

**Project  
On  
Implementation of Smart Contracts Ethereum Blockchain in  
Web-Based Electronic Voting (e-voting)**

**CS201W03**

**For partial fulfillment of the requirement of the degree in**

**BACHELOR OF ENGINEERING  
(Department of Computer Science & Engineering)**

**CUIET**

**By-  
Name: Harshita  
Roll Number: 2110990585  
Cluster: Beta  
Section: 11**

**CHITKARA UNIVERSITY  
PUNJAB, INDIA**

**CHITKARA UNIVERSITY INSTITUTE OF ENGINEERING**

**AND TECHNOLOGY  
July – Dec (2022)**

# Index

Chapter 1	Introduction
Chapter 2	Proposed Methodology
Chapter 3	Results and discussions
Chapter 4	Conclusion
Chapter 5	Bibliography

# CHAPTER 1

## INTRODUCTION

A blockchain is a distributed software network that functions both as a digital ledger and a mechanism enabling the secure transfer of assets without an intermediary. Just as the internet is a technology that facilitates the digital flow of information, blockchain is a technology that facilitates the digital exchange of units of value. Anything from currencies to land titles to votes can be tokenized, stored, and exchanged on a blockchain network.

The first manifestation of blockchain technology emerged in 2009 with the Bitcoin blockchain, a secure, censorship-resistant, peer to peer electronic cash system.

Because Bitcoin is accessible to anyone, it is an example of an open, or a permissionless blockchain.

Today, there are many forms of blockchain technology. Some blockchains have been designed to meet the needs of a finite group of participants, where access to the network is restricted. These are examples of private, or permissioned blockchains.

In addition to the secure transfer of value, blockchain technology provides a permanent forensic record of transactions and a single version of the truth – a network state that is fully transparent and displayed in real time for the benefit of all participants.

Regardless of the type of blockchain protocol that is deployed, blockchain technology holds great promise to transform centuries-old business models, paving the way for higher levels of legitimacy in government and creating new opportunities for prosperity for everyday citizens.

# Key elements of a blockchain

## **Distributed ledger technology**

All network participants have access to the distributed ledger and its immutable record of transactions. With this shared ledger, transactions are recorded only once, eliminating the duplication of effort that's typical of traditional business networks.

## **Immutable records**

No participant can change or tamper with a transaction after it's been recorded to the shared ledger. If a transaction record includes an error, a new transaction must be added to reverse the error, and both transactions are then visible.

## **Smart contracts**

To speed transactions, a set of rules — called a [smart contract](#) — is stored on the blockchain and executed automatically. A smart contract can define conditions for corporate bond transfers, include terms for travel insurance to be paid and much more.

# How blockchain works

## **As each transaction occurs, it is recorded as a “block” of data**

Those transactions show the movement of an asset that can be tangible (a product) or intangible (intellectual). The data block can record the information of your choice: who, what, when, where, how much and even the condition — such as the temperature of a food shipment.

## **Each block is connected to the ones before and after it**

These blocks form a chain of data as an asset moves from place to place or ownership changes hands. The blocks confirm the exact time and sequence of transactions, and the blocks link securely together to prevent any block from being altered or a block being inserted between two existing blocks.

## **Transactions are blocked together in an irreversible chain: a blockchain**

Each additional block strengthens the verification of the previous block and hence the entire blockchain. This renders the blockchain tamper-evident, delivering the key strength of immutability. This removes the possibility of tampering by a malicious actor — and builds a ledger of transactions you and other network members can trust.

## **Benefits of blockchain**

What needs to change: Operations often waste effort on duplicate record keeping and third-party validations. Record-keeping systems can be vulnerable to fraud and cyberattacks. Limited transparency can slow data verification. And with the arrival of IoT, transaction volumes have exploded. All of this slows business, drains the bottom line — and means we need a better way. Enter blockchain.

### **Greater trust**

With blockchain, as a member of a members-only network, you can rest assured that you are receiving accurate and timely data, and that your confidential blockchain records will be shared only with network members to whom you have specifically granted access.

### **Greater security**

Consensus on data accuracy is required from all network members, and all validated transactions are immutable because they are recorded permanently. No one, not even a system administrator, can delete a transaction.

### **More efficiencies**

With a distributed ledger that is shared among members of a network, time-wasting record reconciliations are eliminated. And to speed transactions, a set of rules — called a smart contract — can be stored on the blockchain and executed automatically.

