VoteChain

Ishwarya Varadarajan
Sowmya Viswanathan

Mridula Krishnamurthy
Vishal Praanesh Kole Purushotham

CMPE 272 – Enterprise Software Platforms
San Jose State University, California

***Abstract*— Voting is a fundamental right of every citizen of a democratic country. This is done to choose their government, who work for the betterment of the country. But we come across many instances of frauds happening within the system, from ballot stuffing and voter impersonation, to illegal activities of the electoral officers. Also, there are various ways of casting a vote, be it EVMs or paper ballots, and securing all these methods is challenging. In this paper, we propose a uniform online voting system called VoteChain, using blockchain technology as the security provider. Using this system, the voting process can be secured against vote tampering, as we record all the transactions made.**

I. INTRODUCTION

Blockchain is one of the latest technologies growing exponentially in the digital world. A formal definition of blockchain would be that it is a “distributed database that maintains a continuously growing list of records, called blocks” [1]. Each block contains a timestamp and a link to a previous block, thus maintaining a “chain”. These blocks are nothing but data entered into the system at that time. Blockchains are essentially databases with some inbuilt pre-agreed technical and business logic criteria, kept in sync via peer-to-peer mechanisms and pre-agreed rules about what new data can be added [2]. The digital ledger is maintained on a number of systems, which reflect all the changes and additions done uniformly across the network, instead of maintaining it on a single central system.

Blockchain can be of 3 types: Public, Private or Consortium. Anyone can read and write on a public blockchain, whereas users need permission to read and write on a private blockchain. A consortium blockchain is a blockchain where the consensus process is controlled by a pre-selected set of nodes [3]. In our project, we are making use of the public blockchain as the general public are required to cast their vote using the system. Also, there are various applications of the blockchain. The most popular application is the bitcoin, which is the digital currency exchanged across the network. Some other applications include Zcash, Ethereum, Ripple, and many more. The application used in this project is the Hyperledger fabric, which is an “open-source, modular, multi-channel transaction network” [4] which provides confidentiality and scalability, and enables only permissioned users to access the system.

One of the most important features of blockchain is security for the data entered. We achieve this in blockchain by making the data (votes in our project) relatively immutable [2]. Immutable means that cannot be changed. We achieve this immutability using hashes. A hash function is a type of mathematical function or an algorithm which turns data into a fingerprint of that data called a hash. That is, data is changed into a string of characters when passed through the algorithm. It will be difficult to revert back to the original data from the hash, as well as the hash will change significantly for a change of even a single bit in the original data. This greatly helps us in securing the data from tampering, as it will be evident if anything is changed. Also, timestamps are added to each block of data, helping in maintaining a ledger of all the data added along with the time of addition.

II. ARCHITECTURE OF BLOCKCHAIN

The basic architecture used to build this application is as shown below in Fig 1.

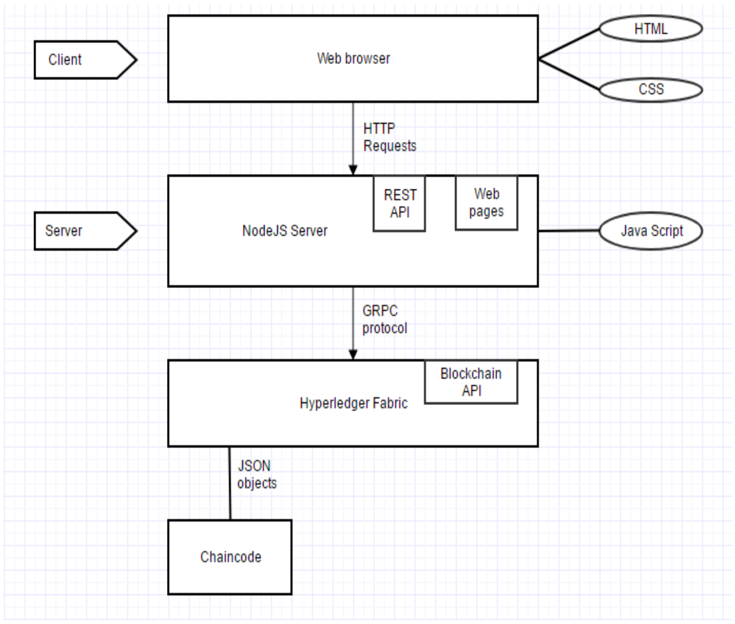


Fig 1. Blockchain architecture

The client in the application is the web browser, which provides multiple web pages to the users based on their functionality. On the client side, we make use of HTML and CSS. The HTTP calls made from the client are forwarded to the server, which in this case is the NodeJS server. We make use of the NodeJS server as it supports JavaScript, which produces dynamic web pages, and is lightweight. JavaScript is the scripting language on the server side, and we have REST APIs here. The server in-turn accesses the Hyperledger fabric using GRPC protocol, which has the Blockchain APIs. The calls are parsed and the input data is converted to JSON objects to access the chaincode.

III. VOTECHAIN

The application we are designing is VoteChain, which is an online voting system which can be accessed remotely from workstations across the network. The application is hosted on the Amazon Web Services (AWS) to enable remote access. The system is provided security against frauds using the blockchain technology. The required functionalities of the application can be divided as functional and non-functional requirements, and are as follows.

