

TAMPER RESISTANT DATABASE LOG UPDATE VIA BLOCKCHAIN



By

**Muhammad Saad Malik
Muhammad Umer
Syed Muhammad Mahdi Raza
Zahra Shiraz**

Supervised By

Asst. Prof. Dr. Fawad Khan

Submitted to the Faculty of Electrical Engineering Department,
Military College of Signals, National University of Sciences and Technology, Islamabad
in partial fulfillment for the requirements of a B.E Degree in
Electrical (Telecommunication) Engineering

June 2019

CERTIFICATE FOR CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

“TAMPER RESISTANT DATABASE LOG UPDATE VIA BLOCKCHAIN”

Is carried out by:

Muhammad Saad Malik

Muhammad Umer

Syed Muhammad Mahdi Raza

Zahra Shiraz

with plagiarism of __ under my supervision and that in my judgment, it is fully ample,

in scope and excellence, for the degree of Bachelor of Electrical (Telecomm)

Engineering from Military College of Signals, National University of Sciences and

Technology (NUST).

Approved by

Asst. Prof. Dr. Fawad Khan

IS Department

Military College of Signals, NUST

DATED: June 2019

ABSTRACT

Blockchain is a new emerging technology which has received great attention in recent years. It is a global online immutable and transparent database which anyone and anywhere with an internet connection can access. The feature that makes it unique is its decentralization. This means the blockchain ledger is shared among all computers around the world, if it is built on public platform, not in one central location. The above discussed features of blockchain i.e. immutability, transparency and decentralization are considered very important in many fields e.g. Maintaining Patient Record, Supply Chain Management, Automotive Industry and Asset Transfer. In this project a specific application of Asset Transfer is made i.e. Registration and Transfer of Vehicle Ownership. All this process is done by writing a *Smart Contract*. As this application is being built on a public platform so we used Ethereum. It is the most used *Smart Contract* framework, which is especially designed to support *Smart Contracts*. It is programmed in Solidity language and prevents the possibility of downtime, censorship, fraud, or third-party interference. By bringing Blockchain into the picture and moving the entire vehicle registration process on to Blockchain risk of attacks and fraud can be reduced as data updates are only possible by authorized personnel using a private key. In fact, any tampering of data can also be easily tracked on Blockchain. The best part is. Blockchain will provide a single and easy view of the vehicle lifecycle which is not available today.

Copyright by

Muhammad Saad Malik

Muhammad Umer

Syed Muhammad Mahdi Raza

Zahra Shiraz

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our Faculty, without whose unflinching support and cooperation, a work of this
magnitude would not have been possible.

And our Parents for their support.

ACKNOWLEDGEMENTS

In the name of Allah, Most Gracious, Most Merciful

We would like to thank Allah Almighty for His incessant blessings which have been bestowed upon us. Whatever we have achieved, we owe it to Him, in totality. We are also thankful to our families for their continuous moral support which makes us what we are.

We are extremely grateful to our project supervisor Asst. Prof. Dr. Fawad Khan from MCS who in addition to providing us with valuable technical help and guidance also provided us moral support and encouraged us throughout the development of the project.

We are highly thankful to all our teachers and staff of MCS who supported and guided us throughout our course work. Their knowledge, guidance and training enabled us to carry out this whole work.

Finally, we are grateful to the faculty of Electrical (Telecom) Department of the Military College of Signals, NUST

Table of Contents

Contents

LIST OF FIGURES	X
LIST OF ABBREVIATIONS	XI
INTRODUCTION.....	1
1.1 OVERVIEW.....	1
1.2 PROBLEM STATEMENT	2
1.3 SOLUTION	3
1.4 APPROACH.....	4
1.5 SCOPE	5
1.6 AIM & OBJECTIVES	6
1.6.1 Research Objectives.....	6
1.6.2 Academic Objectives.....	6
1.6.3 Commercial Objectives.....	6
1.6.4 Other Objectives	6
1.6.5 Organization	6
LITERATURE REVIEW	8
2.1 BACKGROUND	8
2.2 INTRODUCTION TO BLOCKCHAIN	8
2.2.1 Blockchain and Bitcoin.....	8
2.3 WORKING MECHANISM OF BLOCKCHAIN	9
2.4 IS BLOCKCHAIN SECURE?	10
2.5 TYPES OF BLOCKCHAINS.....	11
2.5.1 Public Blockchains.....	12
2.5.2 Private Blockchains	13
2.5.3 Federated Blockchains or Consortium Blockchains.....	13
2.6 APPLICATIONS OF BLOCKCHAIN	14
2.6.1 Financial Services.....	14
2.6.2 Asset Management	14
2.6.3 Insurance.....	15

2.6.4Blockchain Healthcare.....	15
2.6.5Blockchain Internet-of-Things (IoT).....	15
2.6.6Blockchain Identity	15
2.6.7Blockchain Government.....	16
2.7 MAINTAINING VEHICLE HISTORY USING BLOCKCHAIN	16
DESIGN AND METHODOLOGY	19
3.1 SYSTEM DESIGN.....	19
3.2 SMART CONTRACT.....	20
3.2.1 Solidity Language	21
3.2.2 Remix.ide/ Truffle.....	22
3.2.3 Test-rpc/ Ganache.....	23
3.3 DATABASE.....	23
3.3.1 Relational database	24
3.3.2 Distributed database.....	24
3.3.3 Cloud database	24
3.3.4NoSQL databases.....	25
3.3.5 Object Oriented Database	25
3.3.6 Graph database.....	25
3.3.7 Database Integration With VOTS.....	26
3.4 HARDWARE	27
3.4.1Fingerprint Sensor Integration.....	27
3.4.2Hardware Integration with Smart-Contract	28
3.5 DAPP DESIGN.....	30
3.5.1 DApp development.....	30
3.5.2 Front-end for DApp design.....	31
3.5.3 MetaMask Injection for DApp.....	31
3.5.4 Hyper Text Markup Language (HTML).....	32
3.5.5 Cascading Style Sheet (CSS).....	32
3.5.6 PHP	32
3.5.7 Web3.js.....	33
3.5.8 Lite Server.....	34
TEST AND DEVELOPMENT	35
4.1 CONTRACT DEVELOPMENT USING TRUFFLE.....	35

PROJECT ANALYSIS AND EVALUATION	42
5.1 VOTS V.1.0 PROTOTYPE WORKFLOW	43
CONCLUSION	45
6.1 PURPOSE	ERROR! BOOKMARK NOT DEFINED.
6.2 PREVIOUS RESEARCH	ERROR! BOOKMARK NOT DEFINED.
6.3 LIMITATIONS	ERROR! BOOKMARK NOT DEFINED.
6.4 RECOMMENDATIONS	ERROR! BOOKMARK NOT DEFINED.
APPENDICES	48

List of Figures

FIGURE 3.1: SYSTEM DESIGN WORKING DIAGRAM.....	20
FIGURE 3.2: FLOW DIAGRAM OF SENSOR WORKING.....	29
FIGURE 3.3: R307 INTEGRATION WITH RASPBERRY PI.....	29
FIGURE 3.4: STRUCTURE OF A DECENTRALIZED APPLICATION	30
FIGURE 3.5: METAMASK AS A BRIDGE BETWEEN CHROME AND ETHEREUM NETWORK	31
FIGURE 3.6: VOTS FRONT-END.....	33
FIGURE 3.7: FRONT-END OF VOTS AUTHORITY	33
FIGURE 3.8: DAPP ARCHITECTURE.....	34
FIGURE 4.1: CONTRACT COMPILATION ON TRUFFLE	35
FIGURE 4.2: DEPLOYED CONTRACT METADATA.....	36
FIGURE 4.3: BLOCK INFORMATION ON TESTRPC	36
FIGURE 4.4 - INTERACTION WITH SMART CONTRACT USING COMMAND PROMPT	37
FIGURE 4.5: CONTRACT TEST USING SCRIPT	37
FIGURE 4.6 - FRONT END OF DAPP	38
FIGURE 4.7: PERSON REGISTRATION VIA FRONT-END	39
FIGURE 4.8: PERSON REGISTRATION CONFIRMATION VIA EVENT	39
FIGURE 4.9: VEHICLE TRANSFER ON VOTS	40
FIGURE 4.10: VEHICLE TRANSFER UPDATE.....	40
FIGURE 4.11: ERROR FOR VEHICLE TRANSFER.....	41
FIGURE 5.1: VOTS ECOSYSTEM	43

LIST OF ABBREVIATIONS

ABI	Application Byte Interface
CLI	Command Line Interface
DApp	Decentralized Application
ETH	Ether (Crypto-currency)
ETO	Excise and Taxation Office
EVM	Ethereum Virtual Machine
GUI	Graphical User Interface
PoW	Proof of Work
VOTS	Vehicle Ownership Transfer System
SQL	Structured Query Language

CHAPTER 1

INTRODUCTION

1.1 Overview

Asset management is an important aspect of human life; therefore, people spend large amount on buying as well as on maintenance of their assets. Nobody wants their assets to be used by someone else. Obviously, there are so many risks if some unknown person uses someone's asset or property. Government has established different regulatory authorities to ensure proper registration of assets in their database to keep track of all the true owners of every asset. Thus, different organizations work together to register, monitor and resolve conflict between parties. There are multiple types of asset and each of them needs such collaboration to make things happen.

Vehicle is also an important asset on which people invest their savings, hence, need of collaboration between organizations i.e., Excise and Taxation Department (E&T dept.), Justice Department and Insurance Agencies play necessary role for vehicle registration. It is necessary to share updated information of all the vehicles and their owners in all the departments in order to make all the processes transparent.

Presently working system for transference of ownership and registration of vehicles is prone to errors because of the centralized system; error in one system means error everywhere and there is no possible solution to distinguish wrong information from the correct information. Another issue arises from this unreliable updating of data; people who want to buy used vehicles don't have any authentic and trustworthy source of information about the vehicle as well as the past owners and major accidents or criminal activities in which the car had been involved. From these problems of asset management, a solution has been proposed of which the underlying technology is *Blockchain*.

1.2 Problem Statement

George Akerlof a Nobel Prize winning economist briefly explains the problem in his 1970 paper, *"The Market for 'Lemons': Quality Uncertainty and the Market Mechanism"*. In this paper he focuses on different mechanics and explains about the aspects of market where one part (seller) relatively knows more about product as compared to second party (buyer). This non-uniform distribution of information is also termed as *"markets with asymmetric information"*.

An example of this is the used car market. Whenever a car is being sold, the seller, usually, knows more about the car as compared to the buyer. This is a problem for the buyer because the buyer doesn't know what he is buying. To hedge against the risk of buying a so-called 'bad' car, they reduce the price they're willing to pay for the vehicle. This could lead to the entire used car market to disappear.

Thousands of used cars are sold every day and every new owner wants complete and reliable history of the vehicle being bought. Every vehicle has on average 3 to 5 owners during its lifetime, it is clear that 80 percent of vehicle owners do not know anything about their vehicles. This is because there is no system to authenticate the information of centralized database of vehicles. There is always a chance of inaccuracy in centralized database; it can be hacked, the only owner of database can change the information on purpose after taking some bribery or the centralized database can be eradicated. All these situations can result into drastic effects.