# CHAPTER 2

## PROPOSED METHODOLOGY

### Implementation of Smart Contracts Ethereum Blockchain in Web-Based Electronic Voting (e-voting)

#### 1. Introduction

In a large scope like a country, conventional voting systems are still often used. The system is inefficient because it causes various problems such as the length of time required, requires substantial costs, often occurs damaged or invalid ballots, cheating and so on. These problems arise because the voting system is managed centrally. Voting data from voters can easily be manipulated by the manager without anyone knowing. This makes the centralized voting system not safe to do in Indonesia which has a very large population. Blockchain innovation is one arrangement that can be utilized to lessen issues that happen in democratic by e-casting a ballot. Blockchain itself is known to have been used in the Bitcoin and Ethereum transaction database systems. Blockchain is a technology for recording transactions that are interconnected using unique codes in it that are eternal and cannot be changed.

Previously, it was hard to accomplish fair election without gambling extortion and manipulation of results. With the approach of Ethereum, numerous analysts on blockchain

innovation and smart contracts have proposed this is a suitable reason for electronic voting. In

addition, it can have the possibility of making electronic voting progressively worthy and

dependable in the public eye. Smart contracts are computer programs that are run through

blockchain transactions that can maintain status, interact with cryptocurrency in a decentralized

way, and take user input [2]. Smart contracts are a significant code to be incorporated in the

blockchain and executed by timetable at each progression of the blockchain update.

Electronic

voting, then again, is another theme that is trending, however significant identified with online

services. Blockchain with smart contracts, shows up as a decent possibility to be utilized in the

improvement of electronic voting that are less expensive, more secure, more transparent, and

simpler to utilize [3]. Ethereum and its system is one of the most sensible, because of its

consistency, its wide use, and the stock of smart contracts logic.

This e-voting system can overcome problems such as a centralized server that is very likely to be disabled or hacked by hackers, data changes unilaterally, and prevent the possibility

of fraud. This e-voting system can also overcome the general problems of efficiency in the form

of cost, time and so on. This system can also create transparency in the voting process.

In this

work, we have implemented and tested e-voting application examples as smart contracts for the

Ethereum network using the Ethereum network and the Solidity language. With the blockchain



technology, the public can monitor data recording in a database safely without being able to mess up the data. The process of counting of votes which is fairly fast and efficient as well as data from voters that can be easily audited is an added value of this system.

## 2. Research Method

### 2.1 Blockchain

Blockchain technology was discovered in 2009 along with the emergence of Bitcoin cryptocurrency, a virtual currency that is becoming a trend now [4]. Blockchain technology was born in response to the concerns of a number of parties about the workings of centralized or centralized software. Centralized software raises concerns because it is basically designed to send data from one party to another, requiring a centralized server as a publisher and data manager. When there is a disruption to the server, the website cannot be accessed and users cannot automatically use the service optimally. Blockchain is a technology for recording transactions that are interconnected using unique codes in it that are eternal and cannot be changed [1]. The way the blockchain works is that when another transaction or alter to a current transaction enters the blockchain, by and large most node in the blockchain execution must run an algorithm to evaluate and check the history of the recently proposed individual blockchain. In the event that most of

node reach consensus a legitimate signature, another transaction block is inserted into the database and another block is added to the transaction chain. If the majority does not approve the addition or modification of database entries, it is rejected and not added to the blockchain chain.

This distributed model is the thing that permits the blockchain to keep running as a disseminated

database without the requirement for various specialists, the binding together focus says which

exchanges are substantial and (maybe progressively significant) which ones are most certainly

Not.

## There are several types of Blockchain namely:

1. Permissionless Blockchain, as Bitcoin or Ethereum, all can be clients or run a node, anybody can "compose", and anybody can partake in consensus in deciding the validity of the state.
2. Permission Blockchain which is conversely corresponding to the past sort, worked by referred to elements, for example, the consortium blockchains, where consortium individuals or partners in certain business settings work the Blockchain authorization arrange. This Blockchain permission system has the way to distinguish node that can control and update shared information, and frequently has an approach to control who can give exchanges.
3. Private blockchain is a special blockchain that is allowed by one entity, where there is just one trust domain.

## 2.2 Ethereum

Ethereum is the second largest cryptocurrency of all market capitalization which has

extensive documentation and an active developer community [6]. Ethereum

Blockchain is an

open source distributed computing platform that highlights smart contracts (scripting) usefulness.

