GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

ISSUES AND FUTURE ENHANCEMENT FOR BLOCKCHAIN BASED E-VOTING SYSTEM

A Project Report of Capstone Project - 2

Submitted by

ANUBHAV RAJ SINGH
(1613105018)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING WITH SPECIALIZATION OF CLOUD COMPUTING AND VIRTUALIZATION

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

MR. SURENDRA KUMAR, M. Tech
Assistance Professor

APRIL / MAY, 2020

GALGOTIAS
UNIVERSITY

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **“ISSUES AND FUTURE ENHACNEMENT FOR BLOCKCHAIN BASED E-VOTING SYSTEM”** is the bonafide work of **“ANUBHAV RAJ SINGH (1613105018)”** who carried out the project work under my supervision.

SIGNATURE OF HEAD
Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean,
School of Computing Science & Engineering

SIGNATURE OF SUPERVISOR
MR. SURENDRA KUMAR,
M. Tech
Assistant Professor,
School of Computing Science & Engineering

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	5
	LIST OF FIGURES	6
	LIST OF SYMBOLS	7
1.	**INTRODUCTION**	8
	1.1 The Election Process	8
	1.1.1 Voting Process	8
	1.1.2 Introduction of EVMs	9
	1.1.3 Legal Battle	10
	1.1.4 Looming threats of Transparency	12
	1.1.5 International Comparisons	13
	1.2 Blockchain	14
2.	**EXISTING SYSTEM**	17
	2.1 Traditional E-Voting System	17

3.	**PROPOSED SYSTEM**	20
	3.1 Proposed Algorithm	20
	3.1.1 Sharding	20
	3.2 ZK-SNARKS	23
	3.2.1 ZK-SNARK Transaction	24
	3.2.2 A ZK-SNARK Example	25
4.	**IMPLEMENTATION**	28
	4.1 Process	28
	4.2 Sample Code	29
5.	**RESULTS / SCREENSHOT**	
6.	**CONCLUSION**	
7.	**REFERENCES**	

ABSTRACT

Blockchain, the decentralized and immutable modern era technology which is contently being developed and providing solutions to some of the most important fields of technology. Since its outbreak, the blockchain was famous for its valuable contribution to the transaction field but after introducing the concept of smart contracts in Ethereum blockchain the popularity and use of the blockchain have surged. Smart contracts enable us to carry out agreement and transaction trustworthily between two anonymous peers with any central governing authority, because of this feature blockchain can be used to develop a decentralized E-Voting application which can escalate the decreasing level of trust in some of the finest democracies of the world. Replacing the old EVMs with a decentralized blockchain application will also make voting an online event. This paper highlights how the implement the proposed model and what are the issues that can erupt and their possible solution for a future robust system.

LIST OF FIGURES

Figure 1: Sharding Based Consensus Protocol

Figure 2: ZK-SNARKS Transaction

Figure 3: Screenshot of model implemented with issues

LIST OF SYMBOLS and ABBRIVATIONS

Sr. Number	ABBRIVATION	FULL FORM
1.	ECI	Election Commission of India
2.	EVMs	Electronic Voting Machines
3.	VVPAT	Voter Verifiable Paper Audit Trail
4.	PoS	Proof of Stake
5.	DDoS	distributed denial-of-service
6.	zk-snarks	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge
7.	pk	proving key
8.	vk	verification key
9.	prf	proof
10.	P	prover

1. INTRODUCTION

1.1 The Election Process

Ever since gaining Independence from the British colonial rule, India has established itself as a democratic republic nation. Being tagged as the largest democracy in the world with almost 850 million registered voters who use their adult franchise in every election, this festival of democracy i.e; Election is a much talked about and celebrated occasion in the country.

One of the most important aspects of democratic set up in a country with federal structure is having elections at regular intervals. Therefore holding and conducting the elections in a free and fair way is quintessential in any democracy and in a country like India it goes deep in to the root of basic structure doctrine of the Constitution of India. The task of successful conducting of these elections which are voluminous in nature involving a protracted and cumbersome process has been assigned to the Election Commission of India ('ECI'). Involving several lakhs of workforce, series of training sessions to the professionals, dispensation of mass awareness to increase the overall footfall in the elections, accessing even the remotest location, and finally making all this happen as a smooth sail with the result announcement, the ECI, a constitutional entity totally pulls off this daunting errand with a smile. The ECI has been conferred with the responsibility of direction, superintendence and control on each and every Election in India and is successfully conducting various Parliamentary and State Legislative elections from last sixty eight years.

1.1.1 Voting Process:

Voting Process in India has undergone a substantial change over the years i.e; from the use of Ballot Papers to Electronic Voting Machines (EVMs) which is fur- ther followed by EVMs with Voter Verifiable Paper Audit Trail (VVPAT). It is pertinent to mention that in the first two Parliamentary Elections held in the year 1952 and 1957, interestingly the names of the candidate and the symbol were not printed on the ballot papers rather each candidate was allotted a separate ballot box pasted with the name and symbol of the candidate, and the people had to drop a pre-printed ballot paper in the ballot box of the candidate of their choice. However this method of voting had a short run as it emanated fear of manipulations, booth capturing and different forms of tampering with the ballot boxes.

From the year 1960 till 1999, the very famous ballot papers were used in all Elec- tions ranging from State level to General Elections wherein the voter used to stamp against the candidates' symbol of his choice in a pre-printed ballot paper having the names and symbols of all the candidates contesting. Usage of these ballot papers for almost four decades is evident that they had considerable suc- cess in conducting the elections. But with the exponential growth of the regis- tered voters over the period of time spiced up with varied issues inter-alia large scale printing of ballot papers, requirement of numerous ballot boxes, safe stor- age of these ballot boxes, logistic problems, and counting of votes etc. the ECI was finding it hard to cope with these burgeoning challenges leading to various impediments in the electoral process. Furthermore lack of awareness and illiteracy played as a spoiler in rendering large scale invalid votes compelling to explore alternate avenues for the same.

1.1.2 Introduction of EVMs:

Bracing up against the scathing problems ensuing from the ballot papers, for the first time in the year 1977, ECI had advocated the idea of EVMs for conducting

Elections in India and Mr. S.L. Shakdhar, the then Chief Election Commissioner approached Electronics Corporation of India Limited, to start a comprehensive research and sought the practicality of using EVMs in conducting the Elections. By the end of 1979 a prototype was developed by the Electronics Corporation of India and on 6th August, 1980, the ECI exhibited the said prototype to the representatives of various political parties which was duly welcomed by the representatives.

1.1.3 Legal Battle:

After getting a 'go ahead' signal, the ECI in the year 1981 approached Bharat Electronics Limited, a Defence Ministry PSU to manufacture the EVMs. Thereafter on 19th May, 1982 ECI issued a directive under Article 324 of the Constitution of India which envisages superintendence, direction and control to the ECI in con- ducting Elections to use EVMs and conducted a trial run in around 50 polling stations in a bye election of Kerela Legislative Assembly (70-Parur Assemble Constituency). However the said Election was soon embroiled with legal obstacles as it was challenged in the court of law reaching up to the stage at Supreme Court wherein the Court held that "the order of the Election Commission directing casting of ballots by machines in some of the polling stations was without jurisdiction and could not have been resorted to". It further observed that "when the Representation of People Act 1951 and the Conduct of Election Rules 1961, pre- scribed a particular method of voting the Commission could-not innovate a new method and contend that use of the mechanical process was not covered by the existing law and, therefore, did not come in conflict with the law in the field. The Act and the Rules completely excluded the mechanical process which, if resorted to, would defeat in a large measure the mandatory requirements of the Rules".

The Supreme Court made it very clear that without inserting any specific provision in the Representation of People Act, 1951 with regard to voting by machines, Electronic Voting Machines cannot be used in polling booths for con- ducting Elections. Because Section 61 of the Representation of People Act, 1951 mandates that the voting shall only be conducted by the usage of Ballot Paper. This case in effect rendered voting through EVMs unlawful as a result of which Parliament made an amendment in the Act on 15th March, 1989 thereby inserting Section 61A enabling ECI to conduct Elections by using voting machines.

In the late 90s, the EVMs were intermittently used in conducting Elections in the states of Madhya Pradesh, Rajasthan and Delhi which was further substantially increased to various Parliamentary constituencies for General Elections. How- ever its usage and the vires of Section 61-A of the Act was further challenged in the court of law, however Hon'ble Supreme Court of India by dismissing the Petition upheld the constitutional validity of the newly inserted Section and consented to use EVMs for conducting Elections.