Here's how it happens:

- A buyer cannot fully distinguish a good car from a bad one, so they lower the amount they are willing to pay for the vehicle to compensate for possible unknown risks. This reduces the average price of the vehicle
- This leads to higher priced 'good' cars to leave the market, deteriorating the overall quality of the secondary market

- The repercussion is a further reduction in price of available vehicles. As a result, ‘medium’ quality cars are also pushed out of the market
- If the cycle continues the quality of available cars will continue to deteriorate until the buyers pull out of the market completely. This is due to the perception that all second-hand vehicles are of poor quality
- As a result, this will lead to the complete disappearance of the market and can only be stopped by introducing more symmetrical information. This is the nature of markets with asymmetric information

Blockchain comes to the rescue and minimize all these problems. Using *Blockchain*, there is no centralized database. As the blockchain exists on the internet, it becomes decentralized. This means that there is no single controlling authority; every node has a copy of database and if someone tries to make change in one node database all the other nodes have to approve it first. If there is any intruder, he can be caught very easily on *Blockchain* network. These controlling powers are defined in a piece of code that is known as *Smart Contract*. In the current situation data updates are only possible by authorized personnel; therefore, this system will also help to reduce terrorist attacks done by vehicles, other criminal activities and frauds. The best part is, Blockchain will provide a single and easy view of the vehicle lifecycle which is not available today.

1.3 Solution

Transfer of vehicle ownership is a cumbersome process and takes so much time, paperwork and money. Presently, the manual system doing this job is not transparent and hence unreliable. To solve all these problems, we design *Vehicle Ownership Transfer System (VOTS)*, to store detailed information of vehicle in de-centralized and distributed database. As it is a distributed system, so an update made by one department is reflected in all other departments. This solves the problem of outdated data across the network by assuring accessibility of data.

When it comes to asset transfer vehicles are widely used across the world and there are many dealers or normal users that buy and sells their vehicles. By using Blockchain in

such type of dealings would increase transparency of ownership transference and all the process would be trustworthy

Further, this will save time for the tedious tasks and provides with the good management and history of all the previous actions and transfer of ownership.

Data integrity and immutability is also ensured by the proposed solution thus assuring that the complete information stored is accurate and up to date and data is maintained and shared across the network in secure way being cost efficient. Some key features of VOTS are as follows:

- **Improved data integrity and real-time visibility:** Two major features of *Blockchain* are decentralization and transparency will allow the agencies to maintain all the data in a symmetric manner and assure easy access to the database to assess the vehicle ownership
- **Improved collaboration and information sharing:** As it is a distributed system, all the departments and agencies will have exactly same data and no change can be made without the consent of other department. In this way, all the data is updated and there is no confusion about the accuracy of the database
- **Reduced fraud:** Information about all the owners of a vehicle is stored in an immutable fashion. Any fraud or criminal activity done by a vehicle can easily be investigated by the true owner of that vehicle at that time
- **Reduced cost of operations:** Cost of transferring vehicle ownership is around PKR 1000/- and the time and energy saved by this system is another major advantage of this system. As this system is built on a public platform the limited information about any vehicle can be seen by anyone free of cost

1.4 Approach

In this section we will briefly go through the road-map of project and later in Chapter 3 explanation of every step is mentioned. The approach to get desirable results is as follow: Initially research work about *Blockchain* was done. **White paper** published by *Satoshi Nakamoto*[1]in 2009 “**Bitcoin: A Peer-to-Peer Electronic Cash System**”, **White paper**

published by *VitalikButerin*[2] “**Ethereum White Paper: A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM**” and **Yellow paper** published by **DR. GAVIN WOOD**[3] “**ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER**”

- Suitable platform was selected, in our case Ethereum, to proceed further for the development, where the process was initialized by writing smart-contract
- Once a basic smart-contract was written that fulfilled the desired conditions and generated required results, we moved forward to deploy the *Smart Contract* on local test net and started its integration with other parts of project
- To make this system compatible with the already existing system, a centralized database (like NADRA records) was integrated with the *Blockchain*. This database is used whenever a new person is registered in this new system
- The authenticity of a person is checked via his fingerprint; therefore, fingerprint sensor is integrated with raspberry pi and both of these things have to be integrated with *Smart Contract* in order to automate the system
- User Interface is designed so anyone can use system very easily, where he can access the desired information from *Blockchain*
- After final integration of all the parts of the project, the *Smart Contract* is deployed on **Ropsten Network**, then the contract is tested according to different conditions and the results are compared with the desired results
- Based on results, debugging and improvement in contract is made to ensure proper and reliable working of system

1.5 Scope

The scope of this project is to familiarize the world with the internationally introduced new technology *Blockchain*. On international forums, there has been many pieces of work done in this area but in Pakistan people still are not aware of this new technology. It has been said that *Blockchain* will revolutionize all the work fields. It is better to learn and understand it before its implementation in every area. Also, this fully automated system can be used on national level to keep records of all the vehicles in order to aid

general public and to reduce the frauds and criminal activities done using vehicles which are later disowned by everyone and criminal escapes easily.

1.6 Aim & Objectives

VOTS is a prototype solution that aims to provide decentralized system for asset tracking and management and shares symmetric information between two parties and government organizations. Other than this, following objectives are set to achieve.

1.6.1 Research Objectives

- Study and research about Blockchain technology and its different Platforms

1.6.2 Academic Objectives

- Working in the field of cryptography
- Learn *Solidity* by writing *Smart Contract*
- Practice *JavaScript* coding skills to improve *Solidity* skills
- Boost programming skill in HTML and PHP programming to make an appealing and eye-catching user interface

1.6.3 Commercial Objectives

- Design a *Blockchain* based system for vehicle ownership to ensure symmetric information between authorities
- Aid buyer to get maximum and transparent information of vehicle

1.6.4 Other Objectives

- To introduce Blockchain Technology and get familiar with this new technology

1.6.5 Organization

For clear understanding this report is comprised of five sections, short description of each section is:

- The first section includes *Abstract* to describe necessary details of project, followed by *Introduction* to specify problem statement with brief *Approach* of solution to achieve goals and objective of project
- The second section includes Literature Review in which *Blockchain* is introduced as it is relatively a new technology. It contains several features, advantages and disadvantages of *Blockchain* and a portion explains how *Blockchain* is used in this project
- The third section includes Design and Methodology in which project is divided into four parts and detail of all these parts is discussed
- The fourth section includes the explanation of the end product i.e. Decentralized Application on Ethereum Platform for Transference of Ownership or Registration of Vehicles
- The fifth section includes Project Analysis and Evaluation with complete description of the work flow
- The sixth section includes Conclusion and Recommendations for future work that can be carried out by other people in future
- The seventh section includes Bibliography and Appendices with synopsis attached at the end of the document

LITERATURE REVIEW

2.1 Background

In Pakistan, over 62,000 cars were imported in the year 2017. Do the new owners of these vehicles know the complete history of them? Do these vehicles really have all the genuine parts? Were these vehicles used in any criminal activity? Was the previous owner a reliable person? These are some of the questions that everybody needs to know while buying a vehicle. To overcome these problems the idea of Vehicle Verification System has been proposed. It uses blockchain which is today's most popular emerging technology.

2.2 Introduction to Blockchain

Blockchain[4] has impacted every area and domain of the industry and has been adopted everywhere because of its trust building factor in any event occurred. It is distributed ledger required to maintain permanent and tamper-proof record of data. It is decentralized system that excludes the need of any central authority and allows us to trust the outputs of the system. Blockchain provides decentralization because of which the ledger is shared on a public platform and it is not controlled by a single entity. The information is stored in the form of blocks and these blocks are chained together and make a continuously growing database of transactions and are secured cryptographically from any sort of tampering. In short, *Blockchain* is an open distributed ledger which is used to keep the record of transactions between different parties in a verifiable and permanent way.

2.2.1 Blockchain and Bitcoin

During the past year everyone everywhere went crazy about crypto-currencies and in particular these-called crypto gold Bitcoin. Underlying technology of Bitcoin is Blockchain. It was introduced by the anonymous person Satoshi Nakamoto who

introduced Bitcoin in his white paper “Bitcoin: A Peer-to-Peer Electronic Cash System”[1]. The proposed white paper replaces the need of intermediaries or authorities like banks and financial institutions to facilitate transactions:

“What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.”

Because of bitcoin rather depending upon any third trusted intermediaries like banks to do any action like sending or receiving money or making transaction, it maintains a network of different users which are peer to peer connected and any action or transactions are done on the basis of majority, which in this terminology is called consensus mechanism. The set rules of bitcoin govern how the users in this very network will interact with each other. They characterize:

- The conditions under which a transaction is valid.
- Cost of a Transaction for sending money from one party to another.
- Mechanism of using cryptographic token to validate the transactions.
- Rules on how to change current consensus mechanism.

2.3 Working Mechanism of Blockchain

As already mentioned blockchain is a decentralized and immutable system, this portion will discuss about the working of blockchain i.e. how it is immutable, decentralized and how information gets stored in blocks.

Blockchain is just a series of blocks that are linked with each other. So when the word blockchain is used, it is meant that digital data (in form of block) is stored in public database and are linked(chain) with one another.

When new data is added to this public database it is stored as digital information in the form of blocks. When multiple blocks that store information are joined together then it is called blockchain. To add new block in the ongoing blockchain, however, these four things must occur:

1. There must be some transaction made. Take an example of your online shopping on any website let's say Amazon. After finding something you want you will make a transaction by purchasing it.
2. This transaction made for purchasing that product is then needed to be verified. This transaction first must be confirmed by someone after that the purchasing process will be completed for this there must be someone accountable for confirming these transactions or data entries. All of this is much easier when using blockchain as this job is left to be done by the network consisting of many computers (or in case of Bitcoin, around 5 million) connected with each other. When you buy something from Amazon, this network confirms whether the transaction you made occurred as you wanted it to be or not. Details of the transaction are verified regarding amount, total participants and the time at which the transaction is made.
3. This transaction must have to be stored. Once the transaction is made and verified to be accurate, digital signature of buyer and that of Amazon and the amount of transaction are all stored in a block where it joins hundreds and thousands of others like it.
4. There must be a hash generated for a specific block. Once all of the transactions of a block have been verified, a unique number must be given to it, so it can be identified. This exceptional code for identification is called hash. Hash of the previous block is also added in current block and after giving block a specific hash it is then added to blockchain.

After addition of the new block in the ongoing blockchain it become it turns out to be freely accessible for anybody to see and have access to transaction data. Taking example of blockchain of Bitcoin, one has access to transaction information along with data about when, where and by whom he blocks was added to this blockchain.

