Blockchain Based Resilient Data Storage System

Abhishek Bhardwaj
North Carolina State University
abhardw3@ncsu.edu

Madhu Vamsi Kalyan Machavarapu
North Carolina State University
mmachav@ncsu.edu

Sanya Kathuria
North Carolina State University
skathur2@ncsu.edu

Shivam Chamoli
North Carolina State University
schamol@ncsu.edu

ABSTRACT

The Digital world has produced a variety of new innovative products, and close customer relationships globally by the efficient use of mobile, IoT (Internet of Things), social media, analytics and cloud technology to generate models for better decisions. Blockchain was recently introduced and revolutionized the digital world bringing a new perspective to security, resilience, and efficiency of systems. While initially popularized by Bitcoin, Blockchain is much more than a foundation for cryptocurrency. It offers a secure way to exchange any kind of good, service, or transaction. Secure in a way that to create an authorized block, the block needs to be mined with some stringent conditions on how the Hash of the block is created and has to be passed by more than 50% of the peer to peer network, making it extremely resilient to change. Moreover, Blockchain will enable more agile value chains, faster product innovations, closer customer relationships, and quicker integration due to its distributed, decentralized nature.

Keywords

Blockchain, Bitcoins, ethereum, smart-contracts, contracts, decentralized, distributed storage, secured, time-stamping, accessibility, Property records, irreversible transactions.

1. INTRODUCTION

A blockchain is essentially a distributed database of records or a public ledger of all transactions or digital events that have been executed and shared among the participating parties. Each transaction in the public ledger is verified by the consensus of a majority of the participants in the system and, once entered, information can never be erased. The main hypothesis is that the blockchain establishes a system of creating a distributed consensus âĂŃin the digital online world. With public health, wealth, safety, security and environmental protection as a priority, licensing bestows accountability and liability to those developing and operating digital systems and privacy records. We, in this project, would like to address the problem of frauds that take place due to forgery of documents that are present with for example the rotary (like property papers). Another issue that we would like to address is to remove the governing authorities from the use cases like betting, where two parties have to trust a third-party to come through on the deal. Our goal is to make a distributed, decentralized record of potentially fraudulent documents or activities but due to the time-constraints, making a truly generic system which would authenticate the information is unrealistic and hence we would be producing a simple Proof of Concept.

X-XXXXX-XX-X/XX/XX.
DOI:

2. MOTIVATION

2.1 Why not Traditional databases?

Traditional databases are usually maintained by a single organization, and that organization has complete control of the database which includes the ability to tamper with the stored data, to censor otherwise valid changes to the data, or to add data fraudulently. Even if we assume that the responsible organization would never enact a fraudulent change to the database which is already too much to ask, there is still the possibility that a hacker could break in and manipulate the database to their own ends.

2.2 Why Blockchain based technology?

The main motivation for this project is to implement a system that makes the database public and allows anyone to store a redundant copy of the database. This is the best way to ensure that data is safely stored.

2.3 How a Blockchain technology solves the problem?

Blockchain technology solves these problems by creating a network of computers (called nodes) which each store a copy of the database, and a set of rules (called the consensus protocol) which define the order in which nodes may take turns adding new changes to the database. In this way, all of the nodes agree to the state of the database at any time, and no one node has the power to falsify the data or to censor changes. The blockchain further requires that an audit trail of all changes to the database is preserved, which allows anyone to audit that the database is correct at any time. This audit trail is composed of the individual changes to the database, which are called transactions.

2.4 Properties of our blockchain

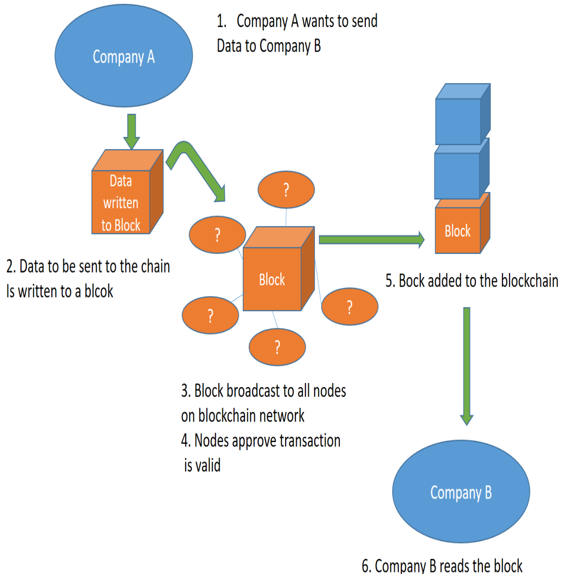

Company A
1. Company A wants to send
Data to Company B
Data
written
to Block
2. Data to be sent to the chain
Is written to a blcok
?
?
Block
?
?
?
3. Block broadcast to all nodes
on blockchain network
4. Nodes approve transaction
is valid
Block
5. Bock added to the blockchain
Company B
6. Company B reads the block

Figure 1: How does Blockchain technology work?

In this short time frame, we could come up with a blockchain constituting of the following properties

2.4.1 *Replication*

Data on the blockchain is copied on every computer that is a part of the P2P network.

2.4.2 *No Central Authority*

There is no central body which governs whether a particular transaction should be recorded or not. This is solved by using consensus amongst all the nodes on the P2P network.

2.4.3 *Irreversibility*

If data/transaction is recorded once on the blockchain, it is very tough to be reversed or updated.

2.4.4 *Accessibility*

Blockchain allows everybody who is a part of the network to read or view any data recorded by anybody.

2.4.5 *Time-stamping*

Timestamping is the process of securely keeping track of the creation and modification time of a document. It allows interested parties to know, without a doubt, that a document in question existed at a particular date and time. In blockchain, every block is time-stamped.

3. USER SURVEYS

In this section, we have described analysis of results for different questions that we asked while taking the survey to understand the utility of implementing the blockchain based resilient storage system. We have described the reasons to ask this question followed by conclusions after collecting the responses.

3.0.1 *Question 1*

What modes of data storage do you generally prefer using for securing information?
We asked this question to understand the trends of current usage for storing data.
Cloud storage system.
Disk storage system.
Blockchain based system
Refer figure 2

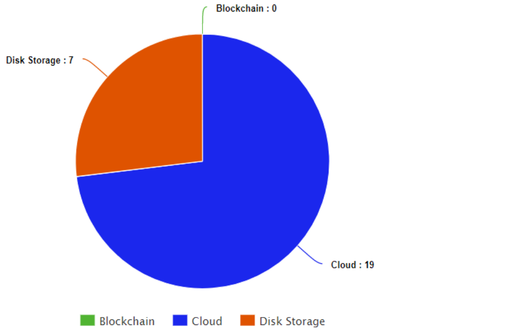


Figure 2: Where do you usually store your data

3.0.2 *Question 2*

Have you or your friend been duped by forged documents relating to property or shares? We asked this question to understand if people are familiar with the situation in which our application suits best. This also helps the customers get aware of the utility of the application is such situations like duped by forged documents relating to property or shares which we intended to work on as our use-cases.
Refer figure 3

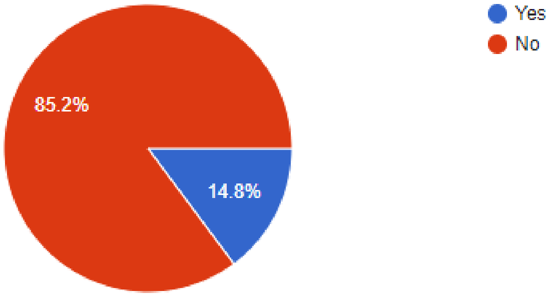


Figure 3: Have you or your friend been duped by forged documents relating to property or shares?

