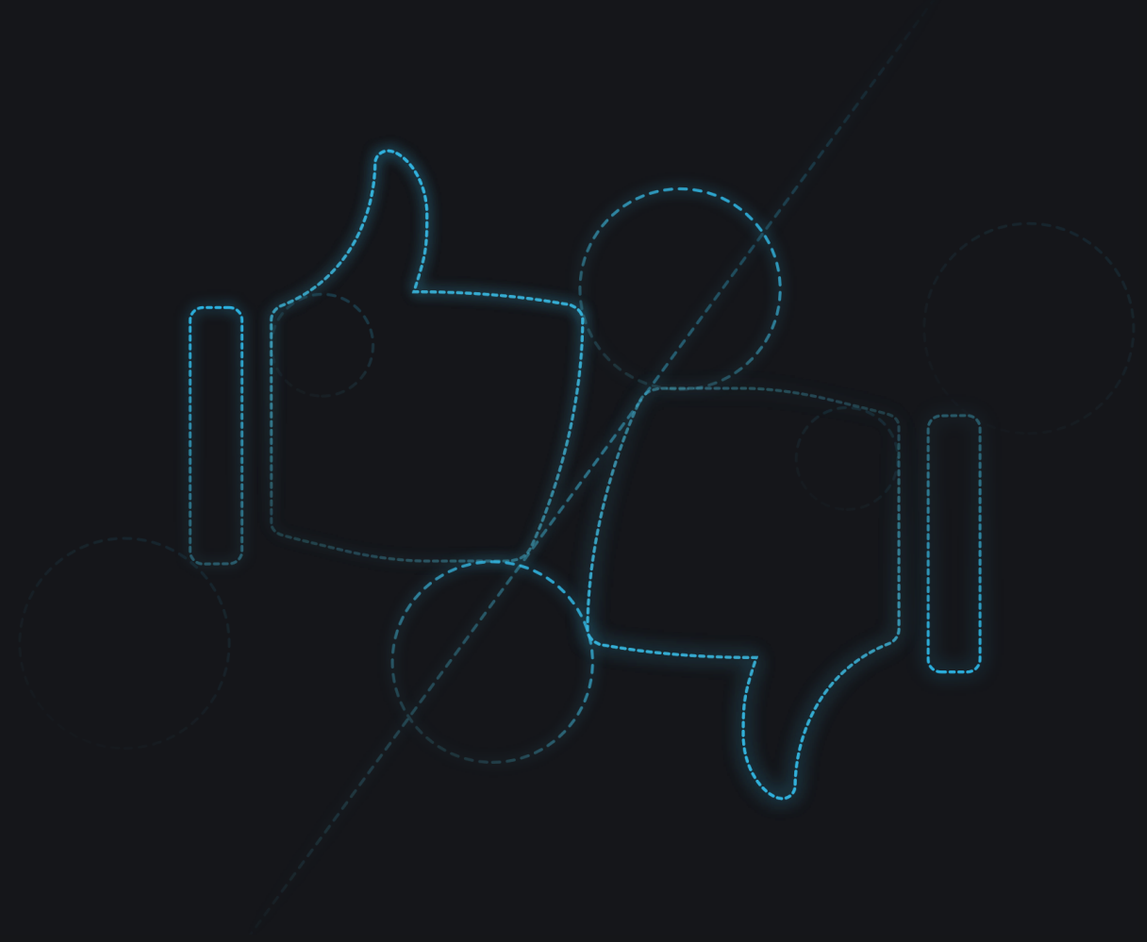
- aragonOS primitive for encoding complex permissions
- Users express their intent and Aragon figures out how to make it happen
- Forwarders can execute aragonOS' EVMScripts when some arbitrary logic checks out