A. Functional Requirements:

- Interactive UI to provide easy access
- Provision every user in the voting process with a web page with related functionality/actions
- Provide each voter with a single vote template to cast their votes
- Track the transfer of votes to detect fraud

B. NON-FUNCTIONAL REQUIREMENTS:

- High performance measurement
- High availability
- Scalability to modify the load taken by the application based on the traffic
- Portability to systems of any architecture
- Fault tolerance so that the system is stable even in case where some functionalities fail
- Maintainability so that the system can be used for a longer period

C. VOTECHAIN ARCHITECTURE

- A Voter Template is created by the Polling Officer for every voter who comes into the Polling Station to vote.
- The Voter Template is then transferred by the Polling Officer to the Polling Station.
- Voter casts his vote in the Polling Station by updating the Voter Template with his details.
- Polling Station Officer transfers the updated Voter Template/Vote to the Counting Station where the counting of the votes take place.
- All the transactions, starting from providing the template to the voter, until the votes transferred to the counting station, are recorded in the digital ledger.

The architecture is as shown in Fig 2.

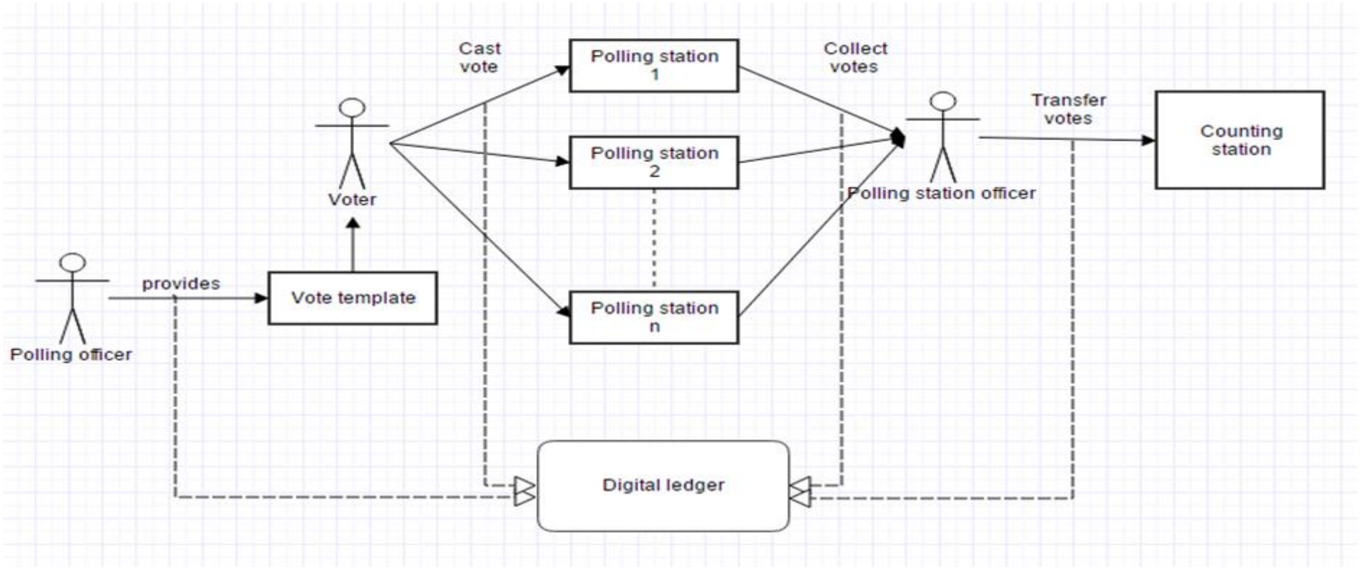


Fig 2. VoteChain architecture

There are 3 main users of the application, whose functions are as shown in Fig 3.

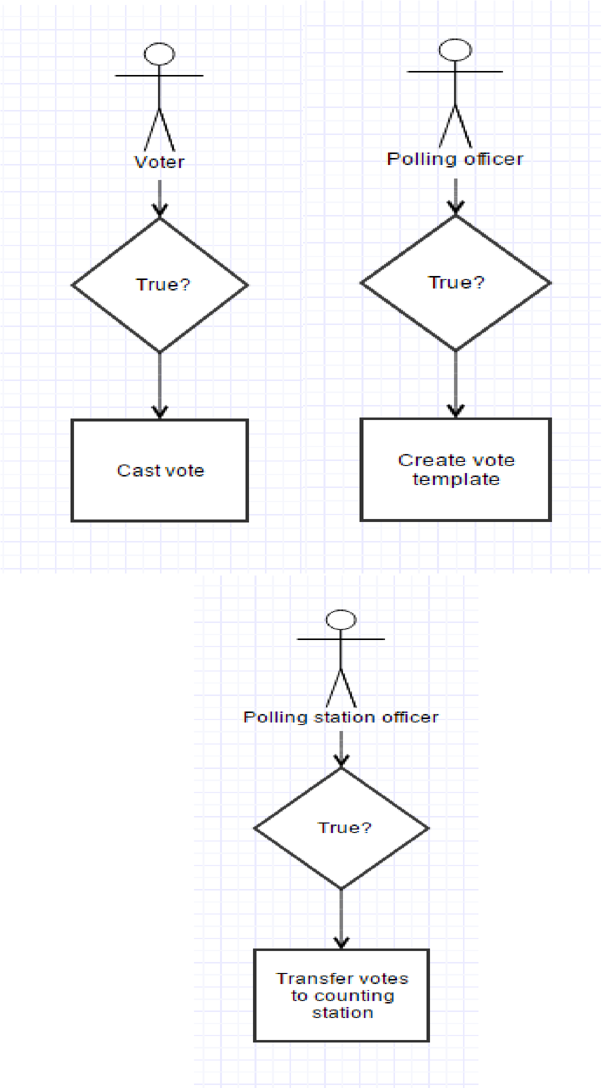


Fig 3. Use cases

IV. VOTECHAIN IMPLEMENTATION

The implementation of VoteChain is as explained here [5].