3.0.3 *Question 3*

Would you consider a blockchain based online public storage for important documents that is hacker and forgery proof? We asked this question to understand the familiarity of the users with this system and how comfortable they will be to explore blockchain based technology online public storage for securing important documents. This helped us

understand the interest of people in using our application for storing data.
Refer figure 4

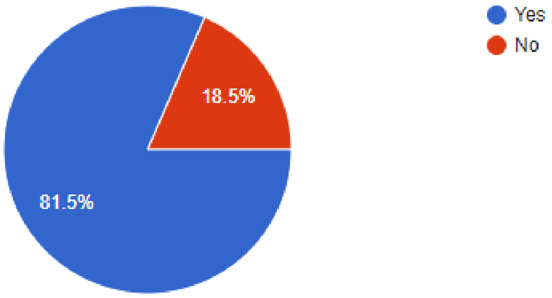


Figure 4: Would you consider a blockchain based online public storage for important documents that is hacker and forgery proof?

4. USE-CASE STUDY

The advantages of Blockchain technology outweigh the regulatory issues and technical challenges. One key emerging use case of blockchain technology involves Digital Blockchain based Property. Digital Blockchain based Property will basically be a computer programs that can automatically execute the terms of a contract. When a pre-configured condition in a smart contract among participating entities is met then the parties involved in a contractual agreement can be automatically made payments as per the contract in a transparent manner. Fraud-resistant data keeping of sensitive information.

4.0.1 Records of property

Maintaining a national register of property ownership is an expensive and labor-intensive operation. Additionally, in countries where there is a history of government corruption, they may not always be trustworthy. Property documents can be forged by miscreants for stealing properties. With blockchain based data storage this can be made really difficult to execute because of the security features baked inherently in the blockchain.

4.0.2 Vehicle renting

The vehicle rental process is often more cumbersome than it needs to be, with insurance documents and identities that need to be verified, and vehicle mileage and damage reports that are still manually verified in many cases. Our project involves implementing a blockchain based data storage such that these cumbersome tasks would be circumvented.

4.0.3 Information of Stock markets

The item can be non-physical such as shares of a company.

4.0.4 Smart Property

Smart Property is another related concept which is regarding controlling the ownership of a property or asset via blockchain using Smart Contracts. The property can be physical such as car, house, smartphone etc. or it can be non-physical such as shares of a company. It should be noted here that even Bitcoin is not really a currency- Bitcoin is all about controlling the ownership of money.

4.1 Advantages of the system according to our use cases

1. Conflict-proof distributed ledger of probable fraudulent transactions like property transfer, stocks etc.

2. Redundancy of sensitive data because of the distributed and decentralized nature.

3. Fraud - proof, because, to change the value of an already existing block in the blockchain, the person has to recompute all the blocks again and gain access of more than 50 % of the p2p network and recompute the nodes again.

4. Further Blockchain provides a lower cost of trade with a trusted contract monitored without intervention from third parties who may not add direct value.

4.2 Disadvantages of the system according to our use cases

Although the blockchain technology has great potential for the construction of the future Internet systems, it is facing a number of technical challenges.

1. People might not be comfortable with such sensitive data online.

2. The block size is limited to 1 MB now while a block is mined about every ten minutes, which implies that the network is restricted to a rate of 7 transactions per second, which makes it incapable of dealing with high-frequency work. But also, larger blocks means larger storage space and slower propagation in the network. This will lead to centralization gradually as fewer users would like to maintain such a large blockchain. Therefore the tradeoff between block size and security will be a tough challenge.

3. Furthermore, current consensus algorithms like proof of work or proof of stake are facing some serious problems. For example, proof of work wastes too much electrical energy while the phenomenon that the rich get richer could appear in the proof of stake consensus process.

4. Highly unlikely, but if the peer-to-peer network is brought down, all the information is compromised, to handle that we need physical records.

5. EVALUATION PLAN

1. Firstly, we will have to create the front-end which would interact with the user (restrict the inputs) so that our mildly generic blockchain application can handle some cases seamlessly.

2. Test the security features of the blockchain.

3. Test the smart contract using Remix

4. Test the webapp integration with the smart contract.

5. Test everything.

6. Repeat 1 - 6 (agile) for improvement.

6. EXECUTION PLAN

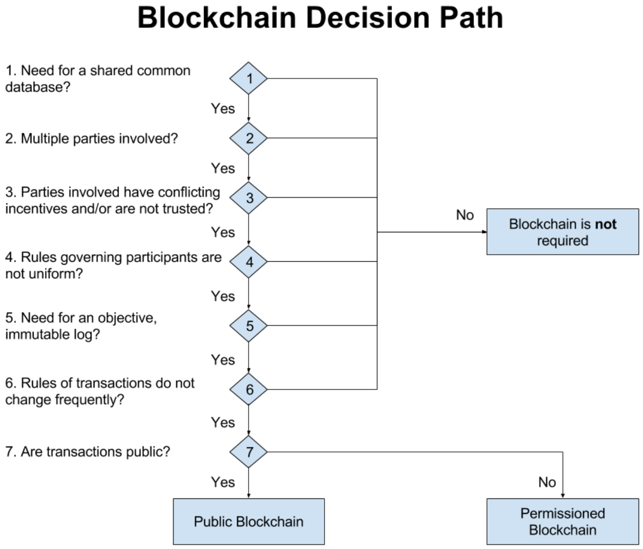
The execution phase for coding includes (Please refer figure 4)

1. What is an application without user-interaction? Hence we have our novel front-end which would take care of all the user information validation and create url to call a suitable controller.

2. Controller in-turn runs server side validations and forwards the information.

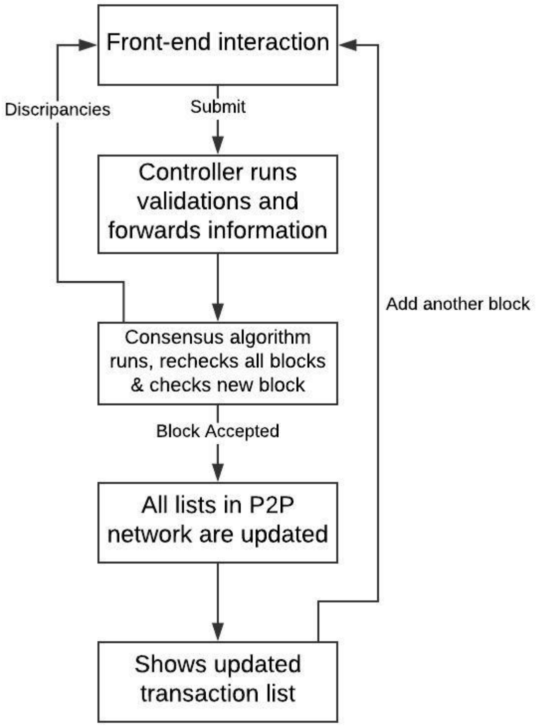
3. The server side validations are taken care by the ethereum blockchain which we interact using the Wb3js api.