Aragon Voting v2

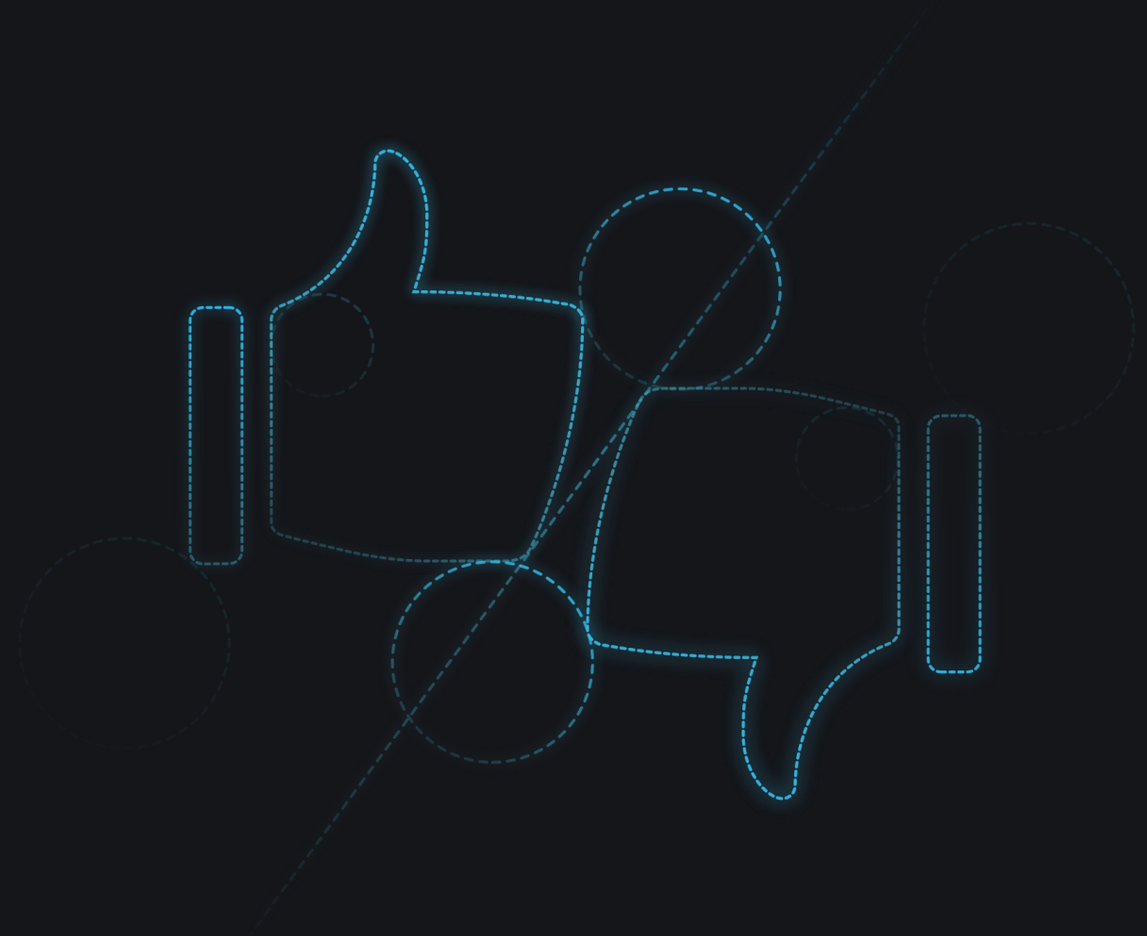


Aragon Voting v2

1. Remove MiniMe requirement



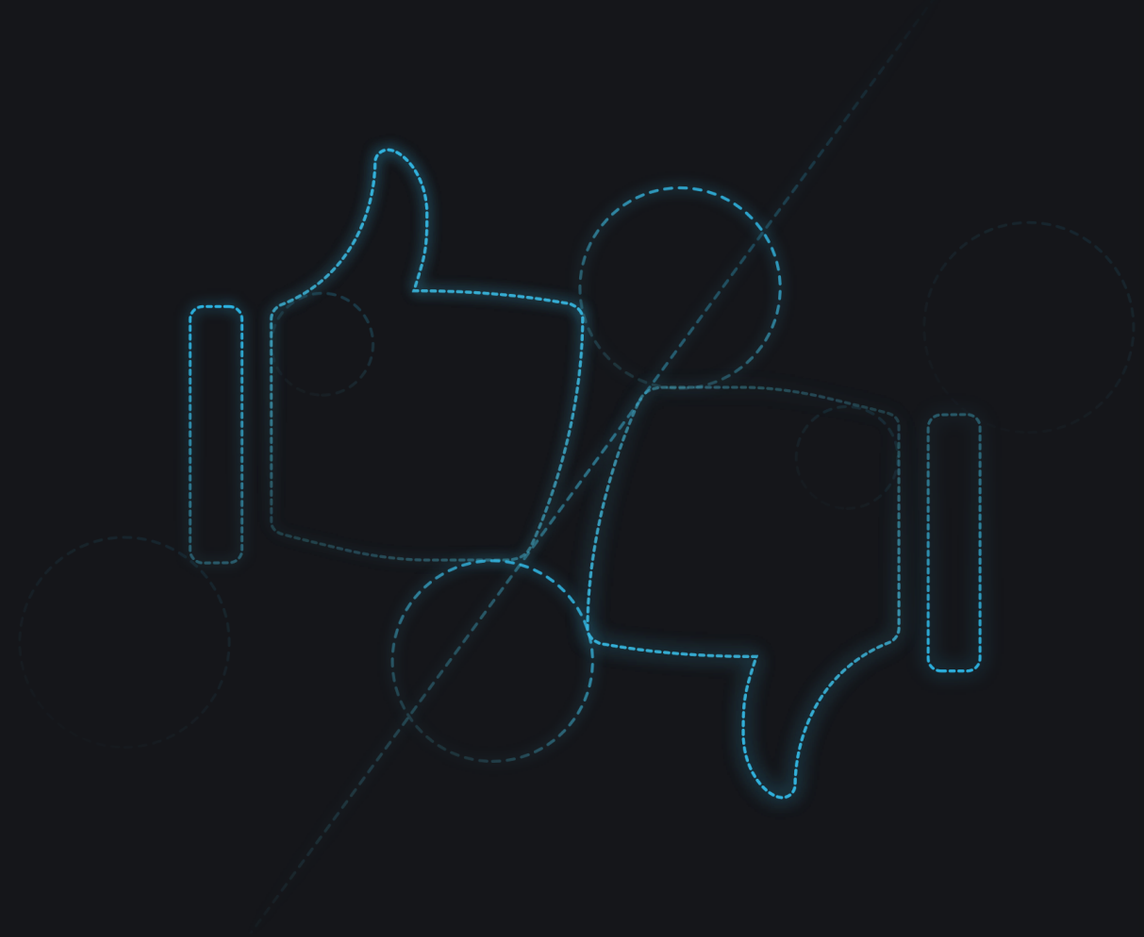
Aragon Voting v2



1. Remove MiniMe requirement
2. Really cheap voting

Aragon Voting v2

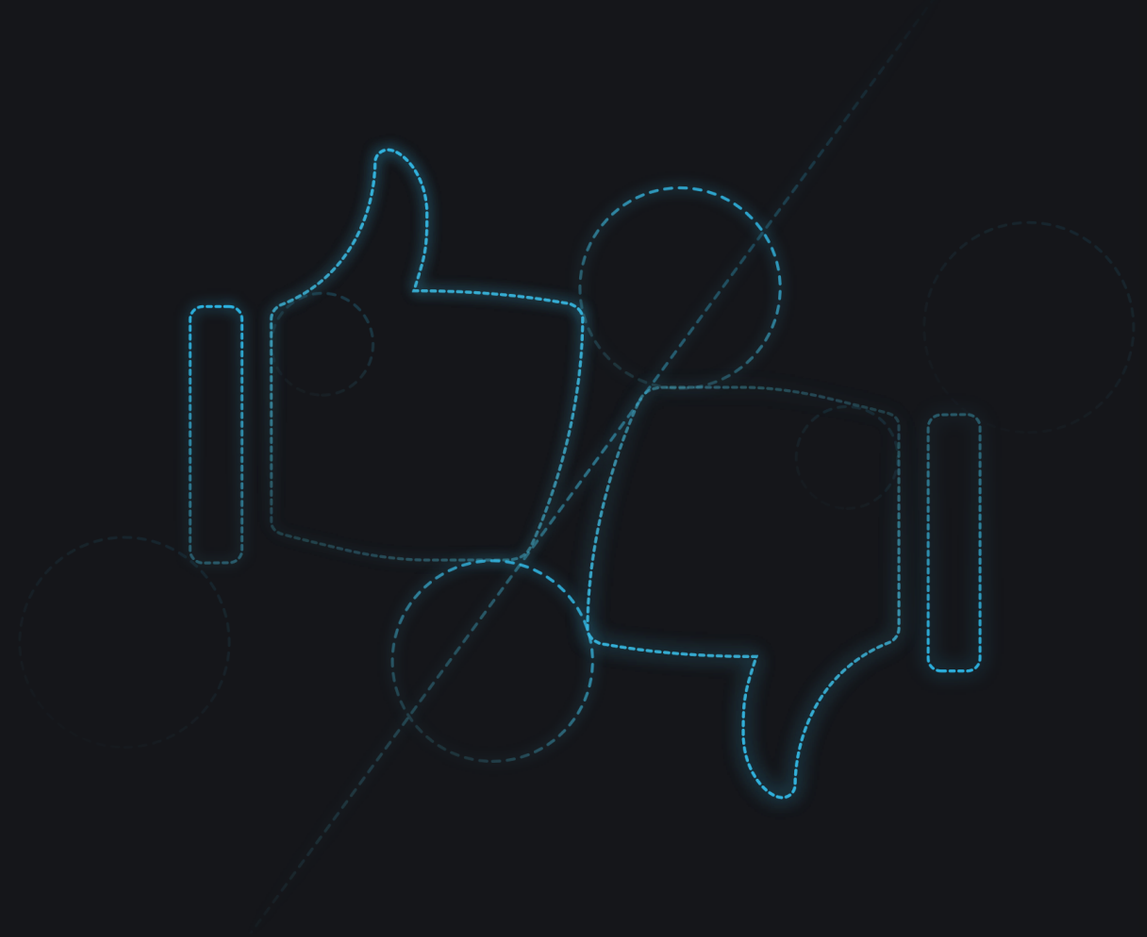
1. Support for all ERC20 tokens*



* tokens that use a standard data structure for storage

Aragon Voting v2

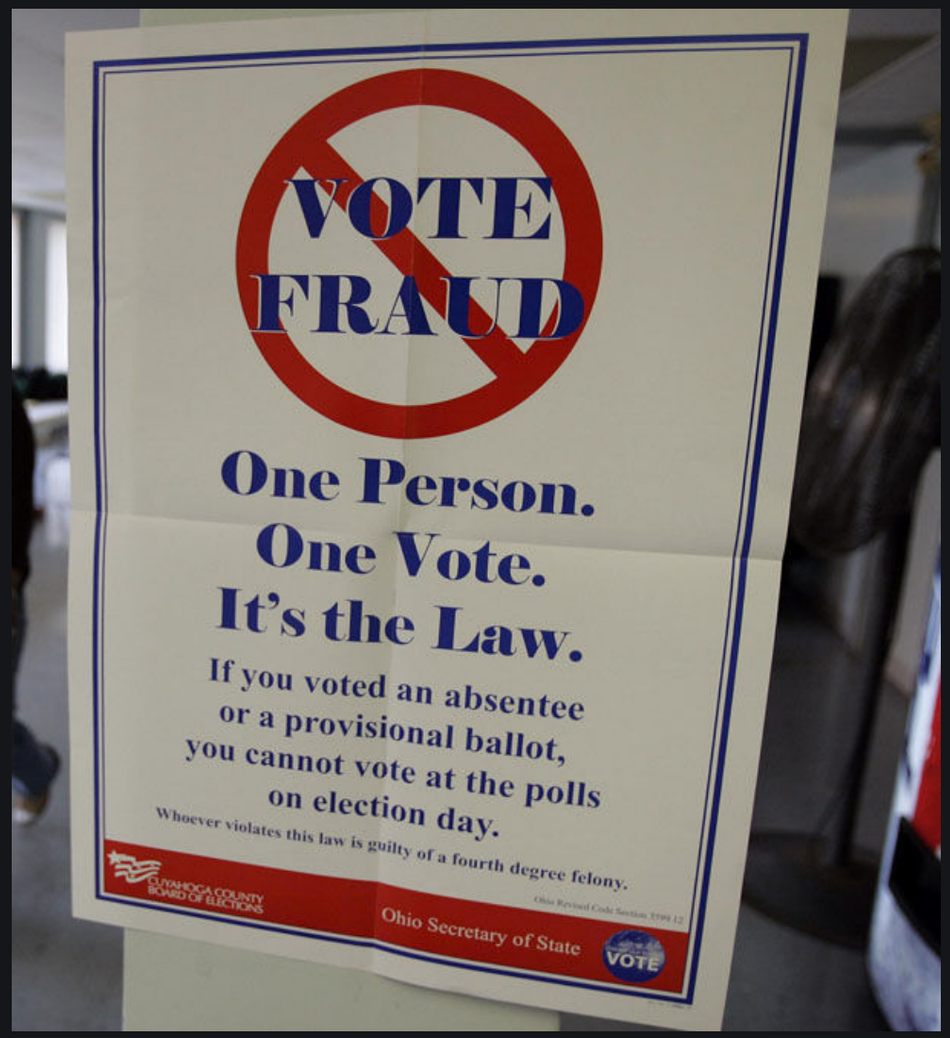
1. Support for all ERC20 tokens*
2. Off-chain vote aggregation



