Decentralized E-Voting System Using Blockchain

Abstract

In 21st century, there is huge political unrests among political leaders in many developing countries, in spite of Boom of Internet users, still the elections are held in semi-old fashion. The cost associated with these elections (i.e. appliances, workforce related, transport etc.) act as burden on tax payer's money & cuts the huge possibilities of R&D in any nation. Specially in south-east nations, Indians (NRIs) move abroad for various reasons; As a result, casting vote for them becomes very difficult, hence we created mechanism where NRIs can vote securely using their passports.

I. Introduction

Originally, blockchain was just known to some CS fellows & researchers for how to structure & share data. Today blockchains are hailed the "5th evolution"[1] of computing. Blockchain (Blockchain) in a Network, where Block allude to the list of transactions noted into distributed ledgers over a given period of time. It has three main components i.e., Size, Period, & Trigger Event[2] for each block.

Chain alludes to a hash that associates one block to another. It's also the magic that glues blockchains together & allowed them to define mathematical trust. Network[3] is composed of "full nodes". A blockchain is a kind of data structure that makes it possible to create a digital distributed ledger of data & share it among a network of independent parties.[4] Blockchain is mainly classified into 3 major types i.e., Public: very large distributed networks running through a native cryptocurrency mostly Bitcoin, Ethereum. Fully Open Source. Permissioned:[5] Large distributed network; control roles for individuals within network. Private: Distributed Ledger Tech i.e., smaller & no token nor any cryptocurrency required.[6]

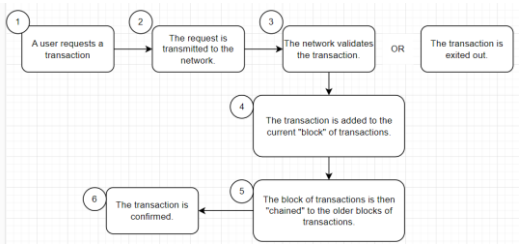

Figure 1: Process of Consensus Algorithm.

The main reason why blockchain becomes word of mouth so quickly is the *"Consensus"*. As mentioned in figure 1, the consensus blockchain creates honest systems where they self-correct themselves (i.e., nodes in a network) without any third-party monitoring.[7] This consensus algorithm is the process of creating an accord or concurrence among group of commonly distrustful shareholders.

Blockchains are also now being used in various industries such as Security: To counter code piracy, securing IOT devices from spoofing & hacking.[8] Government-Agencies:
Maintaining shatterproof Land-Record Systems. ICOs: Smart Contract that allows the issuer to grant token for Investment-Funds related to Banks.[9] The Existing System of general public election is running manually.[10] The Voter has to Visit to Booth to cast their votes. As a result, many people don’t go out to cast their vote which is one of the major drawbacks of current voting system. In democracy, every

citizen of a country must vote.[11] By a new online system which will limit the voting frauds and make the voting as well as counting more efficient and transparent.

II. System Model

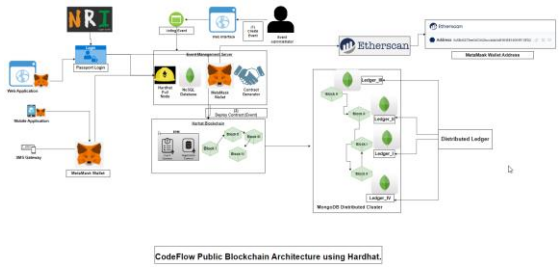

Figure 2: Showing entire Code flow using Hardhat

In Figure 2, we are explaining the complete architecture CodeFlow of our decentralized E-Voting application using Hardhat as blockchain platform.[12] Here, in Front-End Part we designed 3 ways i.e. User can login using web application, mobile application or even via SMS Gateway (for Tier-2 & Tier-3 cities). Also, NRIs can login with passport login, while others can login with MetaMask wallet.[13] After successful login, we can create voter's & candidates. As MetaMask is verified, voter registration can be accepted and also, we can see the voter list.[14] In create new voter, voter can upload photo, user name, address of MetaMask, age. By using hardhat blockchain - block I is connected to Block II and so on.[15] Hardhat blockchain is connected to MongoDB for NRI users and distributed ledger are created. Event Management Server is connected to Etherscan[16] block explorer and analytics platform for Ethereum. All of the process of authentication the Administrator can have all access to allow list of voters and have all the rights for the security reasons. We setup MongoDB Compass as distributed ledger cluster in which each node has its own ledger for maintaining the entire state of cluster as it works on Consensus algorithm.