4. If the block is accepted, All the lists in the P2P network are updated and the user is shown the updated list. (List is shown for project purpose, logically, it makes no sense to show the list to the user).



Blockchain Decision Path
1. Need for a shared common database?
2. Multiple parties involved?
3. Parties involved have conflicting incentives and/or are not trusted?
4. Rules governing participants are not uniform?
5. Need for an objective, immutable log?
6. Rules of transactions do not change frequently?
7. Are transactions public?
1
2
3
4
5
6
7
Yes
Yes
Yes
Yes
Yes
Yes
Yes
No
Blockchain is not required
No
Public Blockchain
Permissioned Blockchain

Figure 5: When to use Blockchain technology?



Front-end interaction
Discripancies
Submit
Controller runs validations and forwards information
Add another block
Consensus algorithm runs, rechecks all blocks & checks new block
Block Accepted
All lists in P2P network are updated
Shows updated transaction list

Figure 6: Execution Flow for our Blockchain technology?

6.1 Technology used for the application

6.1.1 Ethereum Platform

Ethereum is a decentralized platform that runs Smart Contracts: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third-party interference.

These apps run on custom built blockchain, and enormously powerful shared global infrastructure that can move value around and represent the ownership of property.

This enables developers to create markets, store registries of debts or promises, move funds in accordance with instructions given long in the past (like a will or a futures contract) and many other things that have not been invented yet, all without a middleman or counterparty risk.

The project was bootstrapped via an ether presale in August 2014 by fans all around the world. It is developed by the Ethereum Foundation, a Swiss non-profit, with contributions from great minds across the globe.

6.1.2 JavaScript

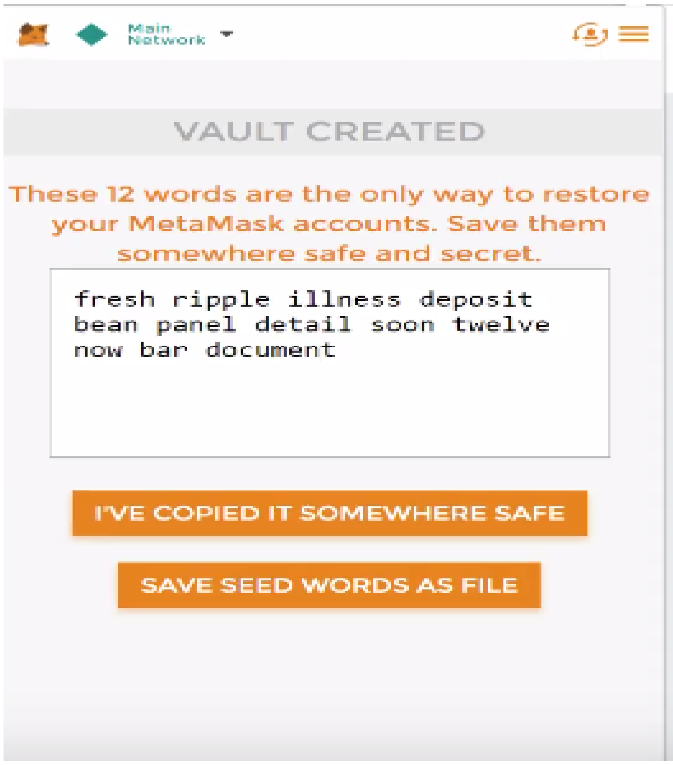
Javascript was used for creating the front-end of the application because of the interface (Web3 API) available for it to interact with the smart contracts. Moreover, its integration with HTML, CSS, JQuery and Bootstrap are very well documented, which made our lives easier while coding this application in such a short frame of time.

6.1.3 *Metamask*

MetaMask is a bridge that allows you to visit the distributed web of tomorrow in your browser today. It allows you to run Ethereum dApps right in your browser without running a full Ethereum node.

MetaMask includes a secure identity vault, providing a user interface to manage your identities on different sites and sign blockchain transactions.

For the sake of our PoC we created 2 new accounts on Metamask and by default the account had no ethers. We mined the ethers using the free ether faucets available for testing on Ropsten Test Network. This allowed us to comprehensively test our application and report few bugs (some of which are still open :P).



These 12 words are the only way to restore your MetaMask accounts. Save them somewhere safe and secret.

fresh ripple illness deposit bean panel detail soon twelve now bar document

Figure 7: creating a new account using metamask

6.1.4 *Solidity*

Solidity is a high-level language whose syntax is similar to that of JavaScript and it is designed to compile to code for the Ethereum Virtual Machine. This is the language we used for creating our smart contracts. There is an online compiler available for this language called Remix, we tested our smart contract there for correctness.

6.1.5 *Web3js*

web3.js - Ethereum JavaScript API. web3.js is a collection of libraries which allows you to interact with a local or remote ethereum node, using a HTTP or IPC connection. See Figure 8 for more information.

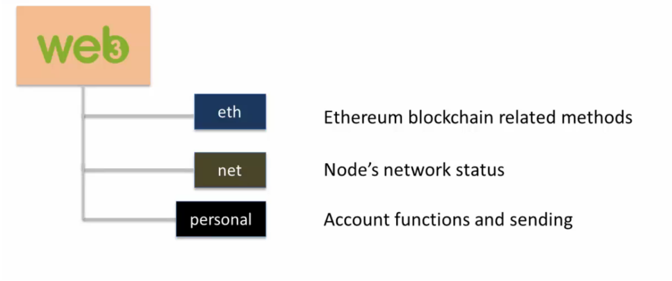

web3
eth
Ethereum blockchain related methods
net
Node's network status
personal
Account functions and sending

Figure 8: How web3js interacts with ethereum blockchain

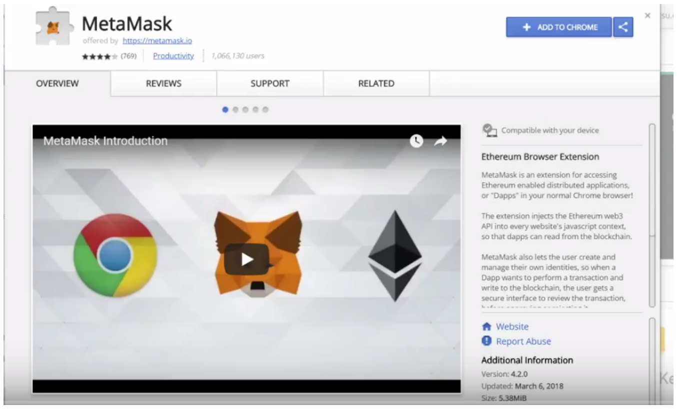
7. METHODOLOGY

7.1 Project Software Development Model

Spiral method of coding was used in this application along with the agile technique of driver and navigator. We first completed the toughest part of the application that was working on the Smart Contract and setting up the ethereum blockchain.

8. PROJECT INSTALLATION

1. Add MetaMask plugin for Chrome or Firefox or Opera. See figure 9.



Ethereum Browser Extension

MetaMask is an extension for accessing Ethereum enabled distributed applications, or "Dapps" in your normal Chrome browser!

