# COVID-19 Database on Consortium Blockchain

Swapnil Kole
swapnilkole7500@gmail.com
**IIIT BHUBANESHWAR**

***Abstract:*** *In the world where everyone is fear struck on the on-going pandemic COVID-19. There are hundreds of patients in India admitted for COVID-19. What about the patients' data? It is getting stored in a centralized database and many people are updating it [Crowd-sourced]. These may be data manipulation or there may be false data updated by faulty nodes. This problem can be solved by storing the metadata in a consortium blockchain and the main data on the Inter-Planetary File System.*

## 1. INTRODUCTION

Everyday reports of patients are coming from many hospitals. These have to be updated as soon as possible to the government trackers to provide the latest updates to people. It might get manipulated or not reach the trackers or it might be hacked and erased from the central database. To reduce such problems we can use a consortium blockchain technology integrated with Inter-Planetary File System. Blockchain is a distributed ledger-based technology where the data stored is confidential and immutable. IPFS is a decentralized database for storing and accessing files governed by some rules and uses content-based addressing.

## 2. OBJECTIVE

In this article, I'm going to develop the decentralized database for COVID-19 using a consortium Blockchain and most of the data will be stored on IPFS and the metadata will be stored on Blockchain mapped with the public key and the aadhar[1] no. of the patient in a smart contract or a chaincode depending on the platform. These technologies are used as it has limited read or write access, no intermediaries, integrity, confidentiality, and secure content based hashing. The smart contract is platform-independent. It can be designed on Ethereum, Co-Equal, Hyperledger-Fabric, etc. This all integrated as a decentralized application [Dapp].

## 3. LITERATURE REVIEW

The main technologies used in this project are:Blockchain, Inter-Planetary File System, Ethereum, Smart Contracts, Ethereum Virtual Machine, Solidity, Consensus Algorithm, and Hashing Algorithm.A brief description of all these technologies follows:

- **Blockchain:** It is a distributed ledger-based technology that uses consensus-based decisions to come to a single point of truth. It involves three main technologies which are private key cryptography, peer-2-peer network, and Blockchain protocol. The data once entered becomes immutable.

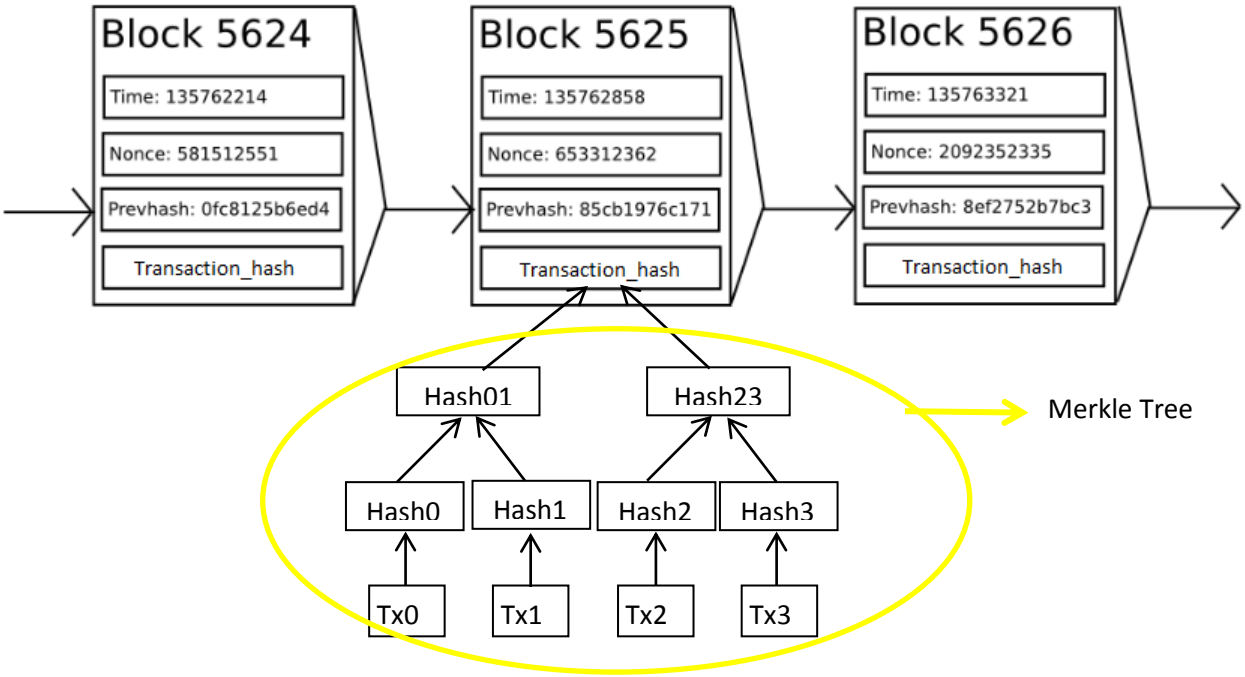



Fig. 1 Basic structure of blocks in Blockchain

Blocks in blockchain hold batches of valid transactions that are hashed and encoded into a Merkle tree. Each block includes the cryptographic hash of the prior block in the blockchain, linking the two. The linked blocks form a chain. This iterative process confirms the integrity of the previous block, all the way back to the original genesis block. Each block also has a timestamp and a nonce associated with it.