* tokens that use a standard data structure for storage

The double voting problem

- ERC20 tokens are fungible
- Can't detect if one particular token already voted, unless:
  - Tokens are locked during the vote location (staking)
  - Token balances can be snapshotted (MiniMeToken)



EVM Storage Proofs

- Prove what the value a given storage slot in a contract was at a past block height
- Derive storage values from a trusted block hash
- Uses `eth_getProof` (EIP1186)

- For Voting, prove the token balance of a holder at a snapshot block

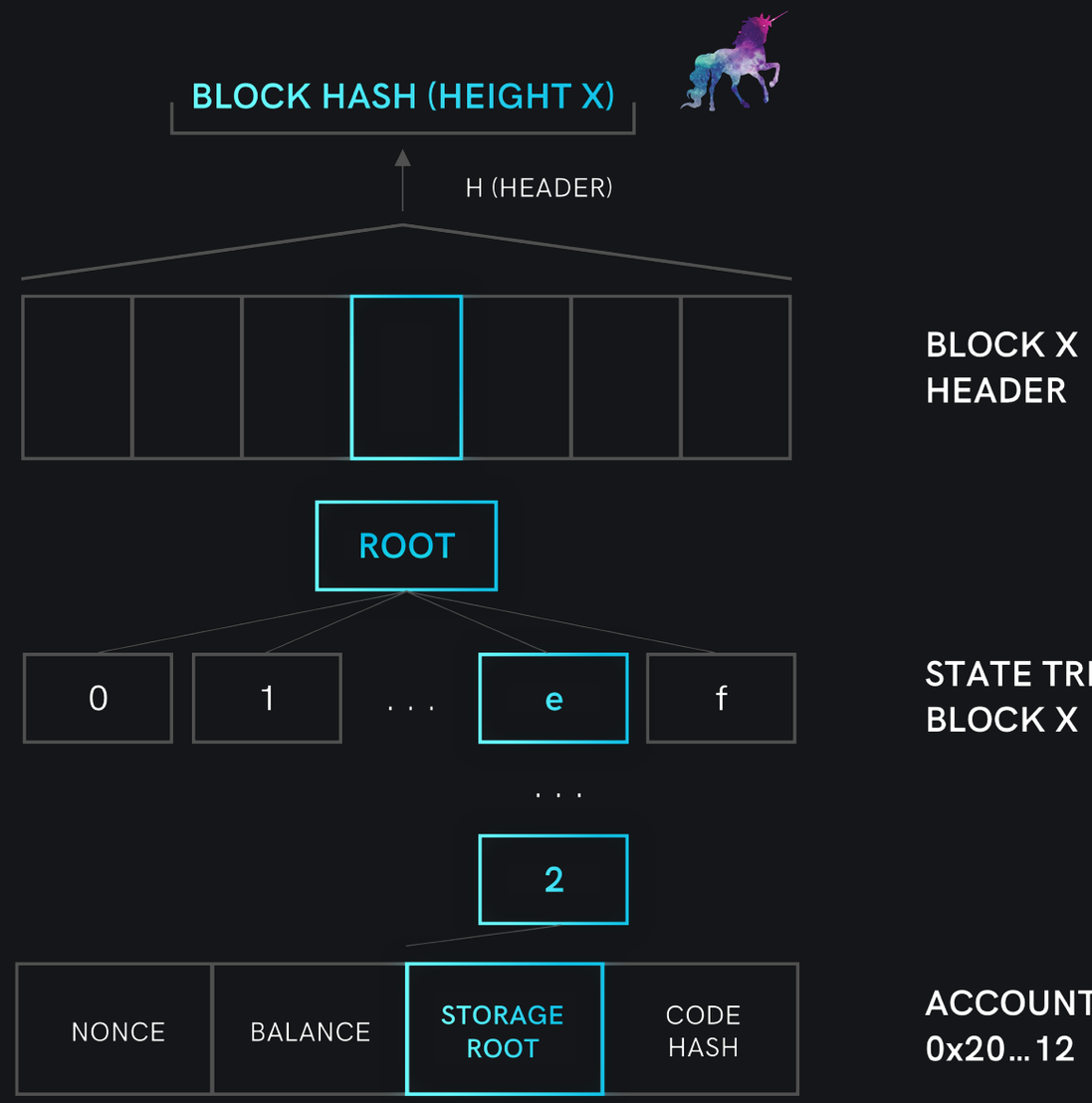
The Ethereum Storage Time Machine

github.com/aragon/evm-storage-proofs

Account proof

Block hash → storage root

- Use BLOCKHASH to get block hash for block X

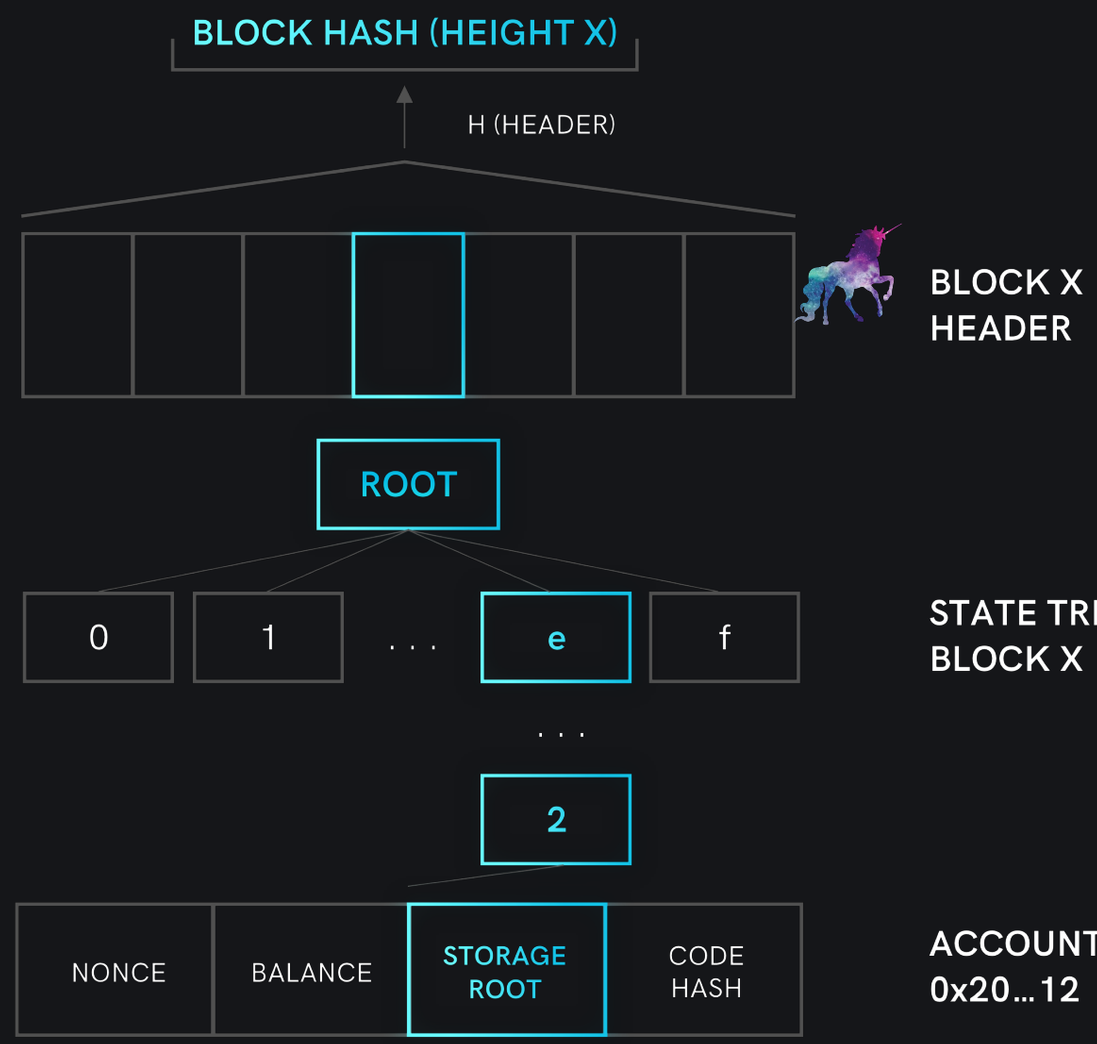


github.com/aragon/evm-storage-proofs

Account proof

Block hash → storage root

- Use BLOCKHASH to get block hash for block X
- Submit the block header and verify its hash matches