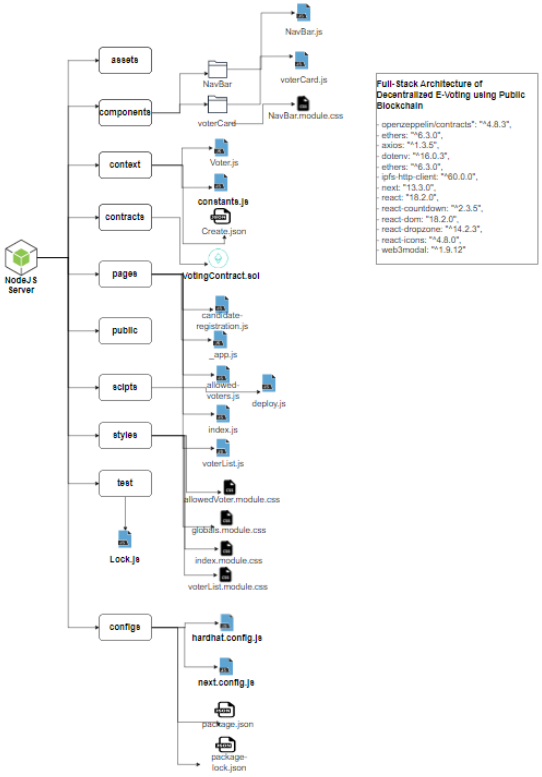

Figure 3: Showing entire directory structure of project

As shown in Figure 3, we explained our Full-Stack directory-level tree structure where we created directories i.e., assets, components, context, contracts, pages, scripts, styles, test & configs.[17] In components, we have Navbar, voter Card which includes NavBar.js, voterCard.js & NavBar.module.css etc. In context we created Voter.js, constants.js & Create.json etc. In pages directory, we created candidate_registration.js, _app.js, allowed-voters.js, voterList.js etc. In scripts we created deploy.js. Under styles directory, we created allowedVoter.module.css, globals.module.css, index.module.css, voterList.module.css. In test directory, we created Lock.js file. In configs file, we hardhat.config.js, next.config.js, package.json, package-lock.json. There are multiple dependencies used in this project such as openzeepelin, axios, ethers, hardhat, ipfs-http-client, dot-env, web3modal. [18]

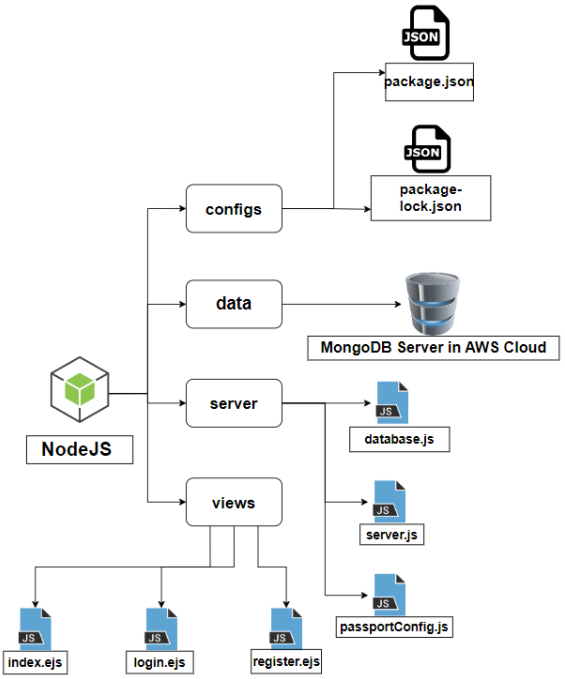

Figure 4: Showing entire directory structure of ***passport authentication system*** for NRIs login.