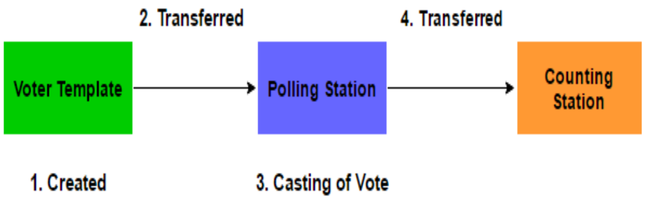


Fig 4. VoteChain implementation

A. OPERATIONS AND FEATURES

i. HOME SCREEN

This page contains the link for various role based functionalities of the application as listed below.

Create and View

- Create Vote Template

Transfer

- I’m a Polling Officer
- I’m a Polling Station Officer

Polling

- I’m a Voter

ii. CREATE VOTE TEMPLATE

- The polling officer creates a uniform voter template for the voter to cast his vote, using the “Create Voter Template”
- Each voter template is tagged to a unique, randomly auto-generated ‘Vote ID’ and polling officer receives a confirmation once the template is successfully created.
- A block will be created and added to the chain of transactions every time when a voter template is created.

The block diagram for creating a vote template is shown in Fig 5.

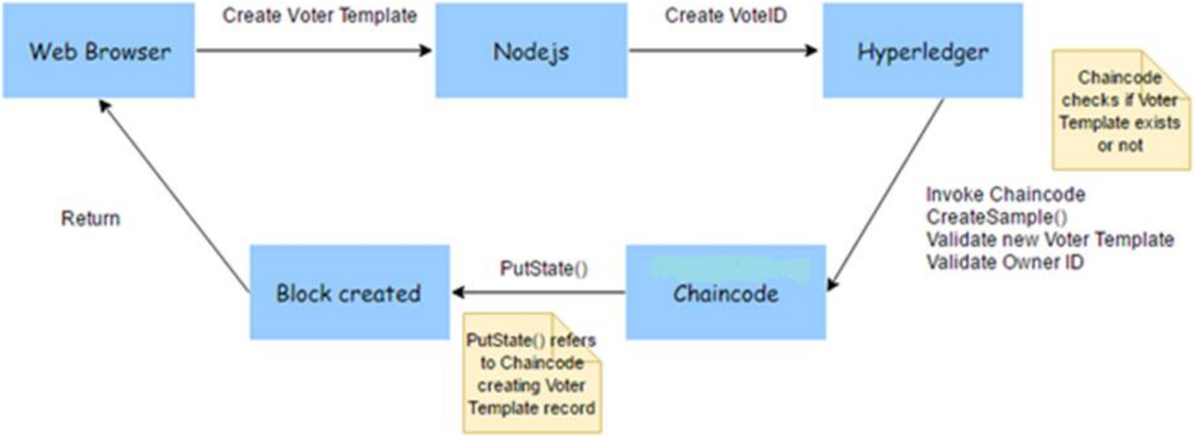


Fig 5. Create vote template

iii. BLOCKCHAIN LEDGER

- This page displays the transactions from the Blockchain that are applicable to the user. It shows a list of every transaction including creation of the Voter Template, update of the template and transfer of the template.
- Every Polling Station is listed as part of the transactions. An admin user can trace a vote right from a Voter Template being created, to update of the voter template and finally to the transfer of the vote. This reduces fraudulent activities and reduces human error.

iv. UPDATE VOTER TEMPLATE

- "Polling Station-Update Asset" is the page where a Voter casts his vote by updating the fields in the Voter Template.

The block diagram for updating the voter template is as shown in Fig 6.

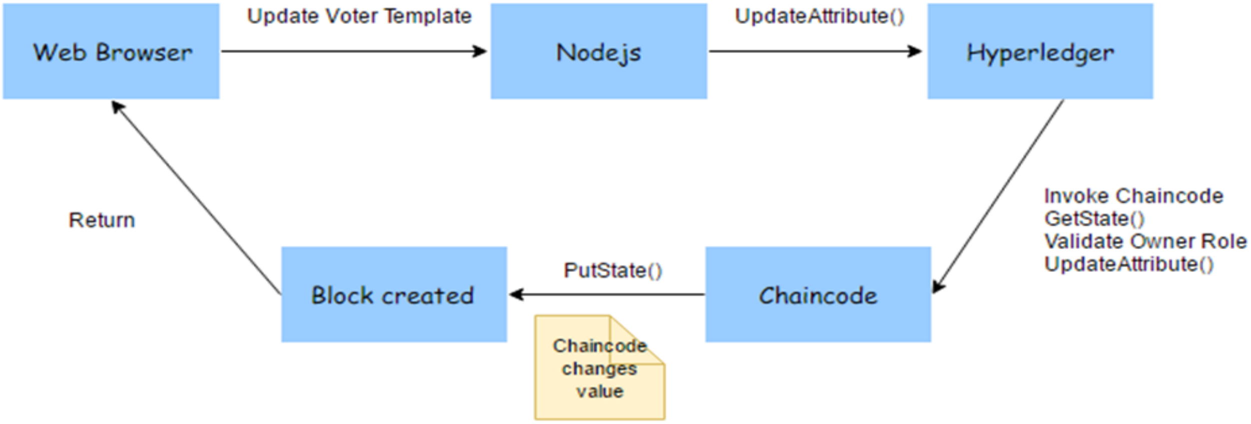


Fig 6. Update vote template

v. TRANSFERRING A VOTE

- “Polling Office-Transfer Asset” is the page where Voter Template is transferred to the Polling Station by the Polling Officer.
- “Polling Station Update-Transfer Asset” is the page where updated Voter template i.e. Vote, is transferred by the Polling Station Officer to the Counting Station.

