Voting in the Colony Governance Protocol

Elena Dimitrova & Dr. Aron Fischer

What is Colony?

Colony is a platform for decentralised governance.

It's about working together: **coordination** and **collaboration**.

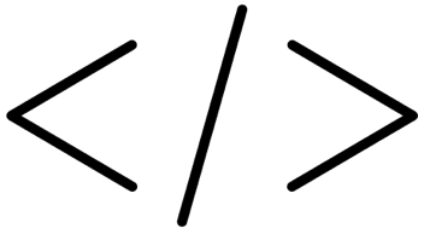
It **distributes authority** throughout a community — **no central point of control required**.

Thousands of people, all over the world, can **build, manage and share in the rewards of a common endeavor**, without trusting (or even knowing) one another.

COLONY

Governance Protocol

Open Source

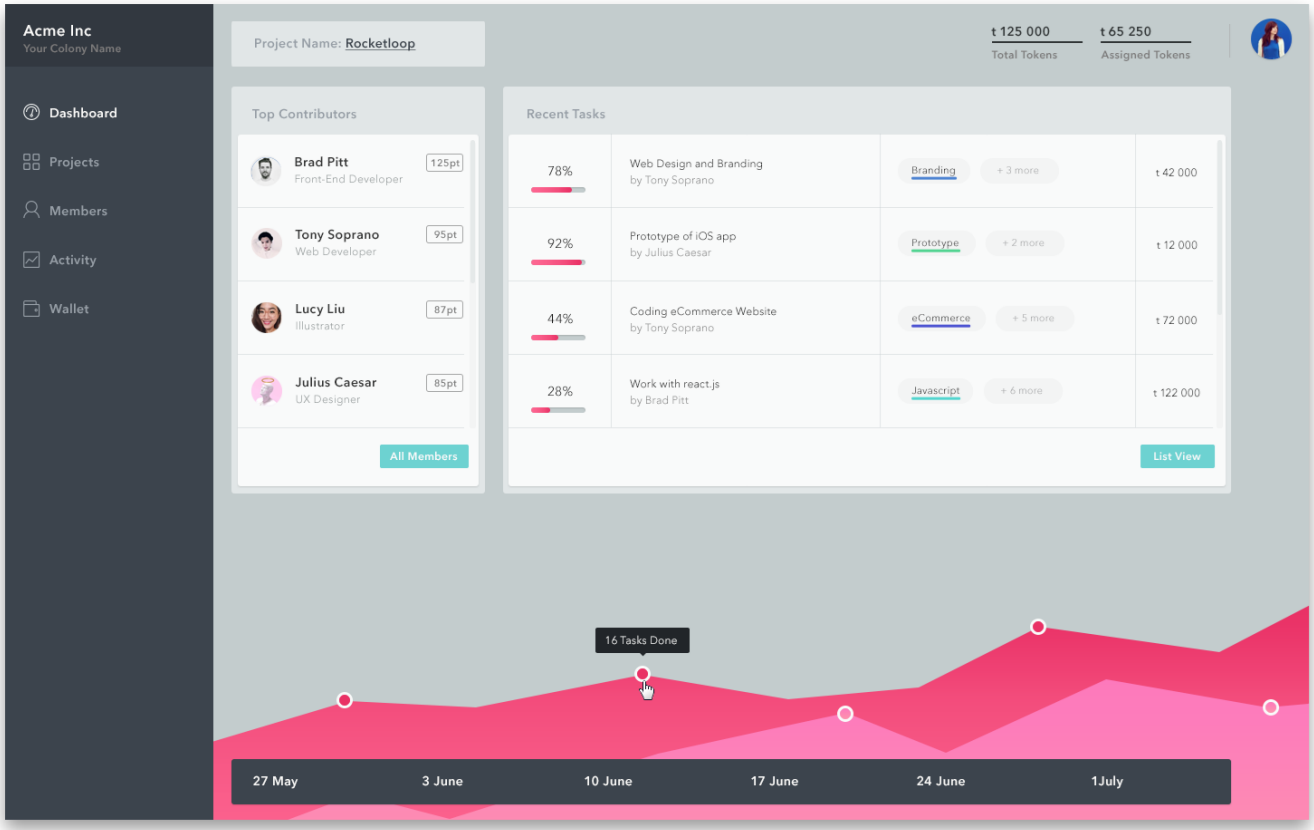


Built for Developers

100% ETH

COLONY

Application



COLONY

Governance

Colony governance is designed to be fluid, efficient and decentralised.

The focus is on ‘getting stuff done’ by:

Distributing authority
Mitigating bureaucracy
Enabling meritocracy

However, we need a security fall-back mechanism: **the Vote!**

COLONY

The Colony Voting Protocol

COLONY

The Problem with Voting

- Traditionally: One vote per person

COLONY

The Problem with Voting

- Traditionally: One vote per person
- But: Who is a “person”?

"On the Internet, nobody knows you're a dog."

COLONY

The Problem with Voting

- Traditionally: One vote per person
- But: Who is a “person”?
- Strict control needed for voter ID

COLONY

The Problem with Voting

- Traditionally: One vote per person
- But: Who is a “person”?
- Strict control needed for voter ID
- dApps have risk of the “**Sybil Attack**”

COLONY

Requirements

1. No double voting

COLONY

Merit weighted voting

Votes weighted by **reputation score**

COLONY

Merit weighted voting

Votes weighted by **reputation score**

Reputation is a **non-transferable attribute** that can only be earned

COLONY

Merit weighted voting

Votes weighted by **reputation score**

Reputation is a **non-transferable attribute** that can only be earned

Derivative of a generic **token weighted voting system**

COLONY

Why token weighted voting?

Solves the Sybil attack double-voting problem

COLONY

Why token weighted voting?

Solves the Sybil attack double-voting problem

- By distributing voting power over tokens representing real-world value

COLONY

Why token weighted voting?

Solves the Sybil attack double-voting problem

- By distributing voting power over tokens representing real-world value
- What matters is *how many tokens* a user owns

COLONY

Why token weighted voting?