As shown in Figure 4, we explained our entire directory-level tree structure where we created directories i.e., configs, data, server & views. In server directory we created database.js, server.js & passportConfig.js. Inside *database.js* file we created the database schema which contains all the collection name & their datatype. We have three main fields i.e., *passport_no, name, email-id & password.* In views directory, we created index.ejs, login.ejs & register.ejs. The. ejs extension is *embedded javascript file.* In scripts we created deploy.js. In configs file, we created package.json, package-lock.json. There are multiple dependencies used in this project such as ejs, express, express-session, local, mongoose, nodemon, passport, passport-local etc.

III. Simulation Results

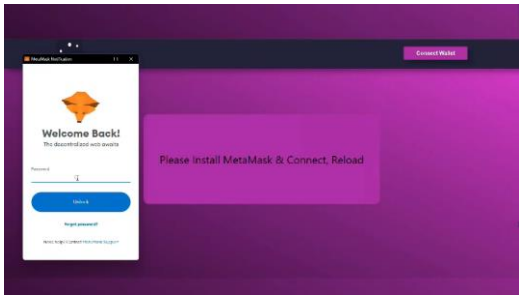

Figure 5: Picture depicting *Connection with MetaMask.*

As shown in figure 5, we are connecting with MetaMask Account by clicking on Connect Wallet. Then, MetaMask will connect it to Etherscan by calling API mentioned in context/constansts.js.

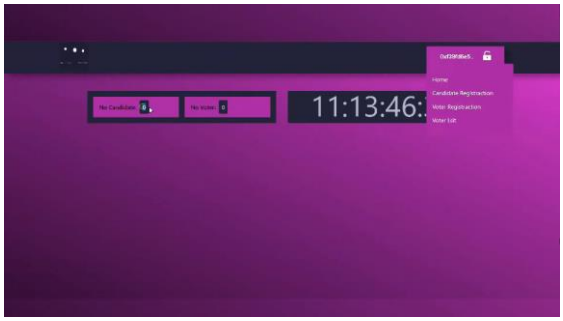

Figure 6: Selecting option of Candidate Registration

As displayed in figure 6, now we are using candidate-register.js file for registering our candidate. Also, in the figure it is showing No. of Candidate's & No. of Voter's in this page.

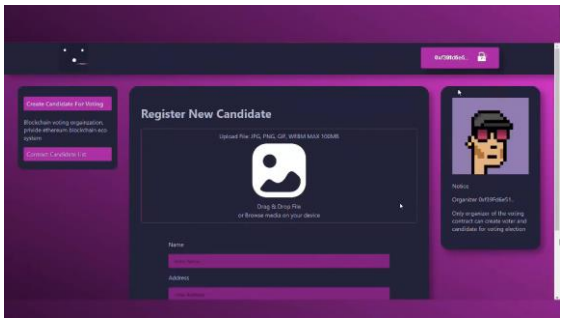

Figure 7: Register New Candidate by filling *Name, Address, Position etc.*

In figure 7, we have created a web form which takes name of candidate, Address from which location it belongs & for the post he/she is contesting for. Here, candidate can also upload his/her image.

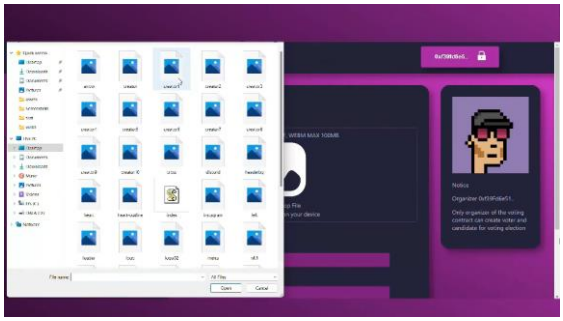

Figure 8: Select *'Candidate Photo'* for upload.

In figure 8, candidate is selecting his/her photo by using index.js file & uploading images present in assets directory. Also, these images are stored in local database.

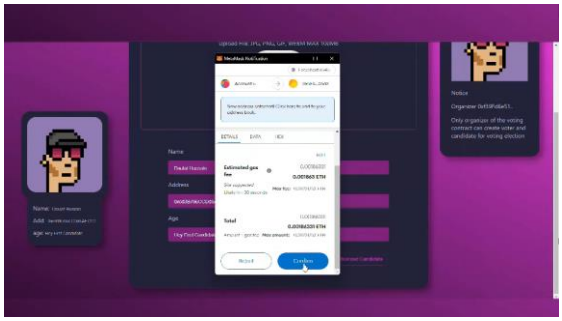

Figure 9: Confirming *Ethereum Gas* for candidate registration.

