



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of the UGC Act, 1956)

Assessment - 3

Technical Answers for Real World Problems

Slot: TG2

Project title

Securing Organ Donation using Blockchain

Team members

Anuradati Kulshrestha (17BCB0110)

Abhirupa Mitra (17BCE0437)

Amisha (17BCB0022)

Proposed Methodology

In this paper, we are proposing a secure method of organ donation over a decentralized platform. This system will be implemented via a web portal that connects organ donors with organ receivers and administered by hospitals. We are trying to completely avoid third-party interference and protecting the integrity of the patient data and identification of the donated organs. This will be attained with the help of smart contracts. Smart contracts will contain the protocols that will govern our organ transaction process and facilitate smooth transactions without intermediaries. These smart contracts will be deployed on a blockchain-based distributed computing platform, Ethereum. All transaction-related information and patient data will be bundled into a smart contract and pushed into the blockchain. We also aim at tracking the location of the organ, during its transfer, with the help of an RFID tag, and continuously monitor its weight to check that the organ container is not being tampered with.

Why are we using blockchain?

Blockchain is a distributed, decentralized, public ledger. Blockchain technology accounts for the issues of security and trust in several ways. First, new blocks are always stored linearly and chronologically. That is, they are always added to the “end” of the blockchain. After a block has been added to the end of the blockchain, it is very difficult to go back and alter the contents of the block. That’s because each block contains its own hash, along with the hash of the block before it. Hash codes are created by a math function that turns digital information into a string of numbers and letters. If that information is edited in any way, the hash code changes as well. In order to change a single block, then, a hacker would need to change every single block after it on the blockchain. Recalculating all those hashes would take an enormous and improbable amount of computing power. In other words, once a block is added to the blockchain it becomes very difficult to edit and impossible to delete.

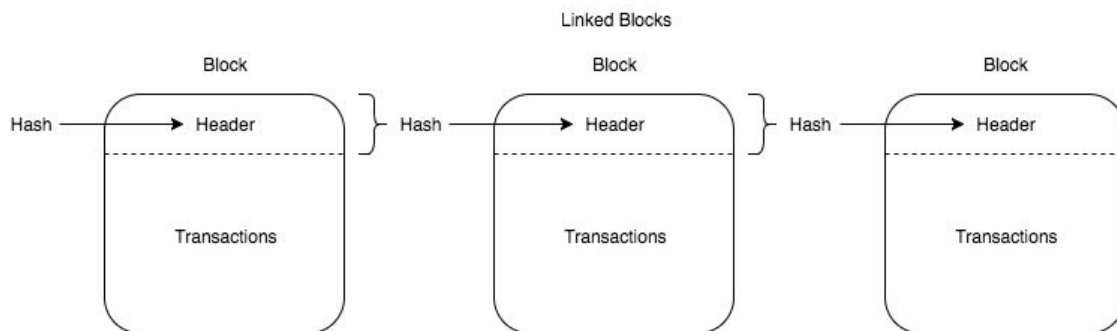


Fig: A basic blockchain architecture

Usage of blockchain provides us with an unparalleled advantage over traditional methods

- **Trust:**
 - Decentralized transparent database
 - Enables participants to have equal access to stored data
 - It can enable the formation of secure ecosystems that groups of companies can use for storing and sharing data.
- **Privacy:**
 - The provenance of each block in the blockchain can be verified and traced back through the chain's history
 - Permanently stores older blocks of data
 - It can be modified to store anonymized data with the help of hashing.
- **Integrity:**
 - Accessing and modifying the data stored on the blockchain is nearly impossible without notifying and seeking consensus from the entire network.
 - Thus, blockchain can be used as a source of truth and privacy.
- **Resiliency:**
 - Highly resilient to hacking and other forms of external attacks.
 - Minimal to no risk of data loss.
 - Highly suited for the storage of vital information, such as health-related data in this case.

Using hyperledger fabric to deploy blockchain-based distribution ledgers:

Hyperledger Fabric is a platform for distributed ledger solutions underpinned by a modular architecture delivering high degrees of confidentiality, resilience, flexibility, and scalability. It is designed to support pluggable implementations of different components and accommodate the complexity and intricacies that exist across the economic ecosystem. At the heart of a blockchain network is a distributed ledger that records all the transactions that take place on the network. A blockchain ledger is often described as decentralized because it is replicated across many network participants, each of whom collaborates in its maintenance.