Solves the Sybil attack double-voting problem

- By distributing voting power over tokens representing real-world value
- What matters is *how many tokens* a user owns
- Not into *how many* accounts a user has divided their tokens

COLONY

Why token weighted voting?

Solves the Sybil attack double-voting problem

- By distributing voting power over tokens representing real-world value
- What matters is *how many tokens* a user owns
- Not into *how many* accounts a user has divided their tokens
- “One Benjamin is just as good as 20 Lincolns”

The problems of token-weighted voting

What happens to the tokens when you vote?

Can tokens be moved freely?

COLONY

Tokens as ballots

If voter must transfer their tokens to vote:

We need to return tokens when the poll closes

&

Voter can only participate in one poll at a time

COLONY

Tokens as ballots

1. User has tokens

COLONY

Tokens as ballots

1. User has tokens
2. Is asked to vote

Yes / No

COLONY

Tokens as ballots

1. User has tokens
2. Is asked to vote
3. Votes...

Yes / No

COLONY

Tokens as ballots

1. User has tokens
2. Is asked to vote
3. Votes by sending tokens

The user

Yes / No

COLONY

Tokens as ballots

1. User has tokens
2. Is asked to vote
3. Votes by sending tokens
4. User is asked another question

The user

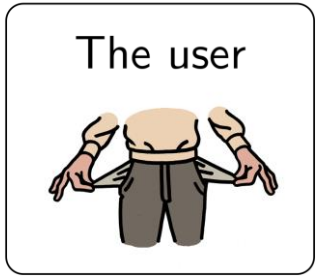
Yes / No

Up / Down

COLONY

Tokens as ballots

1. User has tokens
2. Is asked to vote
3. Votes by sending tokens
4. User is asked another question
5. The user...



Yes / No

Up / Down

COLONY

Tokens as ballots

1. User has tokens
2. Is asked to vote
3. Votes by sending tokens
4. User is asked another question
5. The user can't vote again...

The user

Yes / No

Up / Down

0

COLONY

Tokens as ballots

1. User has tokens
2. Is asked to vote
3. Votes by sending tokens
4. User is asked another question
5. The user can't vote again until *after* the first poll closes and all the tokens are returned.

The user

Yes / No

Up / Down

COLONY

Tokens as weight

If voter sends a voting transaction that *records* their token-weighted vote but the tokens remain with the voter, the voter may double vote.

COLONY

Double voting

1. User has tokens

COLONY

Double voting

1. User has tokens
2. Is asked to vote

Yes / No

COLONY

Double voting

1. User has tokens
2. Is asked to vote
3. Votes...

Yes / No

COLONY

Double voting

1. User has tokens
2. Is asked to vote
3. Votes by sending a message

COLONY

Double voting

1. User has tokens
2. Is asked to vote
3. Votes by sending a message
4. Makes a new account

The user

A "different" user

Yes / No

COLONY

Double voting

1. User has tokens
2. Is asked to vote
3. Votes by sending a message
4. Makes a new account
5. Transfers their tokens

The user

A "different" user

Yes / No

COLONY

Double voting

1. User has tokens
2. Is asked to vote
3. Votes by sending a message
4. Makes a new account
5. Transfers their tokens
6. Votes again...

The user

A "different" user

Yes / No

COLONY

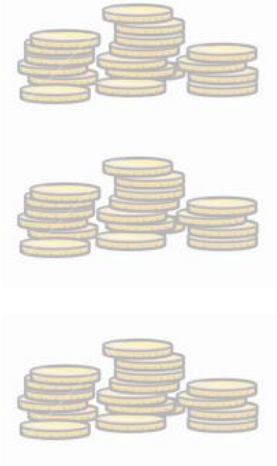
Double voting

1. User has tokens
2. Is asked to vote
3. Votes by sending a message
4. Makes a new account
5. Transfers their tokens
6. Votes again and again...

The user

A "different" user

Yes / No



COLONY

Double voting

1. User has tokens
2. Is asked to vote
3. Votes by sending a message
4. Makes a new account
5. Transfers their tokens
6. Votes again and again and again...

COLONY

Double voting may be prevented by...

Individually accounting for each token transfer as part of the poll evaluation...

COLONY

Double voting may be prevented by...

Individually accounting for each token transfer as part of the poll evaluation...

...or just preventing token transfers. ;P

COLONY

Token locking

1. User has tokens

COLONY

Token locking

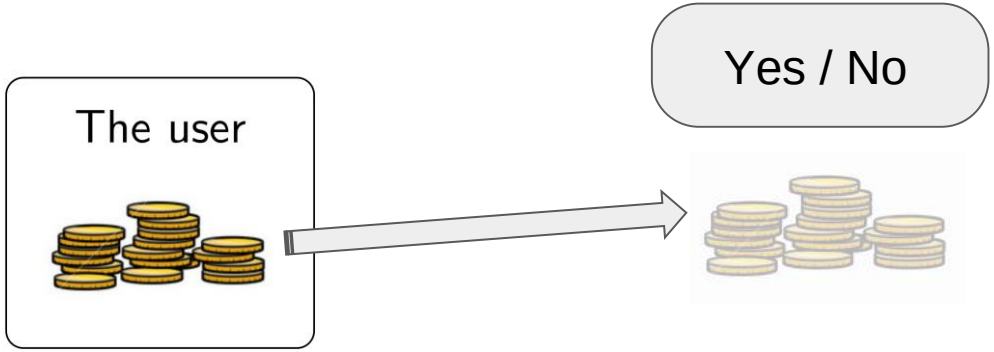
1. User has tokens
2. Is asked to vote

Yes / No

COLONY

Token locking

1. User has tokens
2. Is asked to vote
3. Votes by sending a message



COLONY

Token locking

1. User has tokens
2. Is asked to vote
3. Votes by sending a message
4. Tokens are locked, preventing transfers

Yes / No

COLONY

Token locking

1. User has tokens
2. Is asked to vote
3. Votes by sending a message
4. Tokens are locked, preventing transfers
5. Is asked to vote again