It is shown in Fig 7.

vi. BLOCKCHAIN STATISTICS

This page contains the details of every transaction that is performed during the functioning of the Votechain application. There are two key elements of the statistics page.

Live Stats:

- Last Block - The sequence number of the last block (that is the latest transaction performed)
- Time since the last block was created
- Number of changes to the last block

Blockchain Explorer:

- A chain of numbered blocks is created for every transaction performed during the operation of the application
- Each block contains its corresponding transaction encrypted into a hash value, the hash value of the previous transaction (to create a link to the chain) and the date & time of each transaction

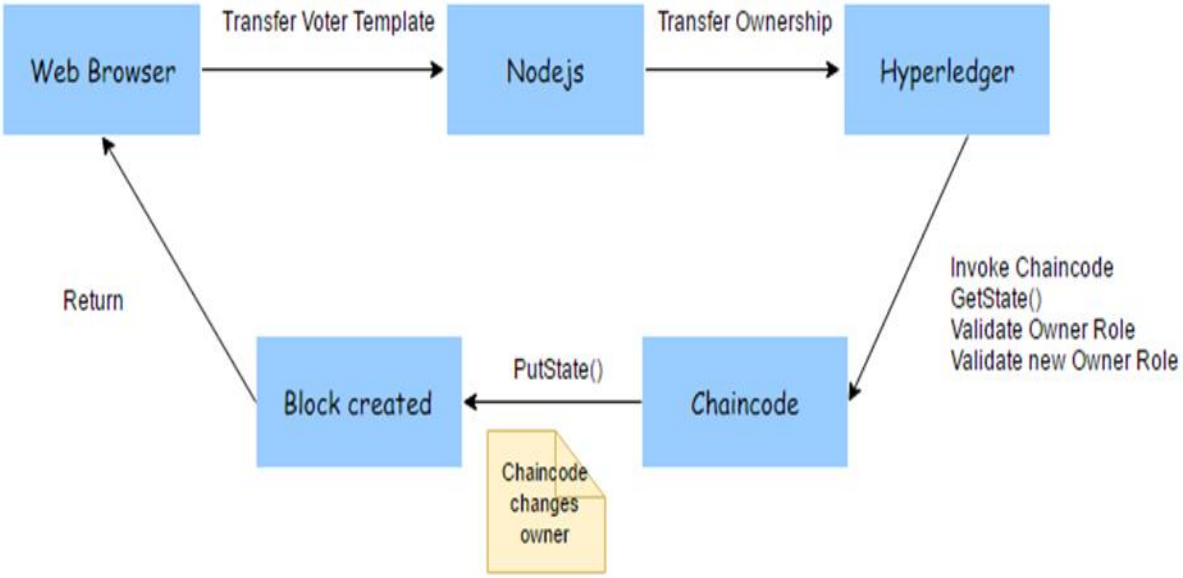


Fig 7. Transferring a vote

V. CONCLUSION

Using the VoteChain application, we can provide the voters with a uniform online voting system, which is made secure using the blockchain technology. The transactions of the entire process is recorded in a digital ledger, enabling transparency of the system. The voters can cast their vote and the transaction is tracked until it is counted in the polling station, so that the users can ensure the votes are not tampered.

REFERENCES

[1] Blockchain Wikipedia (2017, May 14). Blockchain [Online]. Available: https://en.wikipedia.org/wiki/Blockchain

[2] Bits on blocks (2017, May 14). A gentle introduction to immutability of blockchains [Online]. Available: https://bitsonblocks.net/2016/02/29/a-gentle-introduction-to-immutability-of-blockchains/

[3] Types of Blockchains - BlockchainHub (2017, April 25). BlockchainHub [Online]. Available: https://blockchainhub.net/blockchains-in-general/

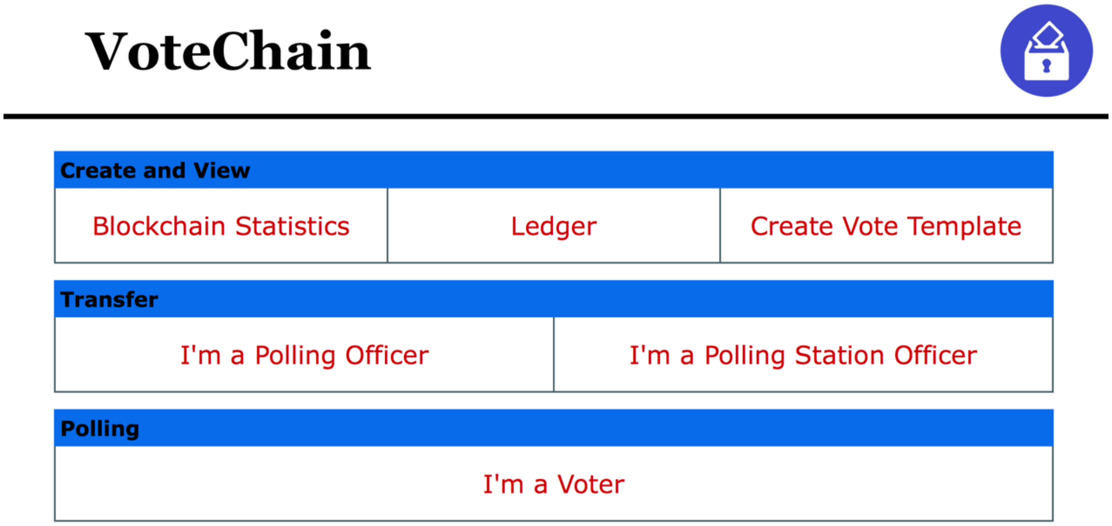
[4] IBM Blockchain (2017). IBM Blockchain - The Hyperledger Project [Online]. Available: https://www.ibm.com/blockchain/hyperledger.html