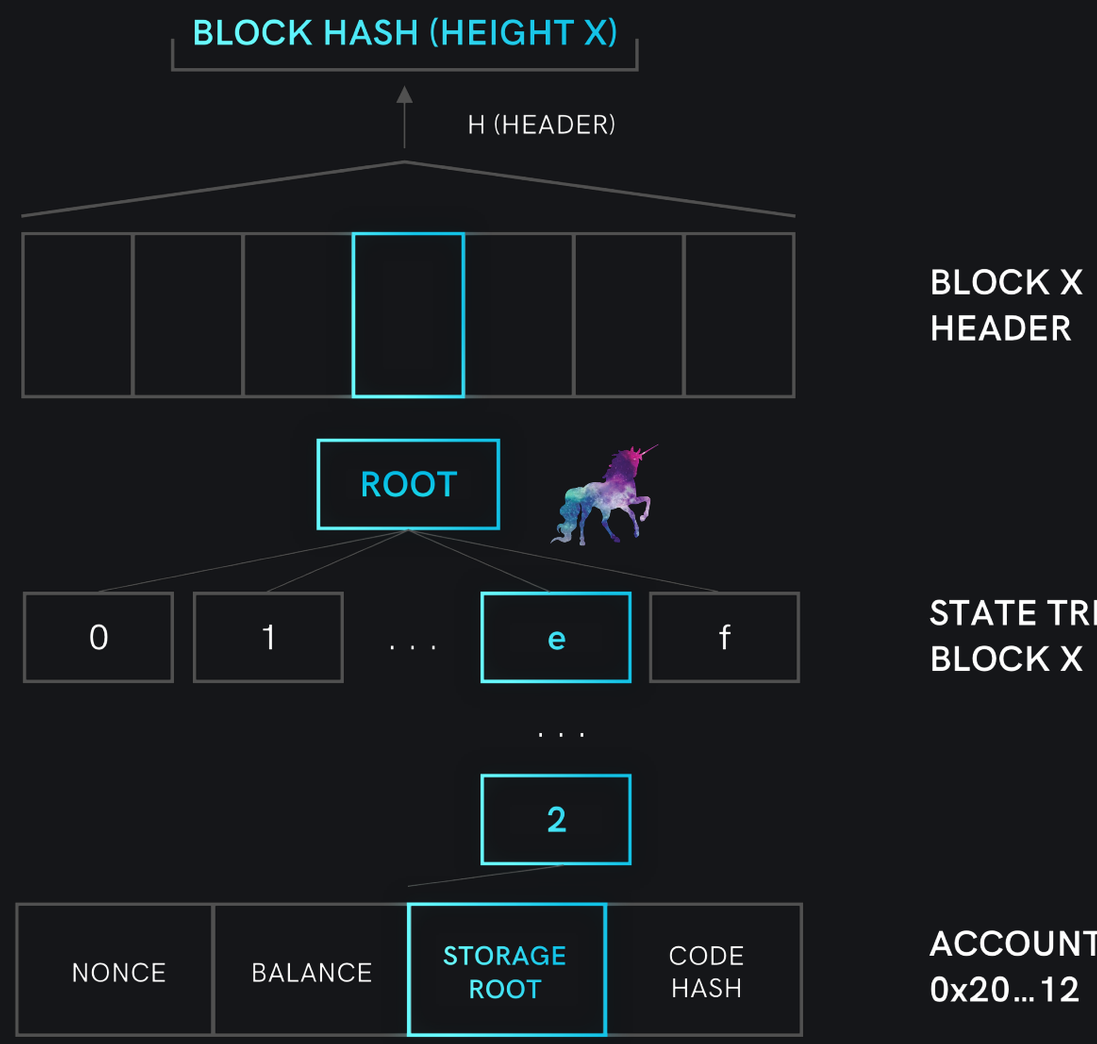


github.com/aragon/evm-storage-proofs

Account proof

Block hash → storage root

- Use BLOCKHASH to get block hash for block X
- Submit the block header and verify its hash matches
- Extract the state root from the block header

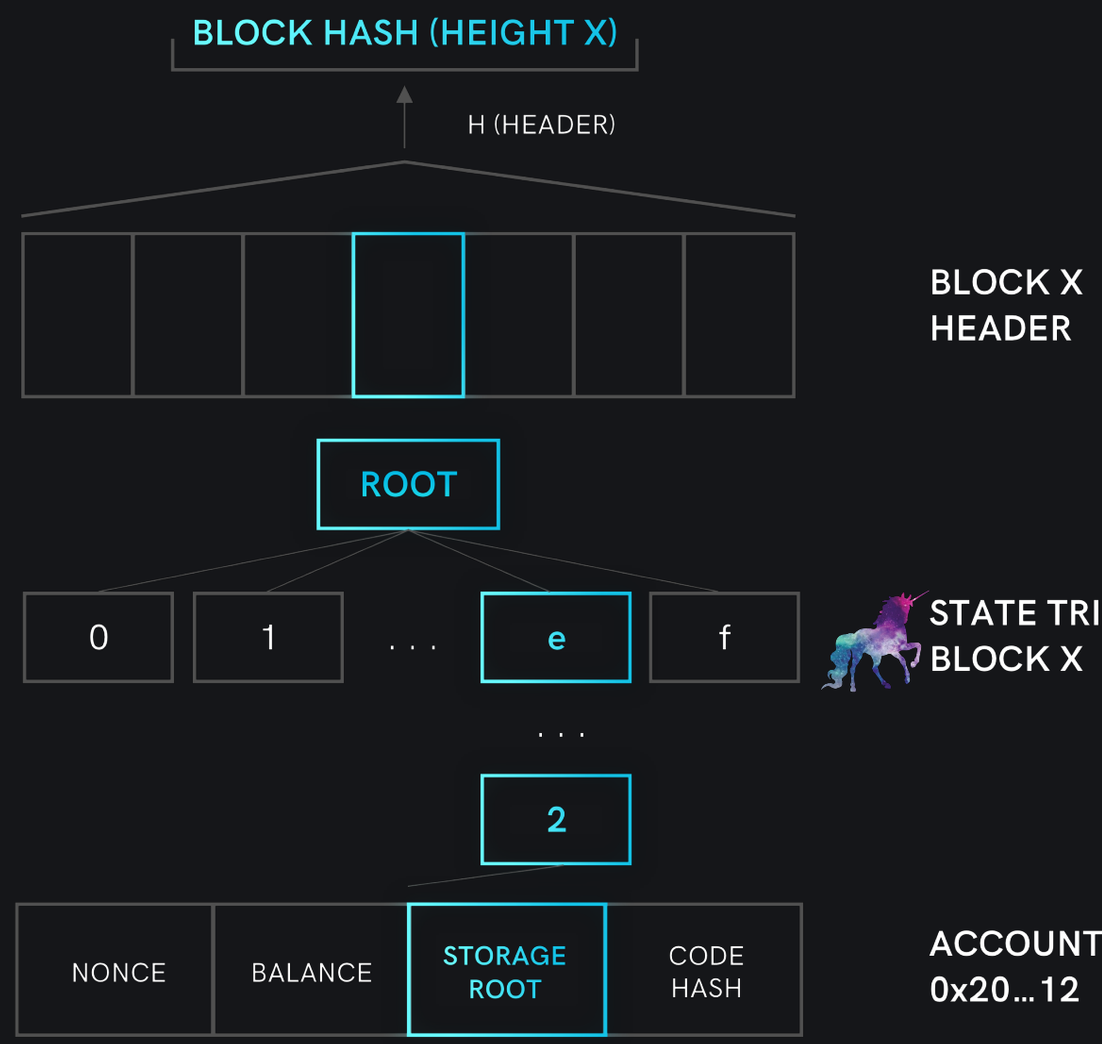


github.com/aragon/evm-storage-proofs

Account proof

Block hash → storage root

- Use BLOCKHASH to get block hash for block X
- Submit the block header and verify its hash matches
- Extract the state root from the block header
- Submit merkle proof of the account state in the state trie