Benefits of the hyper ledger fabric over other models of the hyperledger:

Unified systems for managing the identity of network participants do not exist, establishing provenance is so laborious it takes days to clear securities transactions (the world volume of which is numbered in the many trillions of dollars), contracts must be signed and executed manually, and every database in the system contains unique information and therefore represents a single point of failure.

Using hyperledger Fabric provides us with these technical advantages:

- **Permissioned membership**
 - Participants in the network need to know and have the credibility to get involved in blockchain transactions.
 - Performance, scalability, and levels of trust.
 - Separates transaction process into chaincode, transaction ordering and transaction validation.
 - This leads to fewer levels of trust and verification thus speeding up the entire process.
 - Optimized approach
- **Provides data only when required**
 - Enhanced privacy
 - Supports data partitioning
- **Rich queries over an immutable distributed ledger**
 - Supports full data-rich queries
- **Protection of digital keys and sensitive data**
 - HSM (Hardware Security Module) support is vital for safeguarding and managing digital keys for strong authentication. Hyperledger Fabric provides modified and unmodified PKCS11 for key generation

It's impossible with today's fractured approach to information and process sharing to build a system of record that spans a business network, even though the needs of visibility and trust are clear.

Stages in the development process:

- 1.** The building of smart contracts with clearly defined protocols to carry out the organ transaction process.
 - i.** Setup of Ganache and create a private blockchain that runs locally on our terminal.
 - ii.** Creation of a Solidity smart contract, compilation into JSON and deployment into a private blockchain.
 - iii.** Creation and deployment of smart contracts using the Metamask extension of Chrome. Metamask acts both as an Ethereum browser and as a wallet.

2. Development of a user interface to allow a smooth transaction process.
 - i. This involves creating a separate management portal to manage data for authorized hospital doctors.
 - ii. Develop an interface to facilitate viewing of reports and matching data.
 - iii. Editing information.
 - iv. Managing general information about donation centers and patients.
3. Connecting the blockchain with the user interface using a javascript framework web3.js
4. Deploying the blockchain-powered web portal and consequent testing and modifications to be made.

