On-Premise Voting on Blockchain

Background

The aim of this paper is to propose a methodology for the execution of an on-premise voting process on blockchain in a transparent manner while ensuring the anonymity of voters. This will be applied to the Annual General Meeting (AGM) of the Singapore Cryptocurrency and Blockchain Industry Association (ACCESS) on 10th January 2020.

Requirements

At the AGM, association members will place their votes for future appointment holders and agenda items on a blockchain, utilising the advantages of blockchain technology such as immutability, transparency, and traceability. In addition, anonymity is established through the disconnection between members and their vote, hence a vote cannot be traced back to its originator.

The key requirements are:

1. Due to constraints of the association's constitution, voting is to be conducted on-premise at the AGM.
2. Anonymity. A member's vote should not be trackable to his/her identity.
3. No double voting. Every member gets only one vote.
4. Transparency. Post voting, the process needs to be transparent and auditable.

For this application, we are using Ethereum as the blockchain network and Metamask as the Ethereum wallet to interact with the blockchain.

Process

Access Controls

Access to the smart contracts will be controlled by designating (or whitelisting) pre-defined public Ethereum addresses with the deployment of the contracts. Controls change according to the stage of voting. The actors are: 1. Admin, 2. Voting Machines. There are 3 stages where access controls are changing:

A. Pre-Initialization
   Only the admin can invoke the Code Management and Voting Contracts
B. Voting
   The voting machines can invoke the voting contracts.
   Admin can invoke the end of the voting process
C. Post-Voting
   No further invocation of contract functions
   Open access to voting results

Voting Code Generation and Management

A list of voting codes are randomly generated, hashed and stored in a Code Management Smart Contract, before physical copies are printed as a QR code and randomly distributed to members upon registration at the AGM. It is important to emphasize that the codes are linked to neither voters nor their votes, providing anonymity. Also, it is extremely unlikely to reproduce the same code within the given time frame, hence it is fair to assume the authenticity of all submitted votes.

To ensure voters are only permitted to submit one vote, we track the status of each code in the Code Management Smart Contract too. There are three status, valid, used and invalid. Originally, all codes are valid and can be used for voting. On the other hand, used and invalid codes cannot be used to vote. The used status can only be invoked by the Voting Smart Contract in the event of a successful vote. The invalid status can only be invoked by the Admin account and is used to invalidate spare codes before voting starts.

Voting

The Voting Smart Contract is used to initiate and track votes. Once deployed, the Voting Smart Contract can be initiated by the Admin account. It will contain the positions, candidates running under each position and the number of seats available for each position.

Voters can then send their votes via generic machines (with a fixed blockchain address) to the Voting Smart Contract. Voters have to present an unused code before they are allowed to proceed with selecting candidates.

After all the votes have been received, the admin can terminate the voting process. This will then generate the voting results, with the majority displayed in green, ties in yellow and the remaining in black. In the event of a tie, tied candidates enter another round of voting until a consensus is reached. Voters are advised not to dispose of their code as it will be reused in every round of voting.

The flow of events to set up on-premise voting can be seen in the diagrams below:

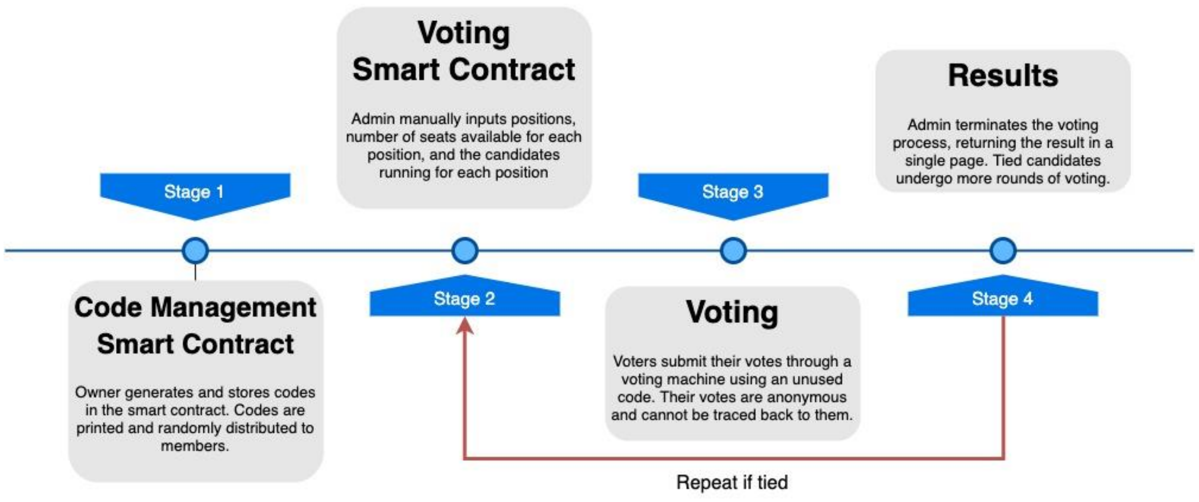

Voting Smart Contract
Admin manually inputs positions, number of seats available for each position, and the candidates running for each position
Results
Admin terminates the voting process, returning the result in a single page. Tied candidates undergo more rounds of voting.
Stage 1
Stage 2
Stage 3
Stage 4
Code Management Smart Contract
Owner generates and stores codes in the smart contract. Codes are printed and randomly distributed to members.
Voting
Voters submit their votes through a voting machine using an unused code. Their votes are anonymous and cannot be traced back to them.
Repeat if tied

Figure 1: Stages of on-premise voting on blockchain

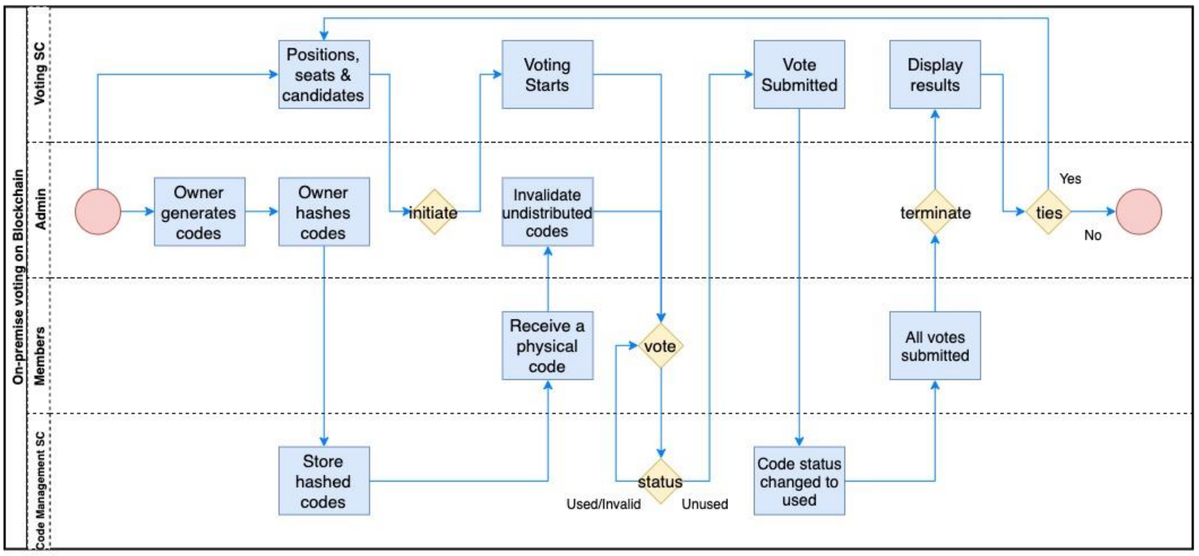


Figure 2: Swimlane diagram of the on-premise voting process

Implementation

The process is implemented via deployment of the smart contracts on the Ethereum Main Net. The user interface is deployed as web applications, there are two UIs one for the admin and another of the voting machines.

The Admin UI (which only the whitelisted admin address can access), allows the admin to initialize voting by keying in the voting items and calling the voting smart contract. It also allows for the invalidation of unused codes and for the termination of voting.

The Voting Machine UI (which only whitelisted addresses for voting can access) allows a voter with a valid voting code to cast a vote via the voting smart contract. For ease of entering the voting codes, a QR code generator is used to create QR codes and printed on voting cards. A QR code reader is implemented on the UI. The code reader will pass the code, the code then gets hashed and checked against the Voting Code Management Contract for the code's validity. Given that the codes are valid, the UI will collect the votes for each voting item and compile them into a transaction that invokes the Voting smart contract.