Engineers can without much of a stretch write decentralized applications at a significant level and

advantage from distributions acquired from Blockchain technology [7].

Ethereum, as Bitcoin, is a hyper ledger of public blockchain cryptocurrency. The thing that matters is the Bitcoin blockchain just stores transactions that trade Bitcoin between

addresses, though Ethereum Blockchain stores addresses with EVM codes.

Transactions that are

recorded on the blockchain are code calls referenced above, and contain data about the

information go to the program as input. These projects are interpreted by a constrained virtual

machine called Ethereum Virtual Machine (EVM) and expressed in the fitting language.

In this section, some basic ethereum concepts are explained: accounts, transactions, and

clients. The basic unit of ethereum is the account. An account is required for everyone who wants

to send any transaction to the blockchain. Ethereum itself includes two types of accounts namely:

Externally Owned Accounts (EOA), users directly send transactions through them, and Contract

Accounts, which are based on the code of the contract if necessary to call another contract then send an internal transaction. Each account in Ethereum is divided by two keys, a private key and a public key. Each account address comes from 20 bytes of a public key which is an important part of every transaction. It is important to understand the difference between a transaction and EOA's private key, the sender of the transaction, and after confirmation of the return of the hash value that we can track all the blockchain transactions. Different sources use call or message conditions for internal transactions .

## 2.3 Smart Contracts

Smart contracts are computer programs that are run through blockchain transactions that can maintain status, interact with cryptocurrency in a decentralized way, and take user input . Smart Contracts are written in the Solidity programming language, which is a mix of C++ and JavaScript. Smart Contract is controlled by peers from the Ethereum organize at regular intervals, and they should be approved by at any rate two different clients to be initiated. From that point forward, contract functions can be executed, and contracts can be imparted to different applicants. To have the option to hold full e-voting, we have to solve the accompanying issues. Straightforwardness, validation and capacity are required in the voting stage. We have to

guarantee that everybody who goes to the election is genuine individuals and  
utilizations the right  
qualifications that we know in the electronic condition, and we should have the option to  
prove  
this whenever, we likewise need our election to be 100% transparent as we wanted.  
Along these  
lines, we have to gather and analyze election information that are signed and  
timestamped.  
Because nobody should have the option to change the vote after the vote is thrown.  
Likewise, we  
need independence in election, with the goal that nobody can pick other candidates.  
These issues  
can be overwhelmed by utilizing distributed blockchain technology. We can  
characterize self-  
executable smart contracts that are required in the blockchain. Just the same as  
composing code,  
we create rules, objects, data models, and in this manner the contracts can start to run.  
After smart  
contracts are deployed, they can't be discharged from the blockchain, and individuals  
can see  
whether the result of the execution of smart contracts are right or not. In Ethereum  
network, there  
is no requirement for focal position to give proof-of-work. All peers can figure contract  
results  
without interference. Indeed, even the Ethereum network can likewise give its very  
own  
figurings.

The use of the Ethereum network in fact for testing experimental software related to the  
development of new smart contracts is quite expensive (because it requires the  
expenditure of  
several ethereum coins) and does not need to occupy large memory in the system.  
Therefore,

Ethereum's private network was created and made available to developers to enable them to test

their software without worrying about actual network congestion.

Ethereum enables developers to utilize smart contracts that will be executed when an

occasion is activated. In this exploration smart contracts are interpreted by the Ethereum web3.js

API and are utilized to control site pages. Clients connect with the Ethereum organize through

Metamask, which is a Chrome extension that can interface with the Ethereum wallet. A client

with cryptocurrency wallet and Metamask extension in their program can interface with the

application and send or get coins. This makes programming on blockchain extremely Conceivable.