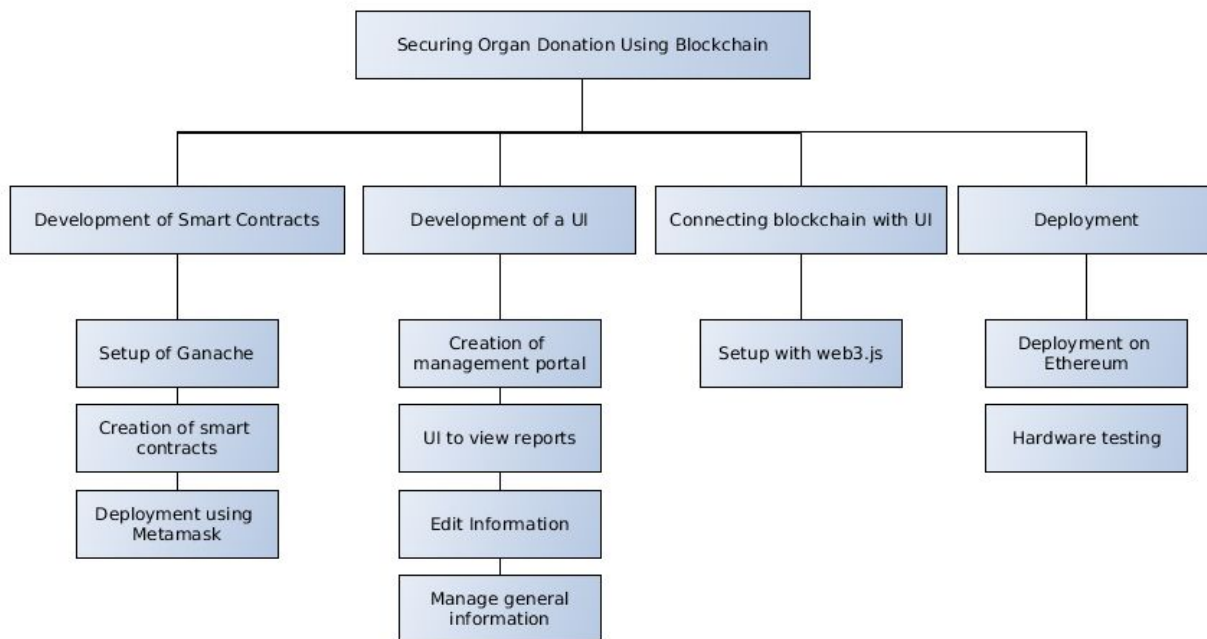


Fig: Flowchart of various modules

A detailed description of the work carried out:

1. Install all dependencies like npm, truffle, ganache.

Node Package Manager

- NPM or Node Package Manager is the world's largest Software Registry containing over 800,000 code packages.
- An online repository for the publishing of open-source Node.js projects
- Command-line utility for interacting with a said repository that aids in package installation, version management, and dependency management.
- Free of cost
- Used extensively for open-source.



Truffle

Truffle Suite is a development environment based on Ethereum Blockchain, used to develop DApps (Distributed Applications). It is a testing framework and asset pipeline for Ethereum. It is quite useful as it has built-in smart contract compilation, linking, deployment and binary management.

Installation: `$ npm install -g truffle`

Other features of Truffle are as follows:

- Configurable build pipeline with support for custom build processes.
- Scriptable deployment & migrations framework.
- Network management for deploying to many public & private networks.
- Interactive console for direct contract communication.
- Instant rebuilding of assets during development.



Ganache

Ganache is a personal blockchain for Ethereum development one can use to deploy contracts, develop your applications, and run tests. It is available as both a desktop application as well as a command-line tool. It is available for various operating systems such as Windows, Mac, and Linux. Ganache is part of the Truffle suite of tools.

When Ganache is launched, the screen will show some details about the server, and also list out a number of accounts each of which are given 100 ether. There are four pages available:

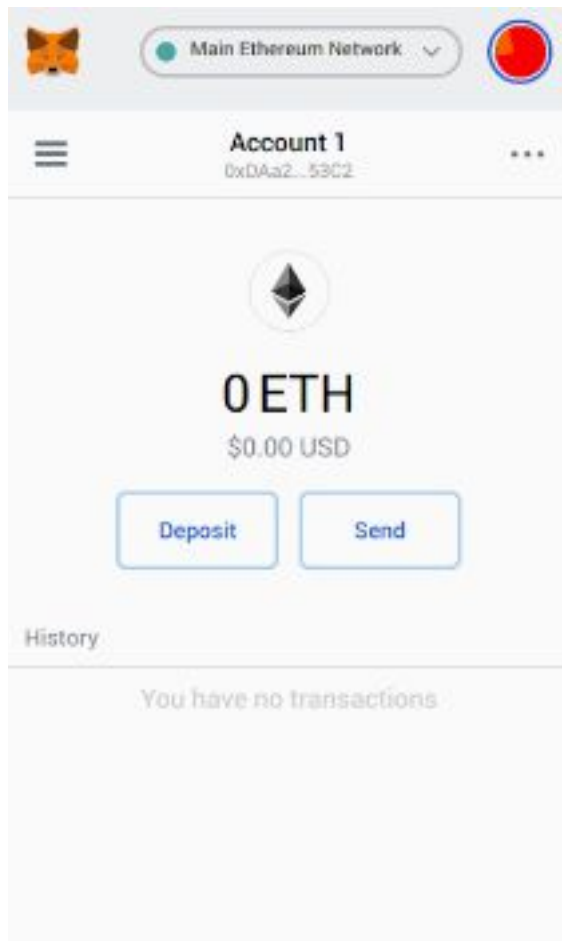
- The Accounts page shows the accounts generated and their balances. This is the default view.
- The Blocks page shows each block as mined on the blockchain, along with gas used and transactions.
- The Transactions page lists all transactions run against the blockchain.
- The Logs page shows the logs for the server, which is useful for debugging.



