VOTING SYSTEM USING BLOCKCHAIN

A Project Report

Submitted in the partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

Department of Computer Science and Engineering

by

G Satya Siva Ramadas (150030288)
M Sai Charan Tej (150030544)

Under the Supervision of
K. Sripath Roy
Assistant Professor

KONERU LAKSHMAIAH EDUCATION FOUNDATION,
Green Fields, Vaddeswaram- 522502, Guntur (Dist),
Andhra Pradesh, India.

November,2018.

KONERU LAKSHMAIAH EDUCATION FOUNDATION
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Declaration

The project Report entitled “Voting System using Blockchain” is a record of bonafide work of G Satya Siva Ramadas (150030288) and M Sai Charan Tej (150030544), submitted in partial fulfilment for the award of Bachelor of Technology in Computer Science and Engineering during the academic year 2018-19.

We also declare that this report is of our own effort and it has not been submitted to any other university for the award of the degree.

G Satya Siva Ramadas (150030288)
M Sai Charan Tej (150030544)

KONERU LAKSHMAIAH EDUCATION FOUNDATION
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that the Project Report entitled **“Voting System using Blockchain”** is being submitted by G Satya Siva Ramadas (150030288) and M Sai Charan Tej (150030544) in partial fulfilment for the award of Bachelor of Technology in Computer Science and Engineering during the academic year 2018-19.

Signature of the Supervisor
K. Sripath Roy
Assistant Professor

Signature of the HOD
V. Hari Kiran
Head of the Department

Signature of the
EXTERNAL EXAMINER

ACKNOWLEDGEMENT

This research project would not have been possible without the support of many people. We are very grateful to them in making our project a successful one. We the entire group members would like to acknowledge the advice, encouragement and guidance of them.We would like to place on record the deep sense of gratitude to the honourable **Prof. L. S. Reddy**, Vice Chancellor, Koneru Lakshmaiah Educational Foundation for providing the necessary facilities to carry on the project with ease of access.

We express our gratitude to Prof. **K. Subba Rao**, Principal, Koneru Lakshmaiah Educational Foundation for inspiring with his words filled with dedication and discipline towards work.

We would like to thank the head of the department, Prof. **V. Hari Kiran**, Professor in Computer Science and Engineering department for his valuable suggestions and his guidance for the timely completion of the project.

It is our immense pleasure to acknowledge our profound sense of gratitude to our project coordinator **G. Pradeepini**, Professor in Computer Science and Engineering department for her inspiring guidance, comments, suggestions and encouragement throughout this project.

Furthermore, we are also very thankful to our guide **K. Sripath Roy**, Assisstant Professor in Electronics and Communication Engineering for his valuable advice and motivation throughout the completion of this project. His technical knowledge was an invaluable asset, but may be more importantly, he has helped to keep us focused and pointed in the right direction.

G Satya Siva Ramadas (150030288)
M Sai Charan Tej (150030544)

ABSTRACT

The project is aimed to design an Voting System using Blockchain. The core idea depicted in this is to combine the block-chain technology with secret sharing scheme and homographic encryption in order to realize the voting application without a trusted third party. As technology has positive impacts on many aspects of our social life, designing a 24 hour globally connected architecture enables ease to a variety of resources and services. It provides a public and transparent voting process while protecting the anonymity of voter’s identity, the privacy of data transmission and verification of ballots during the billing phase thereby making the result of votes highly confidential.

Table of Content

Chapter – 1

Introduction

1.1 Voting System

Voting plays an prominent role in construction of a well democratic society. Voting in olden days are done in such a way that the voters are appointed with some polling stations to cast their vote in and it usually involves more and more expenditure and time and cost along with huge budget.

`The most general and common method of voting followed all over the world in olden days is Voting system using pen and paper based. With the usage of pen and paper based voting system many problems got araised thereby reducing the security to the votes and making the voters to loose the hope over the method of voting. In the Pen and Paper based Voting system votes are counted and calculated based on the amount of papers dropped and filled by the voters in the box present at the polling booths.

Inorder to reduce this problem new technique has came into existence in the early 20th century i.e., Electronic-voting system.It is a new substantial online voting system which is based mainly on the concept cryptography. When compared to the old traditional voting, e-voting is a efficient economic system which addresses on transparency and impartiality.In the method e-voting the voter casts a ballot through a digital system instead of using the paper.The votes are calculated based on the EVM(Electronic Voting Machine) count.The results will be counted automatically and anonymously.This made the voters to trust on the E-voting.After launching of this technique it has been gradually implemented and emphasised by people and attracted many organizations to follow this technique.

But due to the involvement of anti social elements and hackers over the e-voting system many issues have been raised so that problems have provoked again. Ballot System is offering anonymity to the voter but the process of counting the votes is not so transparent. People generally trust the result which is given and provided by an Election commission (or) government. Many and major vulnerability in the current process again made the voters to loose their hope on elections. Scams that are mainly prevailing now-a-days are voter fraud, ballot stuffing and capturing of booths.All these make the e-voting system as corrupted and confused voting system.

For reducing the problem facing in the current e-voting system the new trending and advanced technique “Blockchain” is implemented. It not only gives result accuratley but also secures the votes making the voting system as decentralized one. It doesn’t depend on other governing body and won’t encourage corruption.

1.2 Basic properties of Voting System:

1) **Privacy of the Ballot** simply means that no other third party will know to whom the voter opted his/her vote for. Securing the votes in a highly encrypted format such that no human can decrypt the format is also done in this.
2) **Verifiability** of each and every individual is carried inorder to reduce the conflict of votes being fraud and thus the individual voter can get the possibility to verify the vote.
3) **Individual Eligibility** is the main criteria behind the voting system. From this only legal candidates can enroll in the voting event. Other illegal candidates get easily noticed with help of this property.
4) **Accuracy** will define the efficiency of the voting system using the concept blockchain and how it will secure the votes.
5) **Fairness and Uniqueness** is the primary reason and hidden secret of the voting system which will give a chance to each and every voter.

6) **Robustness** simply refers to the fact that creating a blockchain with the votes is very hard to modify and impossible to corrupt or change.No influence will have an effect over the votes.

7) **Receipt less voting** is the primary and unique concept that will wonder the voters. Using this concept the voters can't receive or build any new receipt after his/her voting.

1.3 Process included in the Voting System using Blockchain:

1) Registering for the vote.
2) Voting process.
3) Verifying the vote.
4) Counting votes.
5) Re-counting votes.
6) Check for invalid and valid votes.
7) Results declaration.

1.4 Advantages of Voting using blockchains includes:

1)Greater and flexible transparency due to open and distributed ledgers.

2)When comes to the concept of anonymity attaining of inherent anonymity is possible with this.

3)In the view of Security and Reliability using blockchain the problem of being threatened gets reduced.

4)Possibility of occurrence of immutability (strong integrity for the voting scheme and individual votes).

5)With the flexibility and accessibility of voting system , voter participation will get increased enormously.

6)Ability to reduce fraud, by eliminating the opportunity for ballot tampering.

7)Independent of location it is very convenient to use by the Election Commission of India.

8)Votes counting is very easy and convenient to the authorities.

9)It may reduce the expenditure of Government.

10)Protects privacy (Ballot information).

Chapter - 2

Literature Survey

2.1 Blockchain Technology

Blockchain is one of the most exciting and emerging technologies in the current world. “Bitcoin” which is a cryptocurrency is the buzzword which made the world take notice of the Blockchain technology. Blockchain has been redefining the way of storage, updating, and moving data across networks. Blockchain technology is highly secure and trustworthy. The core aspect of the technology is that it allows a decentralized and non-hierarchical decision making.

Blockchain technology is actually supported by a very well distributed network which consists of a huge number of interconnected nodes. Each of the nodes that are connected have their own copy of the distributed ledger which contains the overall history of all the transactions that the network has processed. No single authority can control the network. If the majority of the nodes agree, then only they accept the transaction. This network allows users to remain anonymous. A basic analysis of the blockchain technology simply conveys that it is just a suitable basis for e-voting and could have the potential to design the voting system more acceptable and reliable.

Today, the Internet takes a huge part in the way we connect to people in the network, make transactions seamlessly. It has changed the way of money handling and the major bottleneck we face in today’s transactions are –

-Imposing high transaction fees

-Problem of Double Spending

-Fraudulent and Hacking Issues

Literature Survey

Previously, we had to trust the financial institutes and other third parties with our money, not knowing if they are as safe and secured as they claim, but after emergence of Blockchain all these issues can be resolved in a safe, secure and effective manner.

Blockchain is revolutionizing the way of managing data and databases. The traditional database has individual authority that governs it, it is not a decentralized entity like Blockchain.

For example, a bank or any institute which has created a database is owned by them only, they have all access to the database, they can manage and make decisions of storage, deletion, archiving and updating. This can result in two flaws the first one is Single point failure in case of any issue with transactions and second one is Power to only one single central authority.

An agreement between two people occurring over the Internet still requires at least one central authority to approve information. For instance, with a mortgage, banks must approve reserve funds and loans. They can impose charges in case a transaction happens from A to B. Duplication or falsification of transaction can cause double spending problem.

Every central authority imposes a charge of overhead in a mortgage exchange. The transactions in the differed databases takes time, costs money or transaction charges, vulnerable to hacking or can be error prone due to any human intervention. We were subjected to all such difficulties and hassles. The Blockchain solves all these difficulties and challenges.

2.2 Types of Blockchain

Blockchain is Open-Source Technology and is the most Disruptive, Digitized and Decentralized technology. It is majorly used to verify digital currency transactions. In this there is no single brought together specialist however everybody existing in the square can check its genuineness/ authenticity. There are different types of Blockchain –

- Public: Public transactions are done within the existing blocks. For Examples - Bitcoin,Ethereum,Dash,Factom
- Consortium: It is Controlled by a consortium of members. For Examples - Ripple, R3 &Hyperledger1.0
- Private: It Requires an inviation, it means only the Authorised can access it. For Examples - Multichain, Block-Stack.

When Satoshi Nakamoto the founder of Blockchain technology, tried to develop a Peer-to-Peer electronic cash system and gave birth to a Cryptocurrency (Encrypted Currency) known as Bitcoin. It's a digital cash which works as a medium of exchange with in a block. The term is derived since the blocks are been connected one by one like a chain. The Cryptocurrency (Encrypted Currency) uses cryptography to secure and verify transactions which are made by us.

Different types of digital currencies are – Bitcoin, Ether, Ripple, Litecoin, IOTA

Bitcoin – It is the world's first cryptocurrency, it is a first decentralized digital currency. It is the only system that works without a central bank or any other single administrator for transaction in Blocks.

Ether – It is the cryptocurrency of Ethereum. It is a generic open Blockchain platform. The idea proposed was the development of a Turing-complete language unlike bitcoin which turing is incomplete or not completed.

Ripple – It was found by the company 'Ripple', which is a remittance network and currency exchange through RippleNet.

Litecoin – It was forked from the Bitcoin which enables instant payment to anyone in the Block.

IOTA – It developed by IOTA foundation is a open source Distributed Ledger technology to power IOT.

2.3 What is Open Source?

Open Source is referred to any type of programs whose Source Code is made available for the user to use and modification as they use (customization of program or softwares). It's usually developed as a public collaboration and made freely available.