2.4 Is Blockchain Secure?

Blockchain technology represents the issues of trust and security in a few different ways. Initially, new blocks are directly stored in sequential and chronological order. Because of this after a block is mined it is added at the end of blockchain. Size of each block is 1MB. As an example, the bitcoin blockchain has been growing and till September 2018 it is of

size 185GB since 2009 when bitcoin crypto currency was created and first used as virtual money. Blockchain is immutable as nobody can modify data or distributed ledger of all blocks because after creation of new block and to place it at end of whole chain makes it very difficult to go back to some previous block and alter its data. It is also ensured by concept of hashes; each block contain its block hash along with the hash of previous block. Hashes are codes in form of strings that created by complex math functions and algorithms which converts any form of data into a unique alphanumeric string. If information in any of previous block is changed then its hash will also change as well. If somebody alters the information for example in mid of blockchain then he has to change the hashes from mid of block chain to end of blockchain block by block step by step and after it is done this altered blockchain will be entirely different from the one that all the other nodes in the network have on their computers.

Here is for what reason this is critical to security. Let us suppose someone tries to alter your transaction of Amazon so you have to pay for your purchase twice. The hash of block will be changed as soon as the amount is changed. In order to cover this intruder has to change the hash of block and update it but this block contains the hash of the previous block as well. However, in this effort hashes would get changed block by block, and so on.

Thus, in attempt of changing data of a single block the hacker has to change every single block that comes after the block he wants to change. For this he must recalculate all these hashes that demands time and very high computational power. In short once a block is created and added then it is very difficult to edit it and not possible to delete it.

2.5 Types of Blockchains

The thought developed that the Bitcoin blockchain could be utilized for any sort of significant transaction or any sort of understanding, for example, peer to peer protection, trading, insurance and transferability and so on[5]. Colored Coins and Mastercoin attempted to tackle that issue by using Bitcoin Blockchain Protocol. The Ethereum venture decided to produce their own blockchain and deploy it on the main network

which will also be used by others who deploy their smart contracts on the Ethereum blockchain.

Institutions like banks realized that the underlying idea of bitcoin can be used as “Distributed Ledger Technology (DLT)” and establish a permissioned blockchain (private or federated), where the validator is from same organization. Thinking of blockchain as a permissioned private ledger is profoundly dubious and controversial. This is the reason the term “Distributed Ledger Technology (DLT)” developed as more general term.

Private Blockchain[6] may not revolutionize the world, however it holds the possibility to supplant most elements of customary money related establishments with programming, in a general sense reshaping the way the budgetary framework works.

Private Blockchain is important for settling proficiency, security and extortion issues inside conventional money related foundations, however just steadily. It's not in all respects likely that private blockchains will reform the monetary framework.

2.5.1 Public Blockchains

Currently the public blockchain is based on protocols like proof of work, consensus algorithm that are easily accessible and open source. Anyone can make their local device a public node and can be part of consensus by validating the ongoing transactions. Consensus algorithm is process that specifies which block is to be added to the blockchain and tells about the current state. Transactions can be made by anyone around the globe which is added to blockchain after validation. As it is public blockchain, so anybody can have access and see the transaction. Examples of public blockchain include Bitcoin, Litecoin, Ethereum, Dogecoin, etc. Public Blockchain lessens the expenses of making and running “Decentralized application (dApps)” without any need to maintain servers. It has zero infrastructure cost. It has an impact on current business models by intermediation.

2.5.2 Private Blockchains

Private Blockchain is also known as Permissioned Blockchain. In this type of Blockchain, permission is granted to the nodes that want to be a part of the Private Blockchain. This type of Blockchain is built in a case where the owner doesn't want to give read and write access to everyone. In this type of Blockchain, the concept of identity is used. This means that for every node of the network the owner of the Blockchain predefines some set of rules e.g. whether this node can read the data or not. This is possible only if the owner has the identity of all the nodes. Another important and interesting fact is that in private blockchain, everyone knows all the other nodes. If some unwanted activity happens, the whole network is going to know about that hideous person who is trying to tamper the data.

2.5.3 Federated Blockchains or Consortium Blockchains

Consortium Blockchains are somewhat related to Private Blockchains. We can say that these types of blockchains are semi-private. Unlike Public Blockchains, Consortium Blockchains are used in the places where the owner needs to retain the control and privacy of the organization for example banks. As said above it is semi-private blockchain so to mine and add each block on the blockchain the consensus is done, and rights are given to specified nodes that can validate every block other node have not any access to confirm any action or transaction, but they may can read what is happening on this blockchain. These different type of nodes with different rights are defined in predefined set rules.

Consortium Blockchains work under the association of a group. Rather than open Blockchains, they do not give permission to validate transactions to all persons. These are faster with higher adaptability and give more transaction security. Federated blockchains are for the most part utilized in the financial area. The consensus mechanism is constrained to specified nodes. Suppose there is bank that uses consortium blockchain and it has 20 branches and every branch is considered as a node. If 10 branches are given

rights to verify transaction when other nodes cannot participate in this consensus but they may read the blockchain. All this is defined in smart contract

Examples includes Corda, r3 (Banks), B3i (Insurance), Energy Web Foundation (Energy)

It is still controversial that whether this should be considered as blockchain or not. Blockchain is new and advancing technology and much work is yet to be done that's why it is still indistinctive that how it would be accepted and how will it stand. Many people hold the view that private or consortium blockchain may endure destiny of intranets like in late 90s when private organizations and businesses made their own LANs/WANs that are private to them. This is done so as to oppose the use of public internet and administrations which after the appearance of SAAS in Web2 it seems to be pretty much out of date.

2.6 Applications of Blockchain

2.6.1 Financial Services

The benefits that blockchain provides over traditional systems is that the already existing systems is dependent on trusted third party or Intermediaries to do the actions or transactions and run the process but using blockchain makes the process transparent, time efficient and cost effective by giving immutability without being error prone, inconvenient and unmanageable.

Many countries and banks are moving their services to blockchain like Malta, China etc.

2.6.2 Asset Management

In Asset management, trade processing and settlement can be of high-risk and expensive, especially when trade is between two different countries. Both the parties' buyer and seller keep and manage their own record of data, so it is more likely to have significant errors and inefficiencies. By block chain errors are reduce because of immutable data and giving independency to rely on intermediaries

2.6.3 Insurance

In insurance agencies some cases have been seen when person comes to claim insurance with fraud. Insurance agents have to see the whole case as if it is valid or not or if the person who claims had used any abandoned policy or some other wrong way and then it has to be processed manually. It is frustrating tasking and is prone to error. Blockchain prevents false claim or two people claiming a single vehicle by its set rules that use hashes and data cannot be changed once block is added thus giving transparency and immutability. So, providing risk free management, blockchain also save time by making the whole process very easy.

2.6.4Blockchain Healthcare

Individual health records can be stored on blockchain database, so it could be easily accessible and record management would be much easier. A similar procedure could be utilized to guarantee that exploration and research is directed by means of HIPAA laws (in a safe and classified way). Receipts of all the medical procedures done can be stored on a blockchain and can be sent to insurance agencies as proof of delivery. This database can also help for maintaining general health care records, such as drug supervision and regulatory authorities, to manage and test result and healthcare services.

2.6.5Blockchain Internet-of-Things (IoT)

Everyday objects when embedded with computerized and digital devices and connected to internet so it can send or receive data, it becomes Internet of Things. When connected to a network a normal object just not only behaves as it is, but it becomes so called smart objects or devices. These objects are now people connected to other people and objects and devices with one another. According to some analysts there will be more than 26 billion IoT devices.

2.6.6Blockchain Identity

Online companies have all the information about us because we gave them this data. Once we upload anything or give any information on internet it remains there forever after it has been deleted from our side. Some online companies from where we buy or

purchase something or do any transaction shares or sells our details to others which use this data and send us their ads according to known interests or information. The blockchain prevents this by making protected data point from where one can only encrypt the relevant information that is required at certain times. For example, going to bar only needs information about age to be shared to know if he is 21 or not.

2.6.7Blockchain Government

In most of the countries elections and its results are a big issue and most of the time security of the voting system is questioned or sometimes winning party is accused of rigging. This can be prevented by using voting system based on blockchain in which not only the votes will be encrypted but also individuals can cast their vote and confirm it easily. This voting system will be cost and time efficient for both the individuals and the government too

According to report of McKinsey and Company in 2013 the world would be richer by 2.6 trillion USD if freely accessible open government source data is accessible for all the individuals over the internet. New business ventures can use this information to unmask and expose the fraudulent schemes. It can also be used to investigate the side effects of medicines or used by farmers for precise farm-cropping. In the present scenario this information is released yearly which is largely, non-responsive to citizens input.

The blockchain provides open responsive data to the citizens anytime anywhere as they want. The blockchain, as a public ledger, can open this data to citizens whenever and wherever they want.

2.7 Maintaining Vehicle History using Blockchain

Throughout the lifetime of a vehicle, it will be involved in several kinds of historical transactions. From its parts being molded, assembled and tested in a factory, to its transport to its first car lot to be sold to its first owner, vehicle registrations, warranty information, all its oil changes, tire changes, part replacements, safety recalls, major and minor accidents, mileage data, change of ownerships and so on. This project focuses only on registration and transfer of vehicle ownership using Blockchain. This system will store vehicle details in distributed and decentralized manner in a database that is shared with

different organizations that require details of vehicle for any reason. Any update on vehicle details or incident by one department will be simultaneously changed in other departments also that use the same database. This ensures availability of consistent data across various departments. With this system, different stakeholders like manufacturers, dealerships, customers, and motor vehicle agencies can easily collaborate to access and update vehicle details based on their security access. The solution also shares most up to date, complete and accurate information in secure way that is also cost efficient.

2.7.1 Explaining the Project using an example

Let us suppose a new car is sold, now using a smart contract created by its owner, the new owner has this vehicle's un-tampered history from VOTS Blockchain. The owner knows that it is a zero-meter car and there was no previous owner. After several years of use, this owner sells the car to a new owner using the Smart Contract. Now this new owner gets the information of previous owners using VOTS Blockchain. In a similar fashion this transfer of ownership goes on and blockchain will keep record of this. After several years more of use, the car has finally reached the end of its VOTS Blockchain journey. Its final owner passes it to a scrap or junkyard for final decommission via Smart Contract. Once a final junkyard transfer occurs, VOTS Blockchain locking its history in place for all eternity. This car is then "off-the-record" and anything that happens to this car after this is deemed untrustworthy as it doesn't exist on the blockchain.

CHAPTER 3

DESIGN AND METHODOLOGY

In this chapter we will discuss about the system design and briefly explain every tool and language to produce end deliverable. A quick overview of all the prerequisites and development environment is also given in this chapter to familiarize the user with all the things used in this project.

3.1 System Design

VOTS design is divided into four parts that are going to be discussed in detail in next sections, here an overview of each part is listed along with their functionality:

1. *Smart Contract* is a set of logic on which complete system is based. This is the primary thing in designing this project that differs it from existing system.
2. A centralized *Database* for the verification of general public is used to make this system compatible with the already existing system. Information of users is retrieved from this database to confirm that the person is a citizen of this country in order to minimize fraud cases.
3. *Hardware design* is used for verification of person using his biometric fingerprint and on the bases of this majority of the functions are executed in this system.
4. *Front-end* for the design of *dApp* is the last part which will be the end product used by the user. User-friendly graphical user interface is provided to the user for all the functioning.