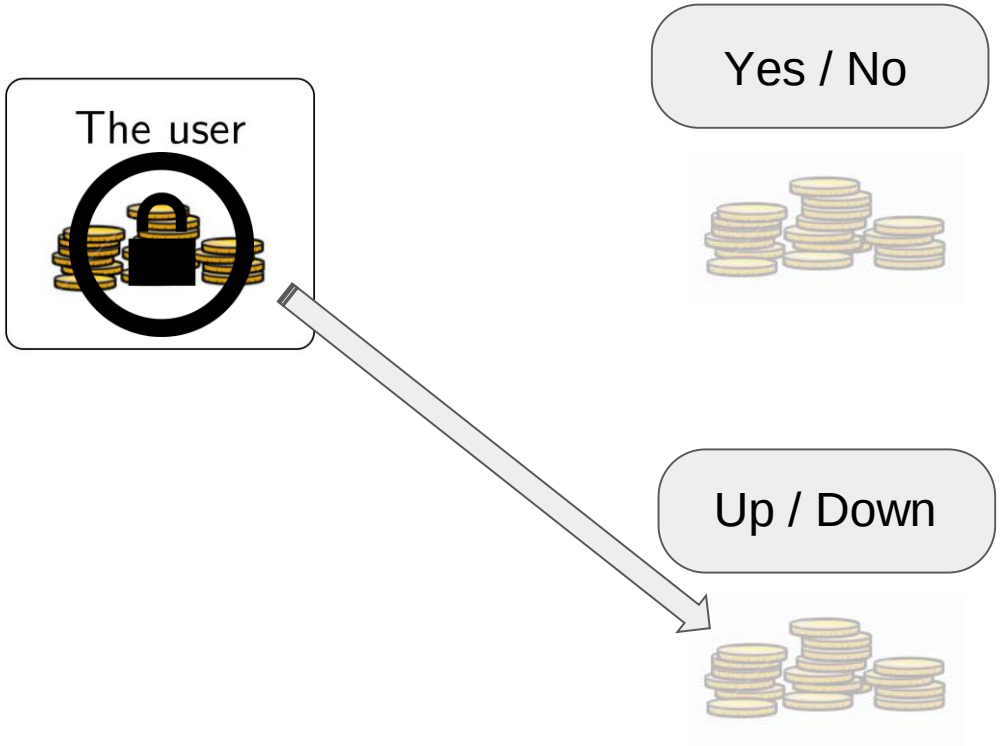
Yes / No

Up / Down

COLONY

Token locking

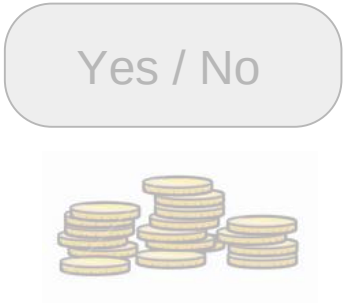
1. User has tokens
2. Is asked to vote
3. Votes by sending a message
4. Tokens are locked, preventing transfers
5. Is asked to vote again
6. Votes again



COLONY

Token locking

1. User has tokens
2. Is asked to vote
3. Votes by sending a message
4. Tokens are locked, preventing transfers
5. Is asked to vote again
6. Votes again
7. When all polls are closed the tokens are unlocked



COLONY

However..

.

COLONY

The problem with token-locking

As locked tokens cannot be moved, voting has a **liquidity cost.**

The **opportunity cost** of the inability to sell tokens until polls close.

Creates an **incentive not to vote** (or to wait until the last minute).

Consequently, **the poll is inaccurate**.

COLONY

Requirements

1. No double voting
2. No skewed incentives

COLONY

The Wisdom of Crowds

The public blockchain is transparent

Running tallies are public in a blockchain poll

But

For the wisdom of crowds to work, votes **must be independent**

So

Running tallies must be secret

COLONY

Requirements

1. No double voting
2. No skewed incentives
3. No Bandwagon effect

COLONY

Q. Why can’t we collect all votes and calculate the result in one go?

A. The *gas limit*.

COLONY

Requirements

1. No double voting
2. No skewed incentives
3. No Bandwagon effect
4. Scalable

COLONY

The Colony Voting Protocol

COLONY

The Colony Voting Protocol

- Prevents **double voting Sybil attacks** by token weighting
- Prevents **token-transfer double-voting** by targeted locking.
- Prevents **skewed incentives** by keeping users in control of their tokens.
- Prevents **bandwagon voting** by masking open polls.

  and

- **Scales** to arbitrary number of voters and simultaneous polls.

COLONY

Vote masking

- The user has tokens

COLONY

Vote masking

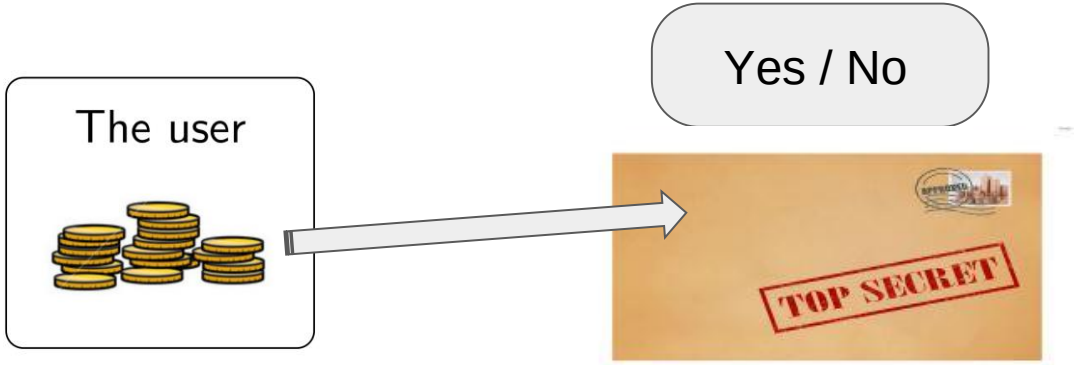
- The user has tokens
- While polls are open...