## CHAPTER 3

### RESULTS AND DISCUSSIONS

#### 2.4 E-Voting

E-voting systems have many advantages compared to conventional voting systems. An e-

voting system requires less time, energy and costs compared to conventional voting systems. This

system eliminates the possibility of invalid or doubtful sounds so the results are far more

accurate. This system is also very environmentally friendly because it saves a lot of trees which

are supposed to be ballot papers. The e-voting system can also prevent the manipulation or sale

and purchase of votes that is common at polling stations. The e-voting system not only modernizes the electoral process but also has the potential to increase interaction between citizens

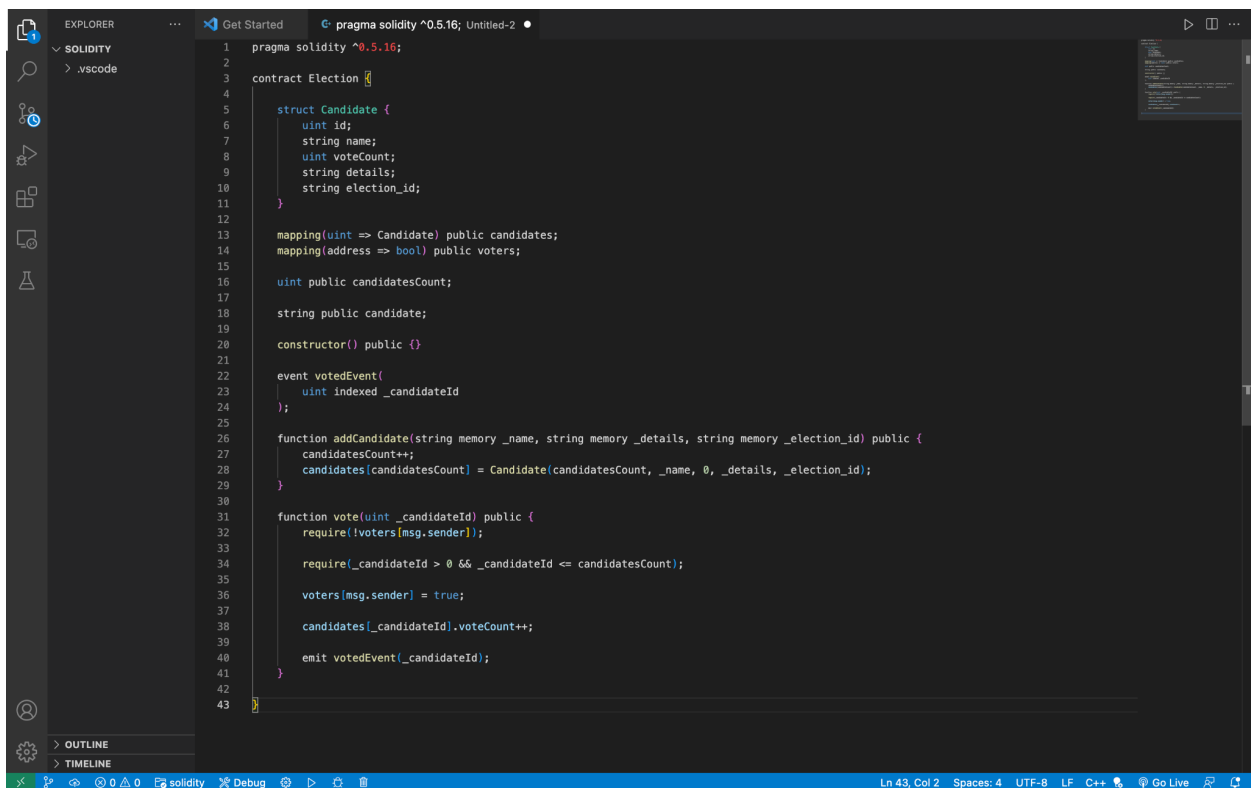
and government, through platforms based on information and communication technology (ICT).

Sophisticated security measures must be put forward to overcome the lack of e-voting, inherent

transparency and to increase confidence in the new system for both voters and electoral authorities [10].