Despite its effectiveness and technological advancement, speculations and doubts persisted both in the political arena and the society regarding the possibil- ities of tampering and rigging with EVMs and questions were raised before vari- ous High Courts between 2001 and 2005 with respect to its transparency. After going through the technological soundness and administrative measures involved, all the High Courts observed that EVMs are tamperproof, credible and reliable. Karnataka High Court praised the way voting changed in India by observing that “This invention is undoubtedly a great achievement in the electronic and computer technology and a national pride”. Madras High Court further observed that “There is also no question of introducing any virus or bugs for the reason that the EVMs cannot be compared to personal computers. The programming in computers, as suggested, has no bearing with the EVMs. The computer would have inherent limitations having connections through Internet and by their very design, they may allow the alteration of the programme but the EVMs are independent units and the programme in EVM is entirely a different system”.

1.1.4 Looming threats of Transparency:

Notwithstanding ECI patting on its back for transforming the traditional, archaic way of voting through postal ballots with voting machines backed by sound technological systems and Indian Courts reaffirming the infallibility and giving credence to the same, threats of its transparency and vulnerability can't be ruled out. There are numerous instances wherein the voters had alleged that despite pressing button for a particular candidate, the red light gleams against a different candidate.

According to a report of BBC in the year 2010, the researchers in the University of Michigan had developed a technique hacking into the system by developing a homemade device connected to the EVM. Facing with severe criticism about the veracity of EVMs, ECI was constrained to throw an open challenge inviting skep- tics to hack its EVMs; however no political party had contested this challenge.

The recent incident pertaining to Bhind bye-election cannot be ignored where it was alleged that on pressing 4 (four) different buttons on the EVM, only the sym- bol of one party were printed. However the enquiry report by ECI denies any transparency flaw, stating the reason that the earlier data was not erased due to the non-adherence of prescribed protocol by the competent officers. Another incident in Dholpur (Rajasthan) Assembly by-poll election emerged where the EVM malfunctioning was alleged but the same have been vehemently denied by ECI.

Time and again various political parties and their leaders have doubted the precision of EVMs. But nowadays the debate has gone a bit notch higher after the recently concluded Assembly Elections of 5 states in 2017. Representatives of as much as 13 political parties met the ECI and submitted their joint representation expressing concerns and apprehensions about the transparency in voting through EVMs.

1.1.5 International Comparisons:

It might come as a bolt from the blue that most advanced nations in the world don't really fancy electronic voting. At least 24 countries have dabbled with electronic voting and today EVMs are under intense scanner. Serious doubts have been casted upon the accuracy, reliability, transparency and security of the Voting Machines.

Netherlands had banned the use of EVMs in the elections from October 2006. In spite of investing millions for the whole setup, the voting machines manufactured by a private Dutch company 'NEDAP' came to a standstill after an independent investigation finding them averse to modern IT and security threats. Ire- land also terminated its use in 2004 citing inadequate technological safeguards.

Germany had discontinued its e-voting machines manufactured from 'NEDAP' as it violated public nature of elections which requires that all essential steps in the Elections are subject to public examinability rendering voting through EVMs unconstitutional.

United States of America, second largest democracy after India does not have any standard system when it comes to voting as it is controlled by states and administered by counties and local government. Voting machines are used in some states while some use punch card systems, hand counted paper ballots etc. A TIME Report quotes the US Election Assistance Commission Chairman Tom Hicks in 2016 saying that the "primary reasons" paper ballots are used in most states are "security and voter preference".

Venezuela had introduced electronic voting in the year 1998 and rose to achieve the feat of becoming the first country in the world to use touch screens for registering votes. This technological advent further helped in eliminating duplication of votes by taking thumb prints of the electorates. Despite such hi-tech setup even this country grappled in a controversy in the recently held Presidential Elections. Smartmatic, the company which provides voting machines in Venezuela had made a statement that the elections have been manipulated by almost 1 million votes. Smartmatic CEO Antonio Mugica at a news briefing in London stated “We know, without any doubt, that the turnout of the recent election for a National Constituent Assembly was manipulated”.

Turning the tables to the nation where electronic voting had become an integral part and is thriving in the true sense is Brazil, which became the first country in the year 2000 to conduct elections completely by electronic voting system. Since then the country has maintained its electronic voting process with furthermore sound technical advancement. This success becomes exemplary keeping in mind a whopping more than 140 million registered voters.

1.2 Blockchain

Blockchain technology that shines like a star after the entrance and widespread acceptance of Bitcoin , the very first cryptocurrency in peoples’ everyday life, has become a trending topic in today's software world. At the beginning, Blockchain was only used for monetary transactions and trade, but studies have started to suggest that it can be used in many more areas over time, because there is a high degree of transparency in this system. For example, in Bitcoin, since the wallets are in a distributed structure, the total amount of coins and instant transaction volume in the world can be followed momentarily and clearly. There is no need for a central authority to approve or complete the operations on this P2P-based system. Because of that, not only the money transfers but also all kinds of structural information can be kept in this distributed chain, and with the help of some crypto logical methods, the system can be maintained securely. Like people's assets, marriage certificates, bank account books, medical information, etc., a lot of information can be recorded with this system with relevant modifications. Ethereum coin (Ether), another cryptocurrency with multipurpose development environments, which emerged a few years after Bitcoin, distinguishes the blockchain in a real sense, revealing that this technology can produce software that can hold information that is structured as described above. The software programs enforced by smart contracts are written into the blockchain and are immutable. They cannot be (illegally) removed nor manipulated once written. Hence, they can work properly, autonomously and transparently forever, without any external stimuli.

Blockchain is a distributed, immutable, incontrovertible, public ledger. This new technology has three main features:

(i) Immutability: Any proposed "new block" to the ledger must reference the previous version of the ledger. This creates an immutable chain, which is where the blockchain gets its name from, and prevents tampering with the integrity of the previous entries.

(ii) Verifiability: The ledger is decentralized, replicated and distributed over multiple locations. This ensures high availability (by eliminating a single point of failure) and provides third-party verifiability as all nodes maintain the consensus version of the ledger.

(iii) Distributed Consensus: A distributed consensus protocol to determine who can append the next new transaction to the ledger. A majority of the network nodes must reach a consensus before any new proposed block of entries becomes a permanent part of the ledger. These features are in part achieved through advanced cryptography, providing a security level greater than any previously known record-keeping system.

With its unique distributed and secure concept, the blockchain technology may address many issues other than digital trade. It might be a very suitable solution for e-voting projects. E-voting is being studied extensively, and many implementations are tested and even used for a while. However, very few implementations are reliable enough and are still in use. Of course, there are many successful examples of online polls and questionnaires, yet we cannot claim the same for online elections for governments and businesses. That’s mainly because, official elections are essential elements of the democracy and democratic administrations, which are the most preferred administrative methodology in the modern world. More, what is most valued in democratic societies is a robust electoral process that provides transparency and privacy. Today, a lot of decisions are being made by people (and members in organizations). Means of such voting systems are used in a lot of fields ranging from the law and act referendums to the TV shows. While most government elections and many organizational elections are held physically using sealed paper ballots, other polls and questionnaires are usually made on the Internet or SMS channels, notarized accounts are counted and publicly announced. But, legacy paper-to box voting systems create some questions; How reliable are the notaries at hand? How can we be sure that the votes people gave are not changed before they are counted on the system? How can we verify the transparency of the system? How can we prevent the tricks that reduce people's trust in the polls? How expensive is to hold an election in one vote center with 1000 voters, including material, logistics and salary costs? What about 1000 vote centers and 1,000,000 voters? And repeating all the setup for each election, considering there are a few each year? These and other similar problems have gradually entered a growth trend.

2. EXISTING SYSTEM

2.1 Traditional E-Voting System

Recent major technical challenges regarding e-voting systems include, but not limited to secure digital identity management. Any potential voter should have been enrolled to the voting system prior to the elections. Their information should be in a digitally processable format. Besides, their identity information should be kept private in any involving database. Traditional E- voting system may face following problems:

Anonymous vote-casting: Each vote may or may not contain any choice per candidate, should be anonymous to everyone including the system administrators, after the vote is submitted through the system.

Individualized ballot processes: How a vote will be represented in the involving web applications or databases is still an open discussion. While a clear text message is the worst idea, a hashed token can be used to provide anonymity and integrity. Meanwhile, the vote should be non-reputable, which cannot be guaranteed by the token solution.