The working diagram of system is shown in *Figure 3.1* where it shows two different persons outside the system using it with different rights to use. In *dApp* different libraries are used for writing the smart contract deployed in EVM, where blocks are continuously adding from different contracts and information of single entity is retrieve from these blocks.

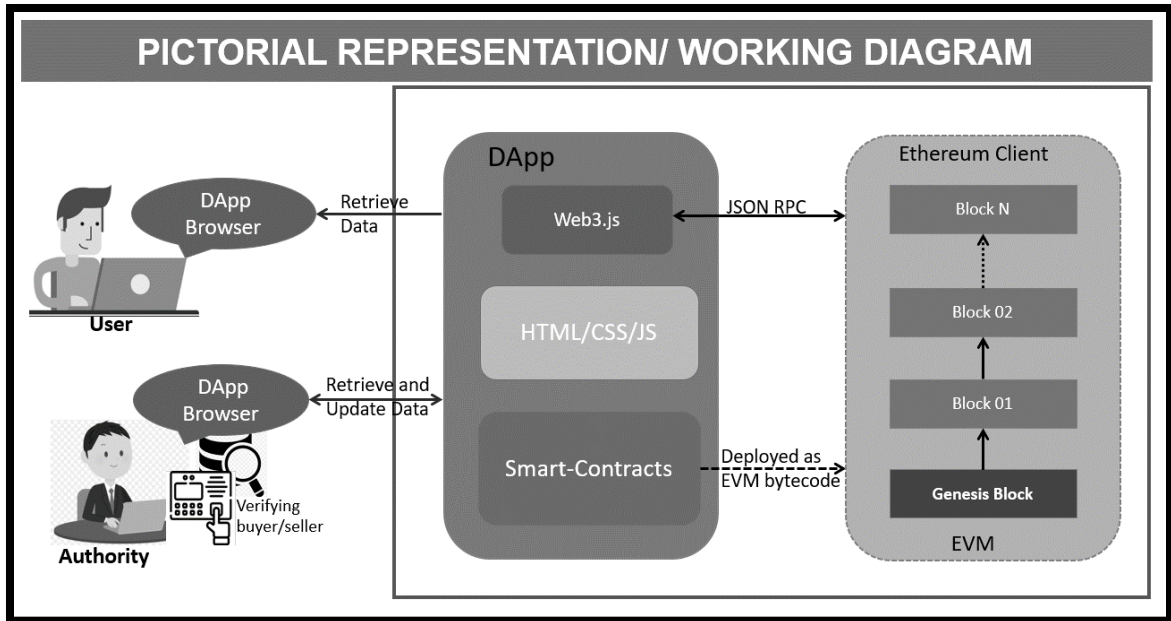


Figure 3.1: System Design Working Diagram

3.2 Smart Contract

A smart contract is the computerized form of business rules and logics, it uses data states to store information on blockchain. It is an automatic way to trigger each event by implementing business logic using high level programming language Solidity[7]. It's smart enough for self-executing and verifying legal obligation.

Finally, smart contract is deployed in its byte-code format on Ethereum Virtual Machine (EVM) with unique address of contract on Ethereum network. Application Byte Interface (ABIs) is used for interfacing with front-end to create DApp using its contract address. It is open source to ensure transparency where anyone can check what code of logic is behind this system. Once it is deployed there is no option to change it i.e. no options are available to change the logic of the system. If one has to change it, a new contract has to be deployed. One of the main features of the *Smart Contract* is that it is self-destructive and transfers its ether and tokens to the address already mentioned in contract. Another important concept of Smart Contracts is *inheritance*. It gives developer an option to deploy large contract in the form of small chunks of contract and then one contract is imported to the other so that both can function in parallel.

In the Smart Contract of VOTS, rules of vehicle registration and transfer are written to ensure that it fulfills the requirements instructed by Government of Pakistan (GoP) and followed by ETO. In our case Smart Contract has following participants that are involved in system:

- VOTS has its beneficiary owner i.e. Government of Pakistan in our case. Some Government Representative will deploy the contract and will have maximum privilege in the system
- Authorities are the most active entities in this system that are appointed by government to work on ground level
- Other participants include people who either have vehicle registered on their name or are planning to buy a vehicle in near future

The functions performed by systems are as follows:

- Registration of all participants and their information is stored on blockchain after proper verification from centralized database
- Adding new and used cars in a system with proper verification by vehicle inspector to ensure transparency where he compares it with previous record and verifies new information that is to be inserted into the system
- Registration and transfer of vehicle ownership by authorities where vehicles and persons are verified by smart contract and after verification transaction gets completed and updated information is added on blockchain
- As blockchain aims to provide transparent information so anyone in the system can retrieve relevant information of concerned vehicle. If person is buying used vehicle he can check complete information and on the basis of this immutable information of vehicle he decides whether to buy or not

3.2.1 Solidity Language

Solidity is a high-level Object-Oriented programming language that is influenced by C++, JavaScript and Python. It is a contract-based language executed on Ethereum Virtual Machine (EVM) and implemented on blockchain network for the development. This is a statically typed scripting language.

In VOTS structures are used for storing data with an array of mapping. Multiple functions with public visibilities are written for execution of source code and some modifiers to execute functions when certain conditions are fulfilled. Inside these functions' events are used for logging. For retrieving information multiple functions are written with different access rights based on the identity of the user. As the authorities have full access rights so they can see all the information residing inside the Blockchain. Normal users will have access to limited information. Complete working of all the functions is explained in the next chapter.

3.2.2 Remix.ide/ Truffle

Remix.ide is web-based IDE to allow users to write smart-contract on solidity, so it was also known as *Solidity Browser* previously. It is an open source powerful tool to write, deploy and run smart-contract on browser. It has three different options for development. *JavaScript* is memory based and local with no connection to any node, *Web3 provider* connection with local node network of test-rpc and *Injected Web3* for making a connection with **Mist** or **Meta-Mask** to connect to real network of Ethereum. This browser provides built in user-interface to interact and terminal for debugging and executing smart-contract.

Truffle is a development environment and testing framework for blockchain using the Ethereum Virtual Machine (EVM), with an aim to make DApp development easier. It's is operated via terminal by writing some easy to remember commands in different stages of DApp development. Built-in commands for compilation and deployment of contract with automated testing script are available. It also provides network management facility of deploying contract on private and public network. The developer does not need to manage artifacts of contract instead the contract will do this task itself automatically. Furthermore, these artifacts will used in front-end to communicate with smart-contract for the deployment on Ethereum network.

Initially smart-contract was deployed using Remix.ide to analyze working of basic smart-contract. After achieving successful results, we moved towards truffle and wrote complete contract in VSCode using Truffle. On truffle the contract was compiled and

deployed using test-rpc and ganache. Both are discussed in next section and testing of contract is explained in later chapter.

3.2.3 Test-rpc/ Ganache

Test-rpc is command line interface (CLI) and Ganache is user-interface for development purposes of smart-contract on Ethereum network. Both provide 10 fake public/private key pair accounts with 100ETH in each account's wallet. These ethers have no value in actual. While all of these accounts are locally available to one node which makes it easier for developer to switch locally between nodes and analyze how it will work in real environment. These fake ethers support developer to write code with no concern of its cost and at the end reduce the transaction price with efficient and effective changes.

As both follow the architecture of local node, so every time for development purposes it starts from the genesis block. This process is required to be repeated to test each case. This is something which is undesirable. While Ganache user-interface updated version (*Ganache v2.0.0*) facilitate developer with an option to create and save multiple workspaces, so testing multiple smart-contracts independently is easy now. The single workspace can be used throughout development with information of previous block in blockchain not erased, thus, blockchain testing starts from where it was left.

Both development networks are used simultaneously during project, where test-rpc is used in the beginning of working on truffle and Ganache is used throughout the development of DApp. Both are local deployment networks and their consensus algorithm use Auto-mining which is not so practical. Therefore, final analysis of working is tested on *Rinkeby*, which works on Proof of Authority consensus algorithm (PoA), and mainly on *Ropsten* network, which works on Proof of Work (PoW) consensus algorithm.

3.3 Database

Database is a collection of data and information that is organized in a form of rows columns and tables so that it can be managed updated and easily access indexes are made so it is easier and time efficient to find the relative information. Database allows querying sorting and manipulation of data that is stored in it.

Databases have many types according to data and information and through the forms of computing. Following are the main types of database

3.3.1 Relational database

Relational databases are in the form of tables in which column represents data category and rows stores the values of these categories. There is no need to reassemble information and fetching data is very easy. The model of this database is made so that it can relate between the data stored.

The Structured Query Language (SQL) is used to communicate with database. Query for information and data gathering from database is done through this language. Relational database is easy to maintain, use and expand without changing the format or data entries of whole database.

3.3.2 Distributed database

In this all the data is not stored in single common computer or server but it is stored in different computers that maybe located in same location or distributed over the network and are independently managed. Distributed databases are logically connected to each other and provides single logical database. It allows multiple users to fetch and update or change data. If one site fails, data can be recovered. Multiple files from dispersed databases must be synchronized.

Distributed databases can either be heterogeneous or homogeneous. In homogeneous type of distributed databases there must be same hardware same running OS and same application in all different location, which in case of heterogeneous distributed database all these do not have to be same in all locations.

3.3.3 Cloud database

It is a database that operates on cloud computing platform which can be private, public or hybrid and its access is provided as a service. This collection of data can either be structured or unstructured. Cloud databases ensure their high availability and provide scalability and it charge user for bandwidth or storage as per use.

Users can access this database through organization local area network. This cloud database which is stored on network accessed only through internet provided as service. It eliminates the need of physical infrastructure

The behavior of cloud database remains same whether accessed through API or through SQL query. However, there may be small difference in response time of cloud-based database and database which is accessed through LAN which is slightly faster because unlike cloud-based database it doesn't require to complete round trip for each interaction

3.3.4 NoSQL databases

NoSQL databases are used when there is large amount of unstructured data. These non-relational databases avoid joins and are horizontally scalable being distributed and open source.

The main advantages of NoSQL are that it can store large volumes of unpredictable data, it is fast, easy to use because of object oriented programming. It is scalable without being expensive

It has further four types of document databases, graph stores, key-value database, wide-column stores

3.3.5 Object Oriented Database

This database stores data in form of objects (real world entity) and objects being instances of classes. It is different from relational databases that store data in tabular form, but it is hybrid of both combining the features of relational database like transaction and recovery, concurrency with the principles of object-oriented programming. Object oriented database provides easy integration between database, OS, artificial intelligence systems, spreadsheets and other applications. It also provides with the feature of referential sharing through inheritance and this sharing of data or information, products or components can be done easily.

3.3.6 Graph database

It uses graph theory for storing, mapping and to query relationships. It is collections of

nodes and edges, here each node means an entity and each edge represents a connection between nodes.