In figure 9, after filling all the form details such as name, address & position, candidate clicked on Authorized Candidate. After that, MetaMask wallet address with registered account gets initiated & Ethereum (ETH) gas fee gets deducted from MetaMask wallet.

Figure 10: Candidate is successfully registered.

In this figure 10, we successfully registered our candidate, also the status of Number of Candidates gets incremented by one.

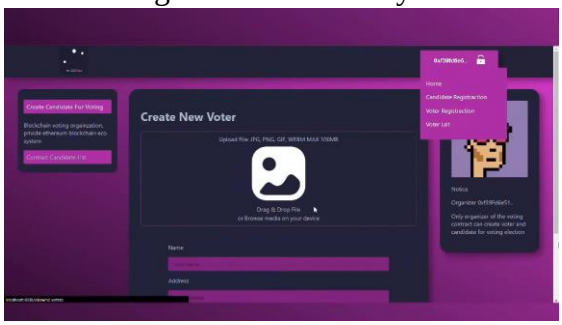

Figure 11: *Create New Voter* by filling *Name, Address, Age.*

In the figure 11, after filling all the form details such as name, address & age, voter clicks on Authorized Voter.

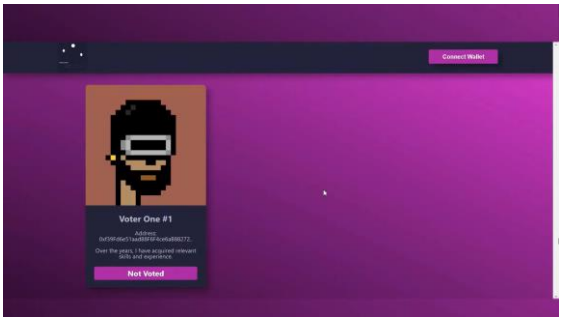

Figure 12: Page showing status of Voter_One#1 i.e. *Not Voted.*

In above figure 12, the Voter is created successfully, yet he/she hasn't voted i.e., displayed as Not Voted. This method of contract is coded in VotingContract.sol file under contracts directory.

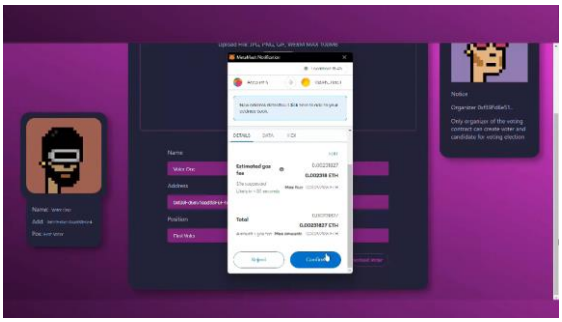

Figure 13: *Voter* successfully casted his vote.

As mentioned in figure 13, MetaMask wallet address with registered account gets initiated & Ethereum (ETH) gas fee gets deducted from MetaMask wallet as voter clicks on Confirm button. The vote gets casted. All these transaction histories can be seen in Etherscan platform.

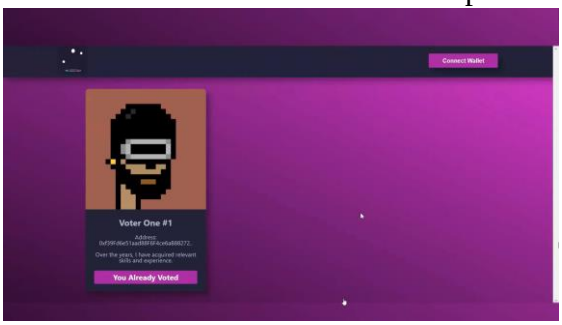

Figure 14: Page showing status of Voter_One#1's vote i.e. *You Already Casted.*

As shown in figure 14, the status of voter from Not Voted to You Already Voted gets

changed. The time taken by showing this status is very less.

