# rv2020 - RChain Voting Dapp

Originally Steve Henley and Jim Whitescarver

## Summary

The voting dapp will be used by coop members to vote for board of directors (BOD) and Items of Business (IOB) during the annual RChain membership meeting in October 2020.

## Requirements

1. Runs on RChain
2. Anonymous voting
3. Employs email address rchain coop identities
4. Yes/no/abstain questions
5. Other question types? Text alternatives, rank, estimate
6. Admin: open, close voting, (multisig/consent, AllPromises)
7. Voter interface
8. Tally of results
9. Open and close voting
10. Decisions by percentages (majority, ⅔, …), medians
11. ~~Collect member information!!! Member survey~~
    a. ~~Rev address (generate one if needed)~~
    b. ~~Discord user~~
    c. ~~Priorities? invest/build/coordinate/~~
    d. ~~keywords~~

## Timeline

- April 7th - requirements gathering
- June 1st - proof of concept (POC)
- July 1st - prototype
- August 1st - complete voting dapp
- August 2nd - start dapp testing
- September 1st - finish testing
- September 1st - call for BODs nominee & IOBs
- September 15th - closing for BOD and IOB submissions
- September 16th - BOD and IOB vetting
- October 24th - annual general membership (AGM) meeting

### Method 1 anonymous voting with ticket (jim)

1. Generate random tickets
2. Distribute tickets randomly to coop members by email
3. Election app verifies ticket and deploys the votes anonymously
4. Association of member email and ticket destroyed after vote closing?
5. Map of ticket hashes votes/answer to each question on rchain
6. Provide website or interface to vote (2 options)

    a. Dappy - download dappy
    b. Tomislav (Metamask) - visit website
    c. Command line voting -
7. Meta information would be how the members votes
    a. Voter can update their vote
8. Import data into CSV file to count results (yes/no or integers) or tally in rholang

## Method 2 functional encryption anonymous voting (theo)

Using https://github.com/fentec-project/CiFEr for anonymous voting of form (A, B, C, D… or neither)

1. Co-op generates a master key.
2. Co-op generates an identity vector [1, 1, …., 1, 1] and encrypts it with the master key.
3. Co-op generates member voting keys that are derived from the master key.
4. Co-op sends member voting keys to members through email (or discord) as a link to a dapp
5. Members client compose an array of form [0, 0, …., 1, 0] that represents the user vote.
6. Members client encrypts the vote making it's content unreadable to anyone else (including owner of the master key)
7. Members client hashes the key to get the unique vote ID hash and uses that to put the encrypted data on chain in a key-value map.
8. Voting period closes.
9. Co-op releases it's master key to the public and puts it on the blockchain.
10. Co-op (or anyone) takes the key-value map values from the blockchain and does an inner product with its own identity vector to ensure votes have at most one ticked option, returning:
    0: member did not vote for any option (can be made oblivious as well)
    1: member voted for one option.
    2+: invalid vote, maximum of one is allowed.
11. Co-op (or anyone), can calculate the final voting results on encrypted data by taking each individual column and doing an inner product with the identity vector, returning a readable number that corresponds to the number of votes for that option. Individual votes remain anonymous.

## Method 3 voting using rev wallets (greg)

Comments from Greg Meredith:

- ~~a REV address is issued to each coop member and is funded with dust.~~
    - Member send us a rev address
- The private and public keys of the addresses are emailed to the members
- A REV address is issued for each possible choice of each ballot item.
- The member then votes by using their wallet to send dust to particular addresses.
- Any wallet can be used for voting. Optionally we can create a browser based app to simplify voting.
- All transactions for all ballot choice addresses are extracted from the transaction history by a script.
- If a voter has multiple votes for one choice then their vote will not count.
- The REV will be returned to the voter after all voting has been completed so that they can vote again in future ballots .

- The coop has a list of all addresses that are eligible to vote and removes addresses that are not available by a script.
- The votes can then be tallied by a script

### Method 4 (raphael)

- Create 2000 key pairs
- Send them by email to members
- Deploy ERC 1155 contract with 2000 tokens (not associated)
- Provide website or interface to vote (2 options)
    - Dappy - download dappy
    - Tomislav (Metamask) - visit website
    - Command line voting -
- Meta information would be how the members votes
    - Voter can update their vote
- Import data into CSV file to count results (yes/no or integers)
    - Export data tally data
    - Tally data online (involve storing the voting on rchain)
- Real-time voting results

### Method 5 Zilip plus rhobot,(~~liquid democracy~~?)

- Metamask login - add rev address to user record, message data, tomasluv?
- Improvements to voting-bot
    - Only admin can add issues and vote alternatives
    - Only registered rev addresses can vote
    - Simplified presentation of all issues in order with simple numeric response
    - Report of all results in order
- Stretch reqs
    - Send encrypted votes to chain
    - Ballot per stream

### Method 6 contract deploy all - tomisluv

- Generic vote contract where you paste in
    -

### Github

https://github.com/rchain-community/rv2020
https://github.com/rchain-community/rv2020/tree/master/o2r moved from rchain-dbr
https://github.com/rchain-community/community

## Gdoc

## Education

How to leverage the voting dapp for educational use