Ballot casting verifiability by (and only by) the voter: The voter should be able to see and verify his/her own vote, after he/she submitted the vote. This is important to achieve in order to prevent, or at least to notice, any potential malicious activity. This counter measure, apart from providing means of non-repudiation, will surely boost the feeling of trust of the voters. These problems are partially addressed in some recent applications. Yet, means of e-voting is currently in use in several countries including Brazil, United Kingdom, Japan, and Estonia.

Estonia should be evaluated differently than the others, since they provide a full e-voting solution that is, said to be, equivalent of traditional paper-based elections.

High initial setup costs: Though sustaining and maintaining online voting systems is much cheaper than traditional elections, initial deployments might be expensive, especially for businesses.

Increasing security problems: Cyber-attacks pose a great threat to the public polls. No one would accept the responsibility if any hacking attempt succeeds during an election. The DDoS attacks are well known and mostly not the case in the elections. The voter integrity commission of the United States gave a testimony about the state of the elections in the US recently. Accordingly; Ronald Rivest stated that “hackers have myriad ways of attacking voting machines”. As an example; barcodes on ballots and smartphones in voting locations can be used in the hacking process. Apple stated that we mustn’t ignore the fact that computers are hackable, and the evidences can easily be deleted. Double-voting or voters from the other regions are also some common problems.

To mitigate these threats, software mechanisms which promise the following should be deployed:

- Prevention of evidence deletion.
- Transparency with privacy.

Lack of transparency and trust: How can people surely trust the results, when everything is done online? Perceptual problems cannot be ignored.

Voting delays or inefficiencies related to remote/absentee voting: Timing is very important in voting schemes; technical capabilities and the infrastructures should be reliable and run at the highest possible performance to let remote voting be synchronous.

The blockchain technology may address many issues regarding e-voting schemes mentioned in above section and make e-voting cheaper, easier, and much more secure to implement. It is a considerably new paradigm that can help to form decentralized systems, which assure the data integrity, availability, and fault tolerance. Some state that "the blockchain technology is bringing us the Internet of value: a new, distributed platform that can help us reshape the world of business and transform the old order of human affairs for the better." .This technology aims to revolutionize the systems. The blockchain systems are formed as decentralized networked systems of computers, which are used for validating and recording the pure online transactions. They also constitute ledgers, where digital data is tied to each other, called the blockchain. The records on the blockchains are essentially immutable.

3. Proposed System

3.1 Proposed Algorithms

This section explains PoS Sharding Protocol and zk-snarks, which are two essential concepts that this paper uses to provide the enhancements that the modern Blockchain-Based E-Voting System requires.

In this chapter, we introduce our proposed method as a possible solution of the blockchain scalability problem. Assume there are *nc* nodes in the network forming *c* groups, therefore each group contains *n* nodes. Two types of blocks are generated in proposed method. The middle blocks are generated by regular node groups and sent to final validation node group. The final blocks are generated by final validation group and broadcast to the network. In order to distinguish, the middle blocks are represented by lower-case "block" and the final blocks are represented by upper-case "BLOCK".

3.1.1 Sharding

The proposed method is mainly based on a sharding protocol and PoS consensus scheme. Assume the initial number of nodes in the network is *cn*. The *cn* nodes form *c* groups, which means each group contains *n* nodes. One of the *c* node groups works as validation node group and the other *c-1* node groups are regular groups. The regular node groups created middle blocks from the transaction shards assigned to them. The middle blocks are then processed in validation node group to produce final blocks which are recorded in the blockchain. To distinguish the two types of blocks created in the processes, the lower case "block" represents the middle block in Step 2 and the upper case "BLOCK" represents the final block in Step 3. Fig. 1. shows the main steps of the proposed method.

Each epoch contains 4 steps:

Step 1: Form node groups. Each node belongs to a group. After a node group is formed, a leader node is chosen randomly and all of the nodes' identities in this group are sent to it. After the group leader gathering all the nodes information in its group, an identity list is generated and broadcasted to other group leaders. This process reduces the communication complexity between nodes from $O(n^2)$ to $O(cn)$.

Step 2: Run internal group consensus. A transaction shard is assigned to a node group randomly. An internal PoS consensus is run in each node group. The node with large coin age (coin amount times holding time) has higher probability to be chosen to generate a new middle block.

Step 3: Generate final BLOCK. The final validation group collects and combines the middle blocks. A PoS consensus is run to generate a final BLOCK which is broadcasted to the whole network.

Step 4: Reshuffle the nodes. After *t* epochs, all of the nodes are reshuffled to form new node groups.

B. Form node groups

First, node groups are formed. Assume a group contains *n* nodes. The identities of the nodes are supposed to be known by others. A simple way is that each node broadcasts its identity to all other nodes. However, this results in $O(n^2)$ message complexity. ***C. Run internal group consensus***

After the node groups are formed, transaction shards are randomly assigned to groups. An internal group consensus is run in each group to generate middle blocks. We choose the PoS consensus mechanism. The node owes the highest coin age are more likely to be chosen to generate a middle block. The middle block is sent to the final validation node group.

D. Generate final block

The final validation node group collects the middle blocks and generates the final BLOCK. A PoS consensus is run to select a node to generate the final BLOCK. A final BLOCK mainly includes two parts: the previous BLOCK hash and new middle blocks.

E. Reshuffle the nodes

Nodes are reshuffled to form new groups every *t* epochs for higher security. Reshuffling could help to reduce the risk of centralization. After new node groups are formed, a new epoch starts from step 1.

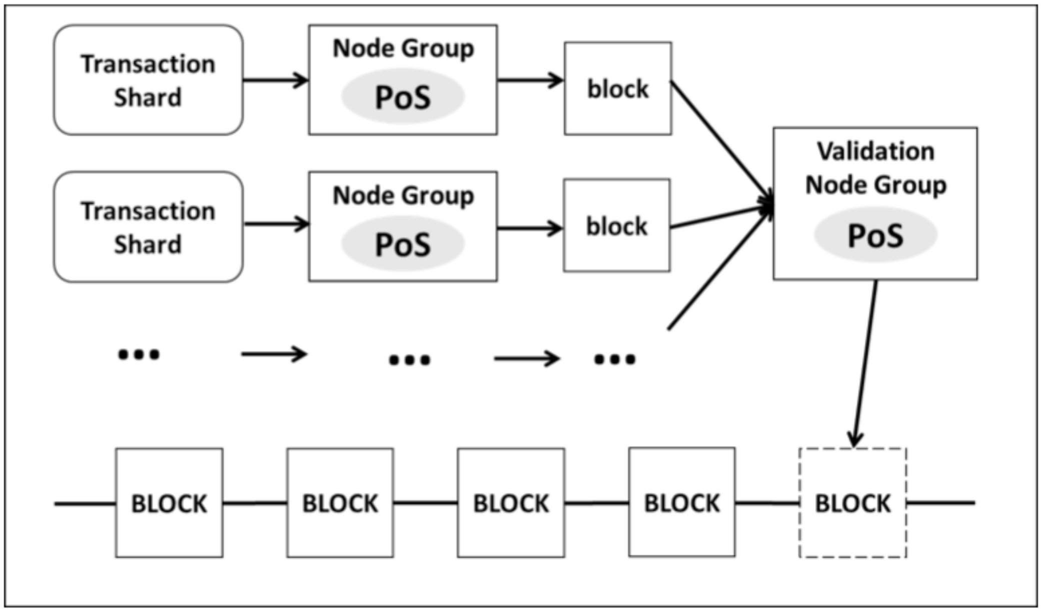


Figure 1: Sharding

3.2 ZK-SNARKS

ZK- SNARK: Data privacy is the most important thing now days. Let's talk about the ZK-SNARK. As the name suggest ***ZK*** stands for Zero-Knowledge and ***SNARK*** stand for "Succinct Non-Interactive Argument of Knowledge".

Zk-SNARK is an acronym for 'Zero-Knowledge Succinct Non-Interactive Argument of Knowledge'.