Graph databases are gaining popularity for the analysis of interconnections. For example, Graphical database can be used by companies to analyze and mine data of customers from social media

3.3.7 Database Integration With VOTS

In this project we are using two databases one for the verification which would be a centralized database. The other is for registration and this database is the system's main decentralized database. The centralized database is playing a very important role. It is making this system compatible with the already existing system. When a person comes for transference of ownership of his vehicle or to register a new vehicle on his name, he must be verified first from the centralized database. This means that the person's information should be present in centralized database and he must be a registered citizen. If this verification gives positive result only then the person would be allowed to continue and do the transaction in the next database. After the registration of the person in the decentralized database, the authority will be able to make transactions on his name. Limited information about vehicles can be viewed by anyone as the used platform Ethereum is a public platform. If there is need to make a change in the decentralized database, all the nodes must approve that change first. This makes this system less prone to hackers.

When a person comes to register his new car or to buy an old one, an immutable account ID will be assigned to that person on Ethereum network for him to make the transaction. For this, smart contract will first send a query to database that either this person is verified and has clean record. This database is decentralized database of law regulatory authorities and governmental organizations. Once the response to query sent is that the person is verified to do this action then smart contract will assign immutable account ID to these users on Ethereum network and set the privileges they can have. Then the event occurs after taking the thumb impressions of both parties and transference of ownership

or registration is done. Data of both persons and their status is then updated on decentralized database that either they are registered with the system and owning a car or not. The information and history of cars is stored on blocks of blockchain which can also be traced back to get information or previous event occurred.

All of this is fully automated. On user end, one just has to affix their thumb impression rest of work is done by smart contract automatically. After getting the finger print it is searched from the databases of law enforcing agencies if you're citizen of (Pakistan) and have a clean record then VOTS will create accounts on Ethereum and the whole process is done. Once the process of registration or transfer of ownership is done then this information is stored on a block and then this block will become part of blockchain.

3.4 Hardware

To ensure that each participant during a transaction is valid person his national identity number is used for verification beside this person is authenticated through his biometric. Hence, we take person fingerprint, compare it with his already register fingerprint in national database and register person in system with authentication. Whereas, fingerprint stored in smart-contract afterward when uses the system. In subsection of this integration of fingerprint sensor and with DApp are covered.

3.4.1 Fingerprint Sensor Integration

We are using the R307 module of this scanner. Fingerprint processing includes two parts: fingerprint enrollment and fingerprint matching. When enrolling, user needs to enter the finger two times. The system will process the finger images two time, generate a template of the finger based on processing results and store the template as in the hash value of the image. As for the biometric verification of the new node in the network R307 module is used as a fingerprint sensor to get the image of the fingerprint. Furthermore, that image is converted in to 256bit hash value using sha256 for storage and reuse when the hash value is transferred to the blockchain.

The fingerprint sensor (R307) is connected to the raspberry pi using a TTL to USB convertor. The pi acts as a microprocessor for R307 and run the executable code to

initialize the fingerprint sensor. After this check whether the fingerprint is connected and communicating as it should be. If all of the above things/procedures are going as they should be then we proceed to the next phase to capture the image of the fingerprint, to do that effectively sensor capture the image two times to get it right, now the image is being capture and the hash is to be calculated. To calculate the hash, we use the *hashlib* in python to calculate the hash of the fingerprint in an effective way. Now we have hash value of the fingerprint of the individual owner of the node.

3.4.2 Hardware Integration with Smart-Contract

In the second phase we are to connect our python code to the HTML GUI of the smart contract so that we can use the hash function to authorize the person's identity. To do that we are to connect the python code to the HTML code of the GUI in consideration with the change of the programming language. In our case we are using the python module flask to make html page and connect the python code with it and the output (hash value) of the code will be sent to the HTML code of the smart contract GUI.

The tricky part is how to do all the stuff we discussed in the above paragraph so that it works like we want it to. First, we use the Flask library of the python to integrate the python code with the HTML and then fetch the hash value of the fingerprint from another python code which is imported here as another library/module. In order to fetch the hash value from the python code and display it on to the screen we first need to get the output of the code in this python script and then jsonify the response to the html file.

Now the hash value can be displayed on the frontend and the function to get the hash value can be run by clicking the button and then use the sensor to give the fingerprint hash to the frontend of the GUI. Hash value is being placed in its place and now we need to send that hash to the function of the person registration of the smart contract to register the person as the new node if he is using the services for the first time. He will write the name, CNIC, click on the button to give his fingerprint hash value and the address of his wallet. When he/she presses the button to enroll his fingerprint the sensor will start working and the person will place his finger on it twice so that the finger is being capture effectively with less noise. Now the image of the fingerprint is converted to the hash

value and sent to function (*vots.Person_Registration.sendTransaction*). The function stores hash value and the address are stored along the person's name and CNIC.

In case of the verification of the existing node the person presses the button and gives the fingerprint hash to the function (). Function has been coded to compare the existing hash to the new recovered hash of the fingerprint, if the hash value is valid and matches the existing hash value then the person is valid if not then person authentication is denied, and he/she can't use the services.

The flow diagram for the process is show below with the figure of the fingerprint sensor connect with the raspberry pi.

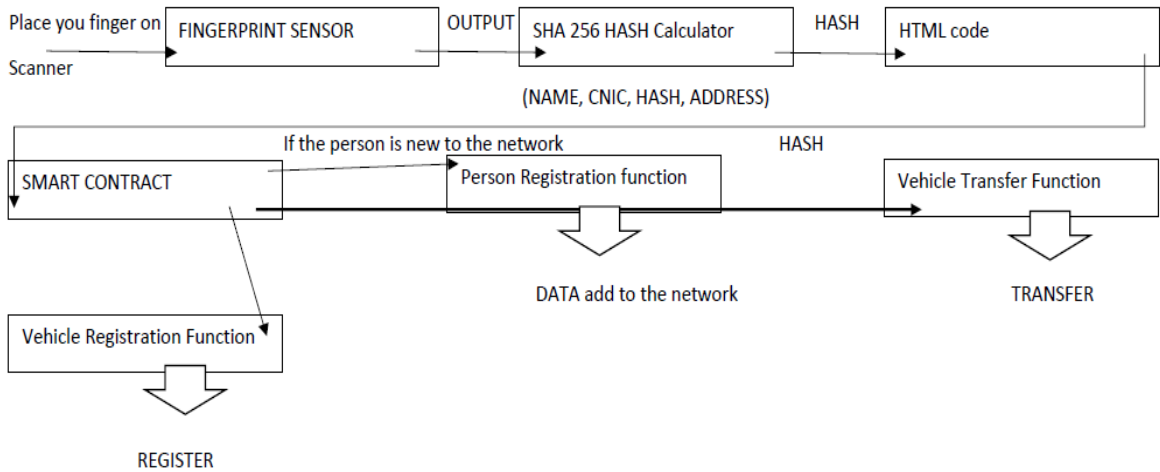


Figure 3.2: Flow Diagram of Sensor Working



Figure 3.3: R307 integration with Raspberry pi

In case the person is already registered in the network then the biometric verification is needed while vehicle registration or the vehicle transfer. When either one of these functions are called the person verify that he is the valid person and no one else is using his account to do the unethical acts. In vehicle registration only, the person owner of the vehicle is going to enter his fingerprint hash to register that particular vehicle under his private domain. While in vehicle transfer both the persons should be the part of the network and at the time of transfer both are going to place their finger to verify the validation of their accounts and also so that their credentials should not be placed under wrong authority.

3.5 DApp Design

3.5.1 DApp development

This section will explain all the necessary steps to develop a decentralized application. DApp is comprised of two parts its smart contract and front-end. Smart contract is discussed in previous section and its deployment on test-rpc, while in this we only cover second part and integration of MetaMask with smart contract and front-end an important extension when working on web2.0.

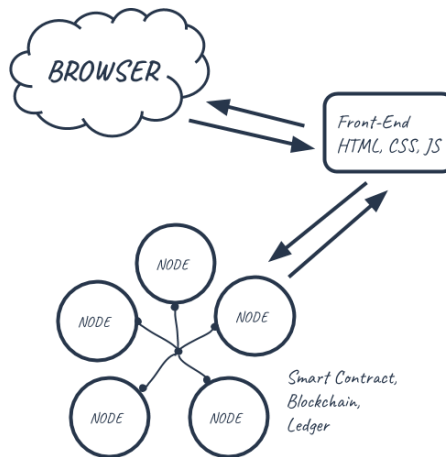


Figure 3.4: Structure of a Decentralized Application

3.5.2 Front-end for DApp design

Front-end of VOTS is a design of website which interact with Ethereum to update and retrieve information on smart-contract. Hence for this ABIs of contract is used with its deployed address only way to integrate smart contract synchronous with front-end and a person can access smart-contract. Since current browser work on web2.0 and DApp use web3.0 so to make these browsers compatible for DApp it's necessary to use MetaMask with browser if web3.0 (Mist) is not available.

3.5.3 MetaMask Injection for DApp

MetaMask is a browser extension which acts as bridge between Web2.0 and tomorrow i.e. Web3.0 browser. It is a software wallet used with Chrome, Firefox and Opera, making easier for user to work on Ethereum. It is used for tracking of transactions and transfer of tokens between accounts. This is also a secure way to connect with ethereum network as it doesn't store private key or seed value of account; also, this offers connection with all networks of ethereum and gives options to connect to multiple custom-rpc to use local nodes in DApp.

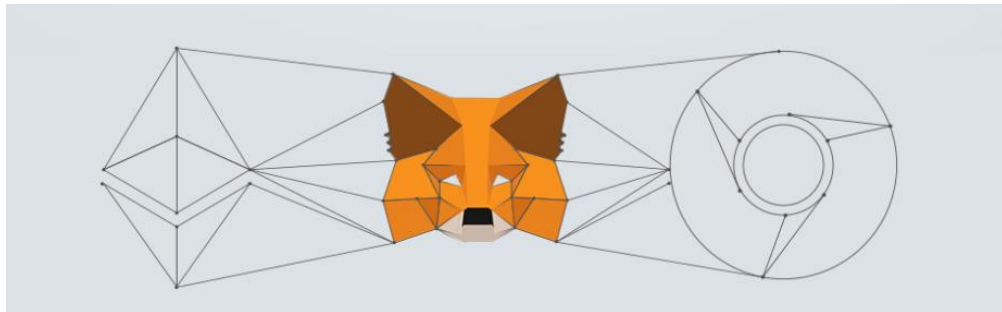


Figure 3.5: MetaMask as a bridge between Chrome and Ethereum Network

Front End Design for DApp

Now smart contract is ready to interact with user friendly front-end, so user communicate. So now we have quick overview on development of any dynamic web applications:

3.5.4 Hyper Text Markup Language (HTML)

HTML is used for designing and developing web applications. It allows the designer to design a user-friendly and eye-catching front end. Different types of forms, images, text, logos, and background colors can be added to make an appealing look. It allows the user to add buttons of different styles and those buttons are used to navigate to some other link. In short, HTML is an easy to code language to develop simple and easy to use yet stylish front end.

3.5.5 Cascading Style Sheet (CSS)

CSS is a language that is used for styling of HTML elements. It decides how the content of front end would look like e.g. either the text would be center aligned or left aligned, image would have border or not etc. Before the invention of CSS all the styling had to be done inside HTML code and often the same type of classes had to be styled again and again which was not considered a good practice. Using CSS, the same styling can be applied to more than one class and even same styling can be applied to a class, id, text etc.