2. Download metamask extension in chrome browser

- MetaMask is a browser extension that acts as a bridge between internet browsers, Ethereum, and decentralized applications built on the Ethereum network.
- It allows us to run Ethereum dApps in our browser without running a full Ethereum node.
- It includes a secure identity vault, providing a user interface to manage your identities on different sites and sign blockchain transactions.
- The MetaMask add-on can be installed in Chrome, Firefox, Opera, and the new Brave browser. Users can store, send, receive, and facilitate interactions with the Ethereum network.
- It does not control any user data, and all data is encrypted on the user's browser and protected by the users MetaMask password.

Fig: Screenshot of our MetaMask Account in Google Chrome



3. Install Ganache and have a local blockchain running.

We will be using Ganache as a virtual blockchain for local testing. This tool is available under two formats:

- Ganache CLI (previously known as TestRPC)
- GUI version

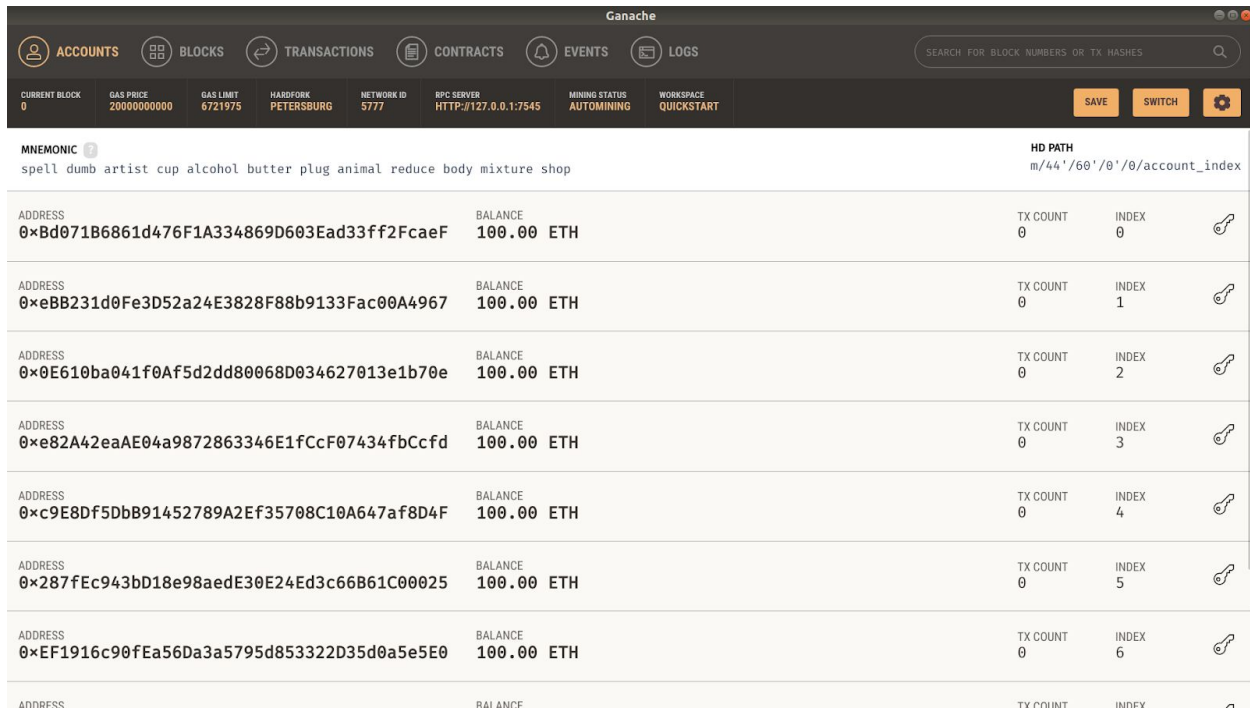
Ganache can be installed as:

```
npm install -g ganache-cli
```


Once installed, open a separate command line or tab and type in the following command:

```
ganache-cli -p 7545
```

When Ganache starts up, it generates 10 accounts (unlocked) preloaded with a balance of 100 ether each, and displays their Ethereum addresses and the corresponding private keys, as shown.