- ***ZERO-KNOWLEDGE:*** if the statement is true, a verifier does not learn anything beyond the fact that the statement is true.
- ***SUCCINT:*** It indicates that the zero-knowledge proof can be verified quickly. This includes proofs with statements that are large. With previous zero-knowledge protocols, the prover and the verifier had to engage in multiple rounds of communication in order to validate a proof.
- ***NON-INTERACTIVE:*** It means that the verifier does not have to interact with the Prover in order to validate a zero-knowledge proof. Instead, the Prover can publish their proof in advance, and a verifier can ensure its correctness.
- ***ARGUMENT OF KNOWLEDGE****:* A computationally sound proof: soundness holds against the Prover that leverages polynomial-time, i.e. bounded computation. The proof cannot be constructed without access to the witness (the private input needed to prove the statement).

3.2.1 ZK- SNARK TRANSACTION:

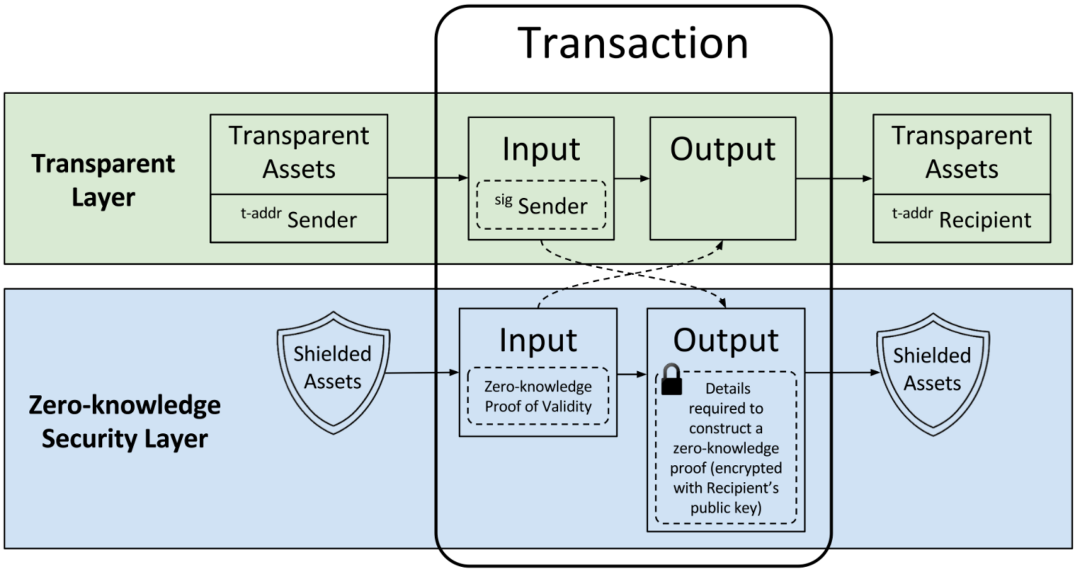


Figure 2: ZK-SNARKS Transcation

In incorporating zk-SNARKS into the Zcash blockchain, the function that determines the correctness of a transaction, in accordance with consensus rules, must return the answer of whether or not that transaction is valid, without disclosing any of the information with which it performed calculations. This is accomplished by encoding some Zcash consensus rules into zk-SNARKs themselves.

A* zk-SNARK *consists of three algorithms G, P, V defined as follows:

The *key generator* G takes a secret parameter lambda and a program C, and generates two publicly available keys, a *proving key* pk, and a *verification key* vk. These keys are public parameters that only need to be generated once for a given program C.

The *prover* P takes as input the proving key pk, a public input x and a private witness w. The algorithm generates a *proof* prf = P(pk, x, w) that the prover knows a witness w and that the witness satisfies the program.

The *verifier* V computes V(vk, x, prf) which returns true if the proof is correct, and false otherwise. Thus this function returns true if the prover knows a witness w satisfying C(x,w) == true.

Note here the secret parameter lambda used in the generator. This parameter sometimes makes it tricky to use zk-SNARKs in real-world applications. The reason for this is that anyone who knows this parameter can generate fake proofs. Specifically, given any program C and public input x a person who knows lambda can generate a proof fake_prf such that V(vk, x, fake_prf) evaluates to true without knowledge of the secret w.

Thus actually running the generator requires a very secure process to make sure no-one learns about and saves the parameter anywhere. This was the reason for the extremely elaborate ceremony the Zcash team conducted in order to generate the proving key and verification key, while making sure the “toxic waste” parameter lambda was destroyed in the process.

3.2.2 A zk-SNARK for our example program

How would Alice and Bob use a zk-SNARK in practice in order for Alice to prove that she knows the secret value in the example above?

First of all, as discussed above we will use a program defined by the following function:

```
function C(x, w) {
  return ( sha256(w) == x );
}
```

The first step is for Bob to run the generator G in order to create the proving key pk and verification key vk. This is done by first randomly generating lambda and using that as input:
(pk, vk) = G(C, lambda)

As discussed above, the parameter lambda must be handled with care, since if Alice learns the value of lambda she will be able to create fake proofs. Bob will share pk and vk with Alice.

Alice will now play the role of the prover. She needs to prove that she knows the value s that hashes to the known hash H. She runs the proving algorithm P using the inputs pk, H and s to generate the proof prf:
prf = P(pk, H, s)

Next Alice presents the proof prf to Bob who runs the verification function V(vk, H, prf) which would return true in this case since Alice properly knew the secret s. Bob can be confident that Alice knew the secret, but Alice did not need to reveal the secret to Bob.

Reusable proving and verification keys:

In our example above the zk-SNARK cannot be used if Bob wants to prove to Alice that he knows a secret. This is because Alice cannot know that Bob didn’t save the lambda parameter, and so Bob could plausibly be able to fake proofs.

If a program is useful to many people (like the example of Zcash) a trusted independent group separate from Alice and Bob could run the generator and create the proving key pk and verification key vk in such a way that no one learns about lambda.

Anyone who trusts that the group did not cheat can then use these keys for future interactions.

zk-SNARKs in Ethereum

Developers have already started integrating zk-SNARKs into Ethereum. What does this look like? Concretely, the building blocks of the verification algorithm is added to Ethereum in the form of precompiled contracts. The usage is the following: The generator is run off-chain to produce the proving key and verification key. Any prover can then use the proving key to create a proof, also off-chain. The general verification algorithm can then be run inside a smart contract, using the proof, the verification key and the public input as input parameters. The outcome of the verification algorithm can then be used to trigger other on-chain activity.

4. IMPLEMENTATION

To introduce a method of secure authentication, our pro- posed system is designed to use electronic ID authentication via Auðkenni, which is an Icelandic service provider for identity verification. Auðkenni utilizes the Nexus software and RFID scanners. When a user registers for an electronic ID, a user chooses a PIN number for its corresponding ID consisting of 6 numbers. A user will therefore identify himself in the voting booth by scanning his ID and providing his corresponding PIN number to authenticate himself to the system.

1. Any computer in any voting district can be used by any eligible voter to vote, since the wallet for the corresponding voter has information on which voting district the voter is supposed to vote from. For a user to successfully authenticate, a valid ID and PIN number needs to be presented at a voting district using a card reader and the nexus software.
2. If the authentication is successful, the corresponding smart contract is prompted for the ongoing election. The ballot for the aforementioned election is a smart contract which has a list of the candidates a voter can choose from.
3. When a voter has selected a candidate and casts his vote, the voter proceeds to sign his vote by re-entering the corresponding PIN number for his electronic ID.
4. After the voter has signed his vote, the vote data proceeds to be verified by the corresponding district node, which the voter is interacting with the smart contract through. If the aforementioned district node accepts the vote data, the vote data must be agreed upon by the majority corresponding district node.
5. If the majority of district nodes agree upon the vote data, consensus for the particular vote has been reached. The user then receives the transaction ID for the corresponding transaction of his vote in the form of a QR-code and the ption to print the transaction ID. When the vote is casted and has been verified, a function in the smart contract adds

one vote to the party which was voted for. This functionality of the smart contract structure is utilized to determine the election result in each of the voting districts. Figure 3 is a visual representation of the steps we just ellaborated.

6. All transactions which were received and verified in the ongoing block time are deployed onto the blockchain after the block time has reached its time limit. With each new block added to the blockchain, each district node updates his copy of the ledger.