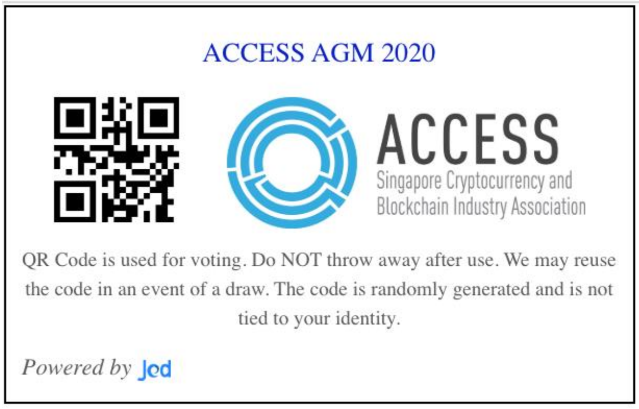
Conclusion

We have demonstrated a process utilizing a random code generator and two smart contracts above that allows voting to be done on-premise and recorded on the Ethereum blockchain with transparency and allowing for anonymity of voters at the same time. The method described allows for use cases where group consensus need to be collected physically on multiple issues and there is a need for a clear audit trail of the voter actions.

As some of the process is offline or off-chain, there is a need to ensure due diligence is done in the generation, distribution and voiding of codes; voting also need to be conducted in a proper manner and should be observed by neutral parties.

This paper is prepared by JEDTrade Pte. Ltd. Any queries may be directed to consultancy@jedtrade.com.

Appendix

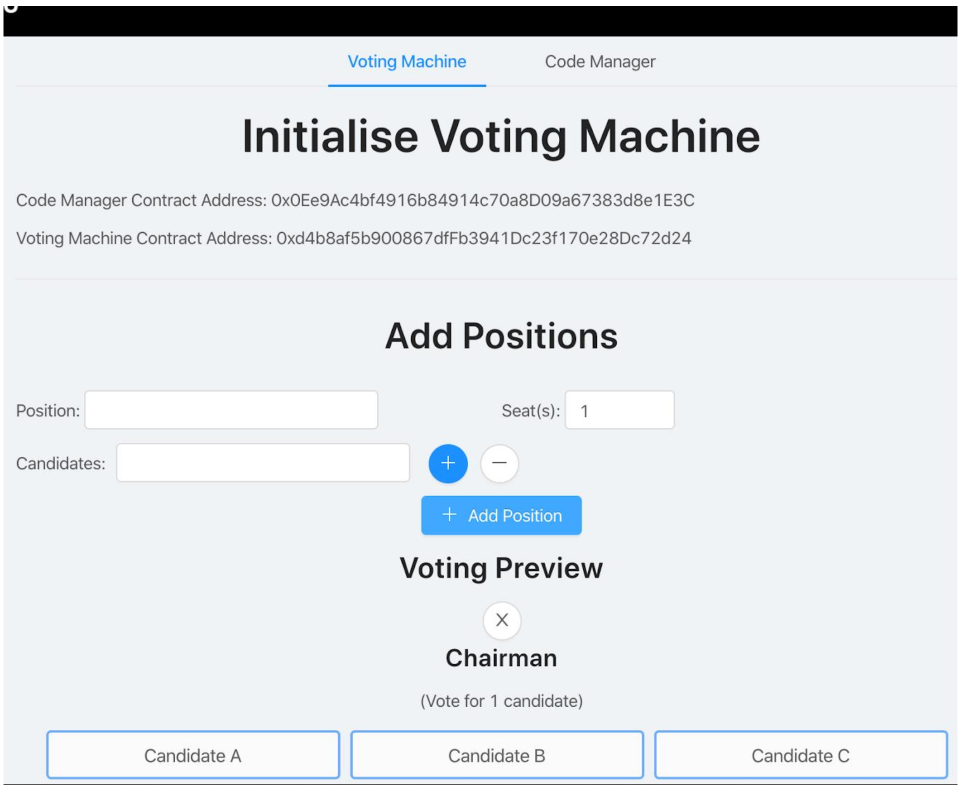

ACCESS AGM 2020

ACCESS Singapore Cryptocurrency and Blockchain Industry Association

QR Code is used for voting. Do NOT throw away after use. We may reuse the code in an event of a draw. The code is randomly generated and is not tied to your identity.