[5] IBM (2016). IBM Blockchain / car-lease-demo [Online]. Available: https://github.com/IBM-Blockchain/car-lease-demo/blob/master/Documentation/Installation Guide.md - deploying-locally

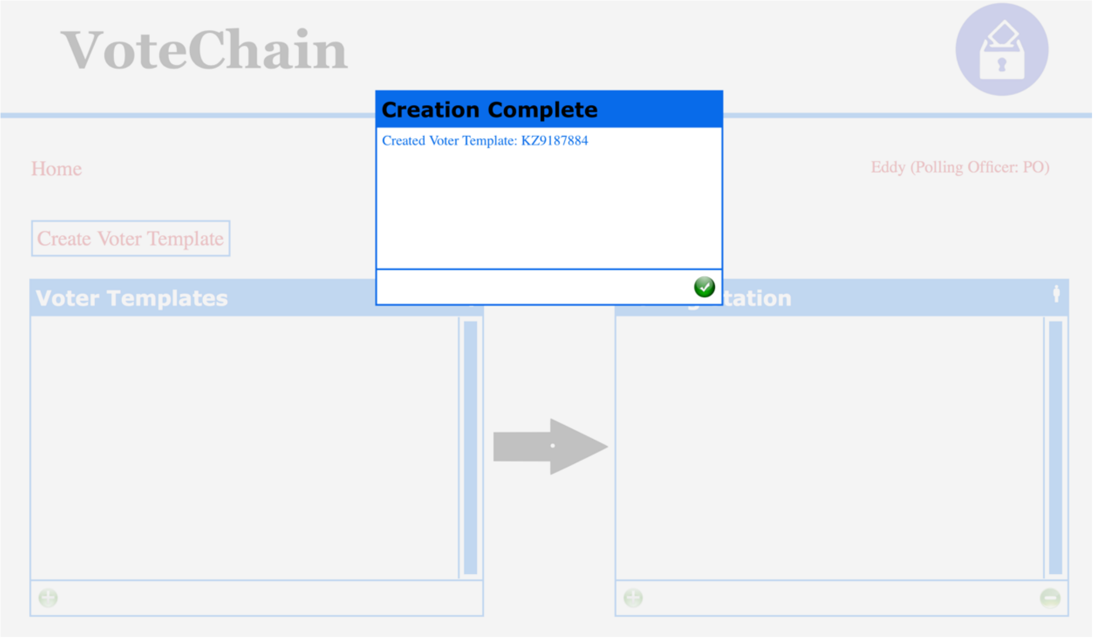
APPENDIX

Snapshots of the VoteChain application

A. Home Screen

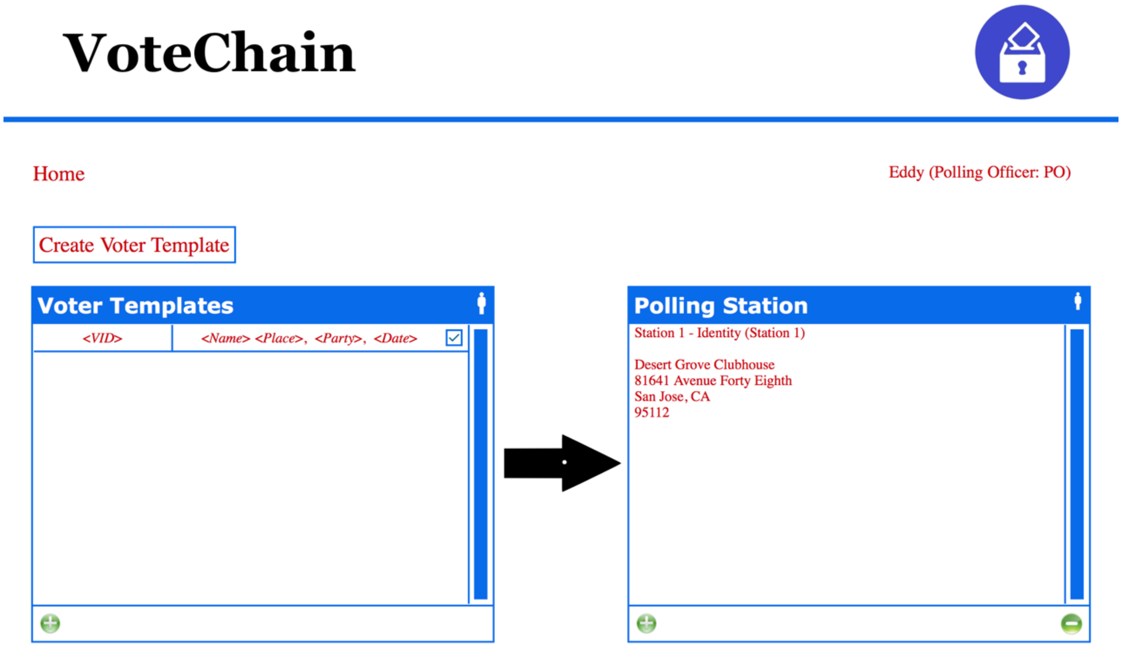


B. Create Vote Template


Creation Complete

Created Voter Template: KZ9187884

C. I’m a Polling Officer – Transferring the vote template to the polling station



Eddy (Polling Officer: PO)