3.5.6 PHP

PHP is Hypertext Preprocessor language is generally used for the development of dynamic web applications. It is a powerful scripting language that is used for designing dynamic and interactive web pages. In this project PHP is used to interact with the centralized MySQL database. Here PHP is used to design the page that verifies the presence of user in the centralized database so that he can be registered in the *Blockchain* database.

Finally, the *Smart Contract* is connected with a user-friendly front end so that users can communicate with the *Smart Contract* in a proper, manageable and easier way.

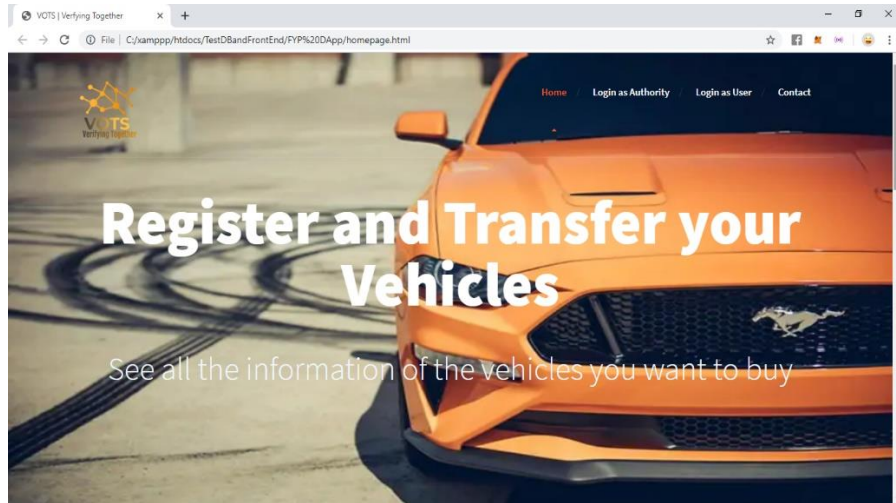


Figure 3.6: VOTS Front-End

When an authority login he views different functions to interact with smart contract user interface of this page is show below, while working of these functions will be cover in next chapter.

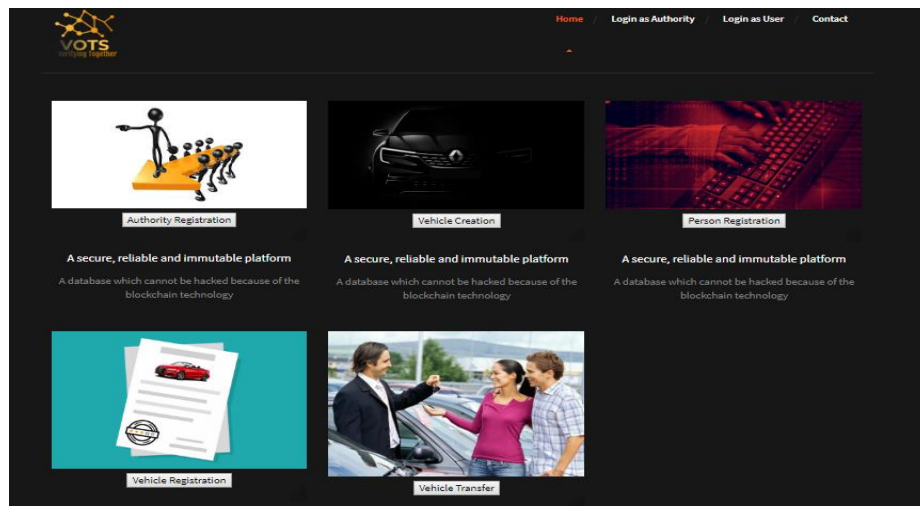


Figure 3.7: Front-End of VOTS Authority

Another major library that has been used in DApp development is **Web3.js**. This tool is especially used to interact with *Smart Contract* and integrating front end with the back-end logic i.e. *Smart Contract*.

3.5.7 Web3.js

Web3.js is an API or in other words it is a collection of libraries developed to interact with Ethereum *Smart Contract*. It needs to be installed in the machine-like usual libraries.

Basically, Web3.js is a tool that is used instead of jQuery to read and write data from web server. Here is pictorial representation Web3.js helps to interact with the *Smart Contract*.

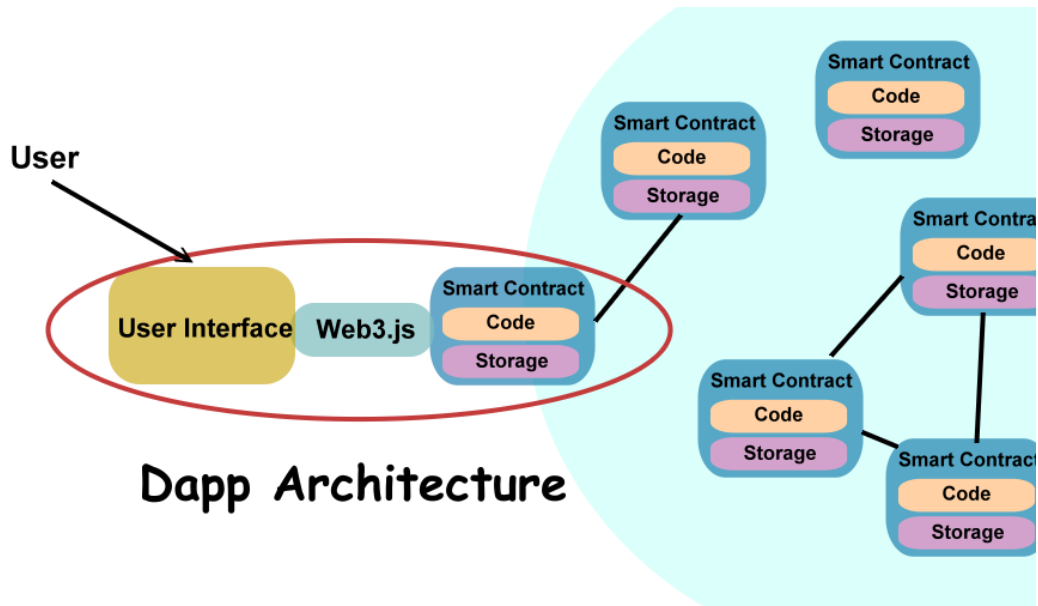


Figure 3.8: DApp Architecture

3.5.8 Lite Server

It is the command that is used to run live server in the browser e.g. Chrome, Firefox, Opera etc. This command is useful because its prerequisite is node.js which is already used in this project, therefore, to run a live server there is no need of extra installations.

TEST AND DEVELOPMENT

This chapter describes how the *Smart Contract* was written and deployed using truffle. It will explain complete working and testing of some of the functions manually by writing commands on CLI as well as testing of some functions using front-end that is connected to MetaMask wallet.

4.1 Contract Development using Truffle

Initially, the Smart Contract was written and compiled using online remix.ide. After analyzing the source code using the online IDE, the development of Smart Contract was moved towards the Truffle Framework which is designed especially for Ethereum Smart Contracts. In truffle there are no issues of storing artifacts and contract address is obtained easily. Following is the step by step explanation for developing a decentralized application:

1. Smart Contract is written in Solidity language and must be converted into byte code to be deployed on EVM hence the contract is compiled using truffle to store its artifacts in *JSON file*. Compilation of contract with solidity version 0.5.0 is shown below.

```
PS D:\truffle\FYP_dem> truffle compile

Compiling your contracts...
=====
> Compiling .\contracts\Migrations.sol
> Compiling .\contracts\VOTS.sol
> Artifacts written to D:\truffle\FYP_dem\build\contracts
> Compiled successfully using:
  - solc: 0.5.0+commit.1d4f565a.Emscripten.clang
```

Figure 4.1: Contract Compilation on truffle

2. Once the contract is compiled it is deployed on Ethereum network.

```
2_deploy_contracts.js
=====
Deploying 'VOTS'
-----
> transaction hash: 0xc3ea57eb5d96e21a9c55380b4147c3a5056e5dd7f7590621a8f29e0611ea0e20
> Blocks: 0
> contract address: 0xe05a5fe9a7d7baa51295cd4acb4d35133bd31fab
> block number: 3
> block timestamp: 1557926594
> account: 0xd3B68e3E649cC37c14b2c1a97c664B2B4ba6e46C
> balance: 99.92655372
> gas used: 3397398
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.06794796 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.06794796 ETH
```

Figure 4.2: Deployed Contract metadata

```
eth_sendTransaction

Transaction: 0xc3ea57eb5d96e21a9c55380b4147c3a5056e5dd7f7590621a8f29e0611ea0e20
Contract created: 0xe05a5fe9a7d7baa51295cd4acb4d35133bd31fab
Gas usage: 3397398
Block Number: 3
Block Time: Wed May 15 2019 18:23:14 GMT+0500 (Pakistan Standard Time)

eth_getTransactionReceipt
```

Figure 4.3: Block information on testrpc

3. Deployed contract can be tested either manually or using script, where we analyze the working of the DApp. In case of manual testing, *truffle console* is used from where development mode of truffle gets started and by creating an instance each function can be validated. In the other case *truffle test* is used to run the script in which each function is validated automatically.

Home Login as Authority Login as User Contact


View all the history of your vehicle

See full history of past owners of your vehicle

"Ever wondered the car you are using may have been used in some illegal activity and you have no idea about it at all. Every vehicle has on average 3 to 5 owners during its lifetime, it is clear that 80 percent of vehicle owners do not know anything about their vehicles. Not only this, many other situations need the history of vehicle; from its origin to the present date. VOTS is blockchain based solution to this problem. It keeps track of all the vehicles. Any vehicle not registered or present in its database will be treated as illegal vehicle and serious action will be taken. The initiative is meant to stop misuse of vehicle registration and transfer, and facilitate vehicle owners and law enforcement agencies to know the origin of every vehicle."


VOTS Team

Why choose us?