2.4 Advantages of Open Source

As the Source code is made available for user, it offers over proprietary solution for all. These are the few in list:

- Security for Solid Information : As with the reliability, OSS (Open Source Softwares) code is often more secure because it is much more thoroughly reviewed at each and every side of the program and make a careful, critical examination of s/w by the community and any issues that do arise tend to be patched more diligently.
- Lesser Hardware Cost : Open Source Solutions should be more than just free s/w, but the fact that they does not require any licensing cost remains a decisive benifits when they looking at the end cost of deploying a solution.
- Agility, Flexibility and Speed : Open Source Softwares (OSS) actives the technology agility, usually offers multiple ways for problem solving. Open Source Software (OSS) helps us to keep our IT organization from getting blocked for a exact capability is not available from a vendor.

It allows us to get the best of the both worlds reality – agility, flexibility and ability to get a quick start and inexpensive, with the capability to mature to a large scale, high supported and functional, and you do not have to go over proprietary licensing hurdles to reach.

- Attraction of better talent : Open Source enables the enterprise the ability to attract talent. Highly and Most professional techies are well known and aware of Open Source. Most of them enjoy in their own project creation and have an ablity to interact with others who were outside their enterprise to develop their own solutions. By giving flexibility to developers and freedom for using the Open Source gives attracting the better talent.

2.5 Applications and Areas of Blockchain

1. Asset Management : Trade Processing and Settlements (Stock Exchange)

   Traditional trade processes within asset management (where parties exchange and overseas resources) can be costly and dangerous, especially with regards to cross-outskirt exchanges. Each party in the process, such as agent/broker, custodian, or the settlement administrator, keeps their own records which makes significant wasteful/inefficiencies aspects and space for error. The blockchain ledger decreases error by encoding/encrypting the records. At the same time, the ledger simplifies the process, while canceling the need for intermediaries.

2. Insurance : Claims Processing (Insurance Policy)

   Claims processing can be a frustrating, disappointing and unpleasant method. Insurance processors have to swim through false cases, fragmented data sources, or abandoned policies for users to state a few and process these forms manually. Room for error is huge. The blockchain provides a perfect system for risk-free management and transparency. Its encryption properties enables safety net providers to capture the responsibility for to be protected.

3. Payments : Cross-Border Payments (Banks)

   The worldwide payments sector is error-prone, exorbitant, and open to illegal tax avoidance. It takes days if not longer for cash to cross the world. The blockchain is as of now giving arrangements settlement organizations, for example, Abra, Align Commerce and Bitspark that offer end – to - end blockchain fueled settlement administrations and services. In 2004, Santander wound up one of the primary banks to consolidate blockchain to an installments application, empowering clients to make universal installments 24 hours every day, while clearing the following day.

4. Smart Property (Land Registration)

   A substantial or elusive property, for example, autos, houses, or cookers, from one perspective, or licenses, property titles, or organization shares, on the other, can have savvy innovation installed in them. Such enlistment can be put away on the record alongside legally binding points of interest of other people who are permitted proprietorship in this property. Shrewd keys could be utilized to encourage access to the allowed party. The record stores and permits the trading of these brilliant keys once the agreement is confirmed.

   The decentralized ledger also becomes a system for recording and managing property rights as well as enabling the smart contracts to be duplicated if records or the smart key is lost.

   Making property savvy diminishes your dangers of running into misrepresentation, intervention expenses, and sketchy business circumstances. In the meantime, it expands trust and productivity.

5. Hospitals/Healthcare (Medical Reports)

   Personal health records could be encoded and stored on the blockchain with a private key which would grant access only to specific individuals. The same strategy could be used to ensure that research is conducted via

HIPAA laws (in a secure and confidential way). Receipts of surgeries could be stored on a blockchain and automatically sent to insurance providers as proof-of-delivery. The ledger, too, could be used for general health care management, such as supervising drugs, regulation compliance, testing results, and managing healthcare supplies.

6. Passport and Personal Identification

The primary advanced international ID propelled on Github in 2014 and could enable proprietors to recognize themselves on the web and off. How can it function? You take an image of yourself, stamp it with an open and private key, the two of which are encoded to demonstrate it is authentic. The visa is put away on the record, given a Bitcoin address with an open IP, and affirmed by Blockchain users.We convey a scope of distinguishing pieces of proof: Our driver's permit, PC secret phrase, character cards, keys, government disability ID, et cetera. Blockchain ID is an advanced type of ID that is designed to supplant every one of these types of physical recognizable proof. Later on, fintech researchers say you'll have the capacity to utilize the one advanced ID for joining at any recorder. It is open source, anchored by the blockchain, and secured by a record of straightforward record.The primary advanced international ID propelled on Github in 2014 and could enable proprietors to recognize themselves on the web and off. How can it function? You take an image of yourself, stamp it with an open and private key, the two of which are encoded to demonstrate it is authentic. The visa is put away on the record, given a Bitcoin address with an open IP, and affirmed by Blockchain users.We convey a scope of distinguishing pieces of proof: Our driver's permit, PC secret phrase, character cards, keys, government disability ID, et cetera. Blockchain ID is an advanced type of ID that is designed to supplant every one of these types of physical recognizable proof. Later on, fintech researchers say you'll have the capacity to utilize the one advanced ID for joining at any recorder. It is open source, anchored by the blockchain, and secured by a record of straightforward record.

7. Birth, wedding, and death Certificates

   Hardly any things could easily compare to archives demonstrating you're conceived, hitched, passed on which open your rights to a wide range of benefits, (for example, casting a ballot, working, citizenship), yet blunder is overflowing. Up to 33% of youngsters younger than five have not been issued a birth declaration, the UNICEF announced in 2013. The blockchain could make record-keeping more solid by encoding birth and passing confirmation and enabling natives to get to this urgent data

It’s vital to take note that for the blockchain to work, the hub to hub or node to node organize must be propelled and consent to work under moral benchmarks. Once, and just if, these guidelines and clung to, the blockchain could turn into an amazing apparatus for enhancing business, leading reasonable exchange, democratizing the worldwide economy, and helping bolster more open and reasonable social orders.

Chapter – 3

Theoretical Analysis

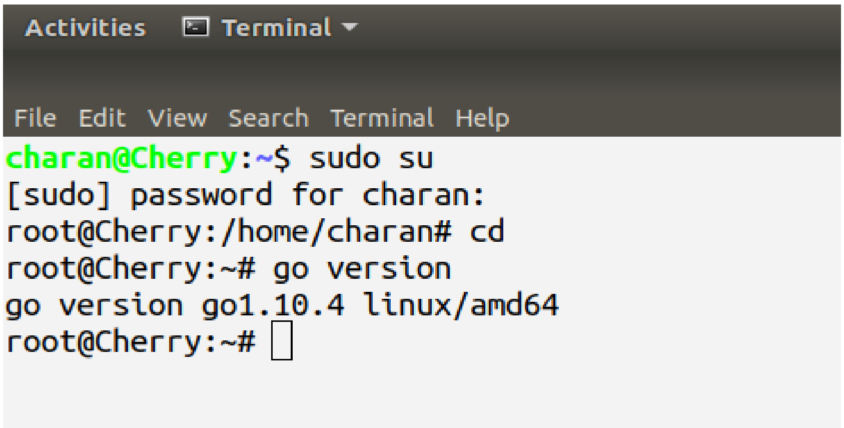
3.1 What is GoLang?

Go Lang (Go Language) is a new, Open Source programming language developed by Google. GoLang is a compiled language. Its a concurrent, garbage-collected, and build fast at scale. It makes easy to build simle, reliable and efficient software. GoLang has become a go to language for developing the decentralized systems. In all other organisation is using the GoLang for their core processing modules, and it also grab and gained a lot of attraction in web development.

3.2 Why Go Lang for Blockchain?

In these days most of the stable blockchain based Dapps and Tools are built using GoLang. All the required packages and libraried are very easy to find.

Because of its a compiled language so that it runs directly with the Operating System (OS). This helps/allows us to build technology like Ethereum Virtual Machine (EVM) in a much efficent way. By using the Go, it gives the feeling of a scripting language and it also has a low startup time, so that it is great for small programs. When compare to Java the Queries per second (QPS) is much better in Go. Hence, GoLang is used to built system to handle high volumes of requests.



```
charan@Cherry:~$ sudo su
[sudo] password for charan:
root@Cherry:/home/charan# cd
root@Cherry:~# go version
go version go1.10.4 linux/amd64
root@Cherry:~# 
```

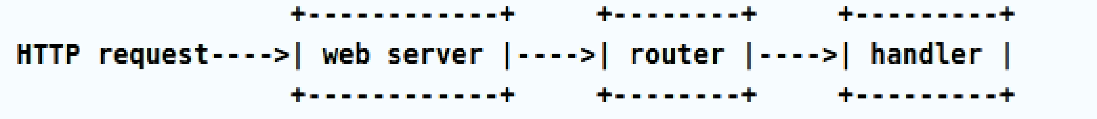
Fig:3.2.1 GoLang version installed in root

After the GoLang installation we need install some more packes like go-spew, gorilla mux, godotenv.

We use the SHA-256 encryption algorithm as a key for the user without involvement of any third party, so that each user will have an unique key which is encrypted. While a particular user accessing the data the encrypted key should match with the data key which is stored in the database.

Go-spew/spew : It implements a profffound pretty printer for Go data structure to help in debugging. A thorough suite of tests with 100% test coverage is given to ensure proper functionality. Go-spew is licensed under the liberal ISC license, so it may be used in open sourece or business ventures.

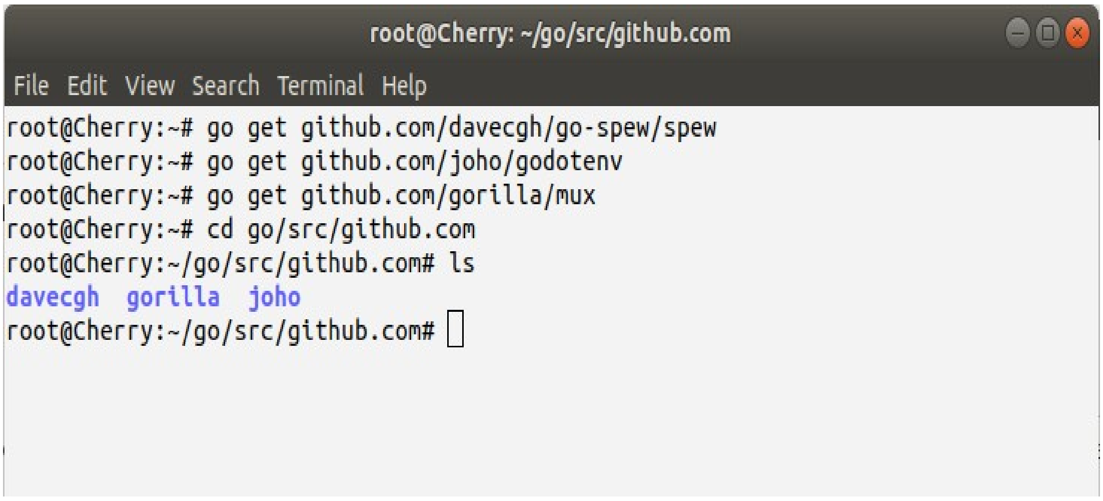
Gorilla/mux : Its a popular package for writing web handlers. It also provvides request routing, validation and other servies. The structure type http: ServeMux is included in Go networking library.



```
                     +------------+     +--------+     +---------+
HTTP request---->| web server |---->| router |---->| handler |
                     +------------+     +--------+     +---------+
```