The extension injects the Ethereum web3 API into every website's javascript context, so that dapps can read from the blockchain.

MetaMask also lets the user create and manage their own identities, so when a Dapp wants to perform a transaction and write to the blockchain, the user gets a secure interface to review the transaction,

Additional Information

Version: 4.2.0
Updated: March 6, 2018
Size: 5.38MiB

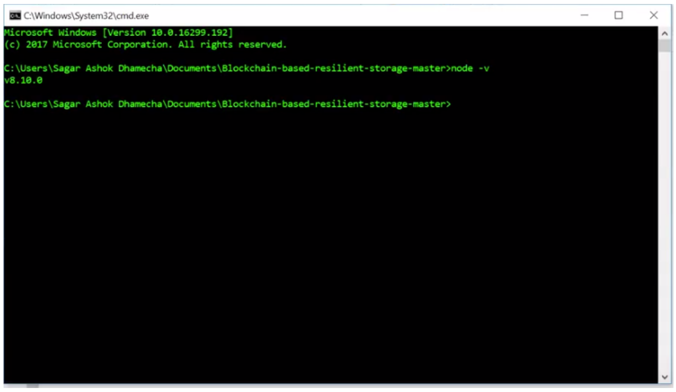
Figure 9: Installing metamask

2. Make 2 accounts and mine 4 Ethers in each account

3. Download the git repository https://github.com/MadhuVamsi/Blockchain-based-resilient-storage

4. Download Node JS from https://nodejs.org/en/ and install it

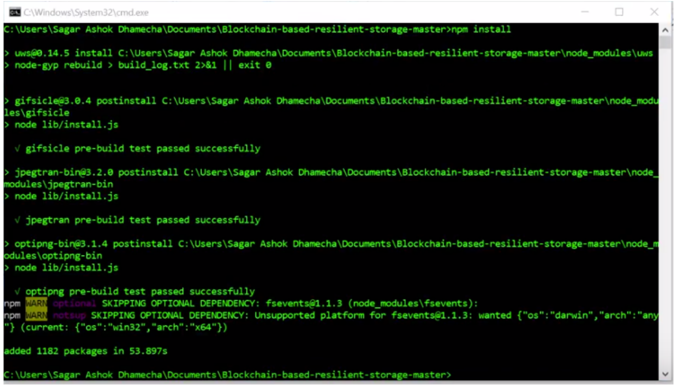
5. Check its presence by typing the node -v command in the terminal. See figure 9.



```
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>node -v
v8.10.0

C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>
```

Figure 10: Checking for installation of node

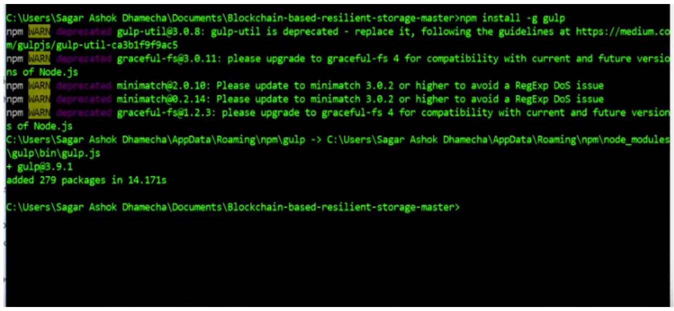
6. run npm install on the root directory of the project. See figure 10.



```
C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>npm install

> uws@0.14.5 install C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master\node_modules\uws
> node-gyp rebuild > build_log.txt 2>&1 || exit 0


> gifsicle@3.0.4 postinstall C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master\node_modu
les\gifsicle
> node lib/install.js

  √ gifsicle pre-build test passed successfully

> jpegtran-bin@3.2.0 postinstall C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master\node_
modules\jpegtran-bin
> node lib/install.js

  √ jpegtran pre-build test passed successfully

> optipng-bin@3.1.4 postinstall C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master\node_m
odules\optipng-bin
> node lib/install.js

  √ optipng pre-build test passed successfully
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.3: wanted {"os":"darwin","arch":"any
"} (current: {"os":"win32","arch":"x64"})

added 1182 packages in 53.897s

C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>
```

Figure 11: Installing Dependencies

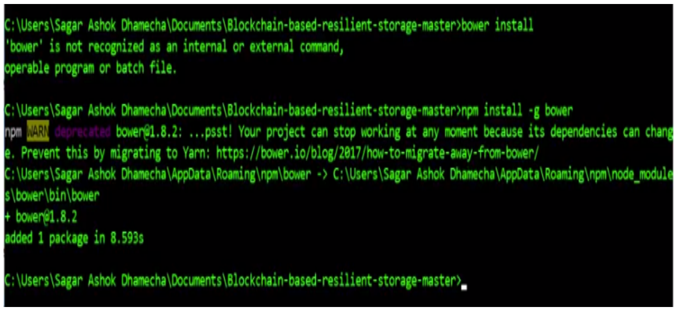
7. run npm install -g gulp. See figure 11.



```
C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>npm install -g gulp
npm WARN deprecated gulp-util@3.0.8: gulp-util is deprecated - replace it, following the guidelines at https://medium.co
m/gulpjs/gulp-util-ca3b1f9f9ac5
npm WARN deprecated graceful-fs@3.0.11: please upgrade to graceful-fs 4 for compatibility with current and future versio
ns of Node.js
npm WARN deprecated minimatch@2.0.10: Please update to minimatch 3.0.2 or higher to avoid a RegExp DoS issue
npm WARN deprecated minimatch@0.2.14: Please update to minimatch 3.0.2 or higher to avoid a RegExp DoS issue
npm WARN deprecated graceful-fs@1.2.3: please upgrade to graceful-fs 4 for compatibility with current and future version
s of Node.js
C:\Users\Sagar Ashok Dhamecha\AppData\Roaming\npm\gulp -> C:\Users\Sagar Ashok Dhamecha\AppData\Roaming\npm\node_modules
\gulp\bin\gulp.js
+ gulp@3.9.1
added 279 packages in 14.171s

C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>
```

Figure 12: Installing gulp

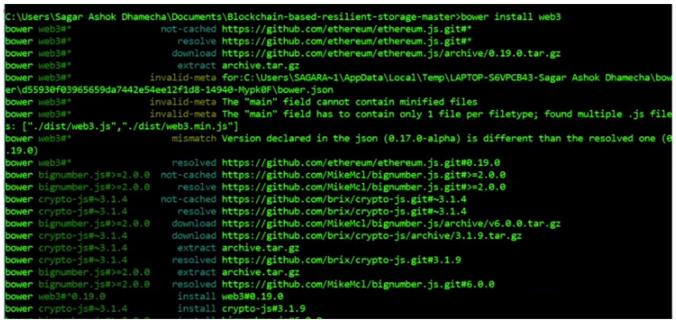
8. run bower install. See figure 12.



```
C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>bower install
'bower' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>npm install -g bower
npm WARN deprecated bower@1.8.2: ...psst! Your project can stop working at any moment because its dependencies can chang
e. Prevent this by migrating to Yarn: https://bower.io/blog/2017/how-to-migrate-away-from-bower/
C:\Users\Sagar Ashok Dhamecha\AppData\Roaming\npm\bower -> C:\Users\Sagar Ashok Dhamecha\AppData\Roaming\npm\node_module
s\bower\bin\bower
+ bower@1.8.2
added 1 package in 8.593s

C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>
```