Powered by Jed

Figure 3: Sample Voting Card with QR Code



Initialise Voting Machine

Code Manager Contract Address: 0x0Ee9Ac4bf4916b84914c70a8D09a67383d8e1E3C

Voting Machine Contract Address: 0xd4b8af5b900867dfFb3941Dc23f170e28Dc72d24

Add Positions

Position:

Seat(s): 1

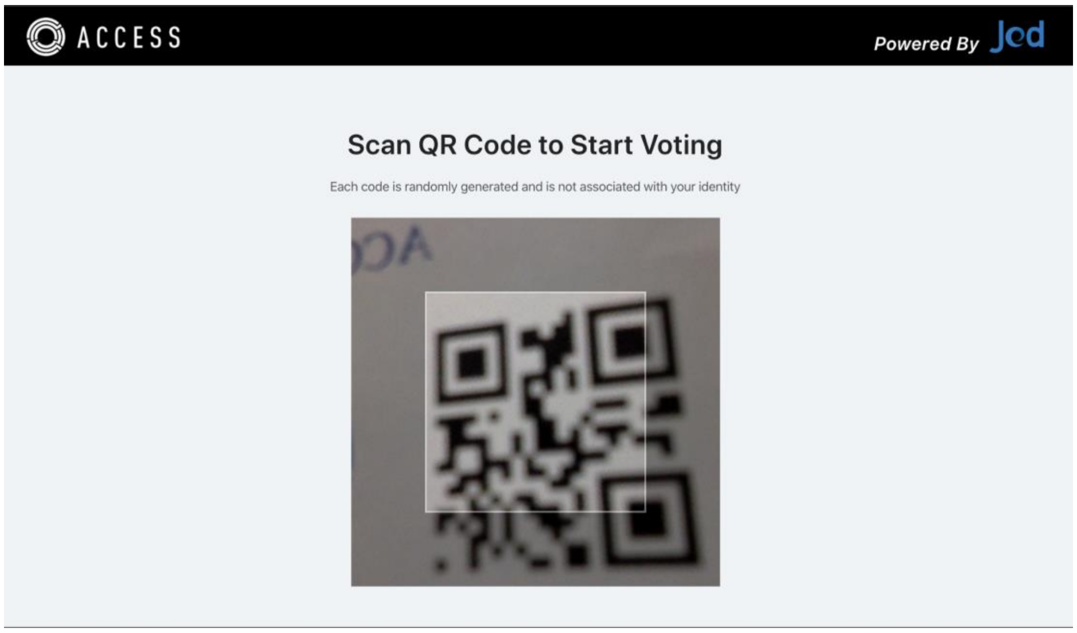
Candidates:

Voting Preview

Chairman

(Vote for 1 candidate)

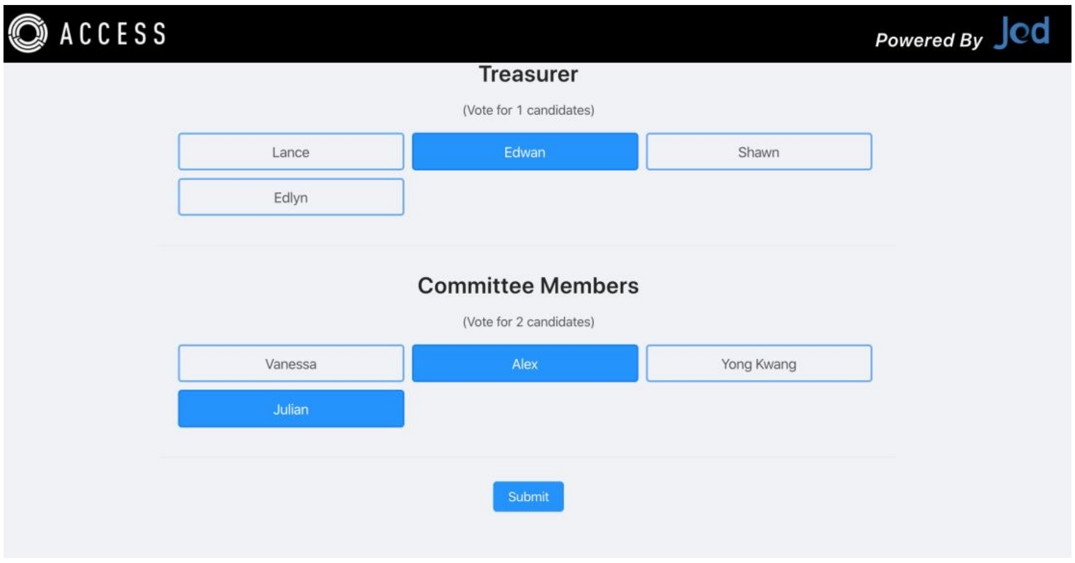
Figure 4: Admin's UI 1: Adding Positions