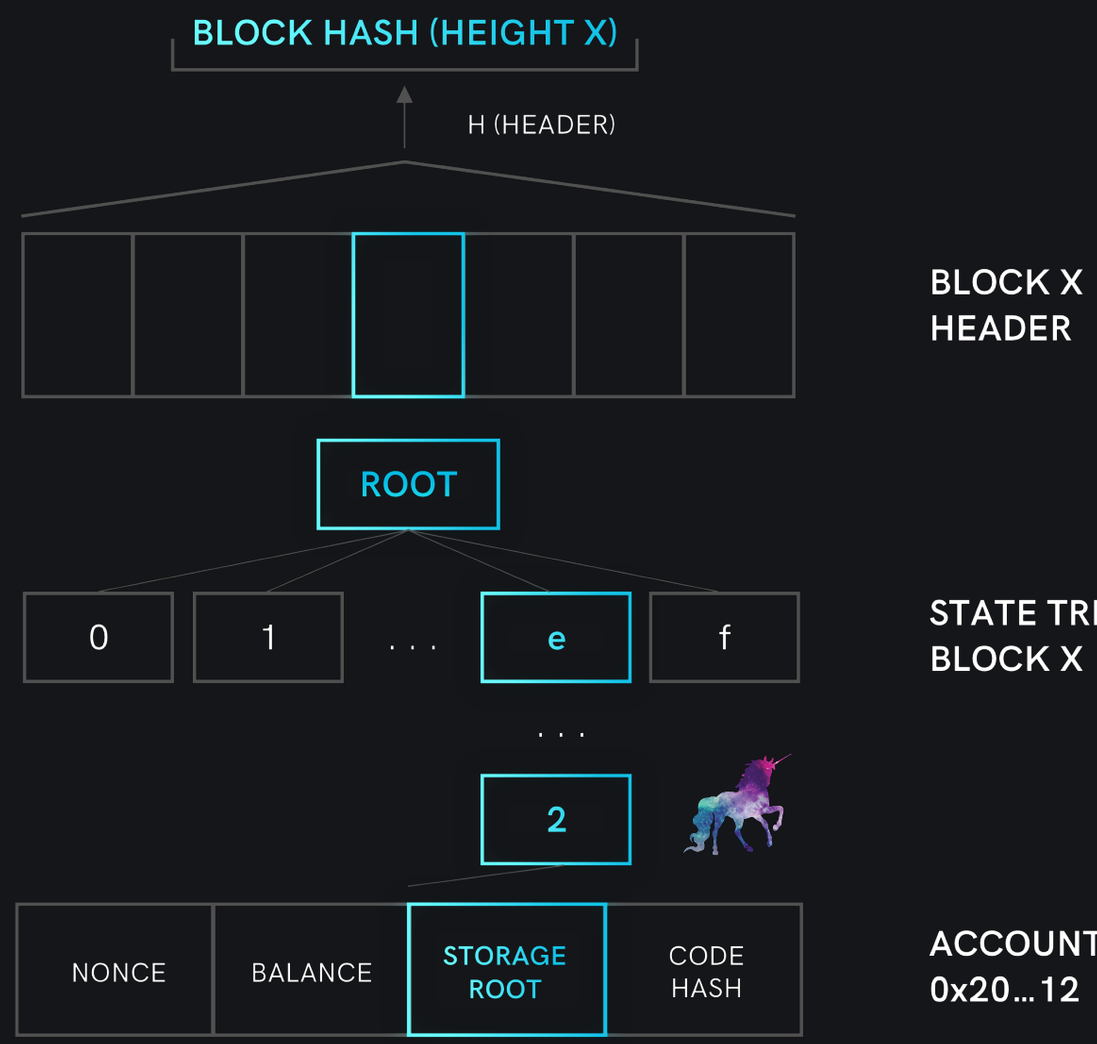


github.com/aragon/evm-storage-proofs

Account proof

Block hash → storage root

- Use BLOCKHASH to get block hash for block X
- Submit the block header and verify its hash matches
- Extract the state root from the block header
- Submit merkle proof of the account state in the state trie
- Verify merkle proof against the state root

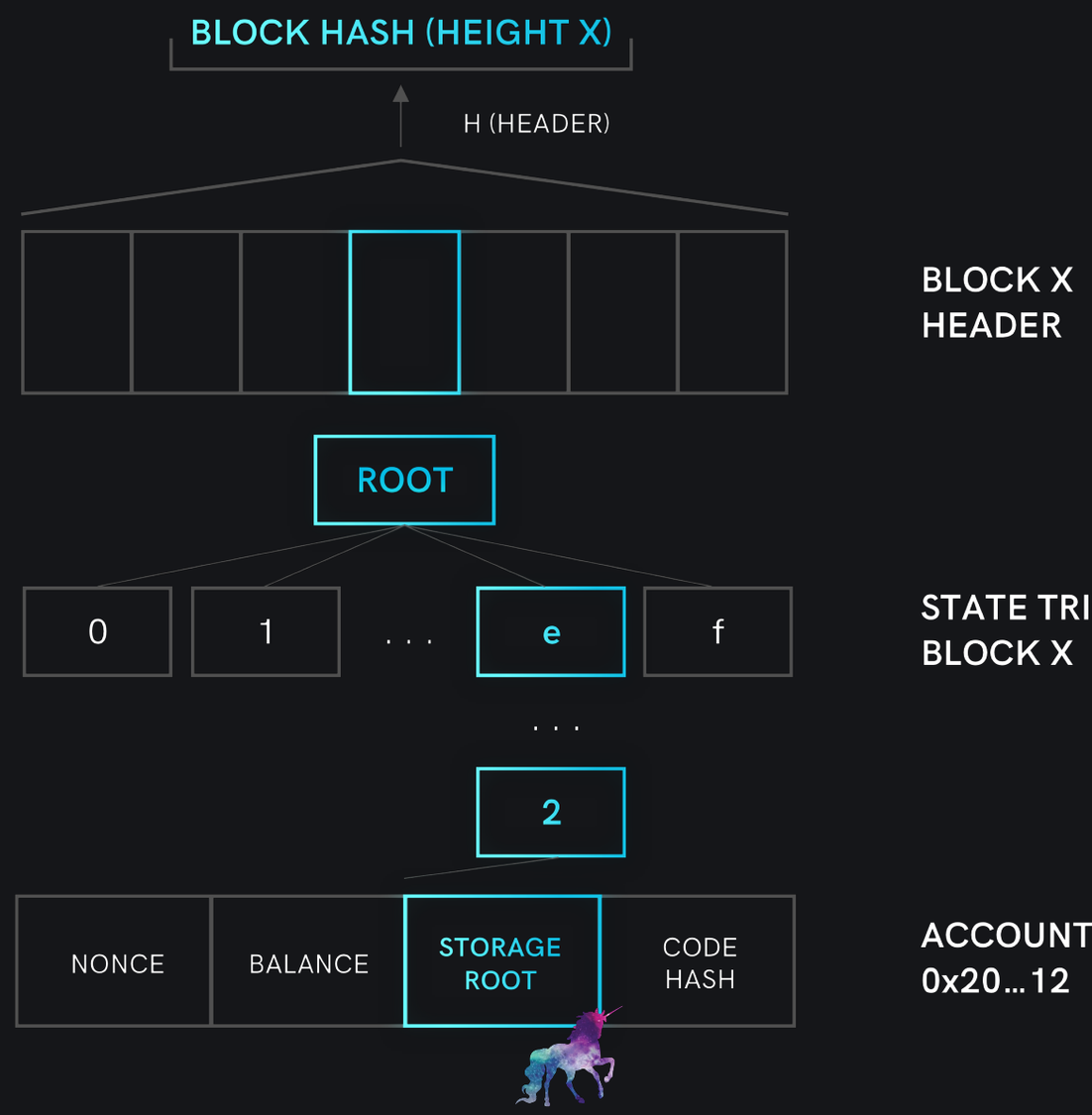


github.com/aragon/evm-storage-proofs

Account proof

Block hash → storage root

- Use BLOCKHASH to get block hash for block X
- Submit the block header and verify its hash matches
- Extract the state root from the block header
- Submit merkle proof of the account state in the state trie
- Verify merkle proof against the state root
- Extract storage root from the account state

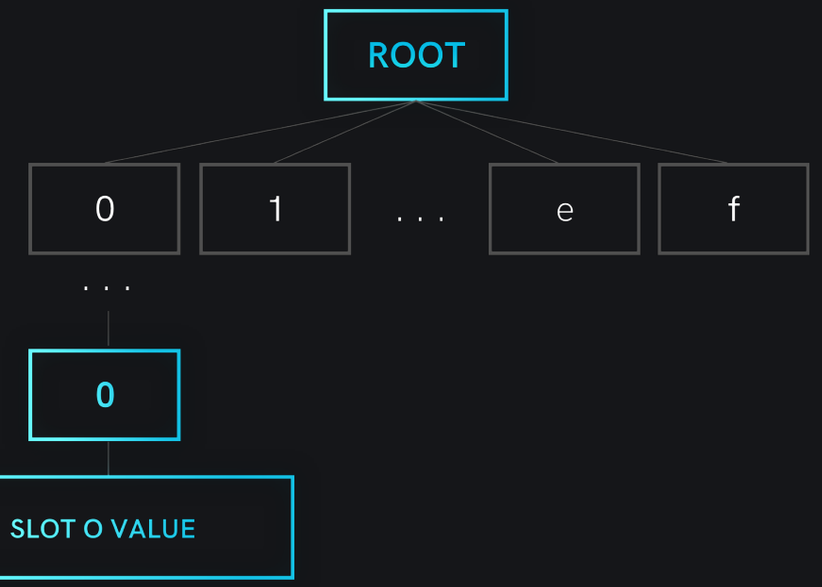


github.com/aragon/evm-storage-proofs

Storage proof

Storage root → slot value

- With an authenticated storage root
- Submit merkle proof of the storage slot value in the storage