Figure 13: Installing bower

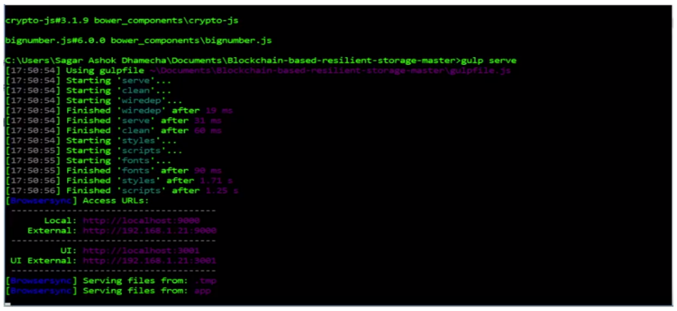
9. run bower install web3. See figure 13.



```
C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>bower install web3
bower web3#*                 not-cached https://github.com/ethereum/ethereum.js.git#*
bower web3#*                    resolve https://github.com/ethereum/ethereum.js.git#*
bower web3#*                   download https://github.com/ethereum/ethereum.js/archive/0.19.0.tar.gz
bower web3#*                    extract archive.tar.gz
bower web3#*               invalid-meta for:C:\Users\SAGARA~1\AppData\Local\Temp\LAPTOP-S6VPCB43-Sagar Ashok Dhamecha\bow
er\d55930f03965659da7442e54ee12f1d8-14940-Mypk0F\bower.json
bower web3#*               invalid-meta The "main" field cannot contain minified files
bower web3#*               invalid-meta The "main" field has to contain only 1 file per filetype; found multiple .js file
s: ["./dist/web3.js","./dist/web3.min.js"]
bower web3#*                   mismatch Version declared in the json (0.17.0-alpha) is different than the resolved one (0
.19.0)
bower web3#*                   resolved https://github.com/ethereum/ethereum.js.git#0.19.0
bower bignumber.js#>=2.0.0   not-cached https://github.com/MikeMcl/bignumber.js.git#>=2.0.0
bower bignumber.js#>=2.0.0      resolve https://github.com/MikeMcl/bignumber.js.git#>=2.0.0
bower crypto-js#~3.1.4       not-cached https://github.com/brix/crypto-js.git#~3.1.4
bower crypto-js#~3.1.4          resolve https://github.com/brix/crypto-js.git#~3.1.4
bower bignumber.js#>=2.0.0     download https://github.com/MikeMcl/bignumber.js/archive/v6.0.0.tar.gz
bower crypto-js#~3.1.4         download https://github.com/brix/crypto-js/archive/3.1.9.tar.gz
bower crypto-js#~3.1.4          extract archive.tar.gz
bower crypto-js#~3.1.4         resolved https://github.com/brix/crypto-js.git#3.1.9
bower bignumber.js#>=2.0.0      extract archive.tar.gz
bower bignumber.js#>=2.0.0     resolved https://github.com/MikeMcl/bignumber.js.git#6.0.0
bower web3#^0.19.0              install web3#0.19.0
bower crypto-js#~3.1.4          install crypto-js#3.1.9
```

Figure 14: Installing web3 using bower

10. run gulp serve to start the server. See figure 14.

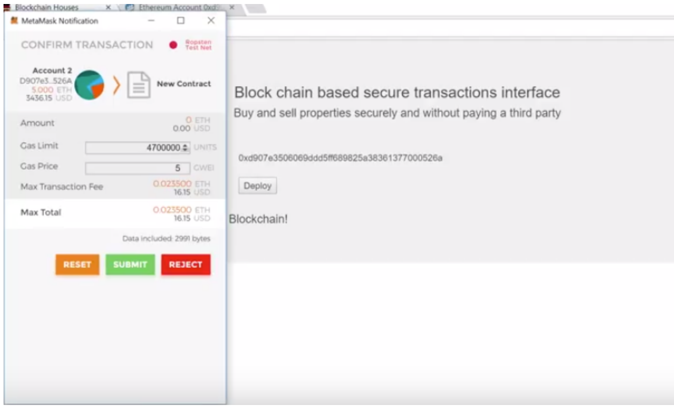


```
crypto-js#3.1.9 bower_components\crypto-js

bignumber.js#6.0.0 bower_components\bignumber.js

C:\Users\Sagar Ashok Dhamecha\Documents\Blockchain-based-resilient-storage-master>gulp serve
[17:50:54] Using gulpfile ~\Documents\Blockchain-based-resilient-storage-master\gulpfile.js
[17:50:54] Starting 'serve'...
[17:50:54] Starting 'clean'...
[17:50:54] Starting 'wiredep'...
[17:50:54] Finished 'wiredep' after 19 ms
[17:50:54] Finished 'serve' after 31 ms
[17:50:54] Finished 'clean' after 60 ms
[17:50:54] Starting 'styles'...
[17:50:55] Starting 'scripts'...
[17:50:55] Starting 'fonts'...
[17:50:55] Finished 'fonts' after 90 ms
[17:50:56] Finished 'styles' after 1.71 s
[17:50:56] Finished 'scripts' after 1.25 s
[Browsersync] Access URLs:
 --------------------------------------
       Local: http://localhost:9000
    External: http://192.168.1.21:9000
 --------------------------------------
          UI: http://localhost:3001
 UI External: http://192.168.1.21:3001
 --------------------------------------
[Browsersync] Serving files from: .tmp
[Browsersync] Serving files from: app
```

Figure 15: Hosting the server

11. run localhost:9000 in the url bar of your chosen browser (only Chrome/Firefox/Opera).

9. APPLICATION FLOW



Account 2
D907e3...526A
5.000 ETH
3436.15 USD

Amount 0 ETH 0.00 USD
Gas Limit 4700000 UNITS
Gas Price 5 GWEI
Max Transaction Fee 0.023500 ETH 16.15 USD
Max Total 0.023500 ETH 16.15 USD
Data included: 2991 bytes

Block chain based secure transactions interface
Buy and sell properties securely and without paying a third party

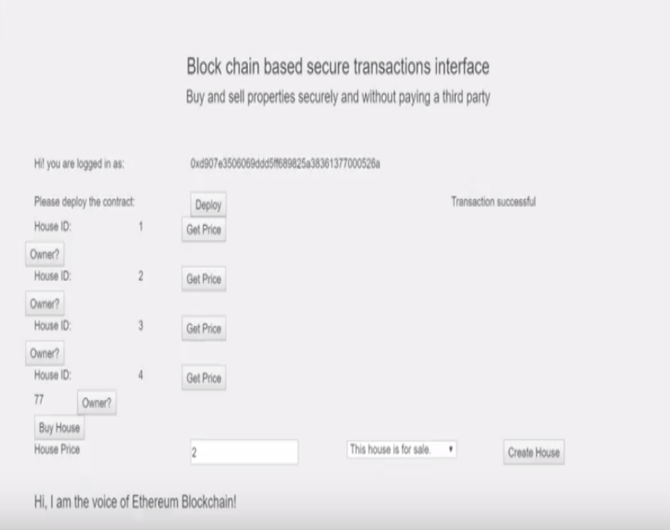
0xd907e3506069ddd5ff689825a38361377000526a

Blockchain!