Fig: 3.2.1 URL request

joho/godotenv : Godotenv gives us a chance to peruse from an .env file that we keep in the root directory so we don’t need to stress things like http ports.


root@Cherry: ~/go/src/github.com
File Edit View Search Terminal Help

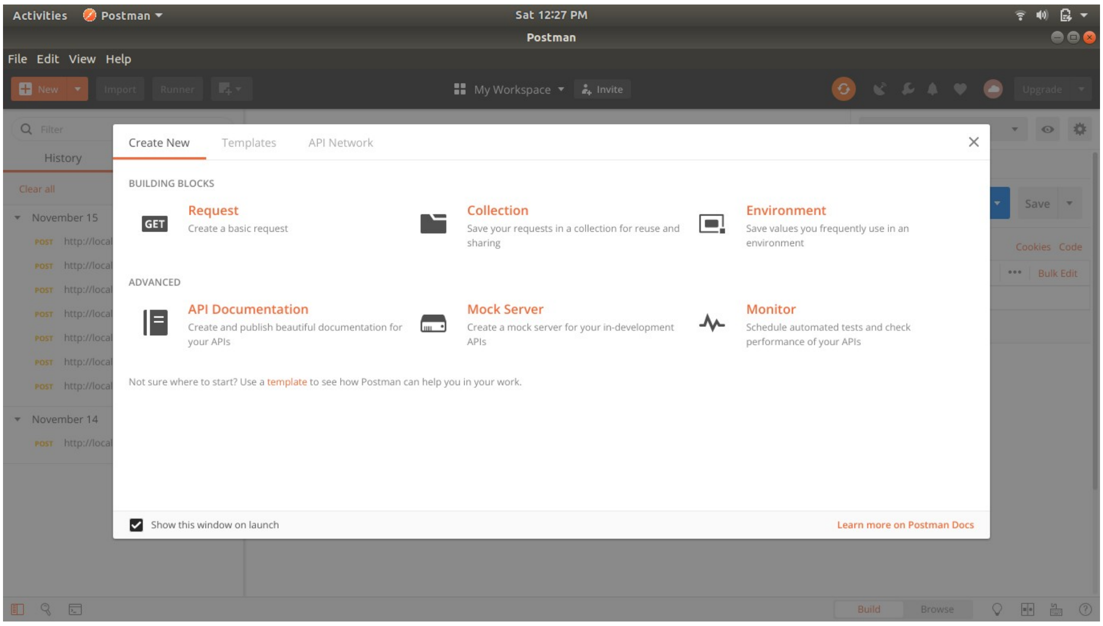
```
root@Cherry:~# go get github.com/davecgh/go-spew/spew
root@Cherry:~# go get github.com/joho/godotenv
root@Cherry:~# go get github.com/gorilla/mux
root@Cherry:~# cd go/src/github.com
root@Cherry:~/go/src/github.com# ls
davecgh  gorilla  joho
root@Cherry:~/go/src/github.com# 
```

Fig:3.2.2 Installation of the required packages

After the installtion of the package we need to create a file label with only '.env' which contains the address of the port as ADDR=8086 or what ever (8080,8081).

Now all we need a local host so that we need a tool called "Postman". Its a great tool for prototyping APIs and comes with some powerful testing features. It tests into your build automation to make it best and attactive.

Postman tool is a Google Chrome's app for communication with APIs and has a friendly GUI for constructing requests and reading responses.



Create a basic request

Save your requests in a collection for reuse and sharing

Save values you frequently use in an environment

Create and publish beautiful documentation for your APIs

Create a mock server for your in-development APIs

Schedule automated tests and check performance of your APIs

Not sure where to start? Use a template to see how Postman can help you in your work.

☑ Show this window on launch

Learn more on Postman Docs

Fig:3.2.3 Initializing the Postman tool

We need to pull the request and initialize the port which we had already given in the '.env' file (i.e., ADDR=8086)

Now open the terminal go to root directory and configure the Go path. Now run the main file by the following command "go run filename.go", so we can see the port is listening with the null block which is created by default while running the code.

3.3 How Blockchain works?

Everyblock has its own number and while creating a block it takes the time stamp of block when created, its also contain some data stored in BPM (beats per minute) as message/data for the block every block is connected with Hashvalues.

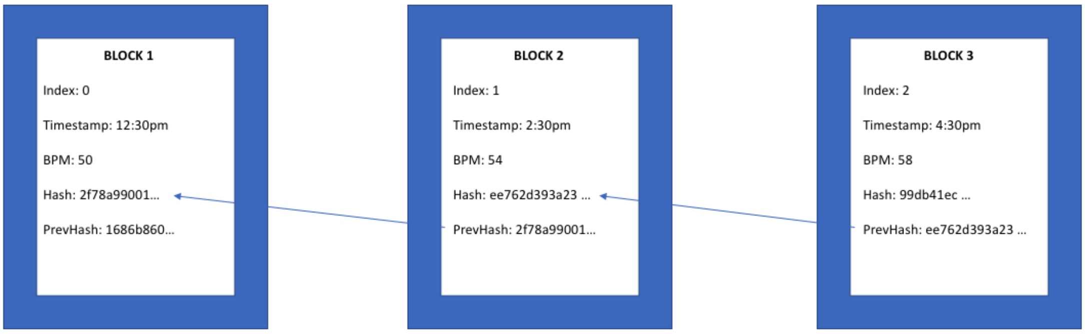
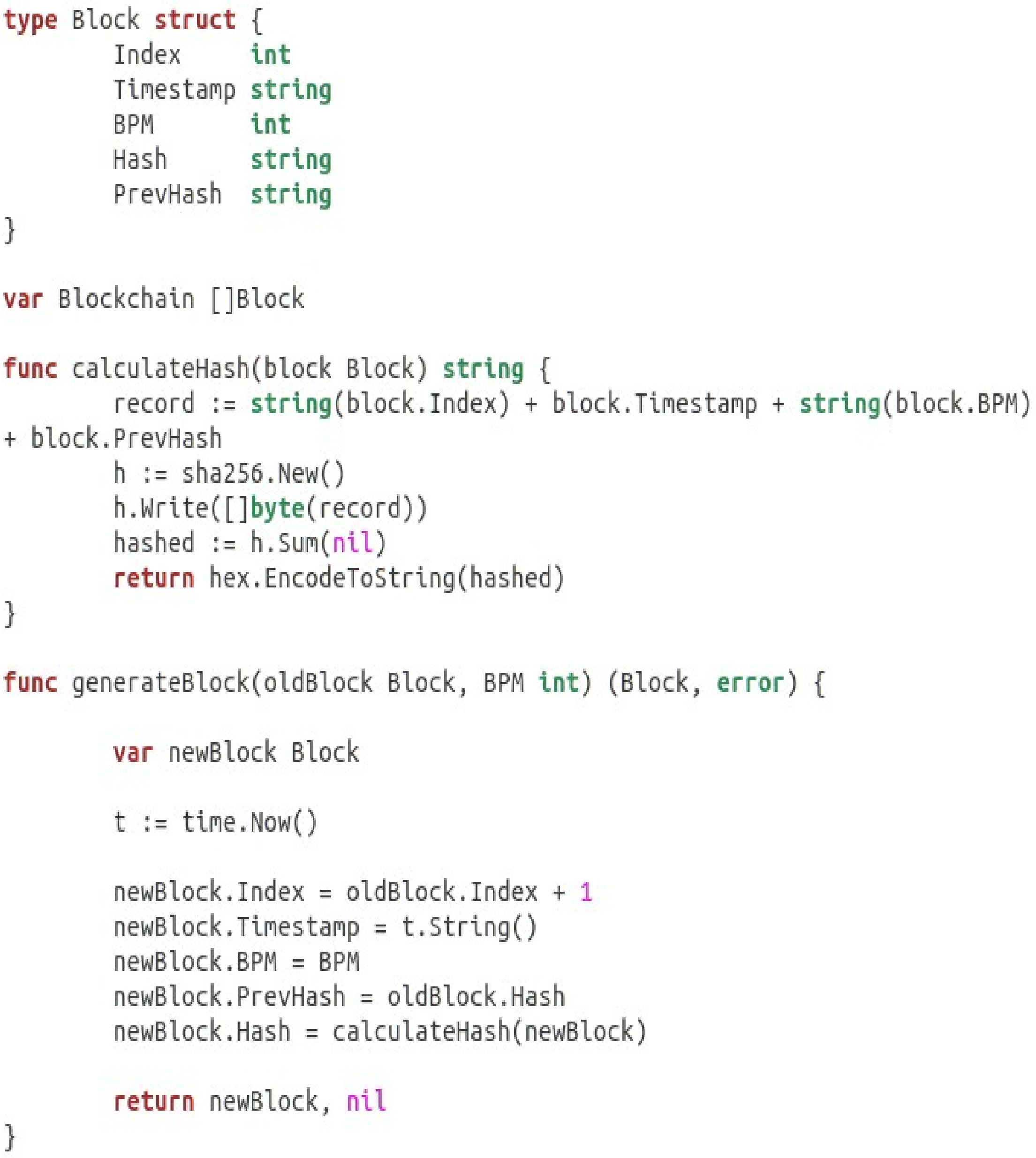


Fig: 3.3.1 Block creation and its interconnections by Hashvalues

By specifying the block structure for every individual block as Index, Timestamp, BPM (beats per minute), Hash, PrevHash. Assigning all variables with their appropriate data types. Initializing the block and calculating the hash values we will generate the block by assigning the Block struct. We will validate the block based on the hash values and followed by index numbers.

GenesisBlock is the most essential piece of the primary capacity. We have to supply our blockchain with an underlying square, or else another square won't have the capacity to contrast its past hash with anything, since a past hash doesn't exist.

We segregate the beginning square into its very own go routine so we can have a partition of worries from our blockchain rationale and our web server rationale. This will work without the go routine yet it's simply cleaner along these lines.