Yes / No

COLONY

Vote masking

- The user has tokens
- While polls are open users can vote by sending *masked votes*



COLONY

Vote masking

- The user has tokens
- While polls are open users can vote by sending *masked votes*
- Tokens are free to move while the poll is open

Yes / No

COLONY

Vote masking

- The user has tokens
- While polls are open users can vote by sending *masked votes*
- Tokens are free to move while the poll is open
- When the poll closes the account is locked

COLONY

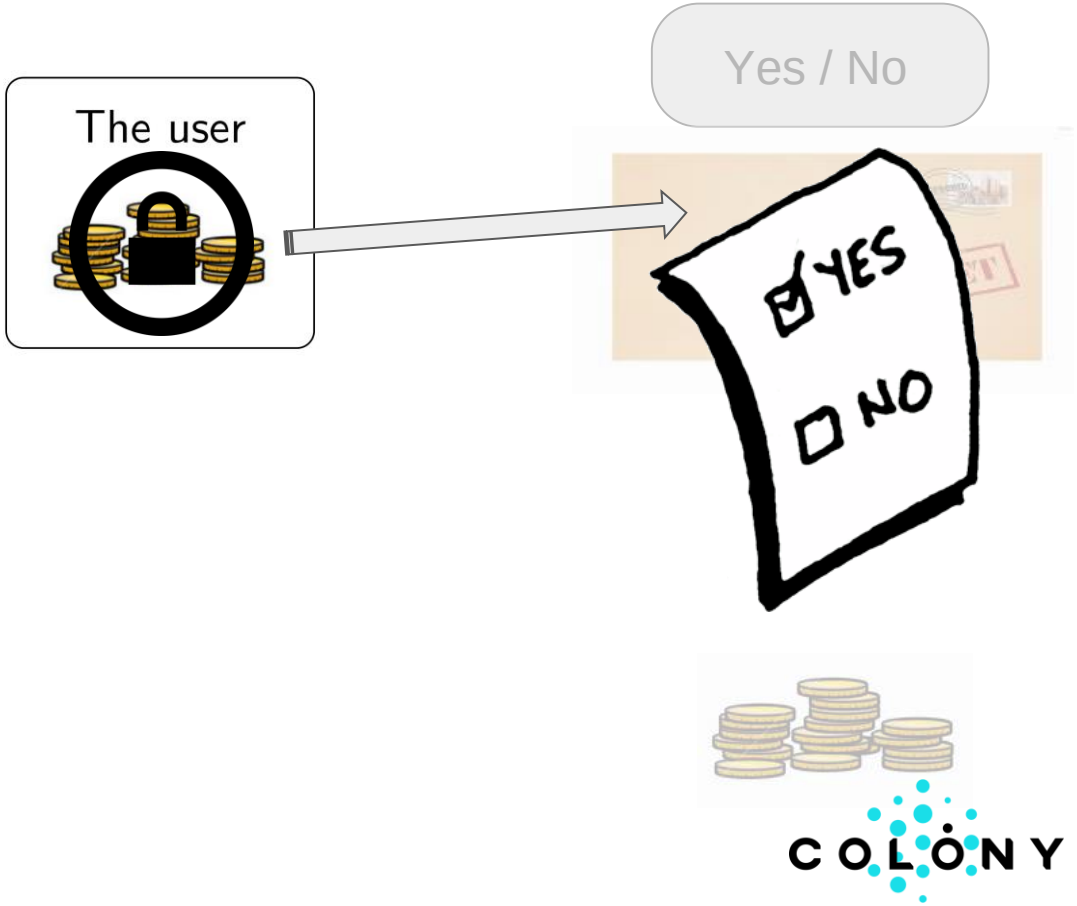
Vote masking

- The user has tokens
- While polls are open users can vote by sending *masked votes*
- Tokens are free to move while the poll is open
- When the poll closes the account is locked
- Tokens can't move but can still vote in other polls

COLONY

Vote masking

- The user has tokens
- While polls are open users can vote by sending *masked votes*
- Tokens are free to move while the poll is open
- When the poll closes the account is locked
- Tokens can't move but can still vote in other polls.
- User can *immediately(!)* unlock account again be revealing and counting their vote

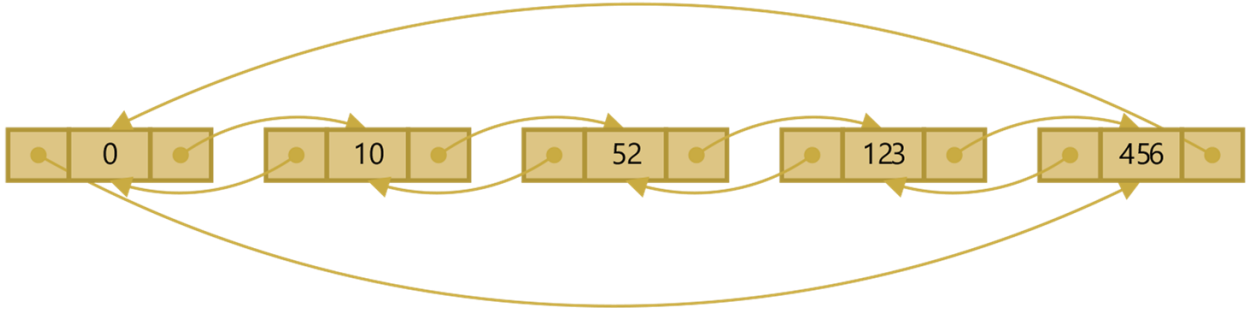


We must be able to:

- Track which polls an account has participated in
- Track which polls are open and which are closed
- Track which votes must be revealed to unlock an account
- Determine whether an account is locked
- Handle incoming transactions to locked accounts