Figure 16: Mining a contract using Metamask

1. At the start the contract is mined on the ethereum blockchain network so that our future transactions can be done safely and securely on the blockchain test network. This requires some gas, that is an incentive for the miner that mines this contract. See figure 15. By default we have created a list of 4 houses assigned to the user that mines the contract. See figure 16 for more information.

2. The contract is then saved on the ethereum blockchain network, for this project we are using the Ropsten test network which is a network for testing the applications before deploying on the main ethereum network.



Block chain based secure transactions interface
Buy and sell properties securely and without paying a third party

Hi! you are logged in as: 0xd907e3506069ddd5ff689825a38361377000526a

Please deploy the contract:

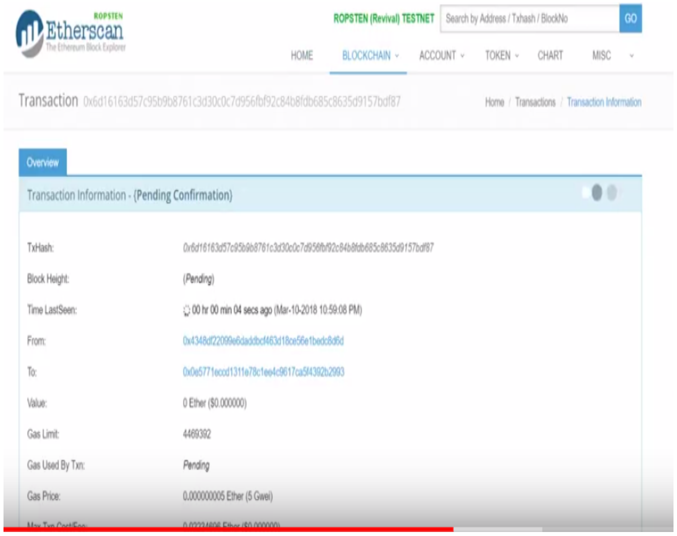
Transaction successful

House ID: 1
House ID: 2
House ID: 3
House ID: 4
77
House Price 2 This house is for sale.

Hi, I am the voice of Ethereum Blockchain!

Figure 17: List of Default houses

3. Now we can try different operations available for example, we can find the current owner of any house. This is a new transaction and again requires gas for completion. We can also see the transaction in progress using the "ropsten.etherscan.io" website. See figure 17 for more information. With the help of this website, if we have the contract address we can see all the transactions that have occurred on that contract too. See figure 18 for more information.

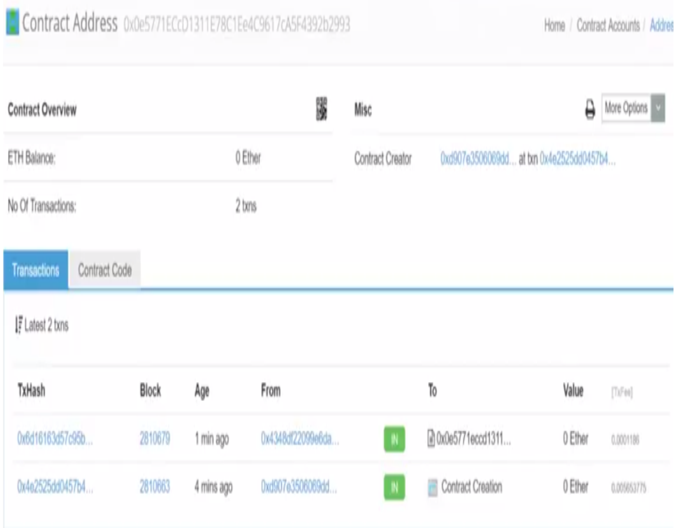

ROPSTEN
Etherscan
The Ethereum Block Explorer
ROPSTEN (Revival) TESTNET
Search by Address / Txhash / BlockNo
GO
HOME
BLOCKCHAIN
ACCOUNT
TOKEN
CHART
MISC
Home / Transactions / Transaction Information
Overview

Transaction 0x6d16163d57c95b9b8761c3d30c0c7d956fbf92c84b8fdb685c8635d9157bdf87

Transaction Information - (Pending Confirmation)

TxHash: 0x6d16163d57c95b9b8761c3d30c0c7d956fbf92c84b8fdb685c8635d9157bdf87
Block Height: (Pending)
Time LastSeen: 00 hr 00 min 04 secs ago (Mar-10-2018 10:59:08 PM)
From: 0x4348df22099e6daddbcf463d18ce56e1bedc8d6d
To: 0x0e5771eccd1311e78c1ee4c9617ca5f4392b2993
Value: 0 Ether ($0.000000)
Gas Limit: 4469392
Gas Used By Txn: Pending
Gas Price: 0.000000005 Ether (5 Gwei)

Figure 18: Viewing transaction on EtherScan.io



Contract Address 0x0e5771ECcD1311E78C1Ee4C9617cA5F4392b2993

Contract Overview
ETH Balance: 0 Ether
No Of Transactions: 2 txns

Misc
Contract Creator 0xd907e3506069dd... at txn 0x4e2525dd0457b4...

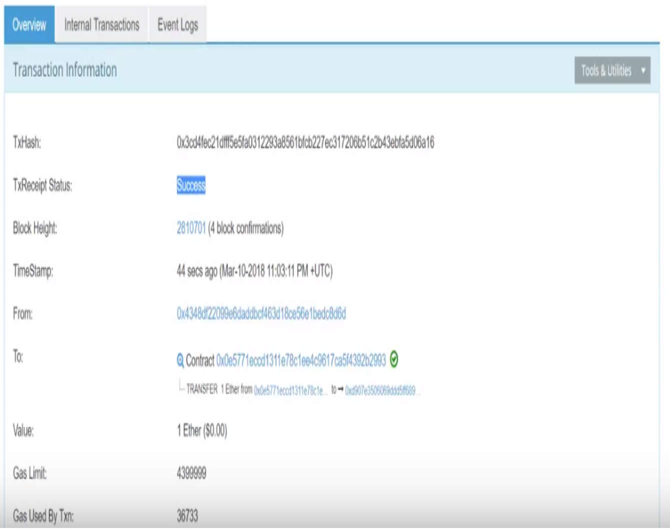
Latest 2 txns

TxHash	Block	Age	From		To	Value	[TxFee]
0x6d16163d57c95b...	2810679	1 min ago	0x4348df22099e6da...	IN	0x0e5771eccd1311...	0 Ether	0.0001186
0x4e2525dd0457b4...	2810663	4 mins ago	0xd907e3506069dd...	IN	Contract Creation	0 Ether	0.005653775

Figure 19: Transactions on a contract

4. So now we can buy a house using the ethers and we can see that transaction and amount of ether transfered as a transaction in the ropsten.etherscan.io website. See figure 19 for more information. This is the main advantage of blockchain, the data is stored on multiple nodes, and for the public to see. So updating any node would be nearly impossible.