Below, we will elaborate the functionalities of a novel ballot and election smart contract for an e-voting system, without the integration of a government indentity verification service.

```
contract ElectionCreation {
address[] public deployedBallots; constructor (bytes32[] candidates, bytes32[] district, uint
hours) public {

   uint expirationDate;
}

Candidates[] public candidates;
address public manager;
bytes32 public votingDistrict; mapping(address => bool) public voters;

modifier restricted() { require(msg.sender == manager); _;

}
constructor (bytes32[] candidateNames, bytes32 district, address creator, uint
amountOfHours) public {

manager = creator; votingDistrict = district;

}

for(uint i = 0; i < district.length; i++){

address newBallot =
new Ballot(candidates, district[i], msg.sender, hours); deployedBallots.push(newBallot);

} }

function getDeployedBallots() public view returns(address[]) {

return deployedBallots;
```

ElectionCreation constructor: Takes in a list of candidates and districts along with the address of the wallet of the creator and the amount of hours the election will take. The constructor then creates a single smart contract for each district provided and puts the address of each smart contract created into the deployedBallots array. getDeployedBallots: returns an array with the address of each created smart contract in each index of the array.

```
contract Ballot {
   struct Candidates {
      bytes32 name;
      uint voteCount;
      uint creationDate;
   uint expirationDate;
}

Candidates[] public candidates;
address public manager;
bytes32 public votingDistrict; mapping(address => bool) public voters;

modifier restricted() { require(msg.sender == manager); _;

}
constructor (bytes32[] candidateNames, bytes32 district, address creator, uint amountOfHours)
public {

          manager = creator; votingDistrict = district;

          votingDistrict = district;
```

Restricted modifier: This modifier is used to restrict functions in the manner of that only the creator of the election can access the information which the functions give.

Ballot constructor: Sets the manager of the ballot smart contract to the address of the wallet which created the election, the voting district of the smart contract to the district which the ElectionCreation contract provided and then proceeds to fill the Candidates struct with the list of candidates provided and the number of votes for each candidate to 0. The constructor also stores the time of the creation of the contract along with the time when the contract is to expire.

```
function vote(uint candidate) public{ require(!voters[msg.sender]);
if(now > candidates[candidate].expirationDate){

revert(); }

candidates[candidate].voteCount += 1;

voters[msg.sender] = true;

}
```

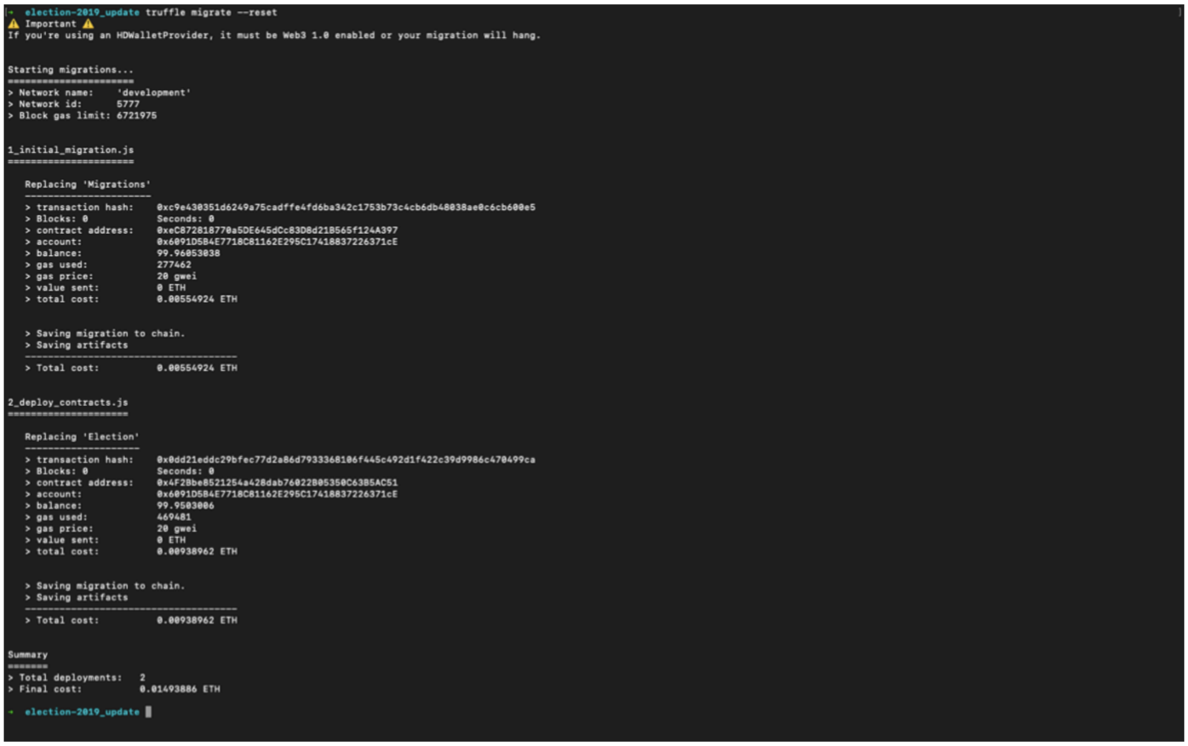
Vote: This function allows voters to vote. The requirement for a voter to vote, is that the mapping of the address of the voter is set to its default, false. If that is the case, the function guarantees that the election time limit has not been reached. If both requirements are satisfied, the contract retrieves the index of which candidate was voted for and increases his vote count by 1 and sets the mapping to true, so that the voter can never vote again in this particular election.

```
function getCandidateName(uint index) public restricted view

{

}
```

```
require(now > candidates[candidate] .expirationDate)
return candidates[index].voteCount;

returns (bytes32)

{

return candidates[index].name; }

function getVoteCount(uint index) public restricted view

require(now > candidates[candidate] .expirationDate

)
}
```

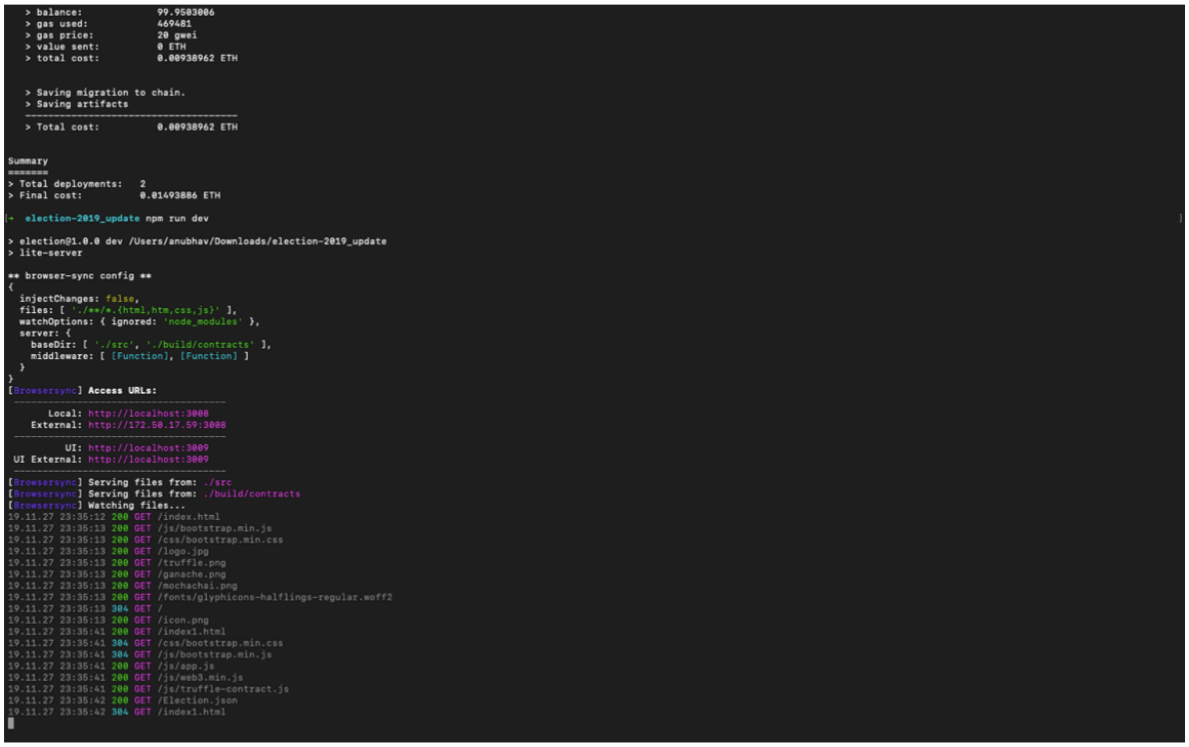
getCandidateName & getVoteCount: Both these functions retrieve the name and amount of votes a candidate has recieved from an index. These functions classify as helper functions to determine the election results after the elec- tion is finished