A decentralized system that excludes the need of any central authority

A database everyone can inspect but which no single user controls



A secure, reliable and immutable platform

A database which cannot be hacked because of the blockchain technology



Terrorist attacks using vehicles and fraud can be reduced as data updates are only possible by authorized personnel

Any tampering of data can also be easily tracked on Blockchain

Figure 4.6 - Front End of DApp

5. Finally, contract is deployed on *Ropsten test network*, where interaction with the contract can be done using front-end. Registration of a person is shown below:

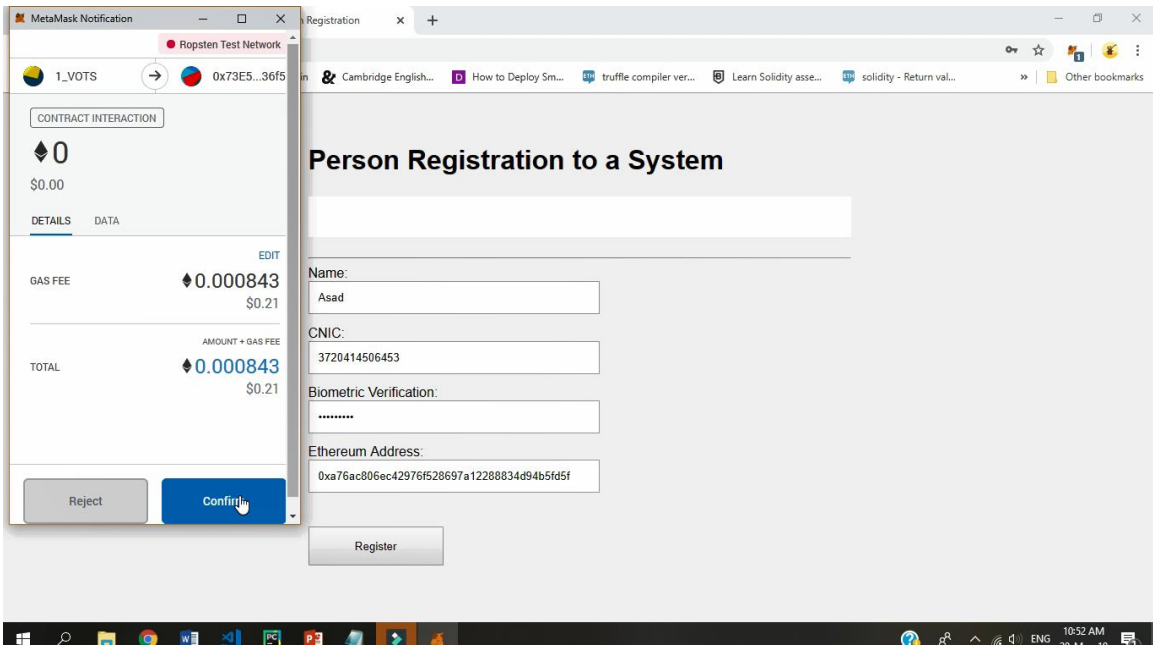


Figure 4.7: Person Registration via Front-end

Using MetaMask user confirms transaction and waits for a couple of minutes to validate the transaction because Pow consensus algorithm is used which takes some time to mine the block. In the figure below, confirmed transaction is shown:

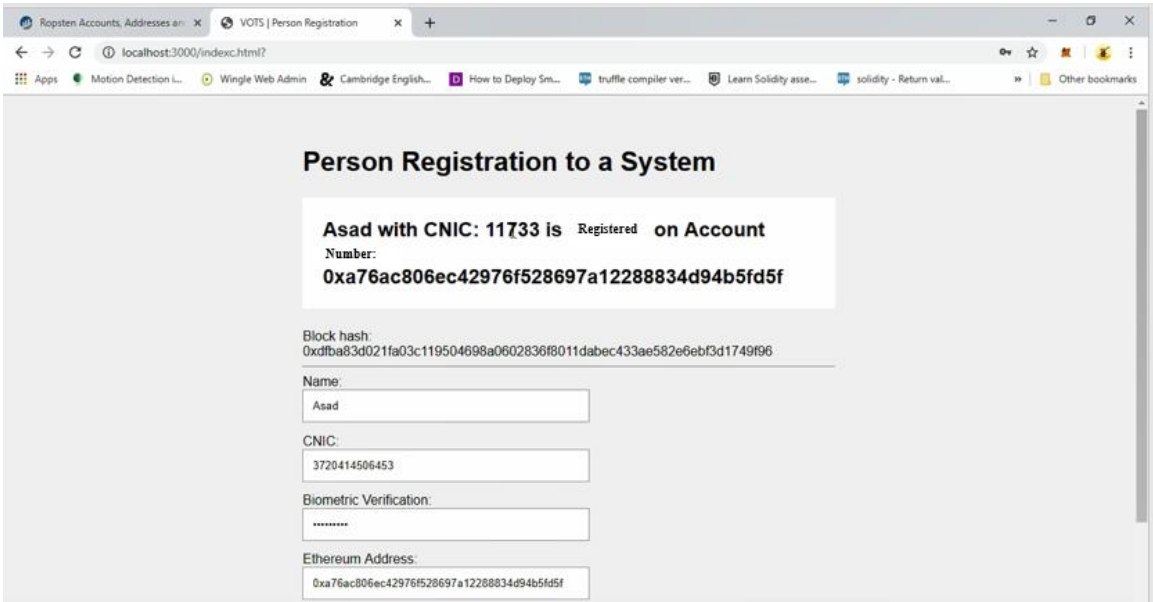


Figure 4.8: Person Registration confirmation via Event

6. Similarly, vehicle is created and registered to a person's name and a vehicle is transferred from person's name to another person's name in case of transfer of ownership of vehicle. Transference of ownership of vehicle is shown below:

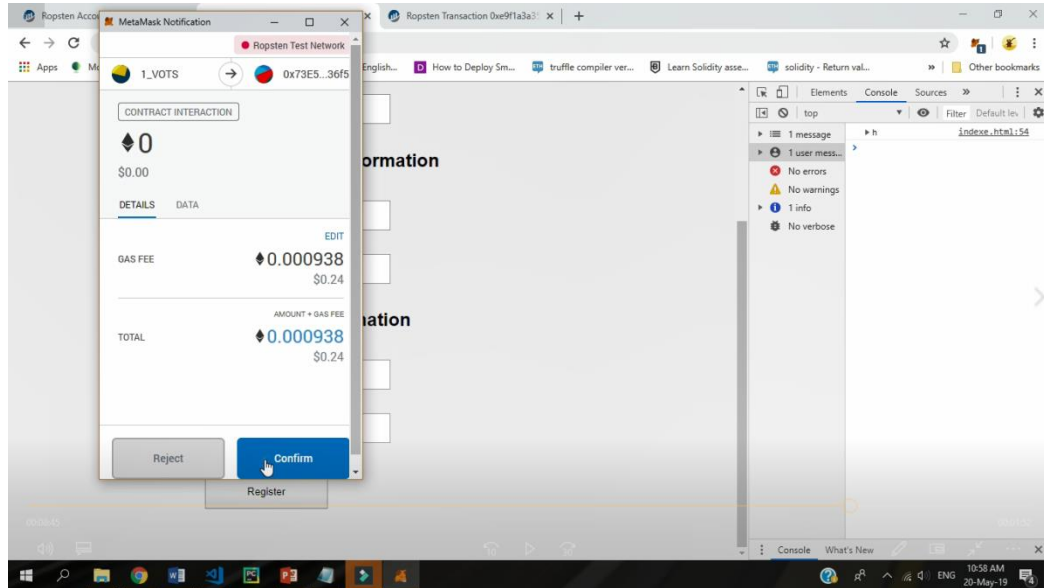


Figure 4.9: Vehicle Transfer on VOTS

Now after confirmation following result is shown on front-end.

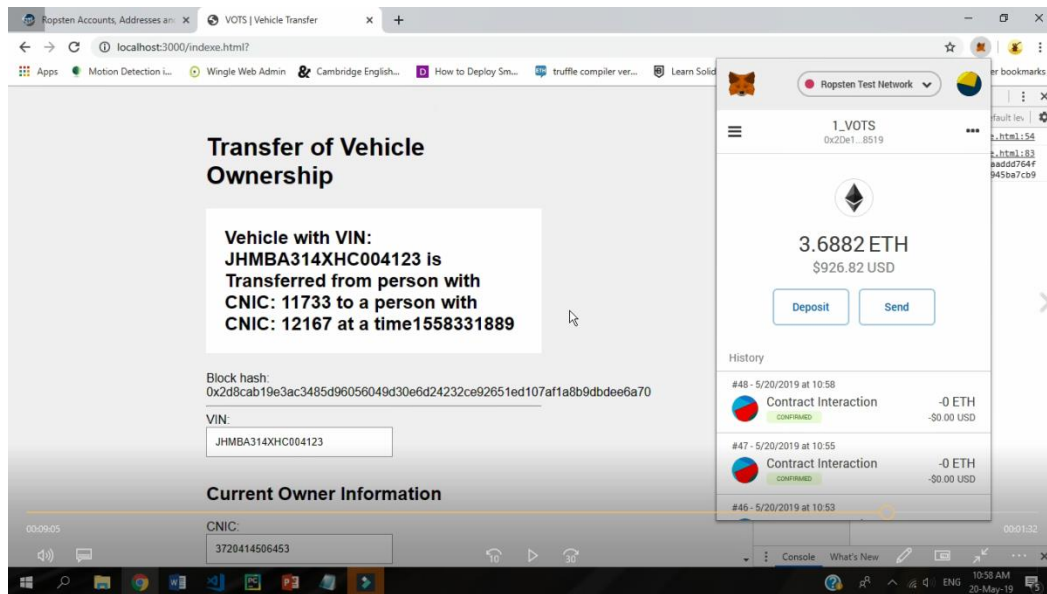


Figure 4.10: Vehicle Transfer Update

7. In case a person tries to do some unethical act e.g. if he tries to transfer the ownership of vehicle which he already has sold to someone else, the system will give error and will not proceed further.

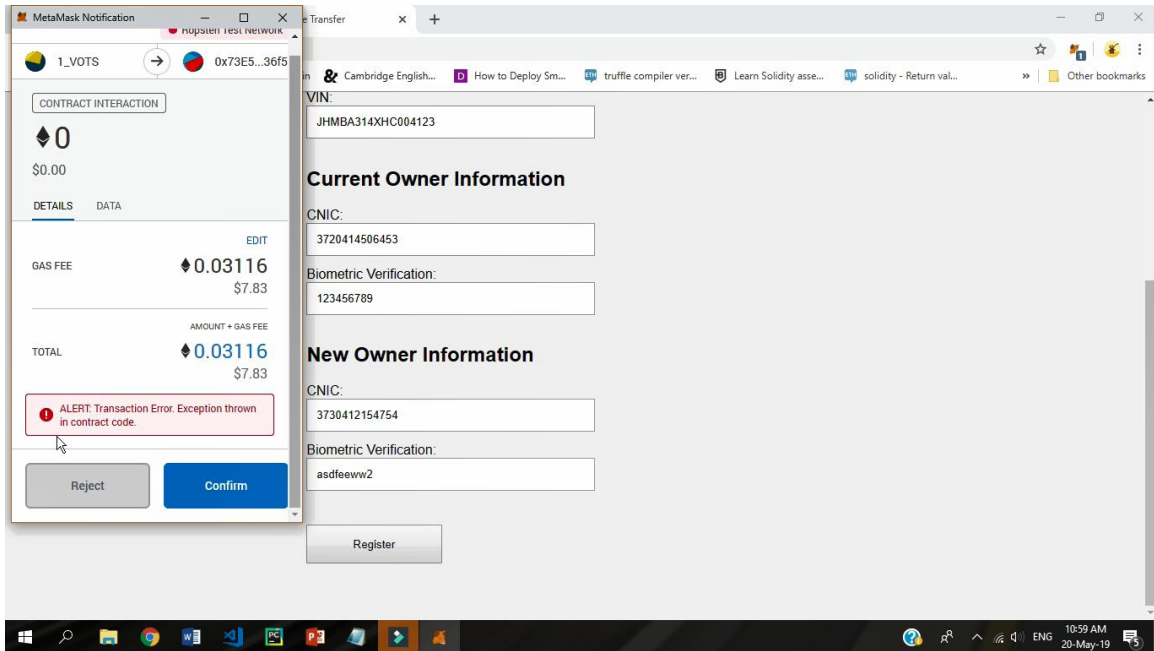


Figure 4.11: Error for Vehicle Transfer

PROJECT ANALYSIS AND EVALUATION

The ecosystem for Transference of Ownership or Registration of a Vehicle is shown in figure below where every participant of the system is acting as a node. All these nodes share same ledger, or we can say every node has a copy of the ledger. The modification made in one node is notified to the other nodes. More than half the nodes must approve that modification. Therefore, it becomes difficult for an intruder to change the important stored information. Hence, using *Blockchain* transparency and availability of data is achieved. In this system all the procedure is being done using the same deployed *Smart Contract*. The business logic of *Smart Contract* is such that only a Government Representative has complete access to the system. This Representative has been made authority and only he can register new vehicles or transfer the ownership of used vehicles. While the other nodes i.e. general public, can only retrieve information in order to make the right decision of buying or selling vehicles. The decentralized behavior of this system creates a synchronous ledger across the network, so errors caused by delay are removed. Hence, problems associated with centralized system are overcome and any update of vehicle from one department is reflected to other department and the owner of the vehicle needs no paper record of vehicle and visit different offices for vehicle related matters and issues.

