Decentralized Voting on WavesPlatform

Raymond Ng
raymond@katalystcoin.com
https://katalystcoin.org
Telegram : @raymondngkh

Abstract

This is a short paper on a quick way to implement decentralized voting on the waves platform. By using existing data and blockchain framework of Waves, one can implement decentralized voting in less than a month with testing.

Decentralized Voting Mechanism

Wavesplatform already has the data infrastructure to implement a decentralized voting mechanism. By leveraging on the existing data framework, one can implement decentralized voting quite readily.

Everyone would be able to start a voting for decisions that matter to them basing on various token ownership parameters.

A typical waves transaction would entail the following information;

```
{
 Transaction id
 Type
 Timestamp
 Block height
 Amount
 Asset
 Asset Name
 Fee
 Fee Asset
 Fee Asset Name
 Sender
 Recipient
 Signature
 Attachment
}
```

By using the following variable in the following manner you can implement a decentralized voting mechanism.

Asset - An asset to denote voting (no decimals) event. Exactly 1 token of asset would be airdropped before the designated date / period of voting.

Attachment - The vote in question. To make it unique, the attachment may start by “Vote : <choice>”

Timestamp - Timestamp is important only insofar if the voting is only allowed within a certain time period. So if the timestamp would act as a filtration mechanism whether to include or exclude the vote.

Recipient - The recipient designated for the voting. There may be a new waves address created for individual and unique voting events.

There is not a need to construct a new “Type” for votes as long as the software protocols search for the occurrence of the string, “Vote :” and also the asset that is designated only for votes.

Vote Counting

For a lot of voting, it is also to take into account of the tokens that a specific voter possesses at a given time. So the timestamp is important here as it details the time which the vote counting would take the token amount into account.

Anonymity

The software protocol defined in this section does not guarantee privacy or anonymity because the vote details is open to all to see in “Attachment”. With the Sender information right in the open for any viewer on the Net.

One may employ the use of a symmetric cipher. The ciphertext can be a concatenation of 3 text.

$ciper = cryptosign($string) // using Recipient’s private key.

The author is unaware of how the Waves cryptographic signing works, so would not be able to provide any more data / conjectures. In theory, only the recipient could have produced that $cipher via its private key signing, and the recipient would be the only reader of the vote decision.

Vote Fraud - Double Voting (Same Address)

To prevent double voting, the eventual counting routine could be programmed to count only the first & earliest voting transaction recorded on the blockchain. All eventual votes are all ignored.

Vote Fraud - Double Counting of Tokens

To prevent the case in which one voter after voting passes his / her tokens to the next voter so that his tokens are counted twice / more, the counting routine must take into account of the token balance at a given predefined time.

If done in this way, the passing of token to another voter after a voter votes would not have bearing on the vote results as the counting routine only counts the amount of a specific token owned at a given time.

Use Case - Curation

The voting can be executed for projects to indicate the number of people with the tokens are supportive towards a project.

There are a few variables considered here. They are as follows;

$tokensupply = Token circulating supply
$votedtokens = total number of tokens actually voted
$yeavote = votes in support
$nayvote = votes not in support

Depending on the desired curating criteria, you can set the voting to be invalid if it is voted by less than a certain number of token holders.

However, upon fulfilling that basic requirement, one may start to have a web plugin to output the star logos upon the following criteria

5 yellow stars litted up for $yeavote/$votedtokens >= 0.80

4 yellow stars litted up for 0.60 >= $yeavote/$votedtokens > 0.80

3 yellow stars litted up for 0.40 >= $yeavote/$votedtokens > 0.60

2 yellow stars litted up for 0.20 >= $yeavote/$votedtokens > 0.40

1 yellow stars litted up for 0.20 >= $yeavote/$votedtokens.

Where there is a need to reduce processing time for the stars to appear on the web, there may be a routine to commit to the blockchain the voting end result so that the web plugin may just look at 1 defining entry on the blockchain instead of processing the results every time the web with the web plugin is loaded.