The screenshot shows the Ganache desktop application. At the top, there's a navigation bar with icons for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. Below this is a status bar with various metrics like CURRENT BLOCK, GAS PRICE, GAS LIMIT, HARDFORK, NETWORK ID, RPC SERVER, MINING STATUS, and WORKSPACE. The main area displays a list of accounts. Each account row includes a mnemonic phrase, an HD path, an Ethereum address, a balance of 100.00 ETH, a transaction count, and an index. The mnemonic for the first account is 'spell dumb artist cup alcohol butter plug animal reduce body mixture shop'.

ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	LOGS
ACCOUNTS					
CURRENT BLOCK: 0 GAS PRICE: 2000000000 GAS LIMIT: 6721975 HARDFORK: PETERSBURG NETWORK ID: 5777 RPC SERVER: HTTP://127.0.0.1:7545 MINING STATUS: AUTOMINING WORKSPACE: QUICKSTART					
MNEMONIC: spell dumb artist cup alcohol butter plug animal reduce body mixture shop HD PATH: m/44'/60'/0'/0'/account_index					
ADDRESS: 0xBd071B6861d476F1A334869D603Ead33ff2FcaeF	BALANCE: 100.00 ETH	TX COUNT: 0	INDEX: 0		
ADDRESS: 0xeBB231d0Fe3D52a24E3828F88b9133Fac00A4967	BALANCE: 100.00 ETH	TX COUNT: 0	INDEX: 1		
ADDRESS: 0x0E610ba041f0Af5d2dd80068D034627013e1b70e	BALANCE: 100.00 ETH	TX COUNT: 0	INDEX: 2		
ADDRESS: 0xe82A42eaAE04a9872863346E1fCcF07434fbCcfd	BALANCE: 100.00 ETH	TX COUNT: 0	INDEX: 3		
ADDRESS: 0xc9E8Df5DbB91452789A2Ef35708C10A647af8D4F	BALANCE: 100.00 ETH	TX COUNT: 0	INDEX: 4		
ADDRESS: 0x287fEc943bD18e98aedE30E24Ed3c66B61C00025	BALANCE: 100.00 ETH	TX COUNT: 0	INDEX: 5		
ADDRESS: 0xEF1916c90fEa56Da3a5795d853322D35d0a5e5E0	BALANCE: 100.00 ETH	TX COUNT: 0	INDEX: 6		
ADDRESS: ...	BALANCE: ...	TX COUNT: ...	INDEX: ...		

4. Configuring Truffle:

Truffle has been configured as follows:

```
networks: {  
  // Useful for testing. The `development` name is special - truffle uses it by default if it's defined here and no  
  // other network is specified at the command line. You should run a client (like ganache-cli, geth or parity)  
  // in a separate terminal tab if you use this network and you must also set the `host`, `port` and `network_id`  
  // options below to some value.  
  development: {  
    host: "127.0.0.1",           // Localhost (default: none)  
    port: 7545,                 // Standard Ethereum port (default: none)  
    network_id: "*",           // Any network (default: none)  
  },  
}
```

Developing Smart Contracts (In Progress):

The code for smart contracts will be distributed into three basic modules with the following functions:

A) Administration

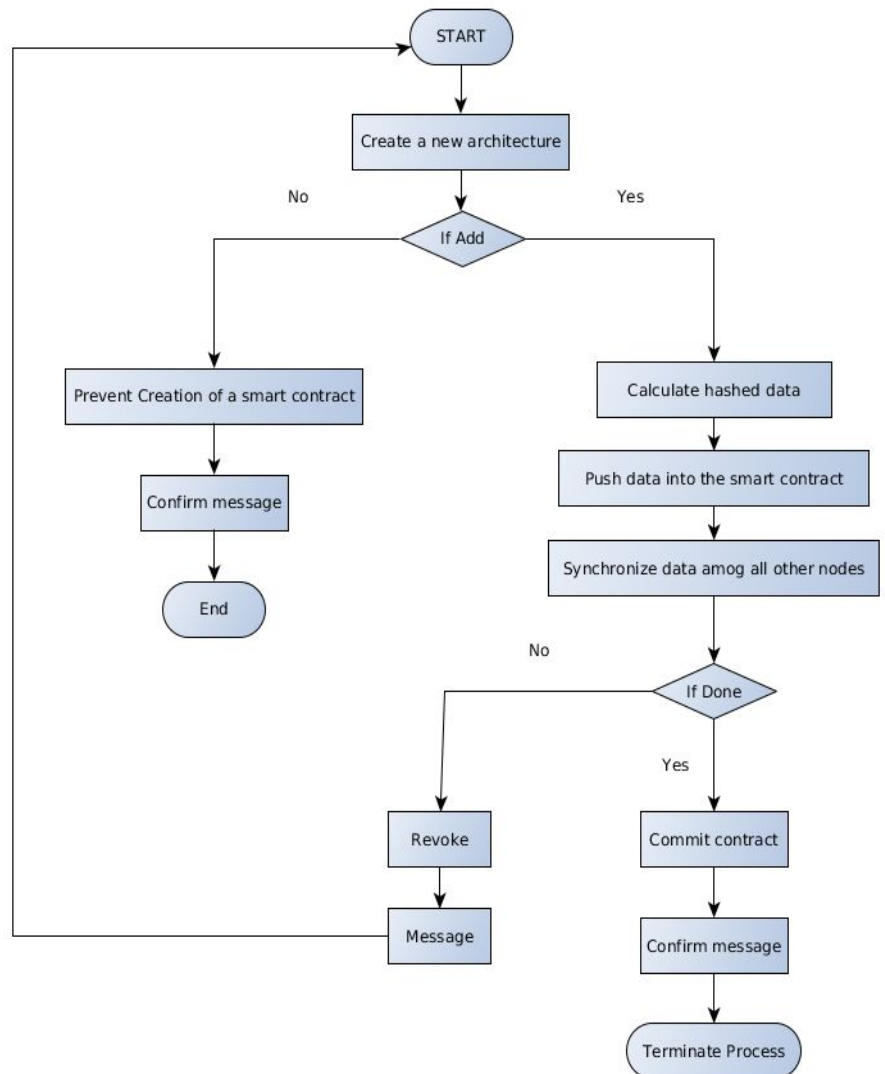
- Manage centers
- Manage general Information
- Donation center
- Manage donation data
- View matching report
- Search

B) Donor

- Apply as donor
- Edit contact information
- View information about Donation
- View map
- Check request status

C) System

- Hashing data
- Distributing data
- Matching data



Innovation / Novelty:

- Implementing blockchain in the given procedure will help us tackle the issue of trust and security. In case of addition of a block it is always stored linearly and chronologically i.e. They are always added in the end. After a block has been added based on the logic of addition of it's own hash along with the hash of the previous block it is very difficult for a hacker to decode it and change. In order to change a single block, the hacker has to change every single block after it on the blockchain.
- We have planned on implementing it through a newer approach within which we take up RFID tags and install them for the unlocking of the Organs which are being transferred. Along with blockchain, this method will in-turn guarantee more security within the entire procedure.
- For the process of development of smart contracts we firstly set up Ganache and it's deployment using Metamask. This will help to make it faster and more convenient. It will help us to fasten the process of organ donation by helping us avoid time delays. This will provide us with a perfect blend of security along with ease of application specifically in cases of exchange of valuable entities.
- All of these features will then further be available for usage by any authority through a very feasible UI platform. It will include a management portal that helps to maintain a track of organs and the authorities who are responsible for it. Along with this, the portal will also have a report viewing section to know which organ is needed as well as when and where. It also has additional features like editing information and managing general information.
- Web3.js is a collection of libraries that allows programmers to interact with these on-chain components, by being able to facilitate a connection to Ethereum nodes. Web3.js is a collection of libraries that allows programmers to interact with these on-chain components, by being able to facilitate a connection to Ethereum nodes. Hence with the application of web3.js we have connected blockchain with UI.
- In the end the deployment will be through Ethereum which runs the Smart Contracts on the EVM for applications that are attributed to being decentralized and are for mass consumption.