5. Basically, the contracts are first deployed on the peer-to-peer network of the Ropsten Test Network. The users interact with the front-end and our application uses Web3 API to redirect any interaction to the peer-to-peer network where the contracts are hosted. The consensus algorithm is run by each node on the peer-to-peer network and an event is raised, informing the front-end about the success or the failure of the transaction.



TxHash: 0x3cd4fec21dfff5e5fa0312293a8561bfcb227ec317206b51c2b43ebfa5d06a16

TxReceipt Status: Success

Block Height: 2810701 (4 block confirmations)

TimeStamp: 44 secs ago (Mar-10-2018 11:03:11 PM +UTC)

From: 0x4348df22099e6daddbcf463d18ce56e1bedc8d6d

To: Contract 0x0e5771eccd1311e78c1ee4c9617ca5f4392b2993 ✓

└ TRANSFER 1 Ether from 0x0e5771eccd1311e78c1e... to → 0xd907e3506069ddd5ff689...

Value: 1 Ether ($0.00)

Gas Limit: 4399999

Gas Used By Txn: 36733

Figure 20: Buying a house transaction

10. TESTING

For testing the solidity smart contract code we used Remix solidity IDE available online fig 20.
For testing the webapp we checked if all the buttons were successfully working and doing the intended task. We also checked by printing the outputs in the browser itself.

10.1 Unit testing

Unit testing of Blockchain data integrity, validating encryption and hashcode computation of each block.

10.2 Front-end testing

It would help us validate methods in our blockchain system. This is essentially similar to API testing where one would use method validations, boundary value analysis, decision tables, test driven development and behavior driven development techniques.

10.3 Integration testing

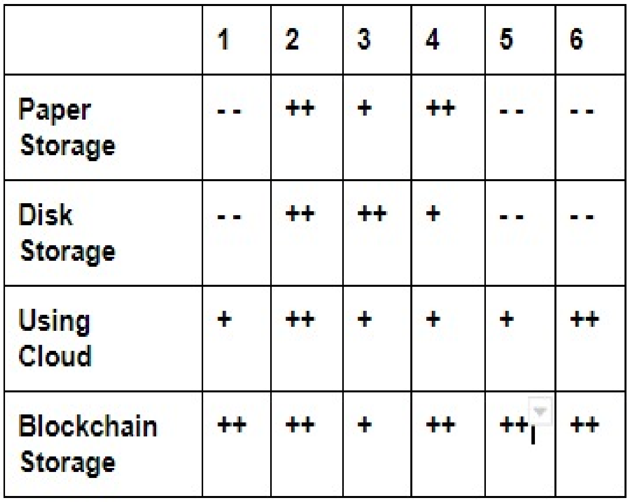
This involves integration testing which emphasizes on the performance and consistency testing between various nodes in the peer to peer network.

11. PROJECT EVALUATION

In this section, we have described analysis of results for evaluating our project by collecting responses using Google forms. We asked different questions while taking the survey to analyse the project on various parameters. We have described the reasons to ask a particular question followed by conclusions after collecting the responses.

11.1 Question 1

How would you rate the application overall ?
We asked this question to understand the overall user experience when they are actually using the application to judge on the basis of various parameters including:
Security
Accessibility
Adaptability
Convenience
Efficiency
Storage Experience.



	1	2	3	4	5	6
Paper Storage	- -	++	+	++	- -	- -
Disk Storage	- -	++	++	+	- -	- -
Using Cloud	+	++	+	+	+	++
Blockchain Storage	++	++	+	++	++	++

Figure 21:

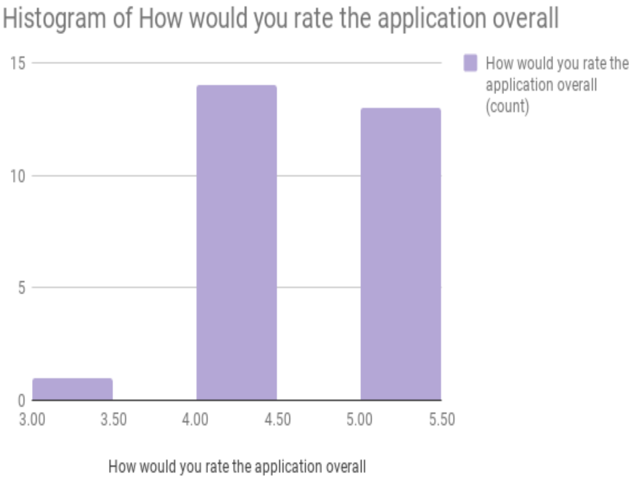


Figure 22:

11.2 Question 2

How appealing is the idea to you? //We asked this question to collect their response to how crazy the idea is and whether they find it worth exploring. We also want to know whether they find it as an effective solution to solve real-world customer problems.

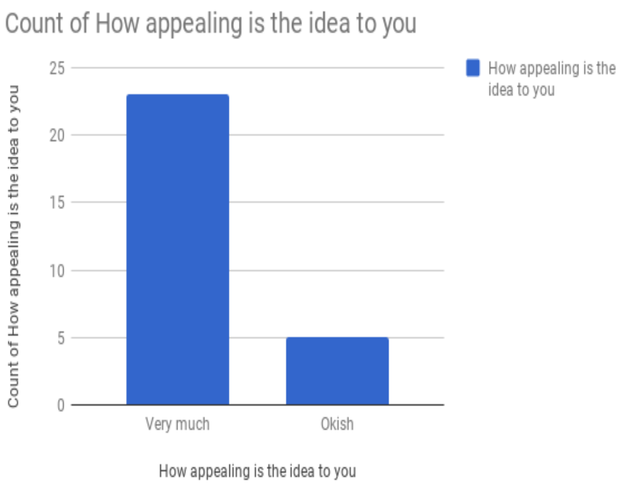


Figure 23:

11.3 Question 3

How useful do you think this project will be?
We asked this question to understand user response of their understanding about the utility of the project and how convenient they find it to use blockchain based storage system. This was done to make sure the users understand the advantages of using this system.

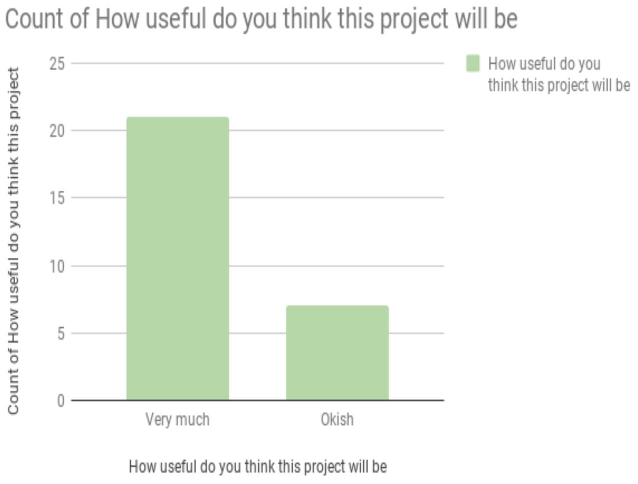


Figure 24:

11.4 Question 4

Are you clear about the project after reading the readme file and watching the video?
We asked this question to understand how intuitive the project is. We added the repository link, the readme file, video link and attached a presentation for the user to be able to understand how to use the application.