5. RESULTS



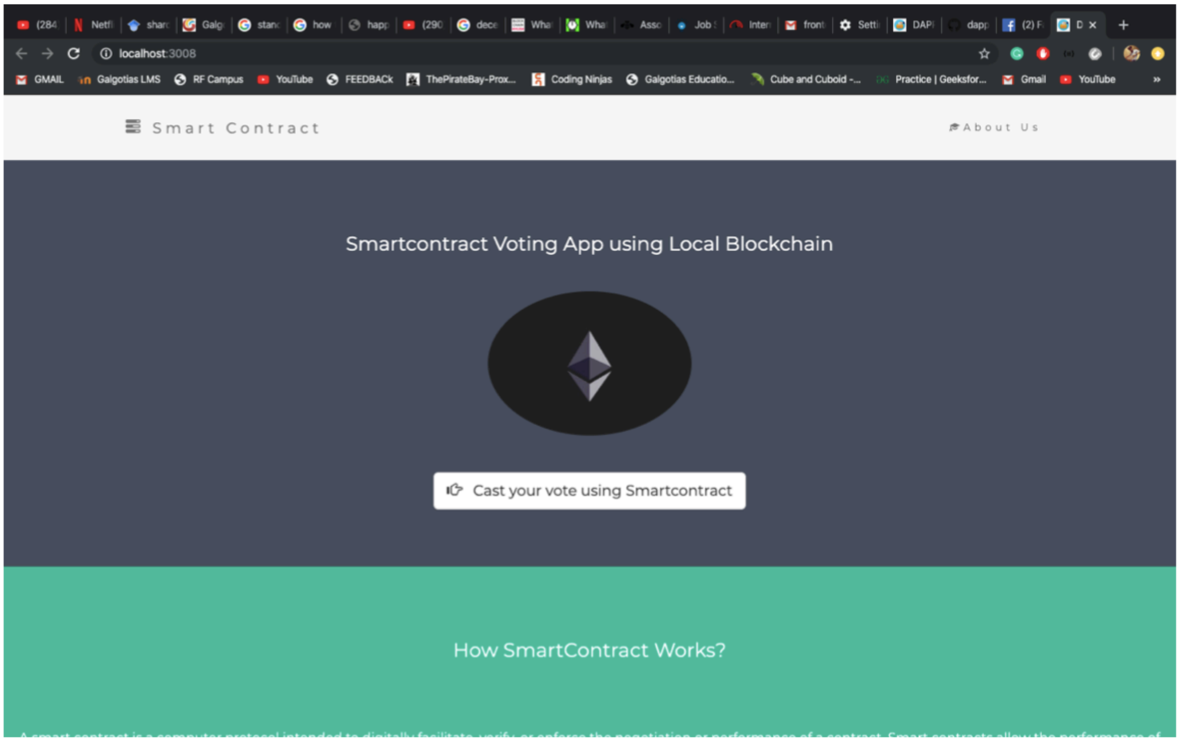
```
→  election-2019_update truffle migrate --reset
⚠ Important ⚠
If you're using an HDWalletProvider, it must be Web3 1.0 enabled or your migration will hang.


Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975


1_initial_migration.js
======================

   Replacing 'Migrations'
   ----------------------
   > transaction hash:    0xc9e430351d6249a75cadffe4fd6ba342c1753b73c4cb6db48038ae0c6cb600e5
   > Blocks: 0            Seconds: 0
   > contract address:    0xeC872818770a5DE645dCc83D8d21B565f124A397
   > account:             0x6091D5B4E7718C81162E295C17418837226371cE
   > balance:             99.96053038
   > gas used:            277462
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00554924 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00554924 ETH


2_deploy_contracts.js
=====================

   Replacing 'Election'
   --------------------
   > transaction hash:    0x0dd21eddc29bfec77d2a86d7933368106f445c492d1f422c39d9986c470499ca
   > Blocks: 0            Seconds: 0
   > contract address:    0x4F2Bbe8521254a428dab76022B05350C63B5AC51
   > account:             0x6091D5B4E7718C81162E295C17418837226371cE
   > balance:             99.9503006
   > gas used:            469481
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00938962 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00938962 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.01493886 ETH

→  election-2019_update
```

Screenshot 1: Migration of Smart Contracts



```
   > balance:             99.9503006
   > gas used:            469481
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00938962 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00938962 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.01493886 ETH

→  election-2019_update npm run dev

> election@1.0.0 dev /Users/anubhav/Downloads/election-2019_update
> lite-server

** browser-sync config **
{
  injectChanges: false,
  files: [ './**/*.{html,htm,css,js}' ],
  watchOptions: { ignored: 'node_modules' },
  server: {
    baseDir: [ './src', './build/contracts' ],
    middleware: [ [Function], [Function] ]
  }
}
[Browsersync] Access URLs:
 -------------------------------------
       Local: http://localhost:3000
    External: http://172.50.17.59:3000
 -------------------------------------
          UI: http://localhost:3009
 UI External: http://localhost:3009
 -------------------------------------
[Browsersync] Serving files from: ./src
[Browsersync] Serving files from: ./build/contracts
[Browsersync] Watching files...
19.11.27 23:35:12 200 GET /index.html
19.11.27 23:35:13 200 GET /js/bootstrap.min.js
19.11.27 23:35:13 200 GET /css/bootstrap.min.css
19.11.27 23:35:13 200 GET /logo.jpg
19.11.27 23:35:13 200 GET /truffle.png
19.11.27 23:35:13 200 GET /ganache.png
19.11.27 23:35:13 200 GET /mochachai.png
19.11.27 23:35:13 200 GET /fonts/glyphicons-halflings-regular.woff2
19.11.27 23:35:13 304 GET /
19.11.27 23:35:13 200 GET /icon.png
19.11.27 23:35:41 200 GET /index1.html
19.11.27 23:35:41 304 GET /css/bootstrap.min.css
19.11.27 23:35:41 304 GET /js/bootstrap.min.js
19.11.27 23:35:41 200 GET /js/app.js
19.11.27 23:35:41 200 GET /js/web3.min.js
19.11.27 23:35:41 200 GET /js/truffle-contract.js
19.11.27 23:35:42 200 GET /Election.json
19.11.27 23:35:42 304 GET /index1.html
```

Screenshot 2: Deployment and Running Local Blockchain Network

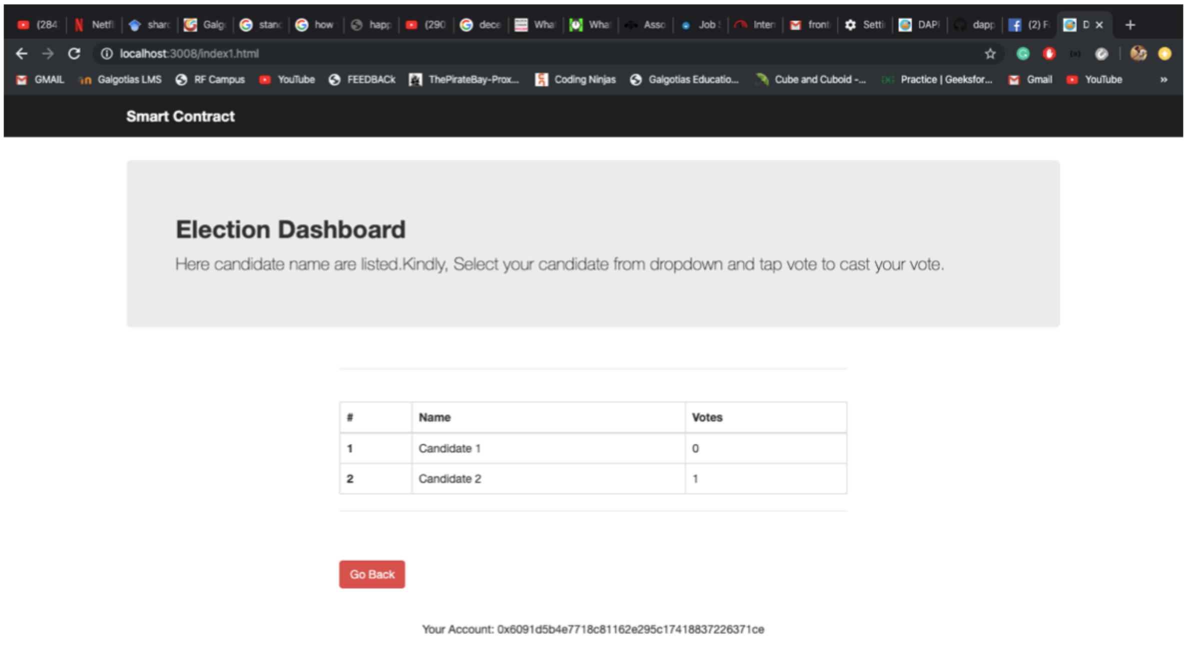


Smartcontract Voting App using Local Blockchain

How SmartContract Works?