- **Inter-Planetary File System:** The Inter-Planetary File System[IPFS] is a protocol and peer-2-peer network for storing and sharing data in a distributed file system. It uses content-addressingto uniquely identify each file in a global namespace connecting all computing devices. It uses content-based addressing and Merkle Directed Acyclic Graph data structure.

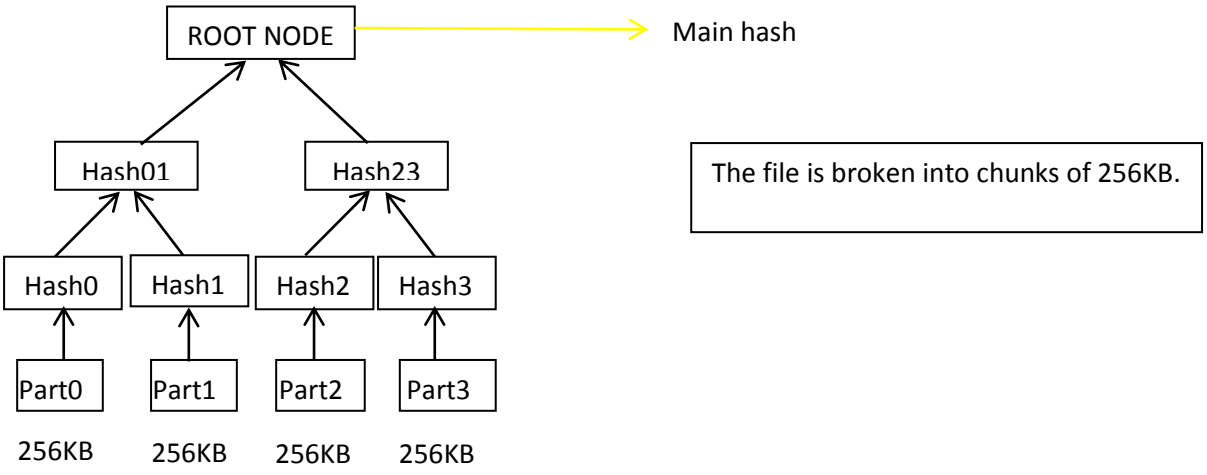



Fig 2: Storing strategy in IPFS via Merkle-DAG

A Merkle-DAG is a DAG where each node has an identifier and this is the result of hashing the node's contents — any opaque payload carried by the node and the list of identifiers of its children — using a cryptographic hash function like SHA256.

- **Ethereum:** Ethereum is an open software platform based on Blockchain technology that enables developers to build and deploy decentralized applications. Ethereum's coding language solidity helps write smart contracts. Its native currency is eth.It was founded by VitalikButerin.

- **Smart Contracts:** A **smart contract** is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. These transactions are trackable and irreversible.