github.com/aragon/evm-storage-proofs

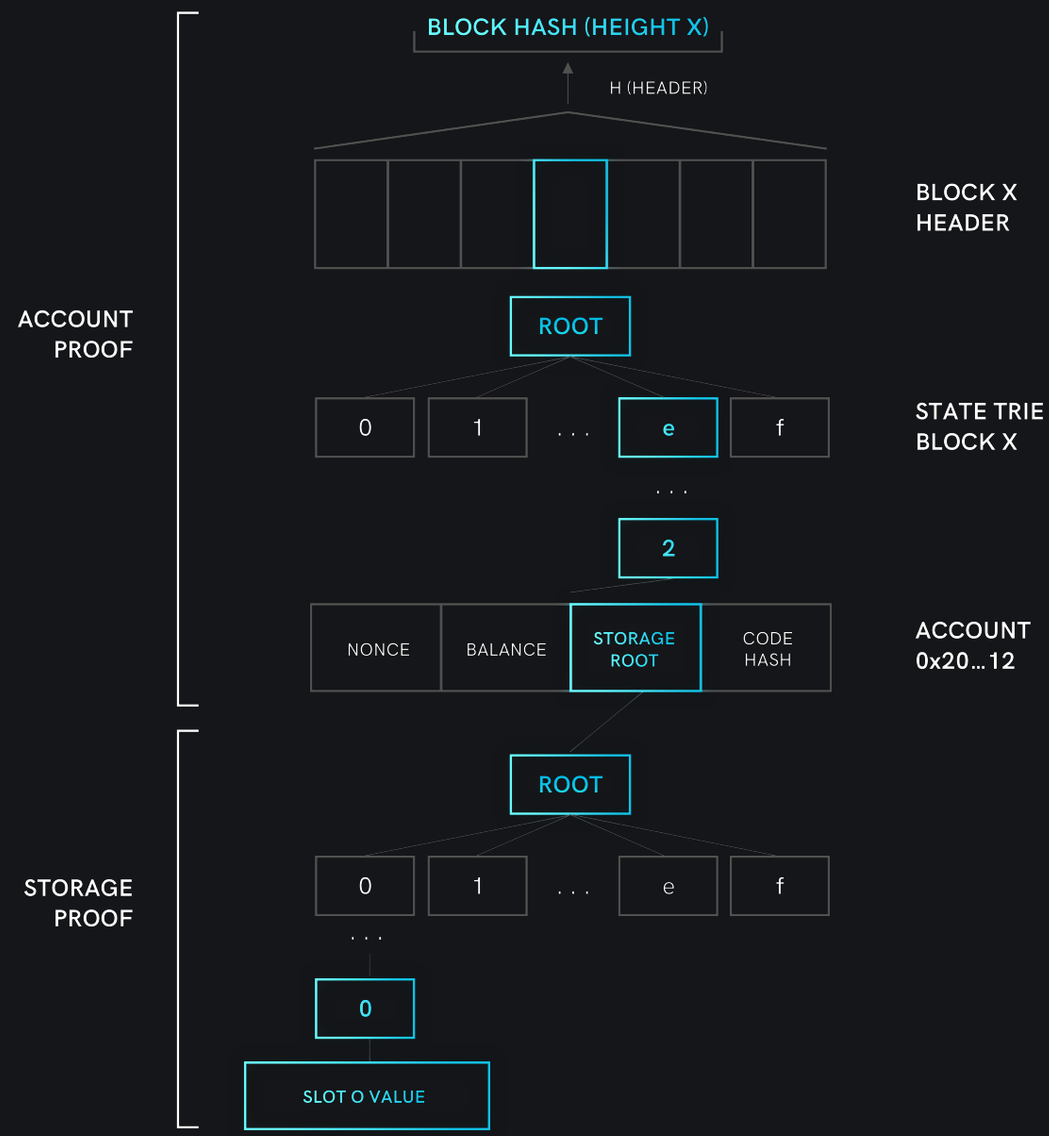
EVM Storage Proofs


```
{ account: "0xc85463bf347deff723feb8985c28348ba9375393",
  proof: [
'0xf901f1a02fb0b1eab84be2d33748f5cd4dac12bb6016578278e33243d8cfbe193ee7efaaa0ab8cdb808c8303bb61fb48e276217be9
770fa83ecf3f90f2234d558885f5abf1a0d8cafe138dc66c0aa91d82be2761cd880fb3c32723dbdc5d54600c6a4b91081c80a0889590b
fac91a9d2f39484939ea971a3c1e9e8e5b1d6e69b94d5e01e795168faa0b5d7a91be5ee273cce27e2ad9a160d2faadd5a6ba518d38401
9b68728a4f62f4a0c1981e61e6d9963fe0e1828d303ef5704d98473d60141b51b49cbacf8a8214e4a0b8a852f9a156d9075d882c3d97a
74194873be074750dd6fcfff2bf8014760313a02e0d86c3befd177f574a20ac63804532889077e955320c9361cd10b7cc6f5809a05d4e
2072db06aa1dfa98bedf5d5cb26776037f4dc9f5e0e096c017304a559c6da0053b94f7a4eeef8caf390ed2d9106c9b2fe36045cf56c53
fc9f2a63a725645aca09d0f23f9b8bef0da7b18f66c2d4c8f1b7ae8004ec5ea39b5982e7a4e13665e22a059550ebc710427a44cf82d66
066497056670f45c4458a32d34741f4f4aeb5021a01b7779e149cadf24d4ffb77ca7e11314b8db7097e4d70b2a173493153ca2e5a0a06
20ebec7cd76a64c51f8ded867e11af84d8c8df8ed358b4834307cab7ab083cba004119e945004b6cf04667a086dcb5682deb73844d7c7
ae0a12aecb4775eaed9280',
  '0xf8518080808080a046fe2ce1a19f07ab65397c7befa09ef5befc138474d41eec9857f06ccfd0153e8080a09171b07a1592254b64
c966edfad141a04a5bdfe446c903fcbbae8b38c2253a2c8080808080808080',
  '0xf869a020cefaab6947ba4a458bed4cc3ed1ee296fa99495da41626cc2155241cc6354db846f8440180a05ac25855e6772df74c5a
5e44622e1dc086155e341b7d43db11ff9b7185a3ba69a06ced21e6f91541030de4e957e4dda9257f67137dd839a7d7b60eccf911cd580
0' ] }


{ key: '0x11c43eaa40d1589106e26f8cc9f1c4999fe8f787317bf324fe2875cd75ffa8fd',
  value: '0x3b9ac9ff',
  proof:
   [
'0xf8518080a0af7d1197a621b2e6c670cdfd1d3fe7860b1c955464b78a7ed6bdb8ea5c0f038c80808080808080a02779e1a68b3368a4
927a782e51491024129784ff1d02c877cdb580e930fcafcb808080808080',

'0xf851a0c33152adc2377a9dcb0824b61a50a1892dffb4aed37081651ba235bdb65b3ad18080a0d036a32f67ea11c4bc5772e2ad3b6a
ad5c784d5280cc365bf83953856832fead8080808080808080808080808080',

'0xe7a020d401f7f21db6cb3bfbfb7eb5f849e7bf2b0c9b8d2e389a20d72f6f2be5659485843b9ac9ff' ] }
```

github.com/aragon/evm-storage-proofs

EVM Storage Proofs for ERC20s

Upsides

- Use all existing ERC20 tokens for Voting
- Snapshotting doesn't impose a cost for all transfers