Scan QR Code to Start Voting

Each code is randomly generated and is not associated with your identity

Figure 5: Voter's UI 1: QR Code Scanning



Treasurer

(Vote for 1 candidates)

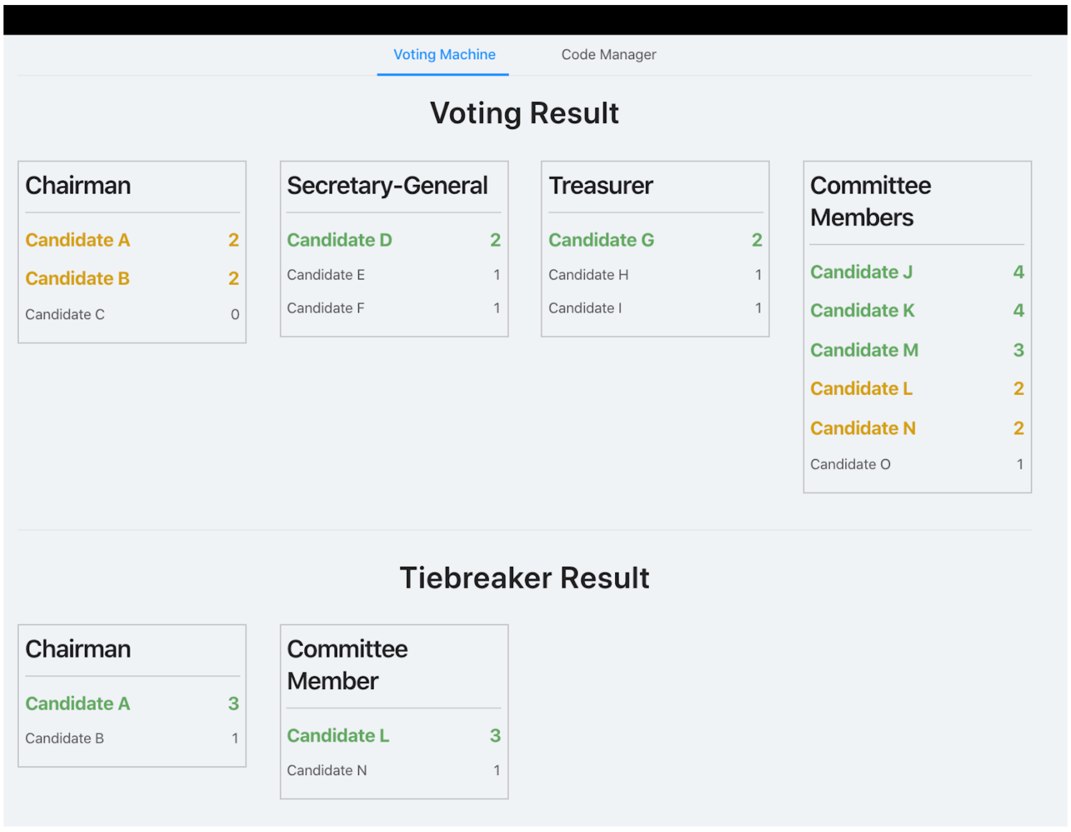
Lance | Edwan | Shawn | Edlyn

Committee Members

(Vote for 2 candidates)

Vanessa | Alex | Yong Kwang | Julian

Figure 6: Voter UI 2: Vote Casting



Voting Result

Chairman	
Candidate A	2
Candidate B	2
Candidate C	0

Secretary-General	
Candidate D	2
Candidate E	1
Candidate F	1

Treasurer	
Candidate G	2
Candidate H	1
Candidate I	1

Committee Members	
Candidate J	4
Candidate K	4
Candidate M	3
Candidate L	2
Candidate N	2
Candidate O	1

Tiebreaker Result

Chairman	
Candidate A	3
Candidate B	1

Committee Member	
Candidate L	3
Candidate N	1

Figure 7: Results Page