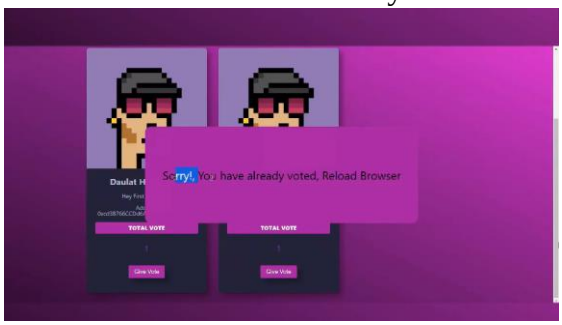

Figure 15: *You Already Voted* status so that *One Person, One Vote.*

As mentioned in figure 15, the solidity file i.e., VotingContract.sol inside contracts directory. It is hard-coded that one voter can only cast vote for single use as each voter has unique wallet address associated with their private key.

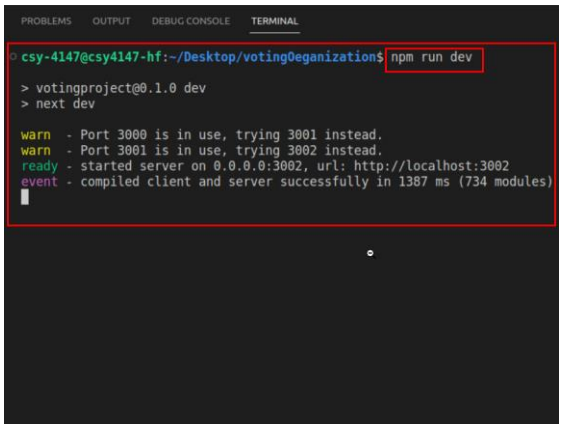

Figure 16: ***Nodejs*** server successfully hosted at localhost:3001

In this figure 16, we are starting our NodeJS server with the use of Node Package Manager(NPM) by providing run as option. Also, our server gets started on https://localhost:3001 as 3001 port number.

Figure 17: Basic passport authentication design containing web pages i.e., ***Register*** & ***Login.***

The figure 17 shows the passport authentication for all the NRIs. Here, we created a basic design to provide an idea how we can cast NRIs vote using their Passport's. This page is linked to MongoDB Compass database in backend.

Figure 18: User Registration is successfully

As mentioned in figure 18, the voter is registering so that he/she can cast the vote but the credentials used are passport related.

Figure 19: Registered User successfully ***login***.

Previously the user/voter gets registered, now in figure 19 the user/voter can login via same credentials & then redirected to

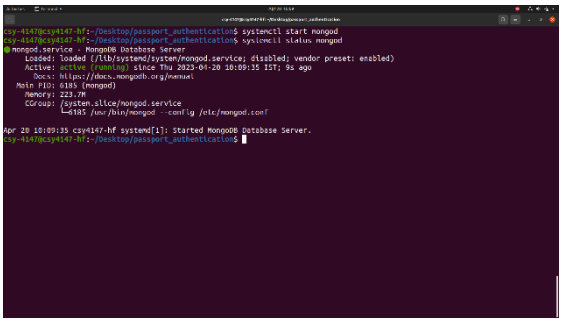

Figure 20: Starting ***mongod*** daemon for deploying database server

As displayed in figure 20, MongoDB Community Server is started in Ubuntu 20.04 using systemctl utility & connected with Passport Authentication Design Module.

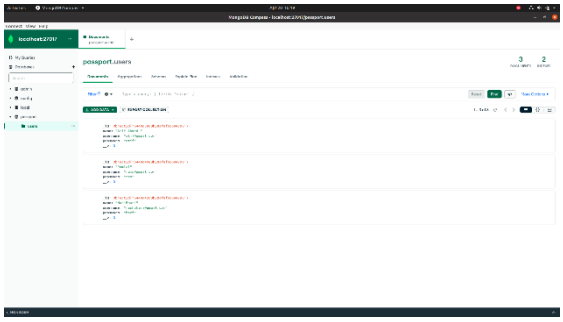

Figure 21: Registered user's data saved in ***Mongodb Compass*** locally.

In this figure 21, all the Registered user's data i.e., name, email & password is stored in MongoDB Database Server where the database name is *passport* & collection name is *users*. All these details are in database.js file.