The e-voting system also has some disadvantages compared to conventional voting

systems. Not everyone will agree with e-voting. Some people will be too afraid to think ahead using more technologically advanced systems. The community will tend to think superficially so that this system will be difficult to implement even though it has far more benefits than conventional systems. Especially for the elderly will be difficult to understand because their youth occurs when traditional times are not familiar with technology. The security of the e-voting system is also questionable because it is very difficult to make a system that is completely safe from hacker attacks.



```
1  pragma solidity ^0.5.16;
2
3  contract Election {
4
5      struct Candidate {
6          uint id;
7          string name;
8          uint voteCount;
9          string details;
10         string election_id;
11     }
12
13     mapping(uint => Candidate) public candidates;
14     mapping(address => bool) public voters;
15
16     uint public candidatesCount;
17
18     string public candidate;
19
20     constructor() public {}
21
22     event votedEvent(
23         uint indexed _candidateId
24     );
25
26     function addCandidate(string memory _name, string memory _details, string memory _election_id) public {
27         candidatesCount++;
28         candidates[candidatesCount] = Candidate(candidatesCount, _name, 0, _details, _election_id);
29     }
30
31     function vote(uint _candidateId) public {
32         require(!voters[msg.sender]);
33
34         require(_candidateId > 0 && _candidateId <= candidatesCount);
35
36         voters[msg.sender] = true;
37
38         candidates[_candidateId].voteCount++;
39
40         emit votedEvent(_candidateId);
41     }
42
43 }
```



```
EXPLORER  ...  Get Started  pragma solidity ^0.5.16; Untitled-2  project (1).sol X
SOLIDITY
> .vscode
Users > harshita0585.be21chitkara.edu.in > Downloads > project (1).sol
1 // SPDX-License-Identifier: GPL-3.0
2
3 pragma solidity >=0.7.0 <0.9.0;
4
5 contract Ballot {
6
7     struct Voter {
8         uint weight; // weight is accumulated by delegation
9         bool voted;  // if true, that person already voted
10        address delegate; // person delegated to
11        uint vote;    // index of the voted proposal
12    }
13
14    struct Proposal {
15        // If you can limit the length to a certain number of bytes,
16        // always use one of bytes1 to bytes32 because they are much cheaper
17        bytes32 name; // short name (up to 32 bytes)
18        uint voteCount; // number of accumulated votes
19    }
20
21    address public chairperson;
22
23    mapping(address => Voter) public voters;
24
25    Proposal[] public proposals;
26
27    /**
28     * @dev Create a new ballot to choose one of 'proposalNames'.
29     * @param proposalNames names of proposals
30     */
31    constructor(bytes32[] memory proposalNames) {
32        chairperson = msg.sender;
33        voters[chairperson].weight = 1;
34
35        for (uint i = 0; i < proposalNames.length; i++) {
36            // 'Proposal(...)' creates a temporary
37            // Proposal object and 'proposals.push(...)'
38            // appends it to the end of 'proposals'.
39            proposals.push(Proposal({
40                name: proposalNames[i],
41                voteCount: 0
42            }));
43        }
44    }
45
46    /**
```

```
EXPLORER  ...  Get Started  pragma solidity ^0.5.16; Untitled-2  project (1).sol X
SOLIDITY
> .vscode
Users > harshita0585.be21chitkara.edu.in > Downloads > project (1).sol
46
47    /**
48     * @dev Give 'voter' the right to vote on this ballot. May only be called by 'chairperson'.
49     * @param voter address of voter
50     */
51    function giveRightToVote(address voter) public {
52        require(
53            msg.sender == chairperson,
54            "Only chairperson can give right to vote."
55        );
56        require(
57            !voters[voter].voted,
58            "The voter already voted."
59        );
60        require(voters[voter].weight == 0);
61        voters[voter].weight = 1;
62    }
63
64    /**
65     * @dev Delegate your vote to the voter 'to'.
66     * @param to address to which vote is delegated
67     */
68    function delegate(address to) public {
69        Voter storage sender = voters[msg.sender];
70        require(!sender.voted, "You already voted.");
71        require(to != msg.sender, "Self-delegation is disallowed.");
72
73        while (voters[to].delegate != address(0)) {
74            to = voters[to].delegate;
75
76            // We found a loop in the delegation, not allowed.
77            require(to != msg.sender, "Found loop in delegation.");
78        }
79        sender.voted = true;
80        sender.delegate = to;
81        Voter storage delegate_ = voters[to];
82        if (delegate_.voted) {
83            // If the delegate already voted,
84            // directly add to the number of votes
85            proposals[delegate_.vote].voteCount += sender.weight;
86        } else {
87            // If the delegate did not vote yet,
88            // add to her weight.
89            delegate_.weight += sender.weight;
90        }
91    }
92
93    /**
```

```
92  /**
93   * @dev Give your vote (including votes delegated to you) to proposal 'proposals[proposal].name'.
94   * @param proposal index of proposal in the proposals array
95   */
96  function vote(uint proposal) public {
97      Voter storage sender = voters[msg.sender];
98      require(sender.weight != 0, "Has no right to vote");
99      require(!sender.voted, "Already voted.");
100     sender.voted = true;
101     sender.vote = proposal;
102
103     // If 'proposal' is out of the range of the array,
104     // this will throw automatically and revert all
105     // changes.
106     proposals[proposal].voteCount += sender.weight;
107 }
108
109 /**
110 * @dev Computes the winning proposal taking all previous votes into account.
111 * @return winningProposal_ index of winning proposal in the proposals array
112 */
113 function winningProposal() public view
114     returns (uint winningProposal_)
115 {
116     uint winningVoteCount = 0;
117     for (uint p = 0; p < proposals.length; p++) {
118         if (proposals[p].voteCount > winningVoteCount) {
119             winningVoteCount = proposals[p].voteCount;
120             winningProposal_ = p;
121         }
122     }
123 }
124
125 /**
126 * @dev Calls winningProposal() function to get the index of the winner contained in the proposals array and then
127 * @return winnerName_ the name of the winner
128 */
129 function winnerName() public view
130     returns (bytes32 winnerName_)
131 {
132     winnerName_ = proposals[winningProposal()].name;
133 }
134
135
136
```