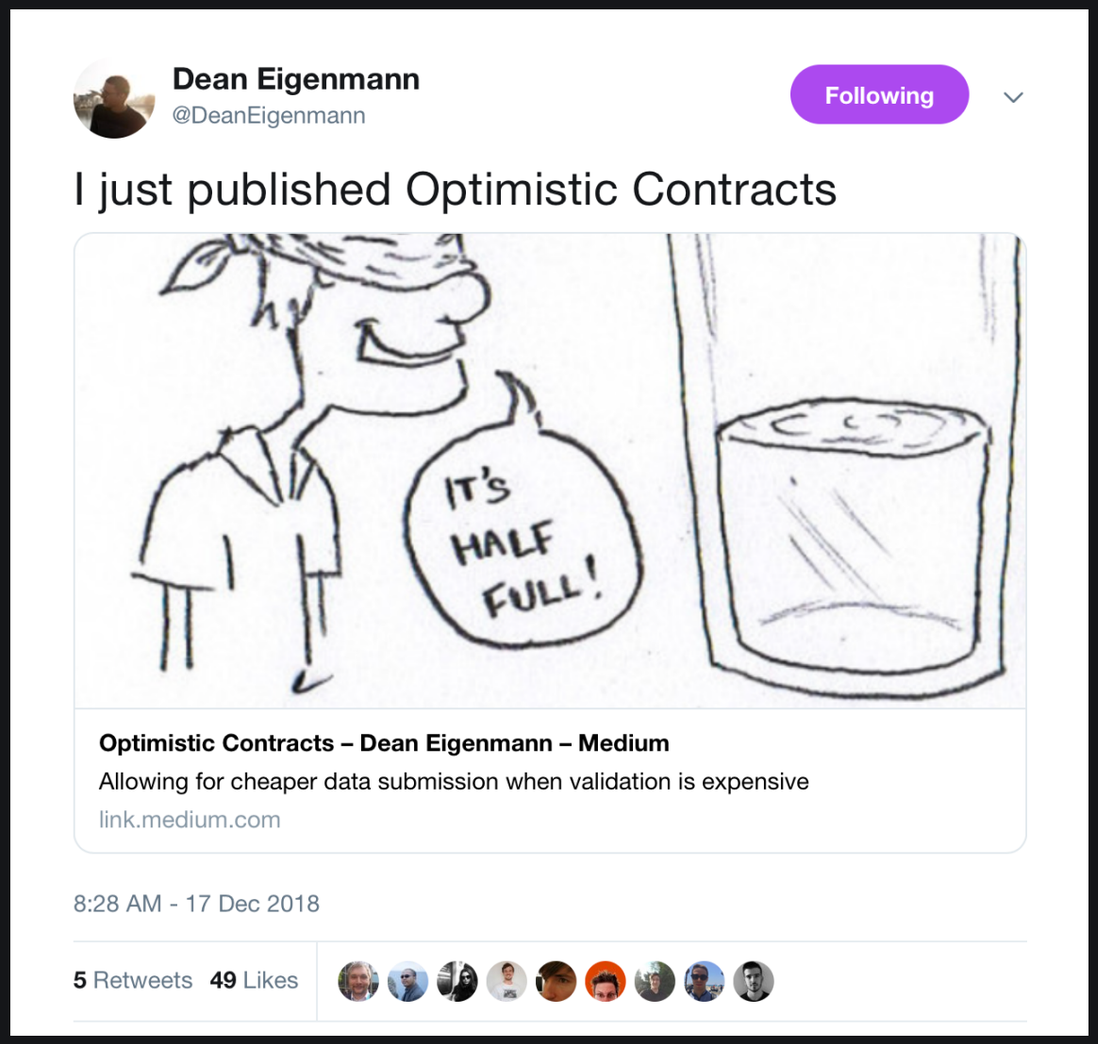
Downsides

- Verifying a full proof uses ~500,000 gas
- `BLOCKHASH` can still only get the last 256 blocks
- Breaks EVM abstraction by checking storage directly, use carefully

github.com/aragon/evm-storage-proofs

Optimistic contracts

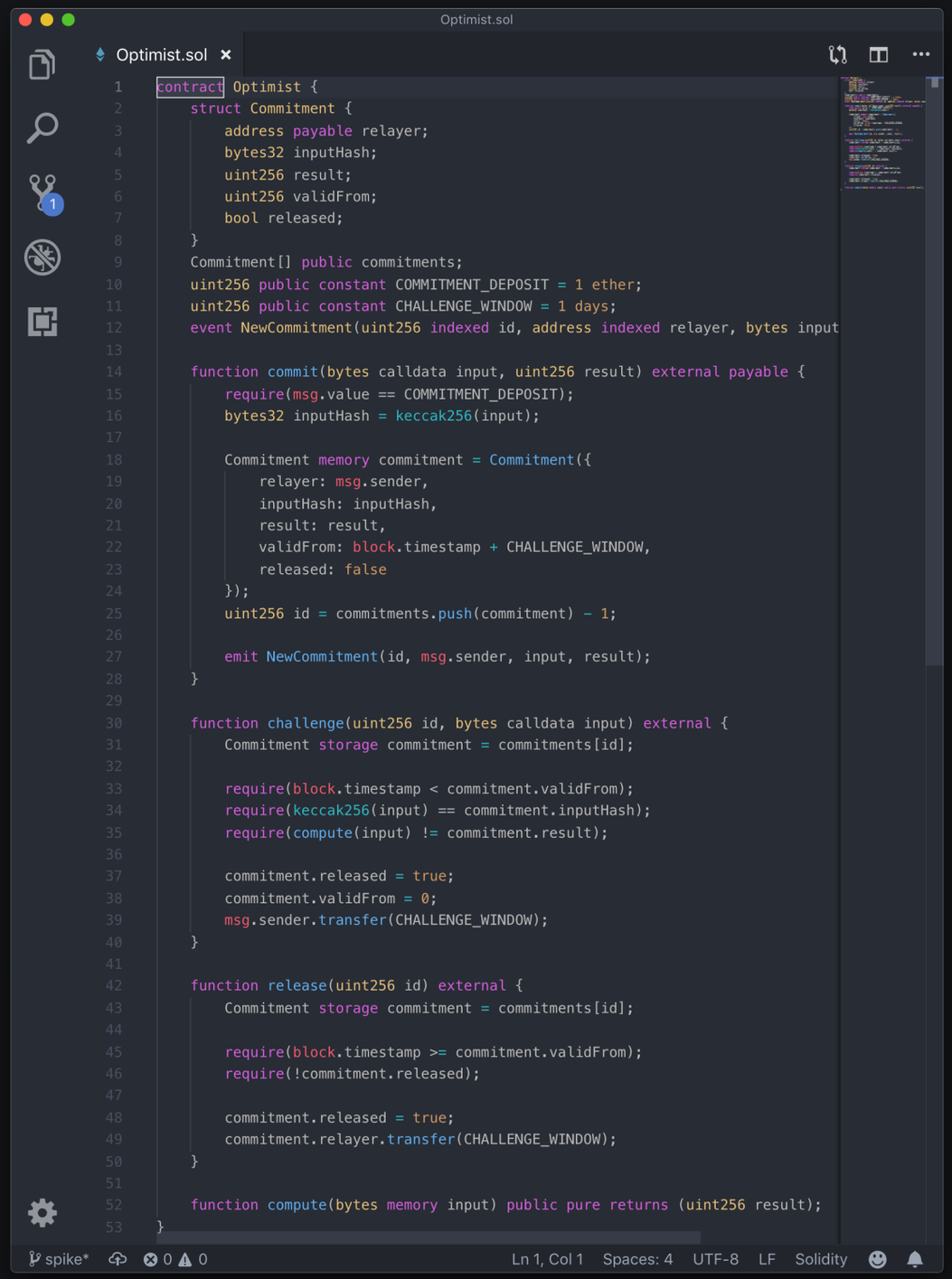
- Offloading computation heavy operations from the network
- Ensure computation correctness with bonded relayers instead of repeating it in every node
- Naïve implementation can scale up to 80x*: use 100k gas for computations that would take the block gas limit (8M)



* for an optimistic execution with no input

Optimistic in 50 LOC

- Commit to the result of executing a pure function with some input
- Challenges can be done during a certain time window, which executes the computation and compares to the commited result
- Challenger is awarded the deposit of the relayer on a valid challenge
- After the challenge window has passed, the result is taken as valid



Voting Relay Protocols (VRP)

- Family of layer 2 protocols that aggregate votes off-chain and submit partial tally **optimistically**
- Update the tally once for many votes without verifying storage proofs
- Individual votes are multiple orders of magnitude cheaper to cast
- Vast design space with different trade-offs
- Voters just sign their vote message and send it to the relayer

Simple VRP (SVRP)

- VRP, except that **only one relayer** is allowed to relay votes at a time
- The relayer slot can be centrally appointed or auctioned
- The relayer stakes tokens as collateral that can be slashed if a fraud proof is submitted
- The relayer may ask for a fee payment to include a vote or sponsor voting

SVRP and centralization

- If the relayer doesn't return a signed inclusion commitment, the voter can submit the vote directly
- If the relayer commits to relaying a vote and they don't, they are slashed
- Data required to prove fraud is available through EVM logs
- If someone is monitoring, the relayer will always be slashed if submitting fraudulent votes

The SVRP bottleneck

- Naïve optimistic relies on the data required to submit fraud proofs being **available with EVM logs**
- In SVRP, the input data is the array of vote mesages being relayed
- Cost per input byte = **88 gas**
- 1 vote message = **68 bytes**
- Hard cap of 938 votes relayed per batch (80% of the block gas limit)
- Compared to Voting v1, SVRP scales voting by **10.72x**

VRP messages

All messages must be RLP-encoded before sending them over any transport protocol or submitting them on-chain.