COLONY

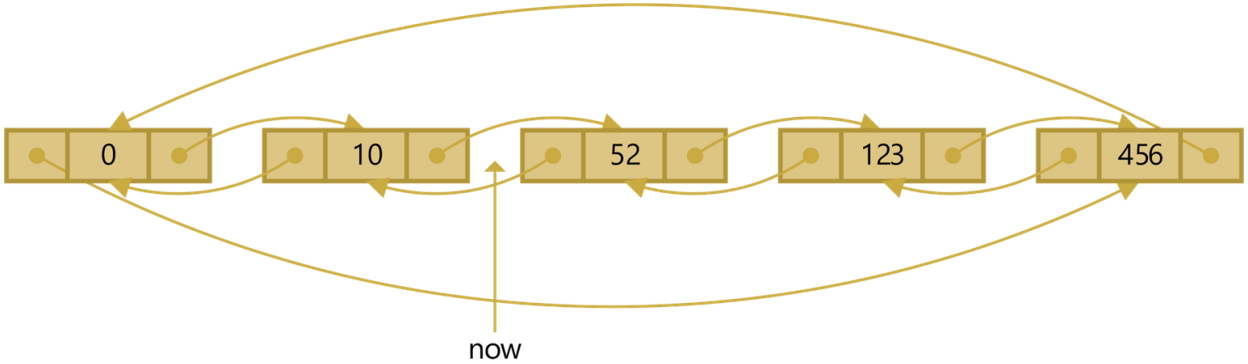
Vote secrets data structure



- Double-linked list of unrevealed vote secrets for a user
- Ordered by poll close time
- Sorting the list is delegated to the user
- Implements a zero-entry item

COLONY

isAddressLocked function



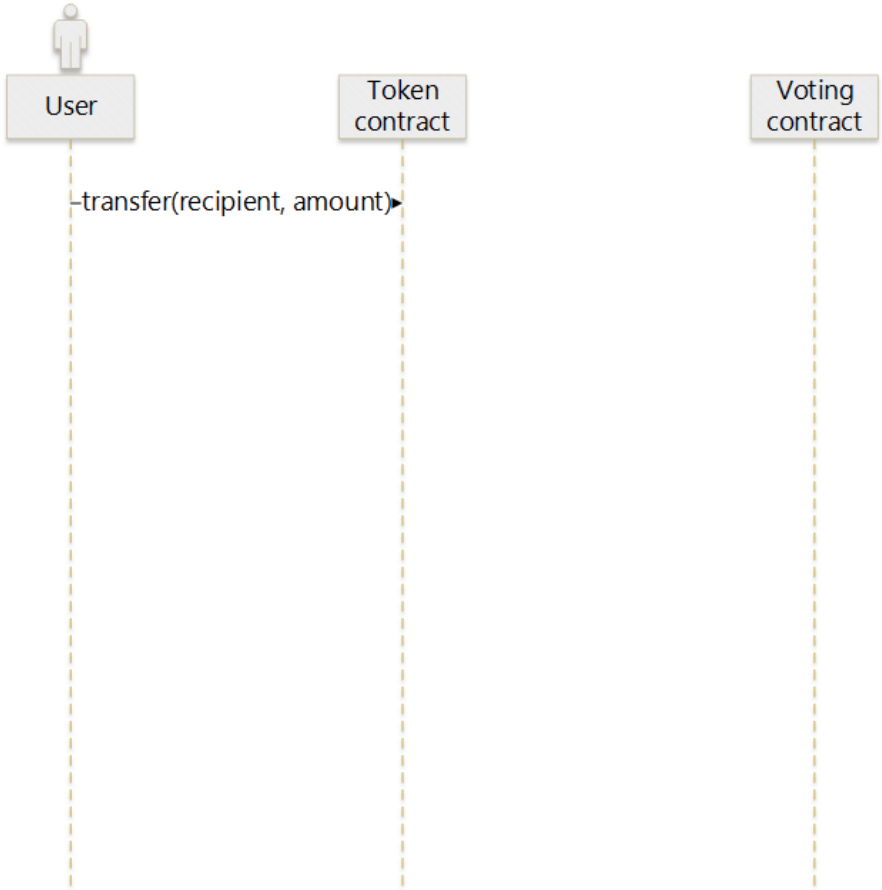

```
function isAddressLocked(address userAddress) constant returns (bool){
    var zeroPollCloseTimeNext = eternalStorage.getUIntValue(sha3("Voting",
userAddress, uint256(0), "nextTimestamp"));

    if (zeroPollCloseTimeNext == 0) { return false; }
    if (now < zeroPollCloseTimeNext) { return false; }
    else { return true; }
  }
```