Voter Templates

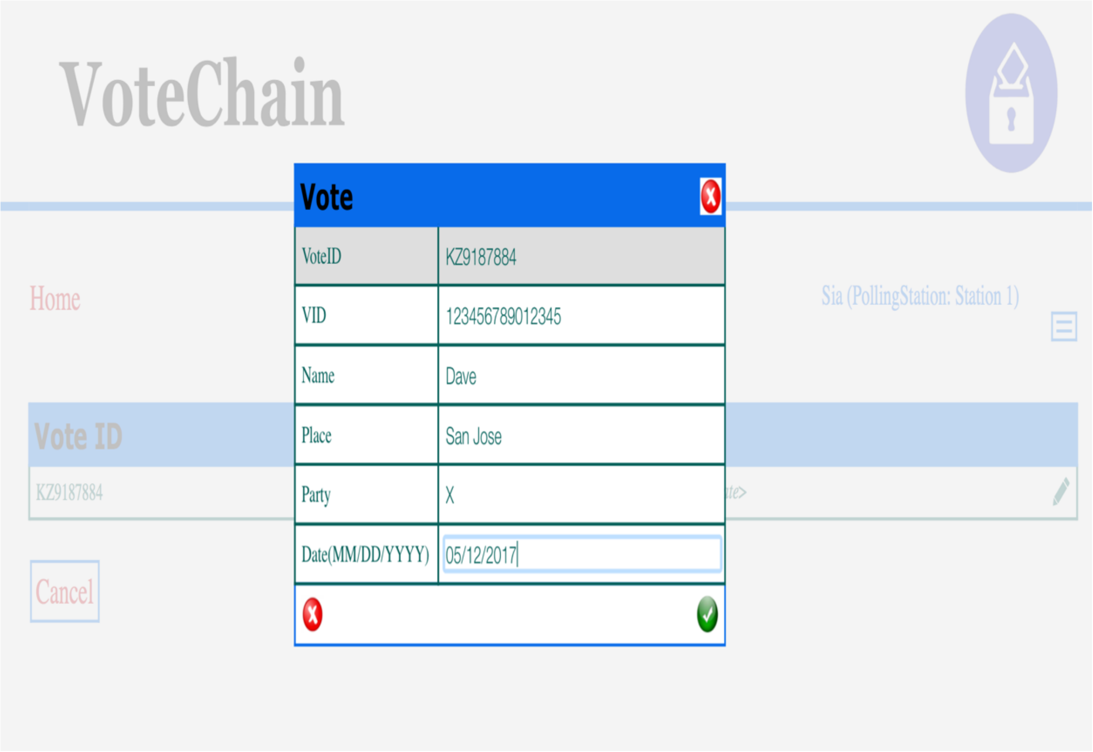
<VID>	<Name> <Place>, <Party>, <Date> ☑

Polling Station

Station 1 - Identity (Station 1)

Desert Grove Clubhouse
81641 Avenue Forty Eighth
San Jose, CA
95112

D. I’m a Voter – Voter casts his/her vote as below



Sia (PollingStation: Station 1)

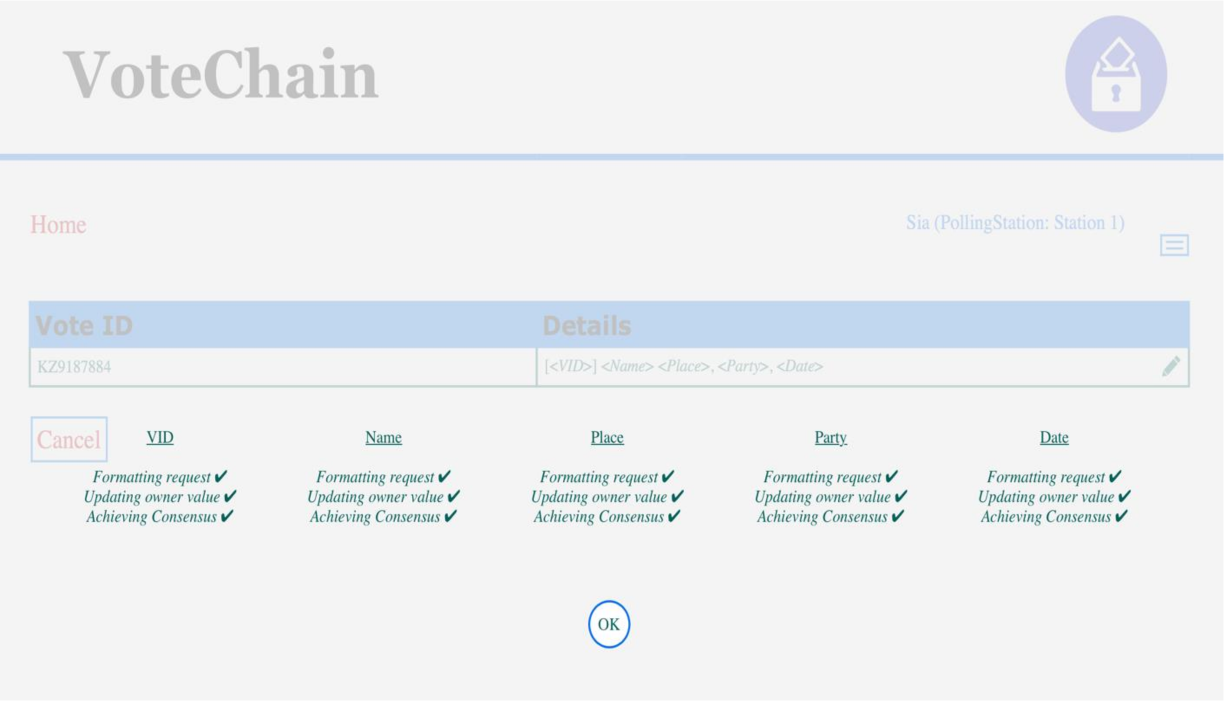
Vote ID

KZ9187884 ...te>

Vote

VoteID	KZ9187884
VID	123456789012345
Name	Dave
Place	San Jose
Party	X
Date(MM/DD/YYYY)	05/12/2017