Vote message

Components:

- Voting app address identifier (4 bytes: `bytes4(hash(address))`)
- Proposal id (3 bytes: 14,706,125 proposals per Voting app)
- Casted vote (1 byte: 256 options)
- Voter balance (16 bytes, claimed user balance in the voting app's token at the snapshot block of vote id)
- Voter ECDSA signature for message hash (64 bytes, with the trailing bit trick)

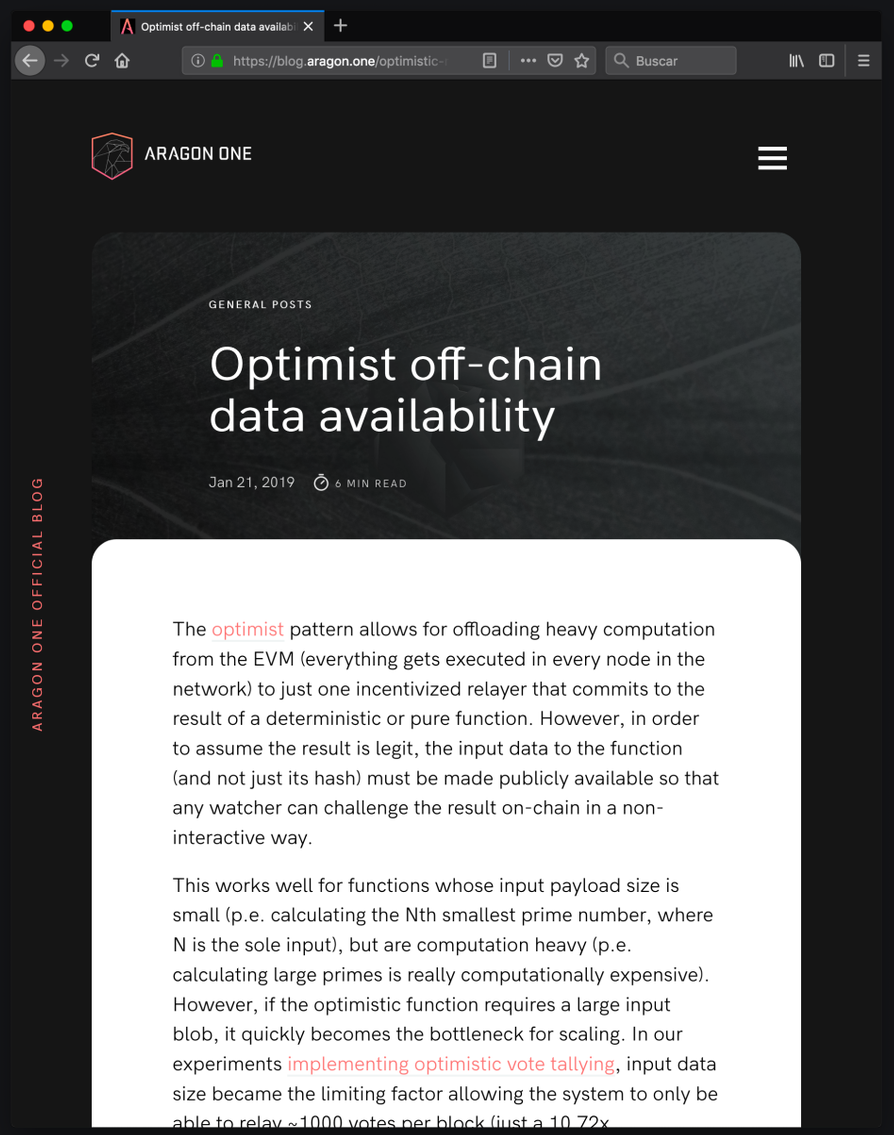
Total batch size: 88 bytes

The data availability problem

- Just submitting the hash on-chain and proving that the hashed data is available is impossible from a smart contract's worldview
- If the data is actually not made available, fraud couldn't be stopped, as the data is required for challenging
- To solve this issue, relayers could be required to submit on demand the full data blob to the chain so a challenge can happen, which can be used for griefing relayers

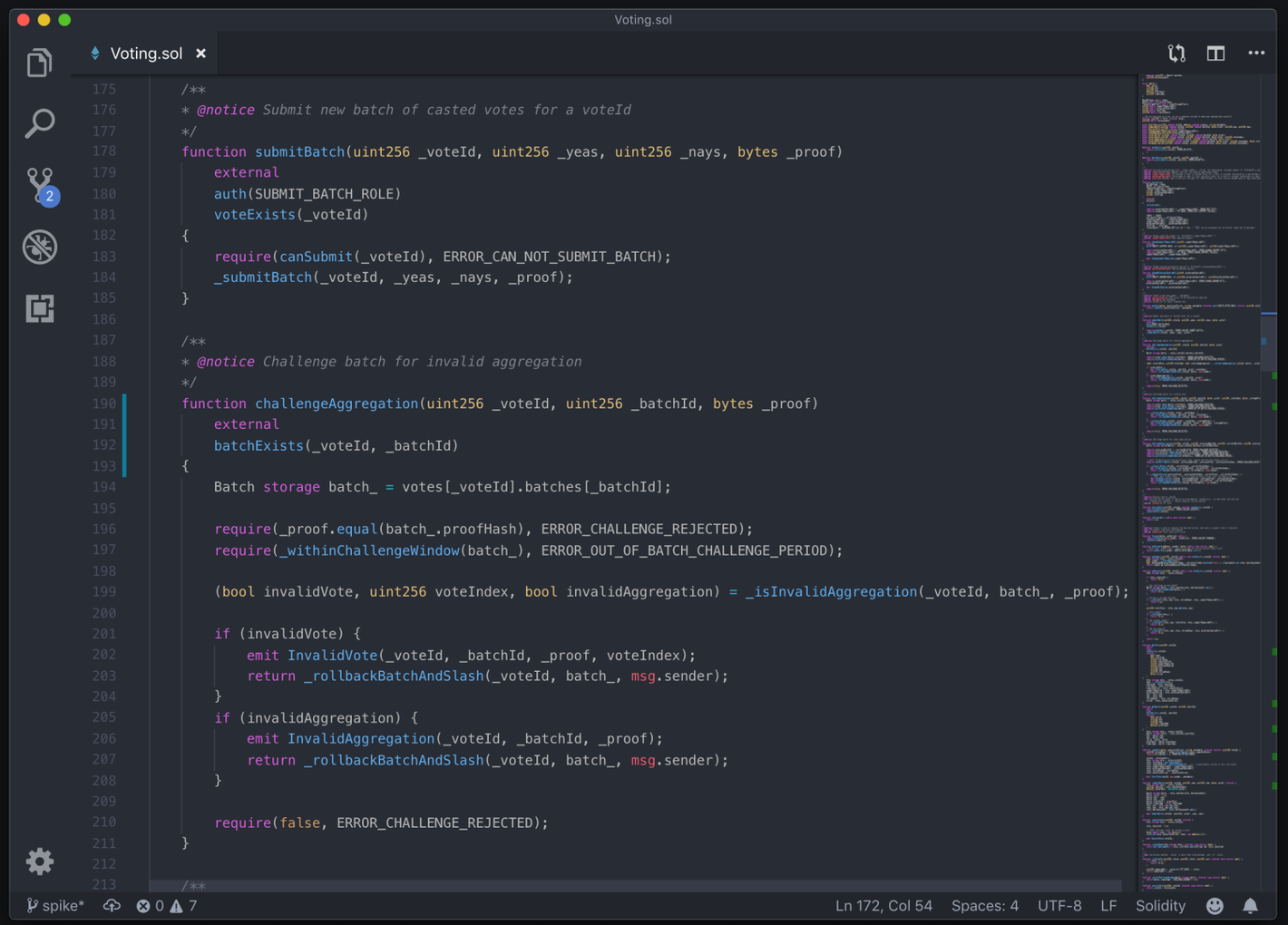
Optimistic data availability

- Used for **Availability VRP** (AVRP)
- A set of **N validators** is selected, but only one of them can relay at a time
- Validators receive relay requests and send them to other validators to receive a signed commitment that they have the data and result
- Later, a validator is randomly appointed as the relayer and they can relay it if they have signatures of **2/3 validators**
- Only one honest validator is needed for the integrity of the system (unless 2/3 validators are malicious and withhold data)



blog.aragon.one

SVRP is now open source



github.com/aragon/srvp

Voting v2: TLDR

- EVM Storage Proofs
    - Use any ERC20 to vote (no staking required)
    - Bring any exisiting token to Aragon for governance
    - github.com/aragon/evm-storage-proofs
- Voting Relay Protocols
    - Aggregate votes off-chain and relay partial tallies
    - Voters don't need ETH, and costs are reduced by 10-1000x
    - github.com/aragon/svrp

- Research discussions: forum.aragon.org
- Come work with us: wiki.aragon.org/jobs

Thanks

aragon.org

@AragonProject @AragonOneTeam @izqui9