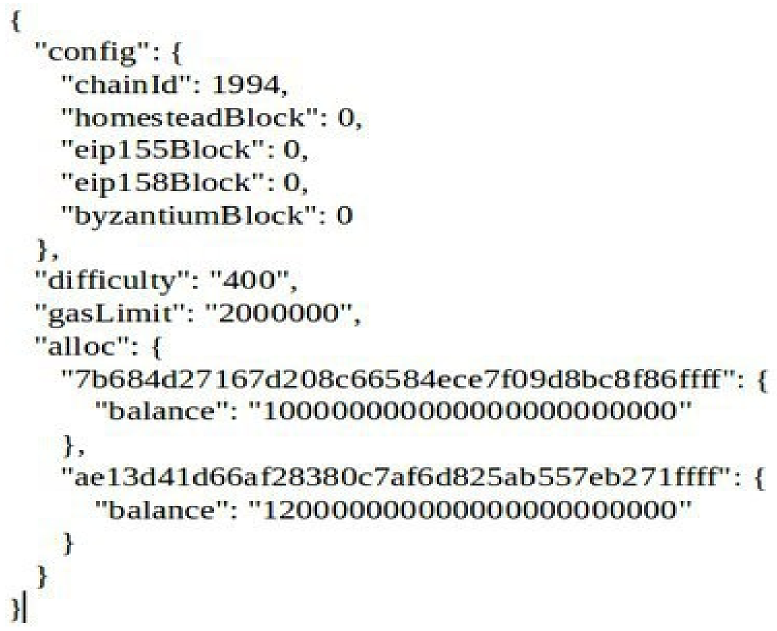


```
type Block struct {
        Index     int
        Timestamp string
        BPM       int
        Hash      string
        PrevHash  string
}

var Blockchain []Block

func calculateHash(block Block) string {
        record := string(block.Index) + block.Timestamp + string(block.BPM)
+ block.PrevHash
        h := sha256.New()
        h.Write([]byte(record))
        hashed := h.Sum(nil)
        return hex.EncodeToString(hashed)
}

func generateBlock(oldBlock Block, BPM int) (Block, error) {

        var newBlock Block

        t := time.Now()

        newBlock.Index = oldBlock.Index + 1
        newBlock.Timestamp = t.String()
        newBlock.BPM = BPM
        newBlock.PrevHash = oldBlock.Hash
        newBlock.Hash = calculateHash(newBlock)

        return newBlock, nil
}
```

Fig: 3.3.2 Block Structure and Block Generation

When comes to networking of blockchain there are many decentralized servers but only in paid version like Amazon CE2. Networking with the blockchain is very critical part in the blockchain technology because when we are using the decentralized servers, if any one of the users data is modified it will be created as a new block so the server will push the request to all other servers which are connected to it.

This is sample genesis algorithm, we can write our own genesis file so that due to any reasons if the chain breaks we can synchronize the missing or lost blocks using this genesis file.



```
{
  "config": {
    "chainId": 1994,
    "homesteadBlock": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0
  },
  "difficulty": "400",
  "gasLimit": "2000000",
  "alloc": {
    "7b684d27167d208c66584ece7f09d8bc8f86ffff": {
      "balance": "100000000000000000000000"
    },
    "ae13d41d66af28380c7af6d825ab557eb271ffff": {
      "balance": "120000000000000000000000"
    }
  }
}
```

Fig: 3.3.3 Genesis Structure

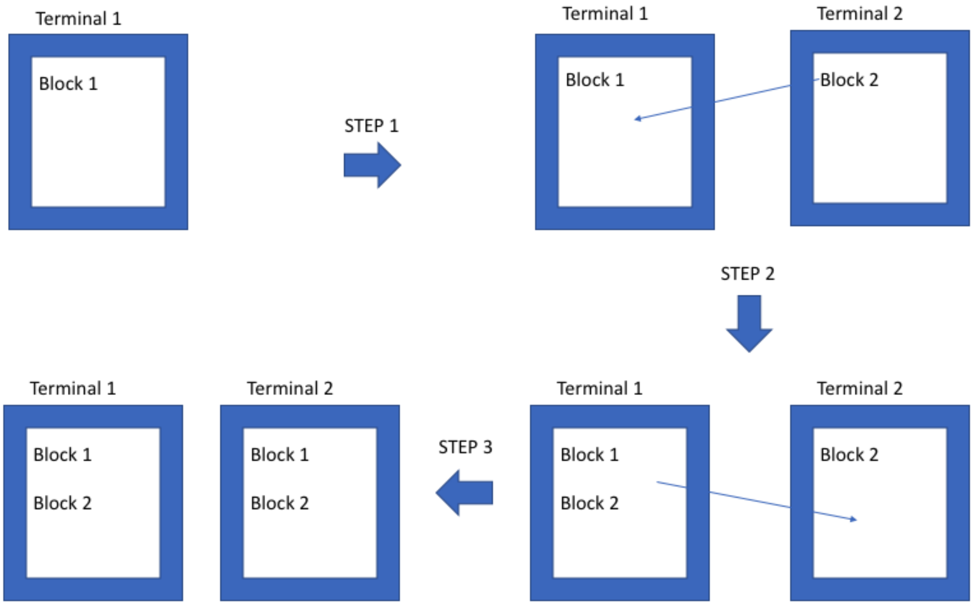


Fig: 3.3.4 Connections between Servers

3.4 About Postman

The one and only complete and flexible API development environment used by developers to establish and support entire Workflow.

Generally modern coding practices use IDE(Integrated Development Environment) which provides comprehensive and flexible facilities for software development.IDE normally comprises of Source Code Editor,Build Automation Tools and Debugger.

Inorder to organize the work and manage collaboration with the rest of API new concept,ADE(API Development Environment) is designed.ADE is a platform that supports and enhances API development.A good ADE will create a single source and streamlie the development process thereby enhacing the collaboration on API's across the organization.

ADE comprises of Collections,Workspaces and Built-in Tools.

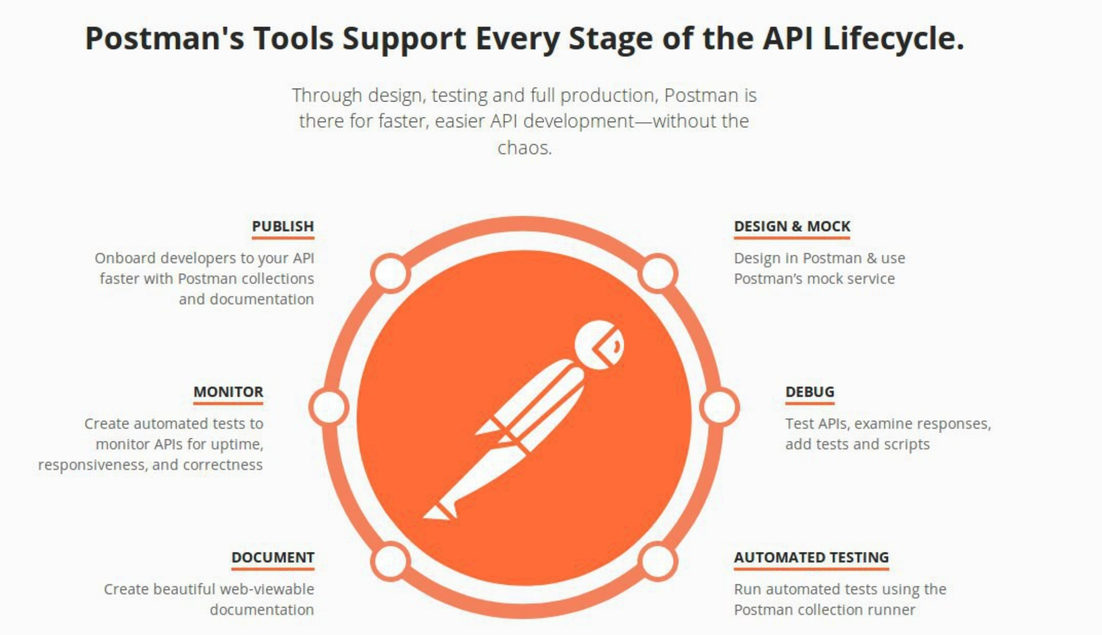
Postman Environment is a convenient way to share team's server configurations, replicate user credentials for testing and to hide personal secrets.It is also used to backup and sync data to the cloud.BY the Postman mock server we can simulate a server response before building out a real point.

SYSTEM REQUIREMENTS:

1)Linux OS (x64 bit)

2)Go version 1.11.1 linux/amd64

3)Postman tool for linux platform



Postman's Tools Support Every Stage of the API Lifecycle.

Through design, testing and full production, Postman is there for faster, easier API development—without the chaos.

PUBLISH

Onboard developers to your API faster with Postman collections and documentation

DESIGN & MOCK

Design in Postman & use Postman's mock service

MONITOR

Create automated tests to monitor APIs for uptime, responsiveness, and correctness

DEBUG

Test APIs, examine responses, add tests and scripts

DOCUMENT

Create beautiful web-viewable documentation

AUTOMATED TESTING

Run automated tests using the Postman collection runner

Fig: 3.4.1 All about Postman tool

3.5 Why Postman?

1) It is so easy to create test suites in Postman than any other tools for Integration Testing.

2)For storing the information for running tests in different environments.

3)For easily backup and sync data to the cloud and use in other tests.

4)For sharing tests and environments to the code repositories by encapsulating collections.

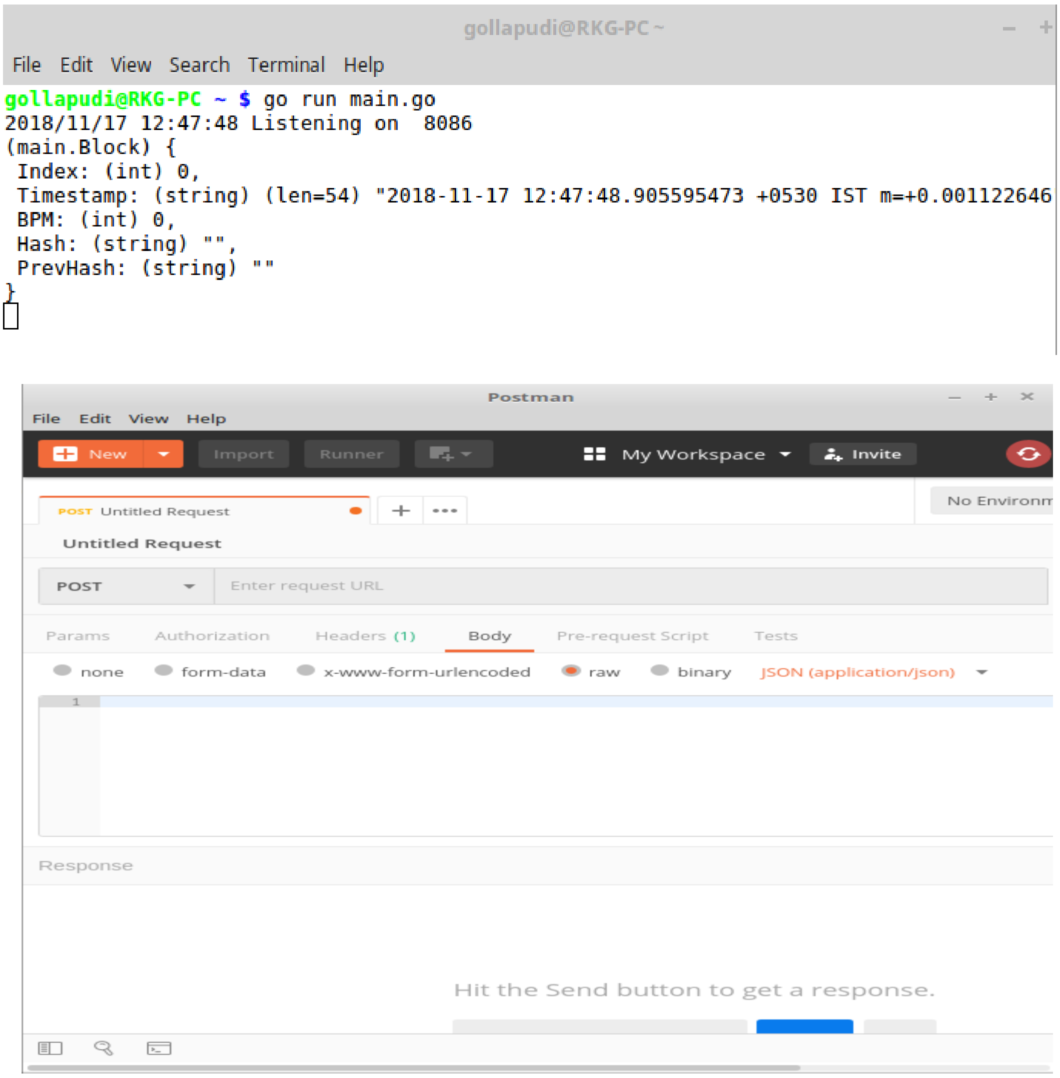
5)It is the best and powerful tool for performing integration testing that allows repeatable and reliable tests which will be useful for both front-end and back-end developers.