E. Values entered by the voter, as above, are updated to the blockchain



VoteChain

Home

Sia (PollingStation: Station 1)

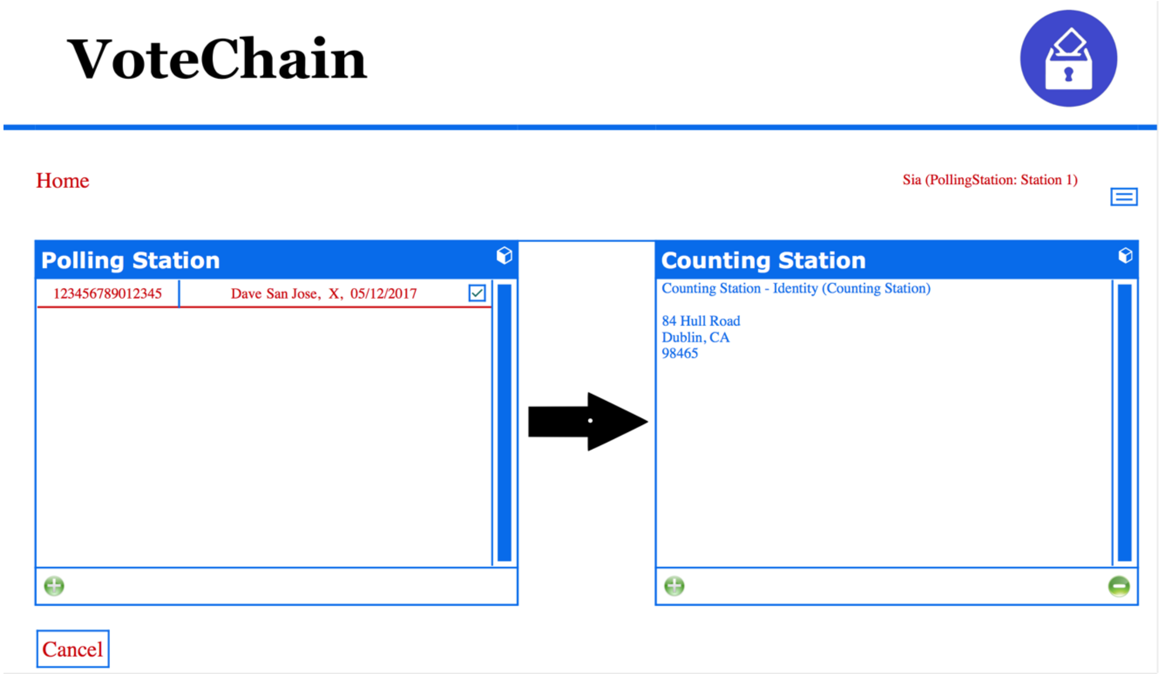
Vote ID	Details
KZ9187884	[<VID>] <Name> <Place>, <Party>, <Date>

Cancel

VID	Name	Place	Party	Date
Formatting request ✔	*Formatting request* ✔	*Formatting request* ✔	*Formatting request* ✔	*Formatting request* ✔
Updating owner value ✔	*Updating owner value* ✔	*Updating owner value* ✔	*Updating owner value* ✔	*Updating owner value* ✔
Achieving Consensus ✔	*Achieving Consensus* ✔	*Achieving Consensus* ✔	*Achieving Consensus* ✔	*Achieving Consensus* ✔

OK

F. I'm a Polling Station Officer – Transfers the vote template updated by the voter from the polling station to the counting station



VoteChain

Home

Sia (PollingStation: Station 1)

Polling Station

123456789012345	Dave San Jose, X, 05/12/2017 ☑

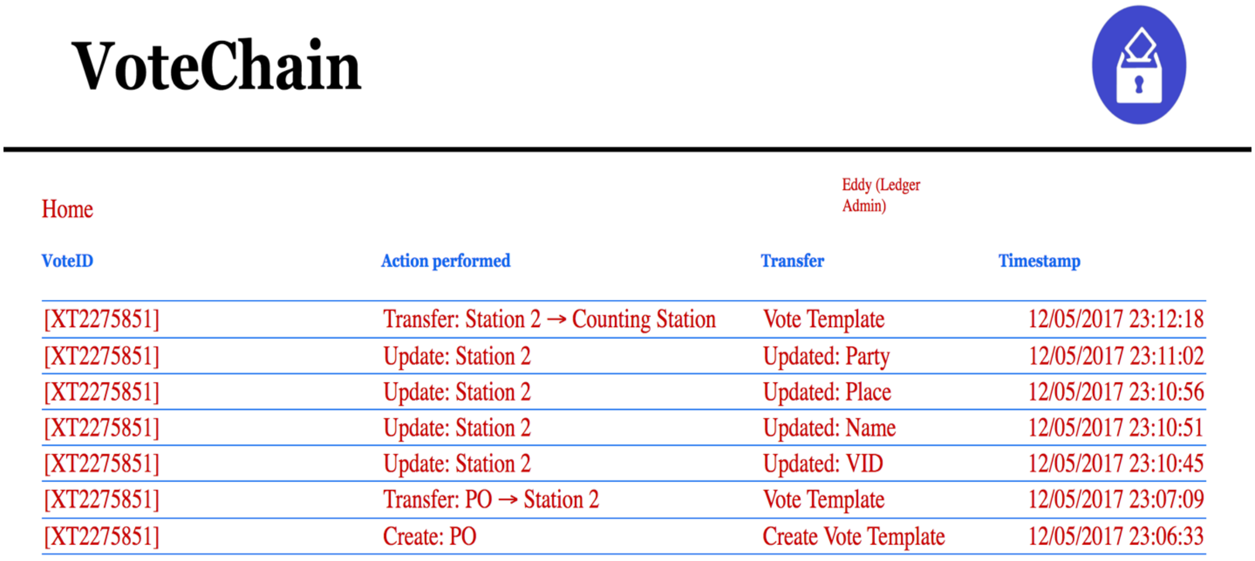
Counting Station

Counting Station - Identity (Counting Station)