3. Results and Analysis

This research requires data of each user who will vote. The data needed is only limited to the private key because the writer focuses only on voting functionality. The private key will later

be used as a unique key for each account. Here is the data of each user.

Table 1. User Account

No
User Account
Private Key
1
Account 1

fbd0da79152ad9a274485edb74387f28d6fc4e3192c45c6b8d6d45e21c8e1  
4f9

2

Account 2

87c8d0b69944a6d225114581a723e18a84545e14033b5a1c4a3252f42be1  
2fcf

3

Account 3

5db6f783aa0677127895aec30575159d6cbab320675128eda8ee462f556a  
8128

4

Account 4

1cfe9bcaa9bfaf441cf2e7f50aedbd2097c5400a471ebff48b50375b5abe31  
dc

5

Account 5

1f97e938e642dc4c774dc33e0c9539e3e00dc304801fa3df2876c98500f92f  
e3

6

Account 6

142a6ce9dc57b672fb32f531c92c768b4dc16bdd5b703139253ca625ae83  
43ba

7

Account 7

8b65e2a096bf66fc9f8b6868cf6fdcfc45ed77a58d441882009d4649208ac2  
40

8

Account 8

9959b7941a57df4bd4b98c3b118f9c81244b13a0d37c4b016263d15537b3  
4190

9

Account 9

6e36a7639b84a01a6e8dcc0a63b3717568e385d6107e288b6e464bd3a20b  
e8b2

10

Account 10

d1cada2966277dfcc64c364df1fc83da16f025d20e63d4346b6e6a9d01163  
804

To vote, there are a number of things that must be prepared before the start of voting.

1. Preparation of an internet connection and running application support programs

such as Ganache, Truffle, and Metamask need to be done

2. After that, confirm whether there is already an active account. If not, then

creating a new account must be done first

3. Then after the account has been active, voters can start voting by first selecting

candidates.

### System Evaluation

To conduct the voting process, voters must go through the following stages:

1. Ensure that the account used is correct by checking in Metamask

2. After selecting the correct account, the user can vote directly by selecting the candidate first then pressing the vote button to vote.
3. After pressing the vote button, a new page will appear to confirm that the voter account will send a Gas Fee as a condition for voting. This transaction is a proof that the voting has been done and entered into the blockchain chain.

Figure 4.

4. Then after confirming, then the votes are sent and immediately displayed on the start page in real-time.
5. Users who have voted can no longer vote. After voting, the candidate's choice and vote button are removed from the main page.

## CHAPTER 4

### CONCLUSION

From the design and implementation of the e-voting system based on the Ethereum

Blockchain, conclusions can be drawn including:

1. The e-voting system based on the Ethereum Blockchain can work well.
2. This e-voting system is able to validate the voter's identity well and prevent repeating the election.
3. This e-voting system can store data safely and reliably.
4. By using this electoral system, the voting process will be much faster and safer.
5. The voting process and the calculation of the number of votes will be faster because the voting process is done in real-time.

# CHAPTER 5

## BIBLIOGRAPHY

1. <https://youtu.be/M576WGiDBdQ>
2. <https://blockchain.ieee.org/standards/>
3. Solidity
4. Imran Bashir, 2018, "Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained", 2nd Edition, Packt.
5. <https://www.linkedin.com/learning/learning-bitcoin-and-other-cryptocurrencies-17179887/the-continuing-appeal-of-cryptocurrencies?autoplay=true&u=92961692>

THANK

YOU