A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of

Screenshot 3: Interface 1



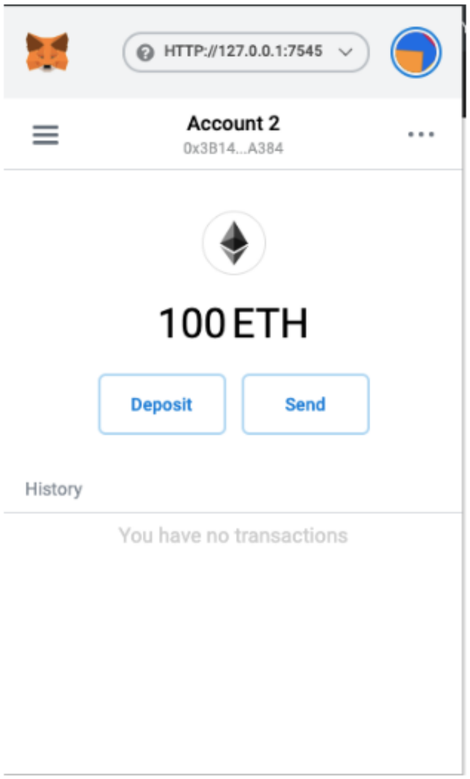
Election Dashboard

Here candidate name are listed.Kindly, Select your candidate from dropdown and tap vote to cast your vote.

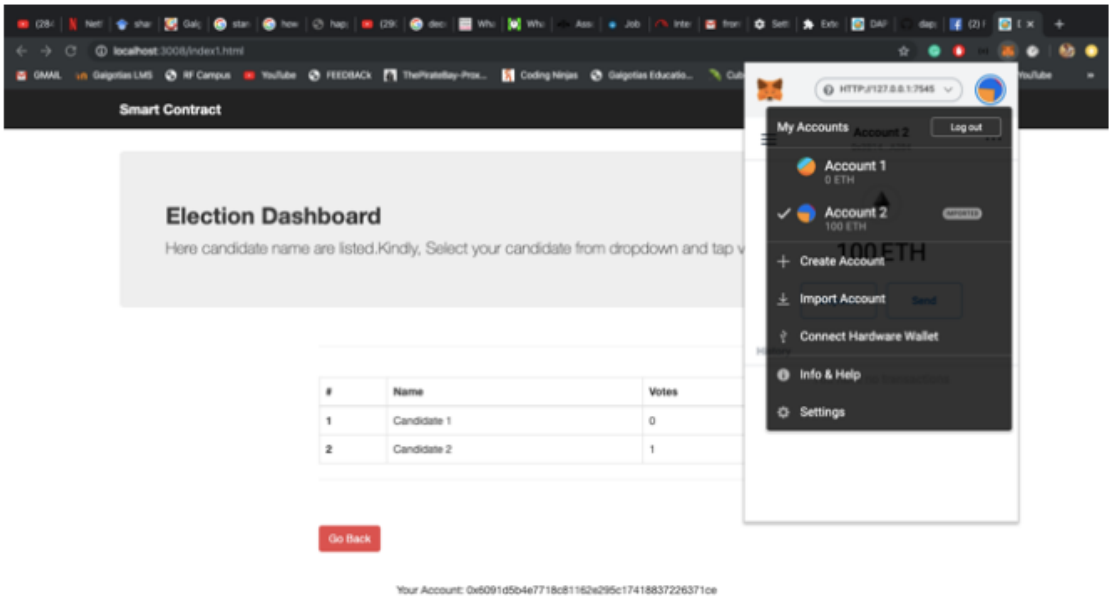
#	Name	Votes
1	Candidate 1	0
2	Candidate 2	1

Your Account: 0x6091d5b4e7718c81162e295c17418837226371ce

Screenshot 4: Interface 2 (Voting Screen)



Screenshot 5: Ethereum Wallet (Metamask)



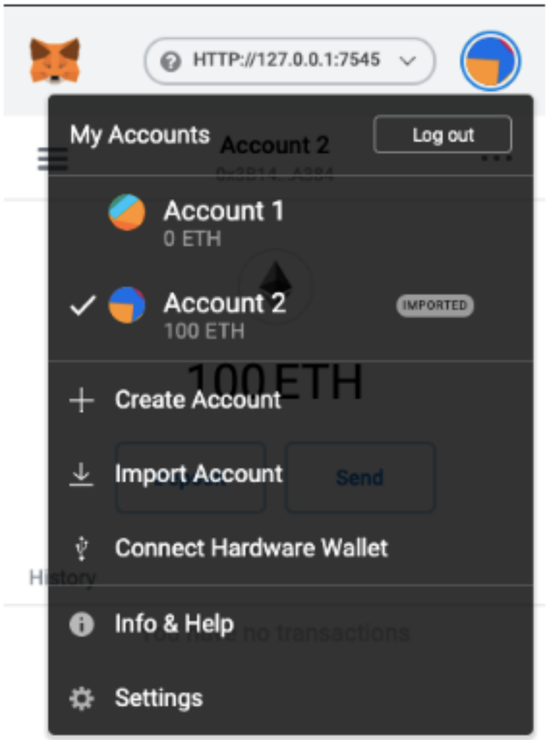
Election Dashboard

Here candidate name are listed.Kindly, Select your candidate from dropdown and tap v

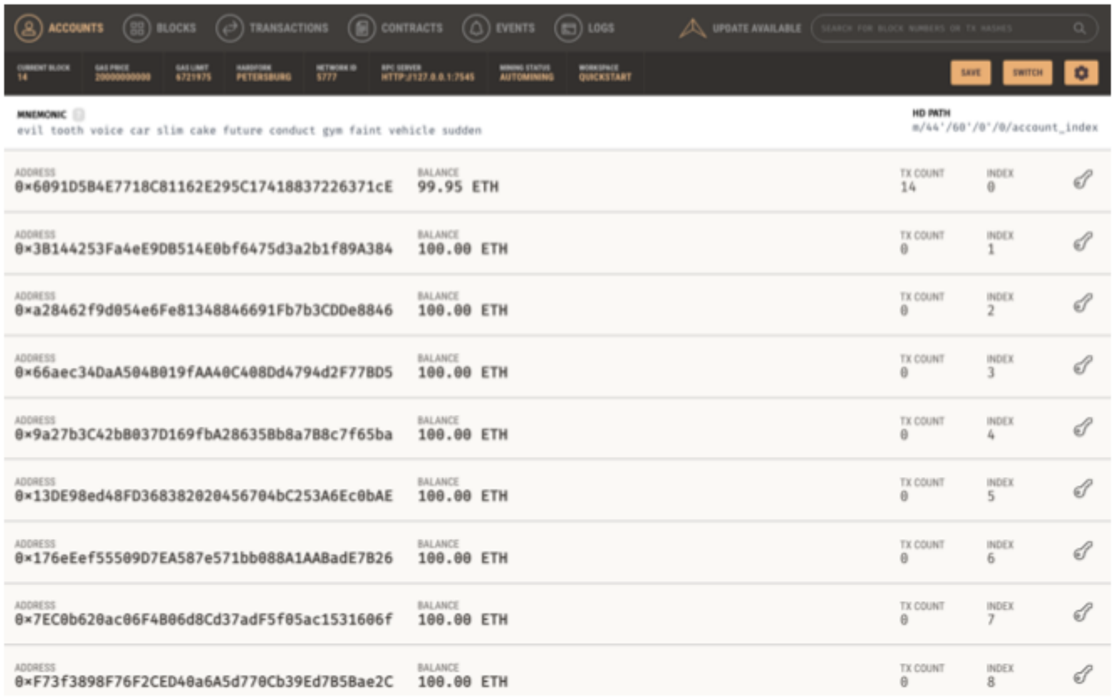
#	Name	Votes
1	Candidate 1	0
2	Candidate 2	1

Your Account: 0x6091d5b4e7718c81162e295c17418837226371ce

Screenshot 6: Interface 3 (Wallet LoggedIn)