84 Hull Road
Dublin, CA
98465

Cancel

G. Blockchain Ledger



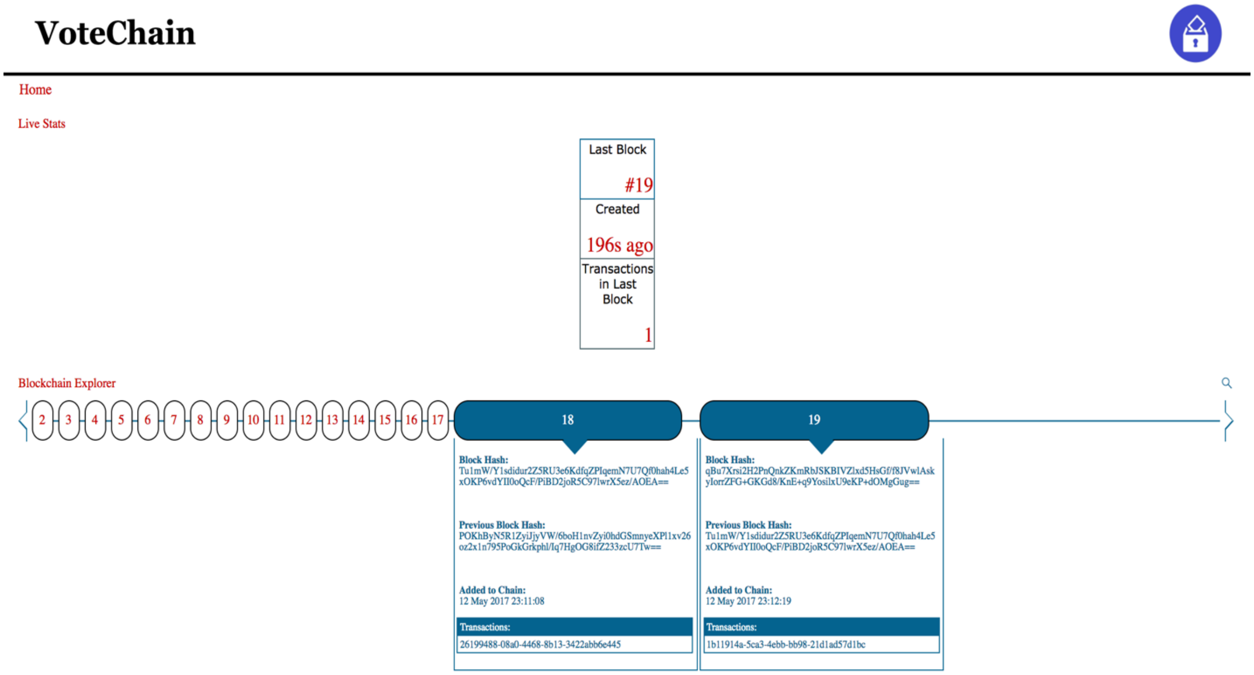
VoteChain

Home

Eddy (Ledger Admin)

VoteID	Action performed	Transfer	Timestamp
[XT2275851]	Transfer: Station 2 → Counting Station	Vote Template	12/05/2017 23:12:18
[XT2275851]	Update: Station 2	Updated: Party	12/05/2017 23:11:02
[XT2275851]	Update: Station 2	Updated: Place	12/05/2017 23:10:56
[XT2275851]	Update: Station 2	Updated: Name	12/05/2017 23:10:51
[XT2275851]	Update: Station 2	Updated: VID	12/05/2017 23:10:45
[XT2275851]	Transfer: PO → Station 2	Vote Template	12/05/2017 23:07:09
[XT2275851]	Create: PO	Create Vote Template	12/05/2017 23:06:33

H. Blockchain Statistics



VoteChain

Home

Live Stats

Last Block #19
Created 196s ago
Transactions in Last Block 1

Blockchain Explorer

Block Hash:
Tu1mW/Y1sdidur2Z5RU3e6KdfqZPIqemN7U7Qf0hah4Le5xOKP6vdYII0oQcF/PiBD2joR5C97lwrX5ez/AOEA==

Previous Block Hash:
POKhByN5R1ZyiJjyVW/6boH1nvZyi0hdGSmnyeXPl1xv26oz2x1n795PoGkGrkphl/Iq7HgOG8ifZ233zcU7Tw==

Added to Chain:
12 May 2017 23:11:08

Transactions:
26199488-08a0-4468-8b13-3422abb6e445

Block Hash:
qBu7Xrsi2H2PnQnkZKmRbJSKBIVZlxd5HsGf/f8JVwlAskyIorrZFG+GKGd8/KnE+q9YosilxU9eKP+dOMgGug==

Previous Block Hash:
Tu1mW/Y1sdidur2Z5RU3e6KdfqZPIqemN7U7Qf0hah4Le5xOKP6vdYII0oQcF/PiBD2joR5C97lwrX5ez/AOEA==

Added to Chain:
12 May 2017 23:12:19

Transactions:
1b11914a-5ca3-4ebb-bb98-21d1ad57d1bc