Chapter – 4

Experimental Investigation

4.1 Experimental Analysis

When we run the code the request is initiated to the port and it starts listening to the specified port which we have declared in the '.env' file. At the initial stage the block contains all NULL values except the index and timestamp. When ever we add block or give the data in the BPM (beats per minute), all the values and size of the block changes.



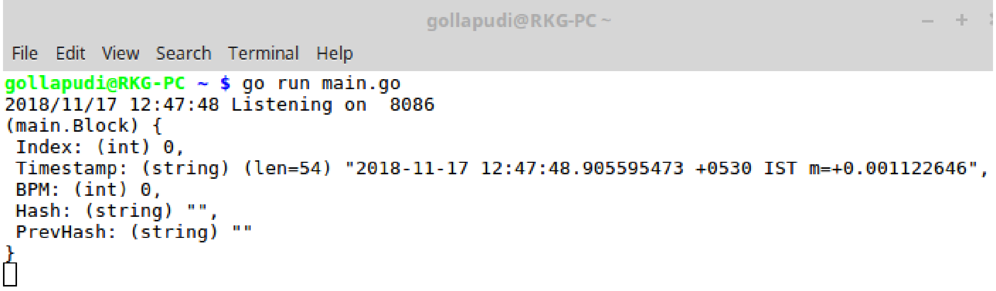
```
gollapudi@RKG-PC ~ $ go run main.go
2018/11/17 12:47:48 Listening on  8086
(main.Block) {
 Index: (int) 0,
 Timestamp: (string) (len=54) "2018-11-17 12:47:48.905595473 +0530 IST m=+0.001122646
 BPM: (int) 0,
 Hash: (string) "",
 PrevHash: (string) ""
}
```

POST | Enter request URL

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON (application/json)

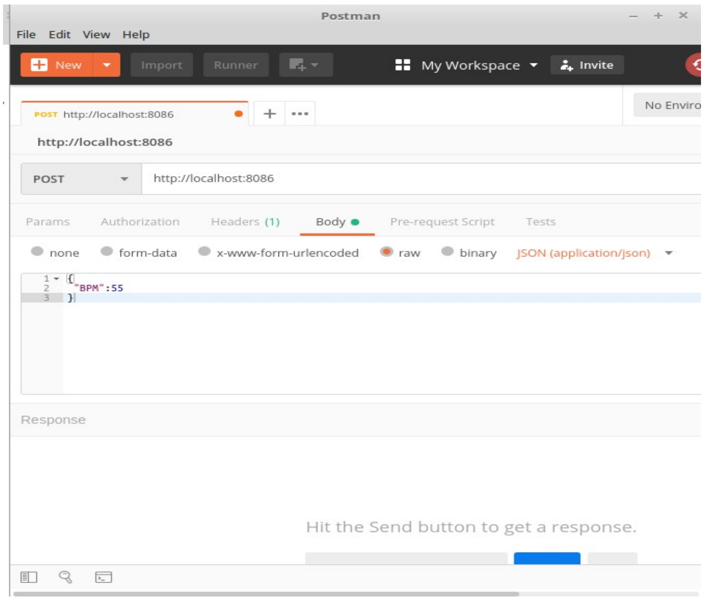
Fig: 4.1.1 Block Initialized with null block

Enter the BPM in the postman tool to add a block as a message in the body as raw data with JSON(application/json) format as default and send the request to the port. Now we can see the data which has some valid block data including the hash values which are connected to the other block's data.



```
gollapudi@RKG-PC ~ $ go run main.go
2018/11/17 12:47:48 Listening on  8086
(main.Block) {
 Index: (int) 0,
 Timestamp: (string) (len=54) "2018-11-17 12:47:48.905595473 +0530 IST m=+0.001122646",
 BPM: (int) 0,
 Hash: (string) "",
 PrevHash: (string) ""
}
```

Fig: 4.1.1.1 Initial blocks with empty raw in the stack



POST http://localhost:8086

http://localhost:8086

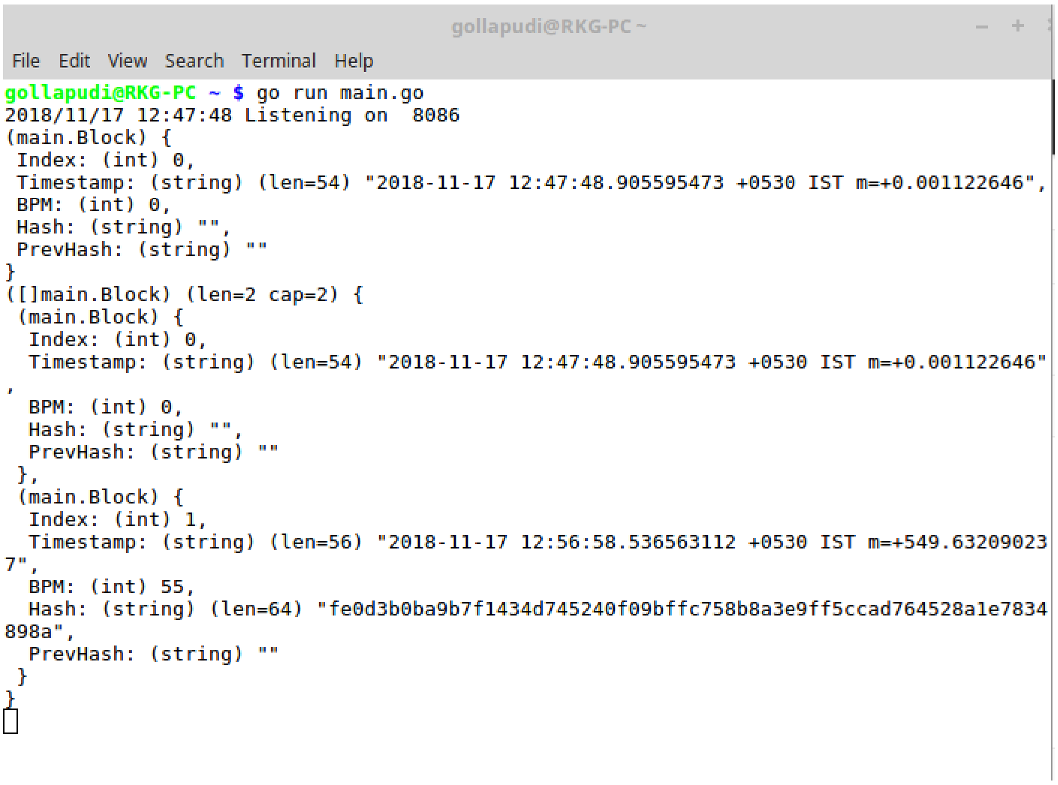
POST http://localhost:8086

Body: ● raw JSON (application/json)

```
{
 "BPM":55
}
```

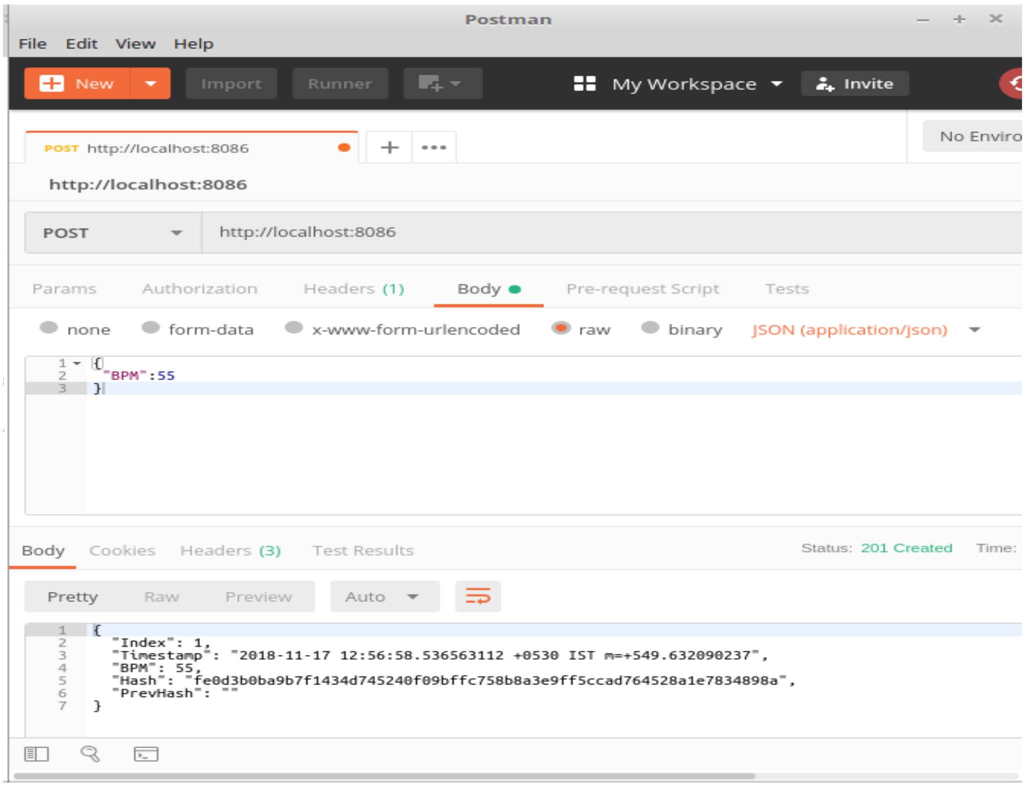
Fig: 4.1.2 Raw data entered in the body as message to add a block.

As we entered new data the hash values are changed in 40 digit format and we can also see that every block contains its own hash value (so called as unique key), and these are connected with other blocks and form a chain.



```
gollapudi@RKG-PC ~ $ go run main.go
2018/11/17 12:47:48 Listening on  8086
(main.Block) {
 Index: (int) 0,
 Timestamp: (string) (len=54) "2018-11-17 12:47:48.905595473 +0530 IST m=+0.001122646",
 BPM: (int) 0,
 Hash: (string) "",
 PrevHash: (string) ""
}
([]main.Block) (len=2 cap=2) {
 (main.Block) {
  Index: (int) 0,
  Timestamp: (string) (len=54) "2018-11-17 12:47:48.905595473 +0530 IST m=+0.001122646"
,
  BPM: (int) 0,
  Hash: (string) "",
  PrevHash: (string) ""
 },
 (main.Block) {
  Index: (int) 1,
  Timestamp: (string) (len=56) "2018-11-17 12:56:58.536563112 +0530 IST m=+549.63209023
7",
  BPM: (int) 55,
  Hash: (string) (len=64) "fe0d3b0ba9b7f1434d745240f09bffc758b8a3e9ff5ccad764528a1e7834
898a",
  PrevHash: (string) ""
 }
}
```

Fig : 4.1.3 Adding blocks



POST http://localhost:8086

http://localhost:8086

POST http://localhost:8086

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON (application/json)

```
{
"BPM":55
}
```

Status: 201 Created Time:

```
{
  "Index": 1,
  "Timestamp": "2018-11-17 12:56:58.536563112 +0530 IST m=+549.632090237",
  "BPM": 55,
  "Hash": "fe0d3b0ba9b7f1434d745240f09bffc758b8a3e9ff5ccad764528a1e7834898a",
  "PrevHash": ""
}
```

Fig: 4.1.3.1 Adding the blocks with a valid data and connecting to the previous block.