IV. Conclusion

Blockchain tech can be utilised in many sectors & services to reduce the initial cost of investment also improves the performance in that particular area. By using BCT individual's vote privacy can be kept secret. Voters can give their vote at their ease of space or according to their comfort zone. Voting system can be helpful for giving vote in the elections happened in the colleges etc. Block chain technology reduces the errors at the time of vote counting. Online voting system will able to take care the voter's information where voter can have access and use their voting rights. Our research paper concludes that usefulness or ease of use are still important to decision makers while implementing EVs, but our research shows that building trust faith is prime.

V. References

[1] Ben Adida and Ronald L Rivest. Scratch & vote: self-contained paper-based cryptographic voting. In Proceedings of the 5th ACM workshop on Privacy in electronic society, pages 29–40, 2006.

[2] Ali Mansour Al-madani and Ashok T Gaikwad. Iot data security via blockchain technology and service-centric networking. In 2020 International Conference on Inventive Computation Technologies (ICICT), pages 17–21. IEEE, 2020.

[3] Ali Mansour Al-Madani, Ashok T Gaikwad, Vivek Mahale, and Zeyad AT Ahmed. Decentralized e-voting system based on smart contract by using blockchain technology. In 2020 International Conferene on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), pages 176–180. IEEE, 2020.

[4] Ahmed Ben Ayed. A conceptual secure blockchain-based electronic voting system. International Journal of Network Security & Its Applications, 9(3):01–09, 2017.

[5] Susan Bell, Josh Benaloh, Michael D. Byrne, Dana Debeauvoir, Bryce Eakin, Philip Kortum, Neal McBurnett, Olivier Pereira, Philip B. Stark, Dan S. Wallach, Gail Fisher, Julian Montoya, Michelle Parker, and Michael Winn. STAR-Vote: A secure, transparent, auditable, and reliable voting system. In 2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 13), Washington, D.C., August 2013. USENIX Association.

[6] Jens-Matthias Bohli, J¨orn M¨uller-Quade, and Stefan R¨ohrich. Bingo voting: Secure and coercionfree voting using a trusted random number generator. In E-Voting and Identity: First International Conference, VOTE-ID 2007, Bochum, Germany, October 4-5, 2007, Revised Selected Papers 1, pages 111–124. Springer, 2007.

[7] Umut Can C¸abuk, Eylul Adiguzel, and Enis Karaarslan. A survey on feasibility and suitability of blockchain techniques for the e-voting systems. arXiv preprint arXiv:2002.07175, 2020.

[8] R Aroul Canessane, N Srinivasan, Abinash Beuria, Ashwini Singh, and B Muthu Kumar. Decentralised applications using ethereum blockchain. In 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), volume 1, pages 75–79. IEEE, 2019.

[9] David Chaum. Secret-ballot receipts: True voter-verifiable elections. IEEE security & privacy, 2(1):38–47, 2004.

[10] David Chaum, Aleks Essex, Richard Carback, Jeremy Clark, Stefan Popoveniuc, Alan Sherman, and Poorvi Vora. Scantegrity:

End-to-end voter-verifiable optical-scan voting. IEEE Security & Privacy, 6(3):40–46, 2008.
[11] David Chaum, Peter YA Ryan, and Steve Schneider. A practical voter-verifiable election scheme. In Computer Security–ESORICS 2005: 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005. Proceedings 10, pages 118–139. Springer, 2005.
[12] David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 24(2):84–90, 1981.
[13] K Dhinakaran, PM Britto Hrudaya Raj, and D Vinod. A secure electronic voting system using blockchain technology. In Proceedings of the Second International Conference on Information Management and Machine Intelligence: ICIMMI 2020, pages 307–313. Springer, 2021.
[14] M Erdenebileg. e-voting anwendung auf ethereum plattform als smart contract. Fachhochschule Campus Wien, 2019.
[15] Rifa Hanifatunnisa and Budi Rahardjo. Blockchain based e-voting recording system design. In 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), pages 1–6. IEEE, 2017.
[16] Friðrik Þ. Hj´almarsson, Gunnlaugur K. Hreiðarsson, Mohammad Hamdaqa, and G´ısli Hj´almt´ysson. Blockchain-based e-voting system. In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), pages 983–986, 2018.
[17] Dalia Khader, Ben Smyth, Peter Ryan, and Feng Hao. A fair and robust voting system by broadcast. Lecture Notes in Informatics, pages 285–299, 2012.