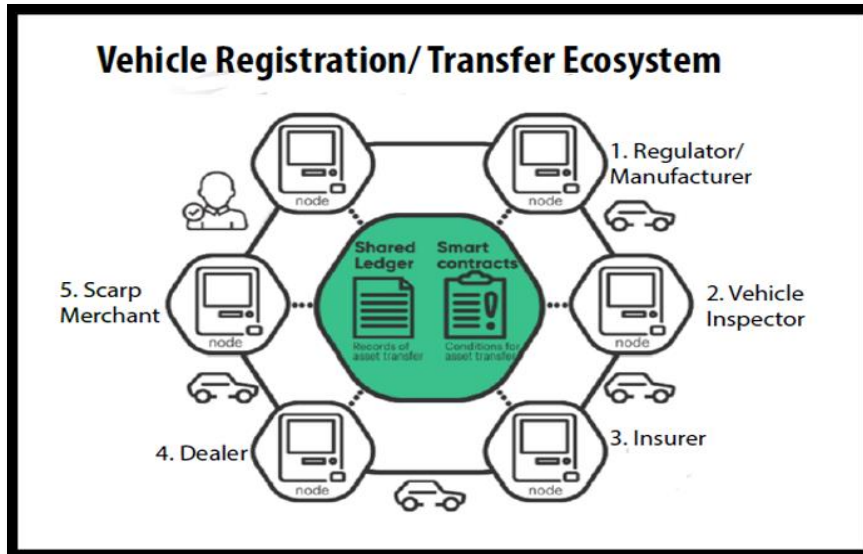


Figure 5.1: VOTS Ecosystem

5.1 VOTS V.1.0 PROTOTYPE WORKFLOW

VOTS V.1.0 is a decentralized system for Transference of Ownership or Registration of a Vehicle on Ethereum network. Here workflow of Transference of Ownership of a Vehicle is explained, and some other functions are also discussed. Workflow for the current system is as follows:

1. Initially, buyer and seller find each other using any online platform e.g., *carfirst.com*
2. With agreement for Transference of Ownership of the Vehicle both parties go to the local authorized person office, who would be a Government Representative, situated in the near area of buyer and seller instead of going to ETO (Excise and Taxation Office).
3. Here Sale and Purchase agreement of Transference of Ownership of the Vehicle is done by an authorized person. Also, he also verifies that both the buyer and the seller are registered on the network and vehicle is actually registered on seller's name.
4. Now if buyer wants vehicle inspection, he refers it to the Vehicle Inspector. He will examine vehicle and compare it with the existing record on *Blockchain* and then he will update information using Decentralized Application's Front End. A report of examining is generated and shared to all parties.

5. After this examination if buyer wants to buy this vehicle, he will notify both the authority and the seller. On the other side both parties will negotiate on price and final price will be told to authorities.
6. Money will be transferred from buyer's to seller's account and the bank also notifies this transfer of money to the authority.
7. With this message authority updates the ownership of Vehicle using smart-contract and notifies both the parties with an email and letter of transference of ownership is posted on postal address within 7 days.
8. Since the system is decentralized therefore this change of ownership is reflected in all the departments of ETO.
9. All law and enforcement agencies, insurance companies and traffic police also get updated with that single click of vehicle transferred.

Conclusion

The aim of this project was to help Law Enforcement Agencies, Insurance Companies and general public in the process of Transference of Ownership or Registration of Vehicles. By using a decentralized application fraud cases of vehicles e.g., fake registration of stolen vehicles, tampering in the history information of vehicle, no record of a vehicle used in a terrorist attack, can be minimized. The cumbersome and time-consuming process of transference of vehicle ownership has been made very easy by this decentralized system. Previously, there was an extensive and hectic procedure to transfer the ownership of vehicle and it was centralized, therefore, transparency and reliability cannot be assured. The major and some of the most important features of Ethereum Blockchain are reduced cost, safe and transparent transit of values and removal of intermediaries or third parties that could interfere during a transaction. Any asset transfer management system must have all these features in order to maintain transparency. Another salient aspect of this project is its compatibility with the existing system. As the current database used by NADRA or any other department cannot be thrown away, therefore, initially all the registrations and confirmations are done using the centralized database which are approved by the law enforcing inspectors and law-making bodies. Gradually, the decentralized database will acquire all the required information and there will be no need of any centralized ledger and the system will be fully decentralized.

Recommendation and Future Work

To make this system more transparent and decentralized some recommendation as a future work are explained below:

1. VOTS can be modified to store hashes in the blocks of the blockchain and access the data or information using Interplanetary File System IPFS which is particularly used for sharing in decentralized world.
2. Examining reports and images of vehicles and their owners can also be stored as a permanent record inside the block.
3. Now participants of VOTS find each other through third party where there is still ambiguity of mislead information, but a platform can be built inside the VOTS system where buyer and seller can interact with each other. In this way sellers can post an ad of their vehicles and information of the vehicle is automatically retrieved from the *Blockchain* ledger, hence buyer gets to know correct information which satisfies him to buy a used car.
4. Using *IoT* is also an attractive way to work on VOTS in future, where vehicle tracking device can be converted into a node of network. This node can update information regularly on blockchain, this information later on can be used by government representative or vehicle manufacture to draw many conclusions.
5. Decentralized Uber service is also one improvement in this system where people can earn from their vehicles when they are not using it and instead of depending on any other entity and giving some percentage of earning to that entity the owner can do it himself using blockchain technology. Although it's different from VOTS but appealing work like this can be added to using this technology.

APPENDICES

TAMPER RESISTANT DATABASE LOG UPDATE VIA BLOCKCHAIN

<p>Extended Title: Vehicle Ownership Transfer System (VOTS)</p>
<p>Brief Description of the Project: The process of vehicle registration has always been cumbersome. It is a time taking process where multiple parties are involved and also poses a risk of information manipulation, data duplication and various errors. In such a scenario, critical information can get highly vulnerable to frauds and data tampering or even become non-traceable. By bringing Blockchain into the picture and moving the entire vehicle registration process on to Blockchain, a lot of these issues can easily be taken care of. Blockchain comes to the rescue by reducing the average turnaround time. Blockchain will enable parties to push data as a smart contract that eventually becomes a single source of immutable data to all parties.</p>
<p>Statement of Work: VOTS will provide a decentralized distributed database of keeping the record of vehicle ownership to give transparency and immutability for the change of ownership.</p>
<p>Academic Objectives: At the end of this project we will have a hands-on practice of:</p> <ul style="list-style-type: none"> • Scripting Language/s • Solidity • Simulators • Smart Contract • Database handling • Sensor Integration
<p>Application: To provide Traceability, Immutability and Scalability to the car registration and ownership process by using the digital identity i.e., biometric verification.</p>
<p>Previous Work Done on the Subject: N/A</p>
<p>Material Resources Required: <i>Solidity</i> will be used for writing the <i>Ethereum</i> smart contract. It is a <i>JavaScript</i> based language. Beside this <i>Sensors</i> will be used for communicating with blockchain and execute smart contract. Cryptographic knowledge is also required for hash function and key pair generation.</p>

No. of Students Required: 4

Group Members:

- ASC Muhammad Saad Malik (Syndicate Leader)
- NC Muhammad Umer
- NC Syed Muhammad Mahdi Raza
- NC Zahra Shiraz

Special Skills Required:

- Scripting language/s
- Database handling
- Smart Contract development

Approval Status:

Supervisor Name & Signature:

Asst. Prof. Dr. Fawad Khan

Assigned to: ASC Muhammad Saad Malik (S.L.), NC Muhammad Umer

NC Syed Muhammad Mahdi Raza, NC Zahra Shiraz

HoD Signature: _____

R&D SC Record Status:

File # _____

Coordinator Signature: _____

BIBLIOGRAPHY

- [1] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).
- [2] Buterin, Vitalik. "A next-generation smart contract and decentralized application platform." *White paper* (2014).
- [3] Wood, Gavin. "Ethereum: A secure decentralisedgeneralised transaction ledger." *Ethereum project yellow paper* 151 (2014): 1-32.
- [4] Crosby, Michael, PradanPattanayak, Sanjeev Verma, and Vignesh Kalyanaraman. "Blockchain technology: Beyond bitcoin." *Applied Innovation* 2, no. 6-10 (2016): 71.
- [5] Zheng, Zibin, ShaoanXie, Hongning Dai, Xiangping Chen, and Huaimin Wang. "An overview of blockchain technology: Architecture, consensus, and future trends." In *2017 IEEE International Congress on Big Data (BigData Congress)*, pp. 557-564. IEEE, 2017.
- [6] Pongnumkul, Suporn, ChaiyaphumSiripanpornchana, and SuttipongThajchayapong. "Performance analysis of private blockchain platforms in varying workloads." In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1-6. IEEE, 2017.
- [7] Parizi, Reza M., and Ali Dehghantanha. "Smart contract programming languages on blockchains: An empirical evaluation of usability and security." In *International Conference on Blockchain*, pp. 75-91. Springer, Cham, 2018.
- [8] Foroglou, George, and Anna-LaliTsilidou. "Further applications of the blockchain." In *12th Student Conference on Managerial Science and Technology*. 2015.
- [9] Swan M. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc."; 2015 Jan 24.
- [10] Zyskind, Guy, and Oz Nathan. "Decentralizing privacy: Using blockchain to protect personal data." In *2015 IEEE Security and Privacy Workshops*, pp. 180-184. IEEE, 2015.
- [11] Raval, Siraj. *Decentralized applications: harnessing Bitcoin's blockchain technology*. " O'Reilly Media, Inc.", 2016.