We can have clear picture now every new data creates a new block and are connected one after the another continuously. All the data of the block are stored in the data so that we can access the particular block using its hash value which is encrypted with SHA-256 algorithm.

Every single insert/add block or modifing the block is termed as a ‘transaction’ in the blockchain terminology.

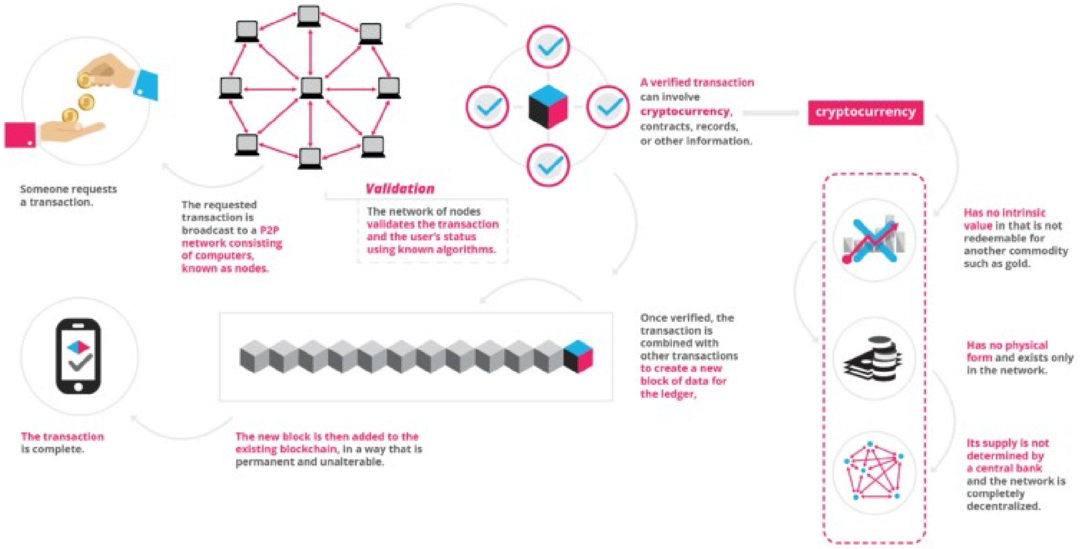
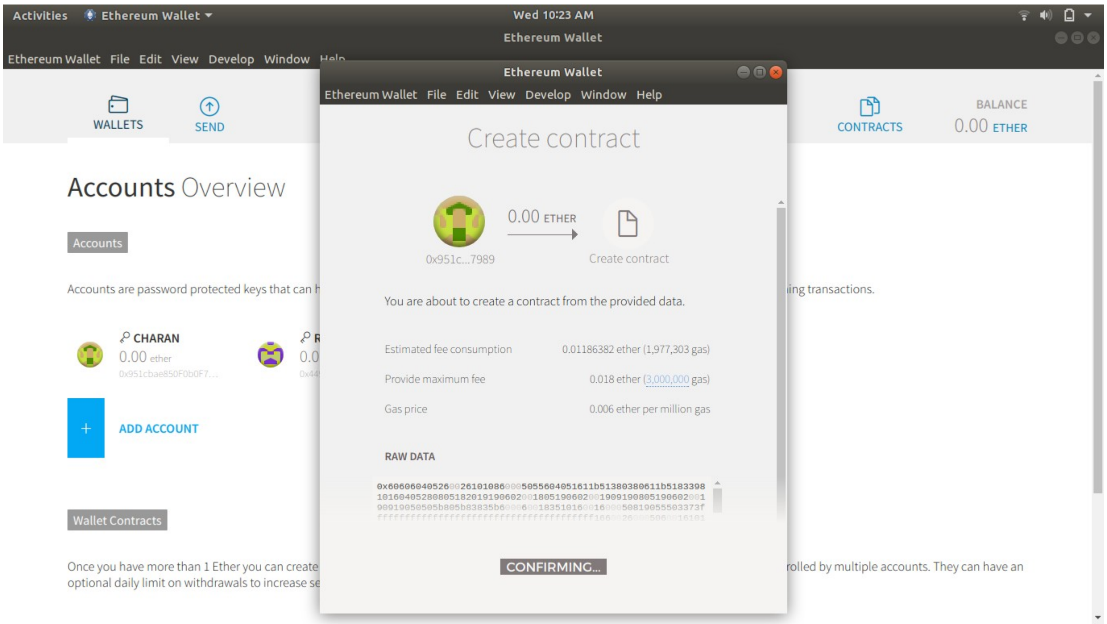

Someone requests a transaction.
The requested transaction is broadcast to a P2P network consisting of computers, known as nodes.
Validation
The network of nodes validates the transaction and the user's status using known algorithms.
A verified transaction can involve cryptocurrency, contracts, records, or other information.
cryptocurrency
Has no intrinsic value in that is not redeemable for another commodity such as gold.
Has no physical form and exists only in the network.
Its supply is not determined by a central bank and the network is completely decentralized.
Once verified, the transaction is combined with other transactions to create a new block of data for the ledger.
The new block is then added to the existing blockchain, in a way that is permanent and unalterable.
The transaction is complete.

Fig: 4.1.4 Clear process of how the transaction are made.

To understand the transactions and connecting to the blocks clearly we use a tool called ‘Ethereum wallet’, it is the one of the free, open source software and very flexible to use. Every user has their own account and their encrypted hash values upto 40 digits.



Accounts Overview

Accounts

Accounts are password protected keys that can h … ing transactions.

CHARAN 0.00 ether 0x951cbae850F0b0F7...

Wallet Contracts

Once you have more than 1 Ether you can create … rolled by multiple accounts. They can have an optional daily limit on withdrawals to increase se…

Create contract

0x951c...7989 → 0.00 ETHER → Create contract

You are about to create a contract from the provided data.

Estimated fee consumption: 0.01186382 ether (1,977,303 gas)

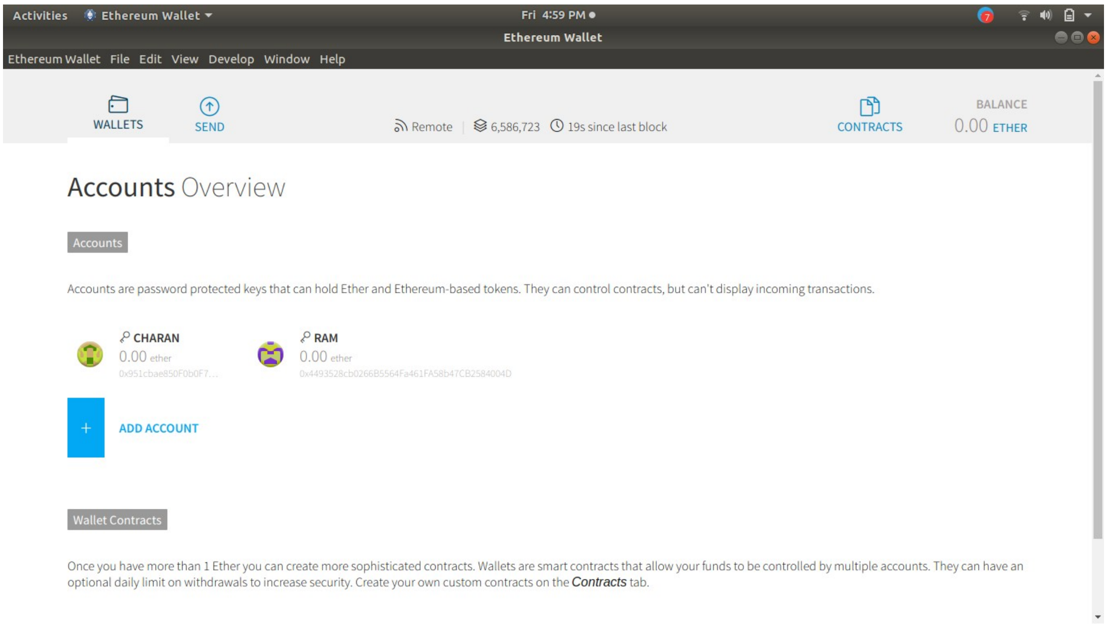
Provide maximum fee: 0.018 ether (3,000,000 gas)

Gas price: 0.006 ether per million gas

RAW DATA

```
0x6060604052600261010860005055604051611b51380380611b5183398
10160405280805182019190602001805190602001909190805190602001
90919050505b805b83835b600060018351016001600050819055503373f
ffffffffffffffffffffffffffffffffffffffff16600260005060016101
```

Fig: 4.1.5 Account Creation and Connecting the block (server)



Accounts Overview

Accounts

Accounts are password protected keys that can hold Ether and Ethereum-based tokens. They can control contracts, but can't display incoming transactions.

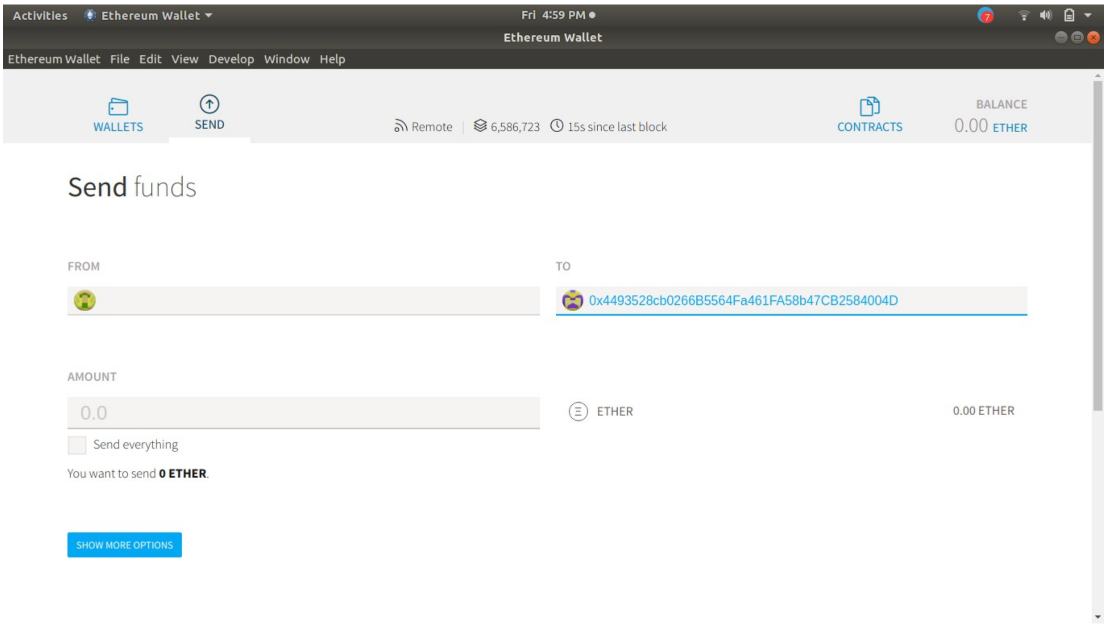
CHARAN 0.00 ether 0x951cbae850F0b0F7...