Screenshot 7: Metamask



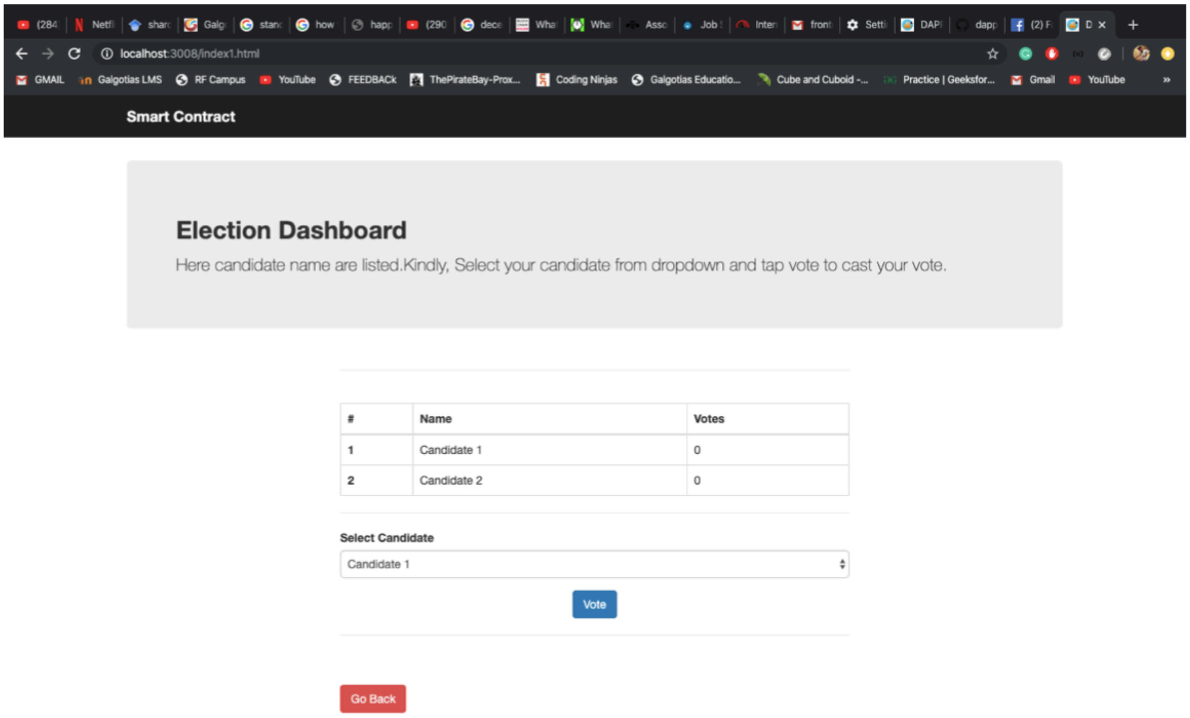
CURRENT BLOCK 14 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK PETERSBURG | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE QUICKSTART

MNEMONIC
evil tooth voice car slim cake future conduct gym faint vehicle sudden

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×6091D5B4E7718C81162E295C17418837226371cE	99.95 ETH	14	0
0×3B144253Fa4eE9DB514E0bf6475d3a2b1f89A384	100.00 ETH	0	1
0×a28462f9d054e6Fe81348846691Fb7b3CDDe8846	100.00 ETH	0	2
0×66aec34DaA504B019fAA40C408Dd4794d2F77BD5	100.00 ETH	0	3
0×9a27b3C42bB037D169fbA28635Bb8a7B8c7f65ba	100.00 ETH	0	4
0×13DE98ed48FD368382020456704bC253A6Ec0bAE	100.00 ETH	0	5
0×176eEef55509D7EA587e571bb088A1AABadE7B26	100.00 ETH	0	6
0×7EC0b620ac06F4B06d8Cd37adF5f05ac1531606f	100.00 ETH	0	7
0×F73f3898F76F2CED40a6A5d770Cb39Ed7B5Bae2C	100.00 ETH	0	8

Screenshot 8: Ganache



Smart Contract

Election Dashboard

Here candidate name are listed.Kindly, Select your candidate from dropdown and tap vote to cast your vote.

#	Name	Votes
1	Candidate 1	0
2	Candidate 2	0

Select Candidate: Candidate 1

Screenshot 9: Interface 4

6. CONCLUSION

The idea of adapting digital voting systems to make the public electoral process cheaper, faster and easier, is a compelling one in modern society. Making the electoral process cheap and quick, normalizes it in the eyes of the voters, removes a certain power barrier between the voter and the elected official and puts a certain amount of pressure on the elected official. It also opens the door for a more direct form of democracy, allowing voters to express their will on individual bills and propositions.

In this project, I introduced a unique, blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost efficient election while guaranteeing voters privacy. I have outlined the systems architecture, the design, and a security analysis of the system. By comparison to previous work, we have shown that the blockchain technology offers a new possibility for democratic countries to advance from the pen and paper election scheme, to a more cost- and time-efficient election scheme, while increasing the security measures of the todays scheme and offer new possibilities of transparency. Using an Ethereum private blockchain, it is possible to send hundreds of transactions per second onto the blockchain, utilizing every aspect of the smart contract to ease the load on the blockchain. For countries of greater size, some measures must be taken to withhold greater throughput of transactions per second, for example the parent & child architecture which reduces the number of transactions stored on the blockchain at a 1:100 ratio without compromising the networks security. Our election scheme allows individual voters to vote at a voting district of their choosing while guaranteeing that each individual voters vote is counted from the correct district, which could potentially increase voter turnout.

7. REFERENCES

[1] S. Bangar, “Journey from ballot papers to evms: How india votes,”International Education and Research Journal, vol. 5, no. 5, 2019.[Online]. Available: http://ierj.in/journal/index.php/ierj/article/view/1877

[2] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2009.[Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[3] D. Vujicic, D. Jagodic, and S. Rani ́c, “Blockchain technology, bitcoin,and ethereum: A brief overview,” 03 2018, pp. 1–6.

[4] M. Alharby, A. Aldweesh, and A. van Moorsel, “Blockchain-based smartcontracts: A systematic mapping study of academic research (2018),” 062019.

[5] S. Shahab and Z. Allam, “Reducing transaction costs of tradable permitschemes using blockchain smart contracts,”Growth and Change, vol. 51,pp. 302–308, 03 2020.

[6] Y. Hu, M. Liyanage, A. Manzoor, K. Thilakarathna, G. Jourjon, andA. Seneviratne, “Blockchain-based smart contracts - applications andchallenges,” 06 2019.

[7] E. Yavuz, A. K. Koc ̧, U. C. C ̧abuk, and G. Dalkılıc ̧, “Towards secure e-voting using ethereum blockchain,” in2018 6th International Symposiumon Digital Forensic and Security (ISDFS), March 2018, pp. 1–7.

[8] F. Sheer Hardwick, A. Gioulis, R. Naeem Akram, and K. Markantonakis,“E-voting with blockchain: An e-voting protocol with decentralisationand voter privacy,” in2018 IEEE International Conference on Internetof Things (iThings) and IEEE Green Computing and Communications(GreenCom) and IEEE Cyber, Physical and Social Computing (CP-SCom) and IEEE Smart Data (SmartData), July 2018, pp. 1561–1567.

[9] H. V. Patil, M. K. G. Rathi, and M. M. V. Tribhuwan, “A study ondecentralized e-voting system using blockchain technology,” 2018.

[10] S. P. A. V. S. T. Anuj Upadhyay, Satya Doulani, “E-voting usingethereum blockchain,” inInternational Journal of Science EngineeringDevelopment Research (www.ijrti.org), vol. 3, November 2018, pp. 30–34. [Online]. Available: http://www.ijrti.org/papers/IJRTI1811006.pdf

[11] F. Hj´almarsson, G. K. Hreiarsson, M. Hamdaqa, and G. Hj´almt´ysson,“Blockchain-based e-voting system,” in2018 IEEE 11th InternationalConference on Cloud Computing (CLOUD), 2018, pp. 983–986.

[12]H.N.YuefeiGao,“Proofofstakeshardingprotocolforscalableblockchains,”inPROCEEDINGS OF THE44TH MEETING OF THE ASIA-PACIFIC ADVANCED NET-WORK,vol.44,2017,pp.1316.[Online].Available:http://journals.sfu.ca/apan/index.php/apan/article/view/214/pdf