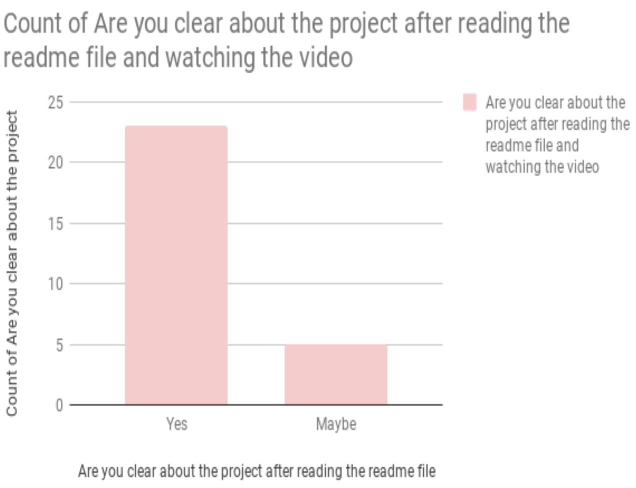


Figure 25:

11.5 Question 5

Were you able to buy a house ?
We asked this question to understand if after following the steps described in our project Readme file, the user is able to follow the steps and use the functionality.

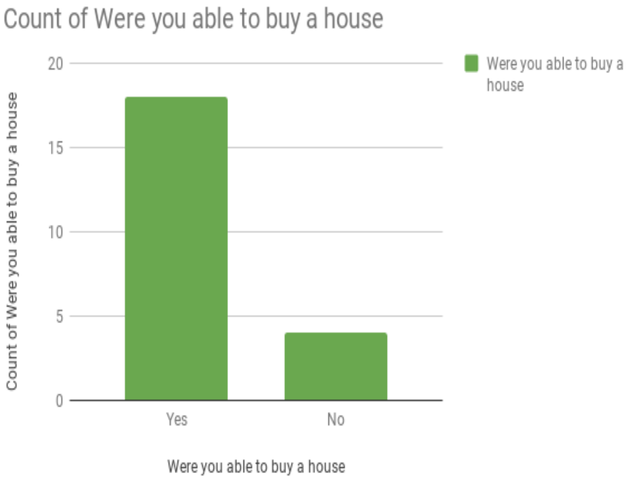


Figure 26:

11.6 SURVEY CONCLUSION

Based on the 31 responses we got for our project evaluation survey, we concluded that there are a large number of people who use unsecured and inefficient methods of data storage such as Paper Storage and Disk Storage. Most of them have migrated to newer methods of storage like Cloud which is much more powerful but comes with a cost. Though it is an efficient way of storing data, people still cannot trust it for storing contracts. In this context, since data on the blockchain is copied on every computer that is a part of the P2P network and there is no central body which governs whether a particular transaction should be recorded or not, it becomes completely secured, accessible, adaptable, efficient and convenient way. Example : Specifically in situations where data can be forged such as property, share market or vehicle renting system. The user response agrees and look forward to using such technologies to create innovate models to store and create contracts for such transactions.

12. POSSIBLE FUTURE WORK

12.1 Vehicle renting

The project can be extended to implement a blockchain based data storage such that many cumbersome tasks would be circumvented.The vehicle rental process is often more cumbersome than it needs to be, with insurance documents and identities that need to be verified, and vehicle mileages and damage reports that are still manually verified in many cases.But the amount of money, cost and labour involved in maintaining this and additionally there is a history of lawsuits of corruption activities which requires protecting this data which can be forged by miscreants for stealing.

12.2 Information of Stock markets

Another use-case which can be implemented using this is the important property which is non-physical such as shares of a company or stocks of a person. Banks, financial institutions and many others are adopting blockchain technology faster than anticipated. It is common to say that real world will range from the increasing demand for distributed ledger technology; reduced total cost of ownership; the rise of cryptocurrencies and their market caps, as well as initial coin offerings; increasing demand for simplified business processes; creating transparency and immutability; quicker transactions and the increasing adoption of blockchain-as-a-service.

13. BLOCKCHAIN SCOPE

The blockchain technology has shown a lot of potential in industry and academia. The following are the possible future directions. There can be main areas of future work combined with Blockchain technology: blockchain testing, big data analytics, and blockchain application.

13.1 Blockchain Testing

Recently different kinds of blockchains have appeared but some developers might falsify their blockchain performance to attract investors driven by the huge profit. So, Blockchain testing mechanism needs to be in place to test different blockchains as users want to combine blockchain into business, they have to know which blockchain fits their requirements. .Blockchain testing could be separated into two categories: standardization phase and testing phase. In standardization phase, all criteria have to be made and agreed. When a blockchain is born, it could be tested with the agreed criteria to valid if the blockchain works fine as developers claim.

13.2 Big Data Analytics

Blockchain could also be combined with big data. Here we roughly categorized the combination into two types: data management and data analytics. As for data management, blockchain could be used to store important data as it is distributed and secure. Blockchain could also ensure the data is original. For example, suppose blockchain is used to store patients health information, the information cannot be tampered and it is hard to steal that private information. When it comes to data analytics, transactions on blockchain could be used for big data analytics. For example, we can extract user trading patterns to predict their potential partners' trading behaviours with the analysis.

13.3 Blockchain Applications

Currently, most blockchains are used in the financial domain but traditional industries could take blockchain into consideration and use the application of blockchain into their fields to enhance their systems. For example, user reputations could be stored on blockchain. At the same time, the up-and-coming industry could make use of blockchain to improve performance. For example, In Raleigh, a ridesharing startup offers an open marketplace where riders connect directly with drivers by leveraging blockchain technology.

14. CONCLUSION

The advent of this distributed, decentralized technology is taking information storage and security to new heights. With our generic front-end platform (restricted for now) which can take a variety of inputs (highly specific for now) for creating a transaction in a set of domains, the user can create a distributed, decentralized and an online transaction block, which is visible to everyone in the p2p network. Moreover, since it is highly secure, none of the agencies need to verify the data once the data is officially verified and added to a block. We believe that with comprehensive research on the consensus algorithm (that is used to accept the block into the transaction list to all the systems) and a robust for verifying documents, this technology can truly change the face of how sensitive information can be made fraud-proof. Data on the blockchain is copied on every computer that is a part of the P2P network and there is no central body which governs whether a particular transaction should be recorded or not, it becomes completely secured, accessible, adaptable, efficient and convenient way. The user response agrees and look forward to using such technologies to create innovate models to store and create contracts for such transactions.

15. ACKNOWLEDGEMENTS

We would like to extend special thanks to our Professor, Timothy Menzies and our mentor, Ken Tu for helping us to come up with this idea and giving us the opp

16. REFERENCES

1. Blockchain Technology Beyond Bitcoin: http://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf
2. Blockchain technology inovations: http://ieeexplore.ieee.org/document/7998367/
3. https://www.quora.com/What-are-the-key-properties-of-the-Bitcoin-blockchain
4. An Overview of Blockchain Technology: Architecture, Consensus and Future Trends: http://ieeexplore.ieee.org/document/8029379/#full-text-section
5. The Digital Privacy Paradox: Small Money, Small Costs, Small Talk: https://papers.ssrn.com/sol3/papers.cfm