RAM 0.00 ether 0x4493528cb0266B5564Fa461FA58b47CB2584004D

Wallet Contracts

Once you have more than 1 Ether you can create more sophisticated contracts. Wallets are smart contracts that allow your funds to be controlled by multiple accounts. They can have an optional daily limit on withdrawals to increase security. Create your own custom contracts on the *Contracts* tab.

Fig: 4.1.5 Transfers Ethers to the Accounts



Send funds

FROM

TO

0x4493528cb0266B5564Fa461FA58b47CB2584004D

AMOUNT

0.0

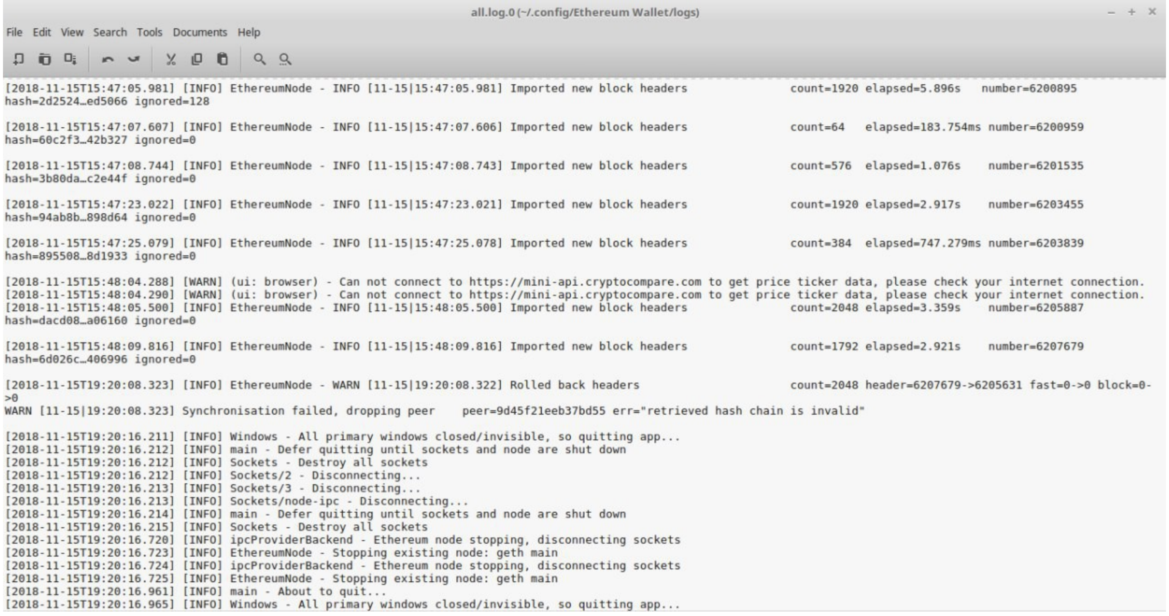
ETHER

0.00 ETHER

□ Send everything

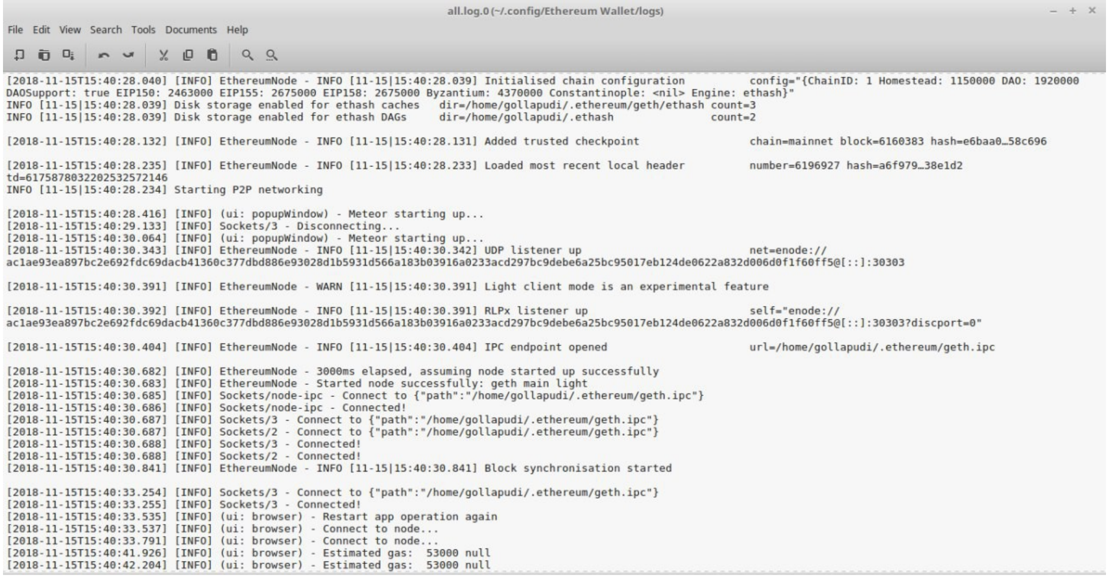
You want to send **0 ETHER**.

Fig: 4.1.6 Sending Ether’s to the other parties


all.log.0 (~/.config/Ethereum Wallet/logs)
File Edit View Search Tools Documents Help

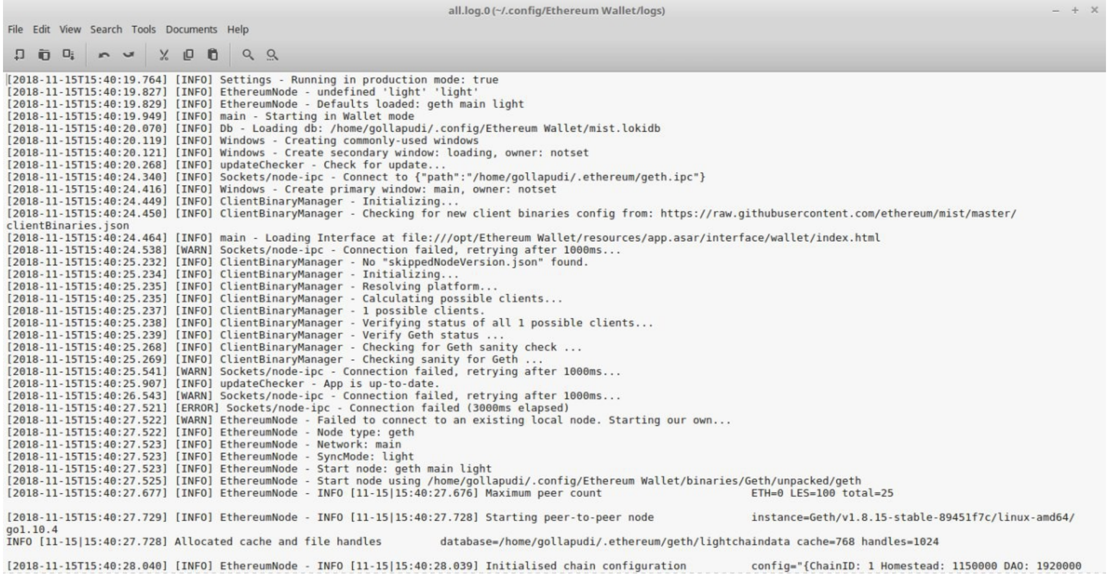
```
[2018-11-15T15:47:05.981] [INFO] EthereumNode - INFO [11-15|15:47:05.981] Imported new block headers              count=1920 elapsed=5.896s    number=6200895
hash=2d2524…ed5066 ignored=128

[2018-11-15T15:47:07.607] [INFO] EthereumNode - INFO [11-15|15:47:07.606] Imported new block headers              count=64   elapsed=183.754ms number=6200959
hash=60c2f3…42b327 ignored=0

[2018-11-15T15:47:08.744] [INFO] EthereumNode - INFO [11-15|15:47:08.743] Imported new block headers              count=576  elapsed=1.076s    number=6201535
hash=3b80da…c2e44f ignored=0

[2018-11-15T15:47:23.022] [INFO] EthereumNode - INFO [11-15|15:47:23.021] Imported new block headers              count=1920 elapsed=2.917s    number=6203455
hash=94ab8b…898d64 ignored=0

[2018-11-15T15:47:25.079] [INFO] EthereumNode - INFO [11-15|15:47:25.078] Imported new block headers              count=384  elapsed=747.279ms number=6203839
hash=895508…8d1933 ignored=0

[2018-11-15T15:48:04.288] [WARN] (ui: browser) - Can not connect to https://mini-api.cryptocompare.com to get price ticker data, please check your internet connection.
[2018-11-15T15:48:04.290] [WARN] (ui: browser) - Can not connect to https://mini-api.cryptocompare.com to get price ticker data, please check your internet connection.
[2018-11-15T15:48:05.500] [INFO] EthereumNode - INFO [11-15|15:48:05.500] Imported new block headers              count=2048 elapsed=3.359s    number=6205887
hash=dacd08…a06160 ignored=0

[2018-11-15T15:48:09.816] [INFO] EthereumNode - INFO [11-15|15:48:09.816] Imported new block headers              count=1792 elapsed=2.921s    number=6207679
hash=6d026c…406996 ignored=0

[2018-11-15T19:20:08.323] [INFO] EthereumNode - WARN [11-15|19:20:08.322] Rolled back headers                     count=2048 header=6207679->6205631 fast=0->0 block=0-
>0
WARN [11-15|19:20:08.323] Synchronisation failed, dropping peer    peer=9d45f21eeb37bd55 err="retrieved hash chain is invalid"

[2018-11-15T19:20:16.211] [INFO] Windows - All primary windows closed/invisible, so quitting app...
[2018-11-15T19:20:16.212] [INFO] main - Defer quitting until sockets and node are shut down
[2018-11-15T19:20:16.212] [INFO] Sockets - Destroy all sockets
[2018-11-15T19:20:16.212] [INFO] Sockets/2 - Disconnecting...
[2018-11-15T19:20:16.213] [INFO] Sockets/3 - Disconnecting...
[2018-11-15T19:20:16.213] [INFO] Sockets/node-ipc - Disconnecting...
[2018-11-15T19:20:16.214] [INFO] main - Defer quitting until sockets and node are shut down
[2018-11-15T19:20:16.215] [INFO] Sockets - Destroy all sockets
[2018-11-15T19:20:16.720] [INFO] ipcProviderBackend - Ethereum node stopping, disconnecting sockets
[2018-11-15T19:20:16.723] [INFO] EthereumNode - Stopping existing node: geth main
[2018-11-15T19:20:16.724] [INFO] ipcProviderBackend - Ethereum node stopping, disconnecting sockets
[2018-11-15T19:20:16.725] [INFO] EthereumNode - Stopping existing node: geth main
[2018-11-15T19:20:16.961] [INFO] main - About to quit...
[2018-11-15T19:20:16.965] [INFO] Windows - All primary windows closed/invisible, so quitting app...
```