COLONY

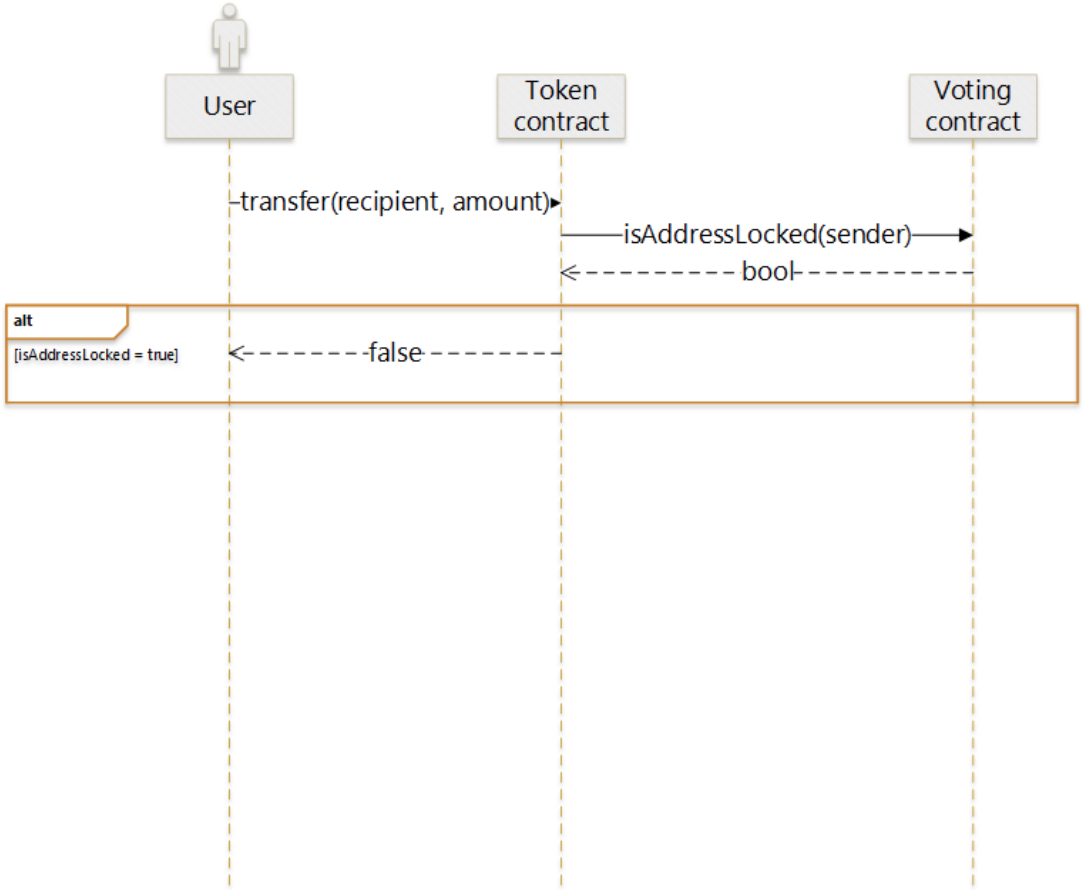
Token Locking

Extension to the standard ERC #20 token implementation



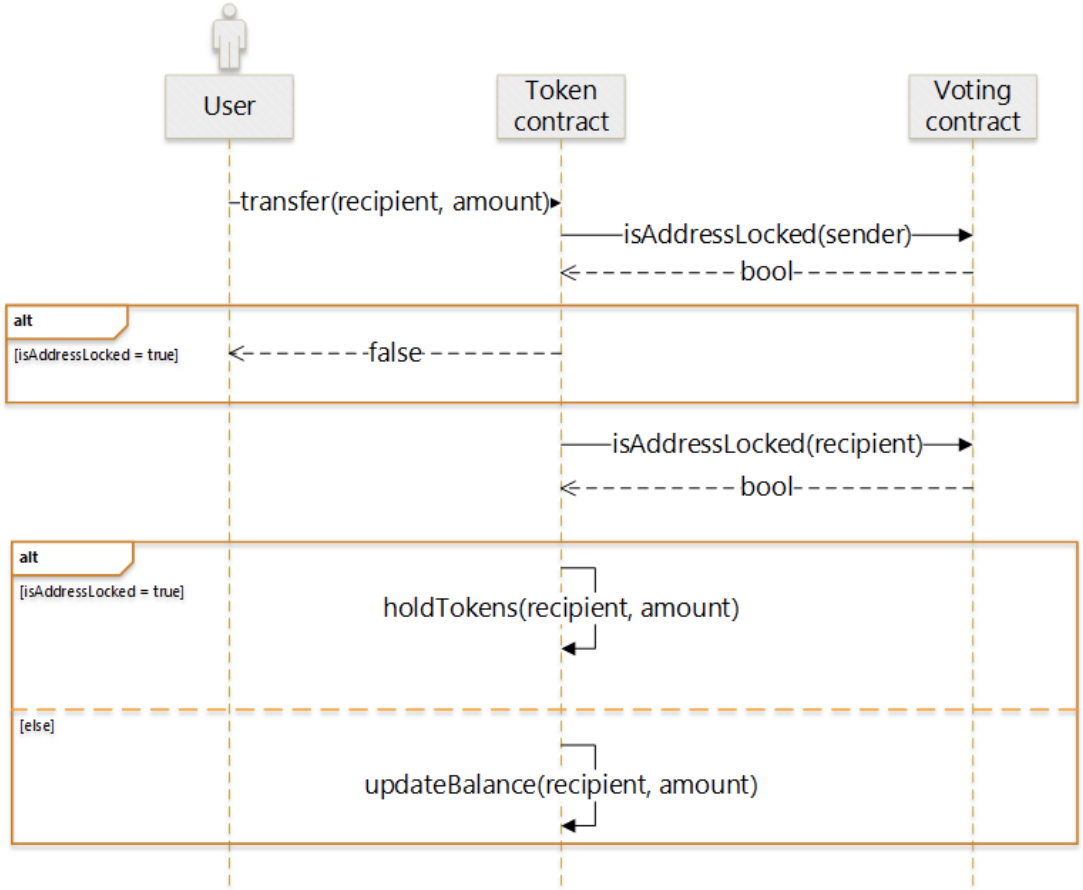
COLONY

Token Locking



COLONY

Token Locking

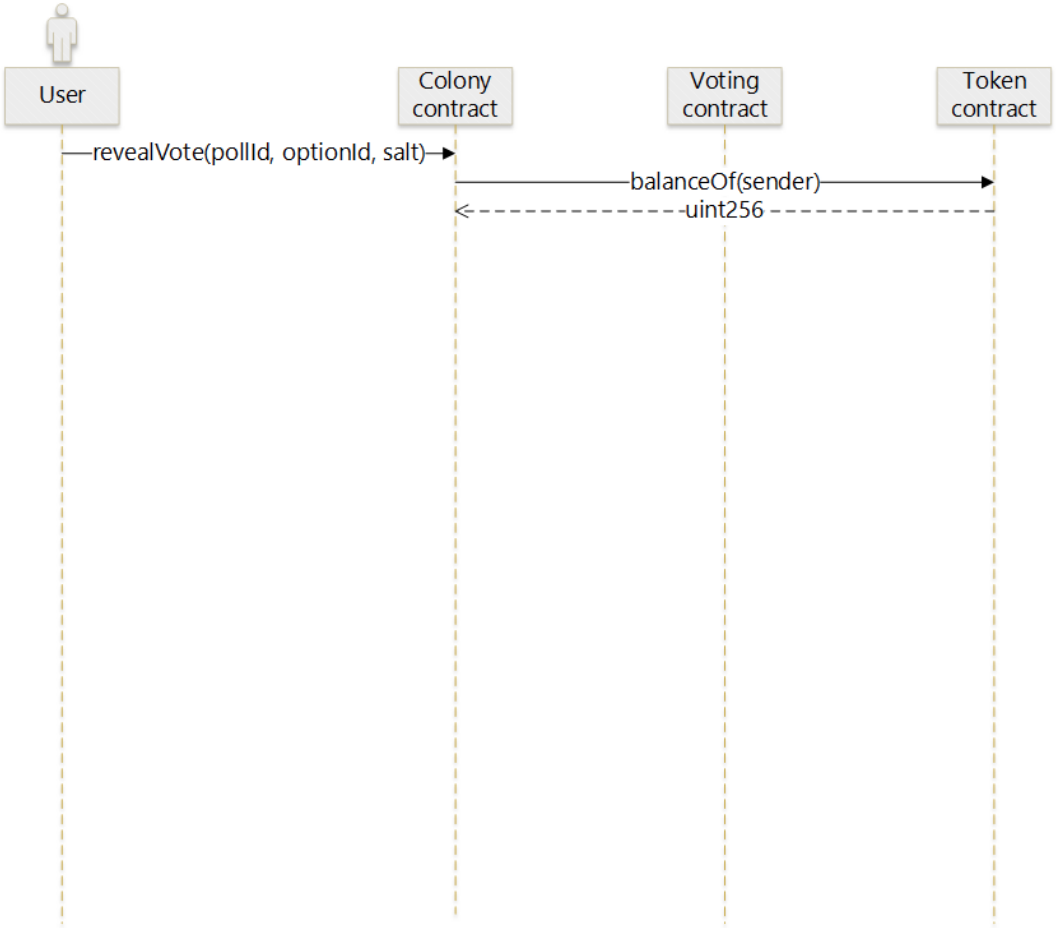


COLONY

Revealing a vote & Unlocking tokens

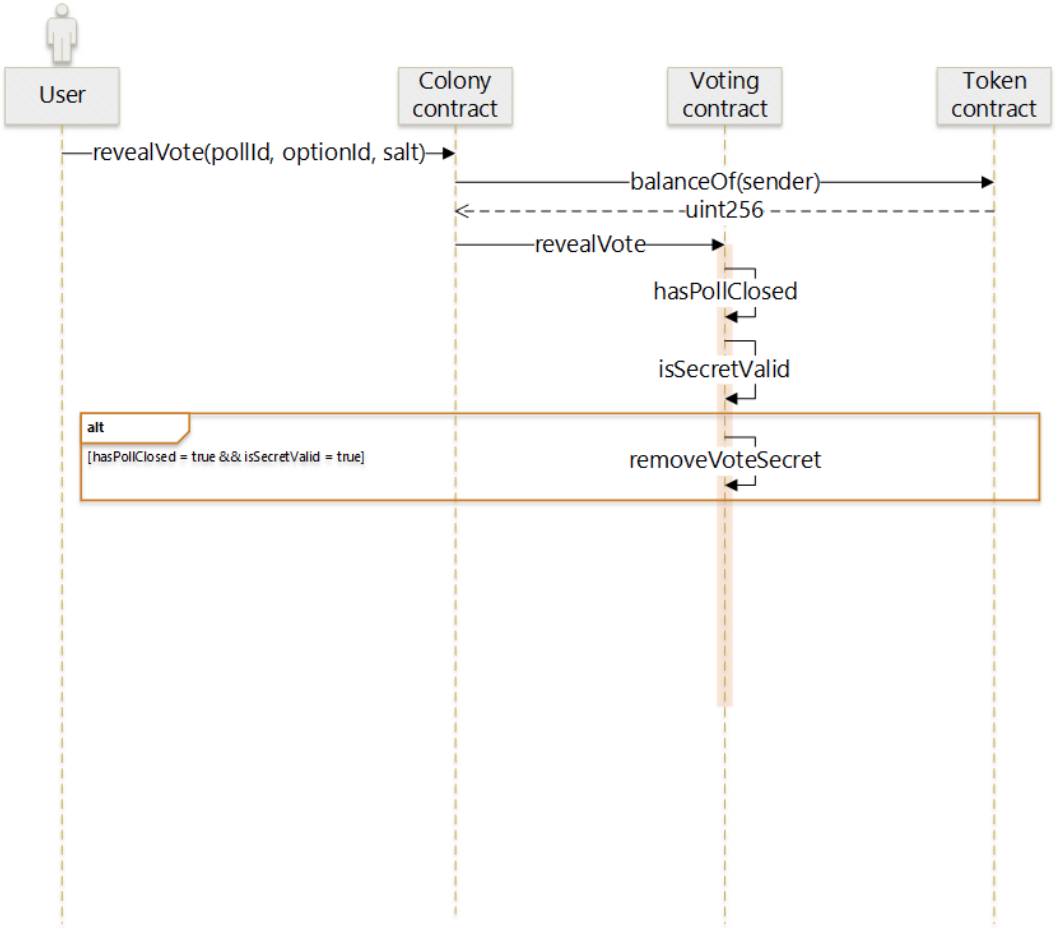
COLONY

Revealing a vote & Unlocking tokens



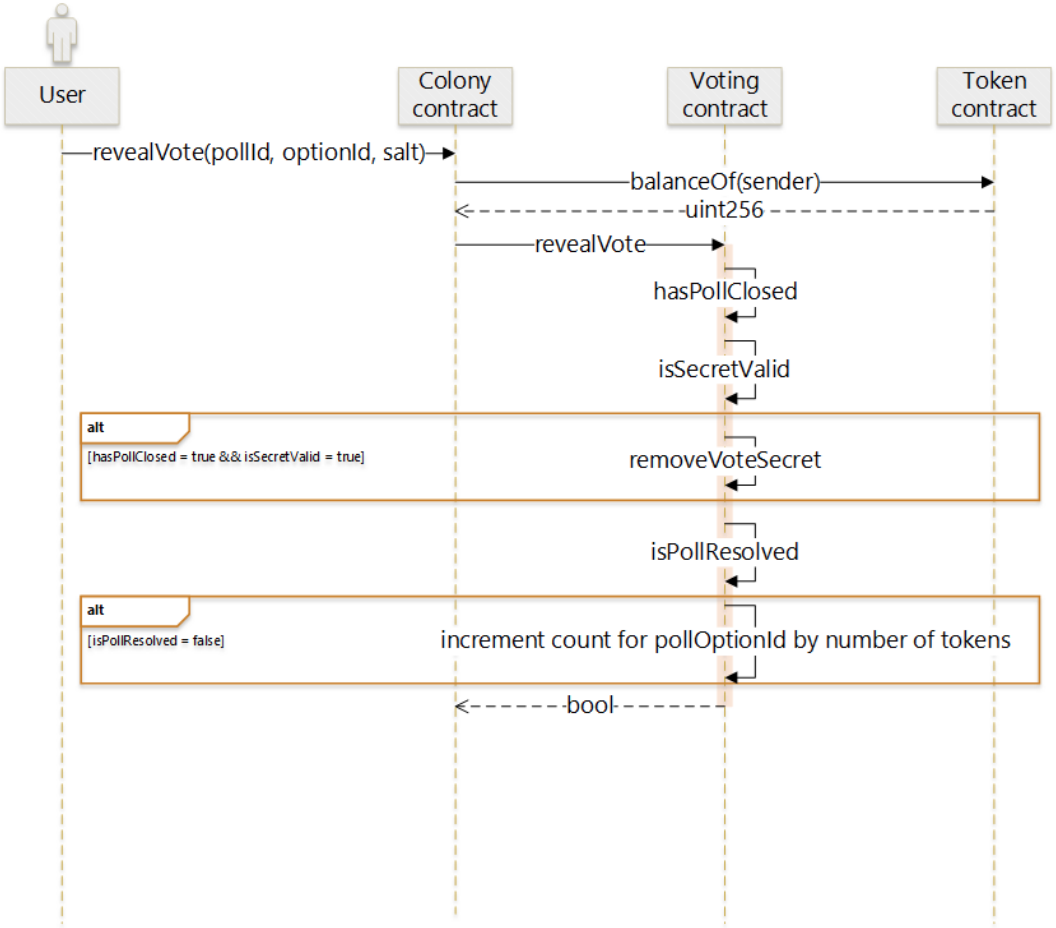
COLONY

Revealing a vote & Unlocking tokens



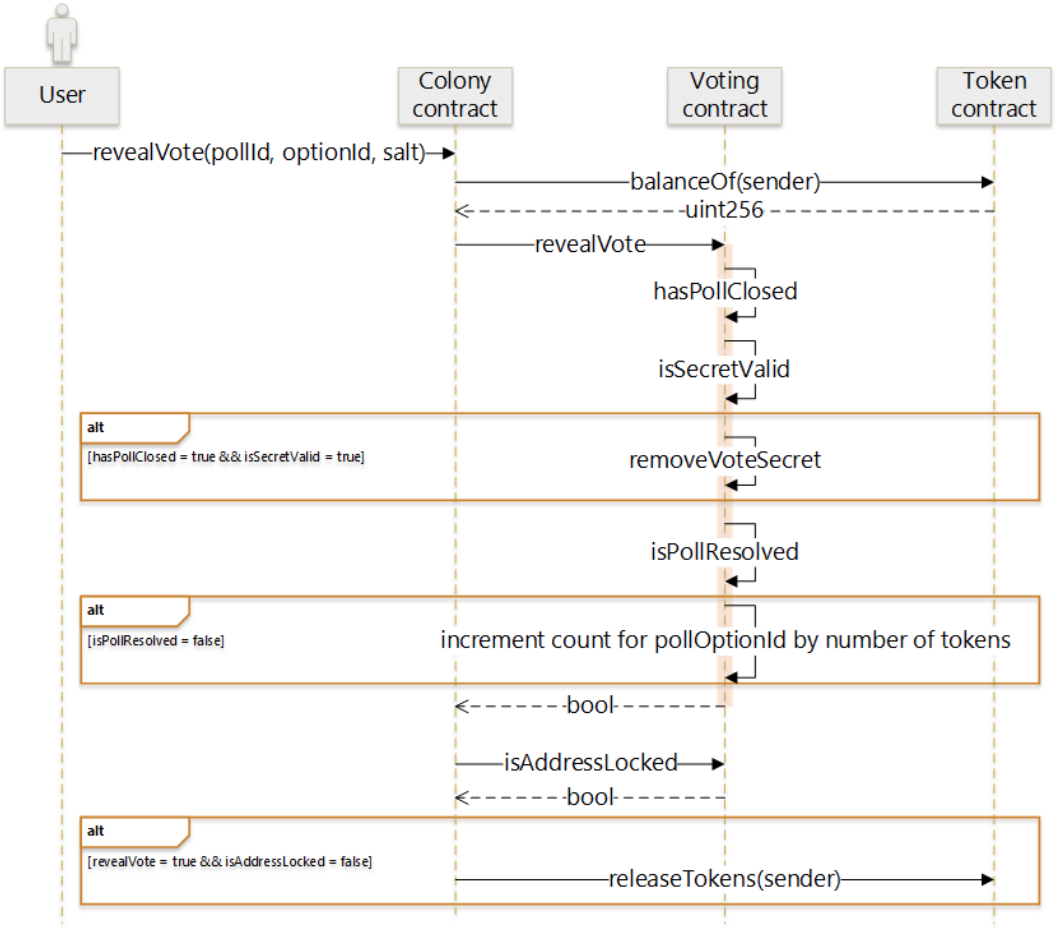
COLONY

Revealing a vote & Unlocking tokens



COLONY

Revealing a vote & Unlocking tokens



COLONY

Where to find us

COLONY.IO

COLONY.FOUNDATION

BLOG.COLONY.IO

HELLO@COLONY.IO

@JOINCOLONY

Recommended by JJ Cohoon, Christian Maniewski, and 19 others

Visions of the Future

rLoop

Colony
Aug 24

Recommended by Aron Fischer, Christian Maniewski, and 12 others

Writing robust smart contracts in Solidity

Before contract function code executes, it's a good idea to validate who triggered it and what inputs are given.

Elena Dimitrova
Aug 10

Recommended by Chris Moyer and 11 others

Why Colony is 100% behind ETH

I was a vocal proponent of the hard fork because for me it represented the best solution to an awful situation we (the Ethereum community...

Jack du Rose
Aug 4

COLONY

We're hiring!

Collin Vine
COO

Jack du Rose
CEO

Dr. Alex Rea
CTO

Thiago Delgado
Engineering

Elena Dimitrova
Engineering

Karol Podlesny
Design

Christian Maniewski
Front-End Dev

Dr. Aron Fischer
R&D

You?
Engineering

You?
Engineering

You?
Engineering

You?
Front-End Dev