Fig: 4.1.7 Connecting to server to validate the node


all.log.0 (~/.config/Ethereum Wallet/logs)
File Edit View Search Tools Documents Help

```
[2018-11-15T15:40:28.040] [INFO] EthereumNode - INFO [11-15|15:40:28.039] Initialised chain configuration         config="{ChainID: 1 Homestead: 1150000 DAO: 1920000
DAOSupport: true EIP150: 2463000 EIP155: 2675000 EIP158: 2675000 Byzantium: 4370000 Constantinople: <nil> Engine: ethash}"
INFO [11-15|15:40:28.039] Disk storage enabled for ethash caches   dir=/home/gollapudi/.ethereum/geth/ethash count=3
INFO [11-15|15:40:28.039] Disk storage enabled for ethash DAGs     dir=/home/gollapudi/.ethash              count=2

[2018-11-15T15:40:28.132] [INFO] EthereumNode - INFO [11-15|15:40:28.131] Added trusted checkpoint                chain=mainnet block=6160383 hash=e6baa0…58c696

[2018-11-15T15:40:28.235] [INFO] EthereumNode - INFO [11-15|15:40:28.233] Loaded most recent local header         number=6196927 hash=a6f979…38e1d2
td=6175878032202532572146
INFO [11-15|15:40:28.234] Starting P2P networking

[2018-11-15T15:40:28.416] [INFO] (ui: popupWindow) - Meteor starting up...
[2018-11-15T15:40:29.133] [INFO] Sockets/3 - Disconnecting...
[2018-11-15T15:40:30.064] [INFO] (ui: popupWindow) - Meteor starting up...
[2018-11-15T15:40:30.343] [INFO] EthereumNode - INFO [11-15|15:40:30.342] UDP listener up                         net=enode://
ac1ae93ea897bc2e692fdc69dacb41360c377dbd886e93028d1b5931d566a183b03916a0233acd297bc9debe6a25bc95017eb124de0622a832d006d0f1f60ff5@[::]:30303

[2018-11-15T15:40:30.391] [INFO] EthereumNode - WARN [11-15|15:40:30.391] Light client mode is an experimental feature

[2018-11-15T15:40:30.392] [INFO] EthereumNode - INFO [11-15|15:40:30.391] RLPx listener up                        self="enode://
ac1ae93ea897bc2e692fdc69dacb41360c377dbd886e93028d1b5931d566a183b03916a0233acd297bc9debe6a25bc95017eb124de0622a832d006d0f1f60ff5@[::]:30303?discport=0"

[2018-11-15T15:40:30.404] [INFO] EthereumNode - INFO [11-15|15:40:30.404] IPC endpoint opened                     url=/home/gollapudi/.ethereum/geth.ipc

[2018-11-15T15:40:30.682] [INFO] EthereumNode - 3000ms elapsed, assuming node started up successfully
[2018-11-15T15:40:30.683] [INFO] EthereumNode - Started node successfully: geth main light
[2018-11-15T15:40:30.685] [INFO] Sockets/node-ipc - Connect to {"path":"/home/gollapudi/.ethereum/geth.ipc"}
[2018-11-15T15:40:30.686] [INFO] Sockets/node-ipc - Connected!
[2018-11-15T15:40:30.687] [INFO] Sockets/3 - Connect to {"path":"/home/gollapudi/.ethereum/geth.ipc"}
[2018-11-15T15:40:30.687] [INFO] Sockets/2 - Connect to {"path":"/home/gollapudi/.ethereum/geth.ipc"}
[2018-11-15T15:40:30.688] [INFO] Sockets/3 - Connected!
[2018-11-15T15:40:30.688] [INFO] Sockets/2 - Connected!
[2018-11-15T15:40:30.841] [INFO] EthereumNode - INFO [11-15|15:40:30.841] Block synchronisation started

[2018-11-15T15:40:33.254] [INFO] Sockets/3 - Connect to {"path":"/home/gollapudi/.ethereum/geth.ipc"}
[2018-11-15T15:40:33.255] [INFO] Sockets/3 - Connected!
[2018-11-15T15:40:33.535] [INFO] (ui: browser) - Restart app operation again
[2018-11-15T15:40:33.537] [INFO] (ui: browser) - Connect to node...
[2018-11-15T15:40:33.791] [INFO] (ui: browser) - Connect to node...
[2018-11-15T15:40:41.926] [INFO] (ui: browser) - Estimated gas:  53000 null
[2018-11-15T15:40:42.204] [INFO] (ui: browser) - Estimated gas:  53000 null
```

Fig: 4.1.8 Socket Connection


all.log.0 (~/.config/Ethereum Wallet/logs)
File Edit View Search Tools Documents Help

```
[2018-11-15T15:40:19.764] [INFO] Settings - Running in production mode: true
[2018-11-15T15:40:19.827] [INFO] EthereumNode - undefined 'light' 'light'
[2018-11-15T15:40:19.829] [INFO] EthereumNode - Defaults loaded: geth main light
[2018-11-15T15:40:19.949] [INFO] main - Starting in Wallet mode
[2018-11-15T15:40:20.070] [INFO] Db - Loading db: /home/gollapudi/.config/Ethereum Wallet/mist.lokidb
[2018-11-15T15:40:20.119] [INFO] Windows - Creating commonly-used windows
[2018-11-15T15:40:20.121] [INFO] Windows - Create secondary window: loading, owner: notset
[2018-11-15T15:40:20.268] [INFO] updateChecker - Check for update...
[2018-11-15T15:40:24.340] [INFO] Sockets/node-ipc - Connect to {"path":"/home/gollapudi/.ethereum/geth.ipc"}
[2018-11-15T15:40:24.416] [INFO] Windows - Create primary window: main, owner: notset
[2018-11-15T15:40:24.449] [INFO] ClientBinaryManager - Initializing...
[2018-11-15T15:40:24.450] [INFO] ClientBinaryManager - Checking for new client binaries config from: https://raw.githubusercontent.com/ethereum/mist/master/
clientBinaries.json
[2018-11-15T15:40:24.464] [INFO] main - Loading Interface at file:///opt/Ethereum Wallet/resources/app.asar/interface/wallet/index.html
[2018-11-15T15:40:24.538] [WARN] Sockets/node-ipc - Connection failed, retrying after 1000ms...
[2018-11-15T15:40:25.232] [INFO] ClientBinaryManager - No "skippedNodeVersion.json" found.
[2018-11-15T15:40:25.234] [INFO] ClientBinaryManager - Initializing...
[2018-11-15T15:40:25.235] [INFO] ClientBinaryManager - Resolving platform...
[2018-11-15T15:40:25.235] [INFO] ClientBinaryManager - Calculating possible clients...
[2018-11-15T15:40:25.237] [INFO] ClientBinaryManager - 1 possible clients.
[2018-11-15T15:40:25.238] [INFO] ClientBinaryManager - Verifying status of all 1 possible clients...
[2018-11-15T15:40:25.239] [INFO] ClientBinaryManager - Verify Geth status ...
[2018-11-15T15:40:25.268] [INFO] ClientBinaryManager - Checking for Geth sanity check ...
[2018-11-15T15:40:25.269] [INFO] ClientBinaryManager - Checking sanity for Geth ...
[2018-11-15T15:40:25.541] [WARN] Sockets/node-ipc - Connection failed, retrying after 1000ms...
[2018-11-15T15:40:25.907] [INFO] updateChecker - App is up-to-date.
[2018-11-15T15:40:26.543] [WARN] Sockets/node-ipc - Connection failed, retrying after 1000ms...
[2018-11-15T15:40:27.521] [ERROR] Sockets/node-ipc - Connection failed (3000ms elapsed)
[2018-11-15T15:40:27.522] [WARN] EthereumNode - Failed to connect to an existing local node. Starting our own...
[2018-11-15T15:40:27.522] [INFO] EthereumNode - Node type: geth
[2018-11-15T15:40:27.523] [INFO] EthereumNode - Network: main
[2018-11-15T15:40:27.523] [INFO] EthereumNode - SyncMode: light
[2018-11-15T15:40:27.523] [INFO] EthereumNode - Start node: geth main light
[2018-11-15T15:40:27.525] [INFO] EthereumNode - Start node using /home/gollapudi/.config/Ethereum Wallet/binaries/Geth/unpacked/geth
[2018-11-15T15:40:27.677] [INFO] EthereumNode - INFO [11-15|15:40:27.676] Maximum peer count                      ETH=0 LES=100 total=25

[2018-11-15T15:40:27.729] [INFO] EthereumNode - INFO [11-15|15:40:27.728] Starting peer-to-peer node              instance=Geth/v1.8.15-stable-89451f7c/linux-amd64/
go1.10.4
INFO [11-15|15:40:27.728] Allocated cache and file handles        database=/home/gollapudi/.ethereum/geth/lightchaindata cache=768 handles=1024

[2018-11-15T15:40:28.040] [INFO] EthereumNode - INFO [11-15|15:40:28.039] Initialised chain configuration         config="{ChainID: 1 Homestead: 1150000 DAO: 1920000
```

Fig: 4.1.9 Allocation cache and file handler

CHAPTER-5

Conclusion

By implementing the E-voting system using blockchain, the method and voting process across the world will be eventually get changed thereby grabbing the attention of developers towards this new methodology.Not only voting system but also many other techniques will be developed using the concept blockchain. Many real world problems get rectified and solved using this methodology. With the usage of the technology blockchain there is a wonderful and better possibility for democratic countries to enhance and make improvements from the old and paper based election schemes and methods to the less cost and efficient time saving methodology thereby increasing the security as well as transparency.

CHAPTER-6

Future Scope

Blockchain is used not only for the e-voting but also for many other primary purposes. In order to solve the real world problems in a quick manner this type of voting system can be the best reference.For development of many other management systems this process is a good example.In the field of science the concept blockchain is used for development of new technologies. Developers will get a chance to make improvements and can create new techniques to implement new blockchain methods.For developing new programs and codes “Go” language used here is the best reference. The concept Postman will be useful to create new inventions.

CHAPTER-7

References

1) K. A. M. F. M. Kirby, “Votebook : A proposal for a block-chain based electronic voting system,” 2016.
2) P. McCorry, S. F. Shahandashti, and F. Hao, “A smart contract for boardroom voting with maximum voter privacy,” in International Conference on Financial Cryptography and Data Security. Springer, 2017, pp. 357–375.
3) A. B. Ayed, “A conceptual secure block-chain based electronic voting system,” International Journal of Network Security & Its Applications, vol. 9, no. 3, 2017.
4) L. C. Schaupp and L. Carter, “E-voting: from apathy to adoption,” Journal of Enterprise Information Management, vol. 18, no. 5, pp. 586–601, 2005.
5) https://www.researchgate.net/publication/325396308_Evoting_with_the concepts_of_the_technique_Blockchain_An_Evoting_Protocol_with_the Decentralisation_and_the Voter_Privacy.
6) http://mfa.ee/sites/default/files/content-editors/2015%20Parliamentary %20elections%20Internet%20voting%20system.pdf.
7) https://www.lawfareblog.com/ secure-vote-today.