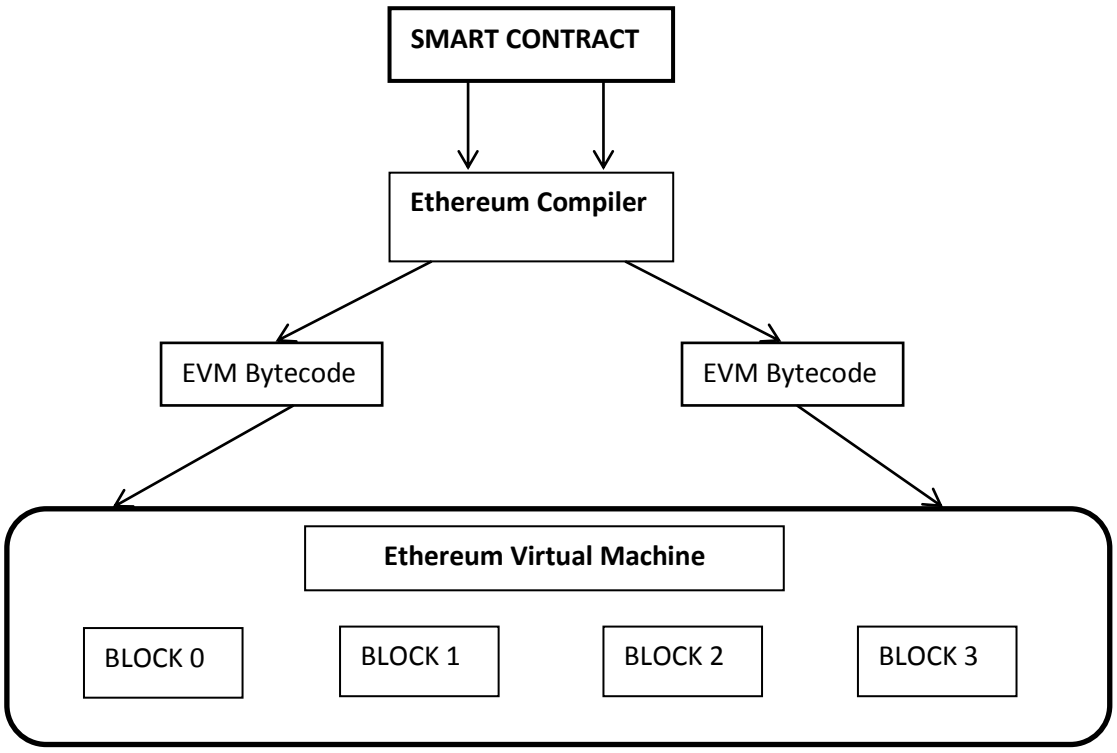



Fig 3: Execution of a smart contract

- **Ethereum Virtual Machine:** An Ethereum virtual machine provides a run anywhere obstruction layer for the smart contract. A smart contract written in HLL is translated into EVM bytecode and then deployed on EVM. Every node will host the smart contract codes on the EVM.

- **Dapplication:** A decentralized application is a computer application that runs on a distributed computing system. They have distributed ledger[DLT] based technology. It has a web-front and blockchain back-end and the smart contract connecting both.

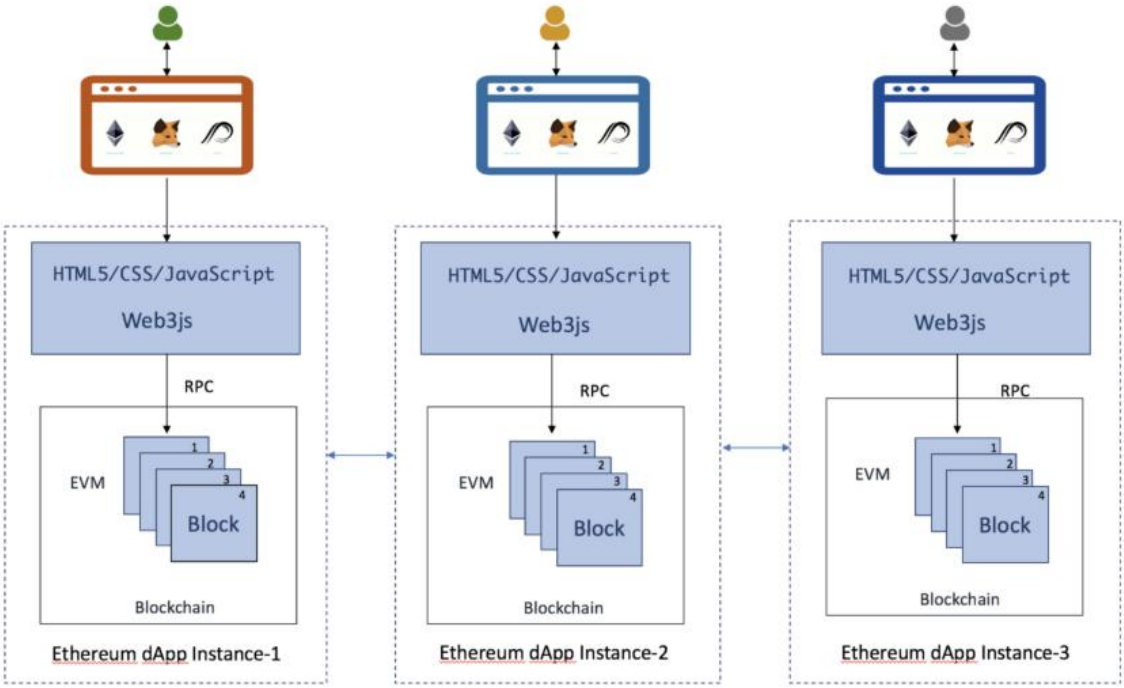



Fig 4: EthereumDapp instance

- **Solidity:** Solidity is a contract oriented language. It is designed to target the Ethereum Virtual Machine. It is statically typed language, supporting inheritance, libraries and complex user-defined types.

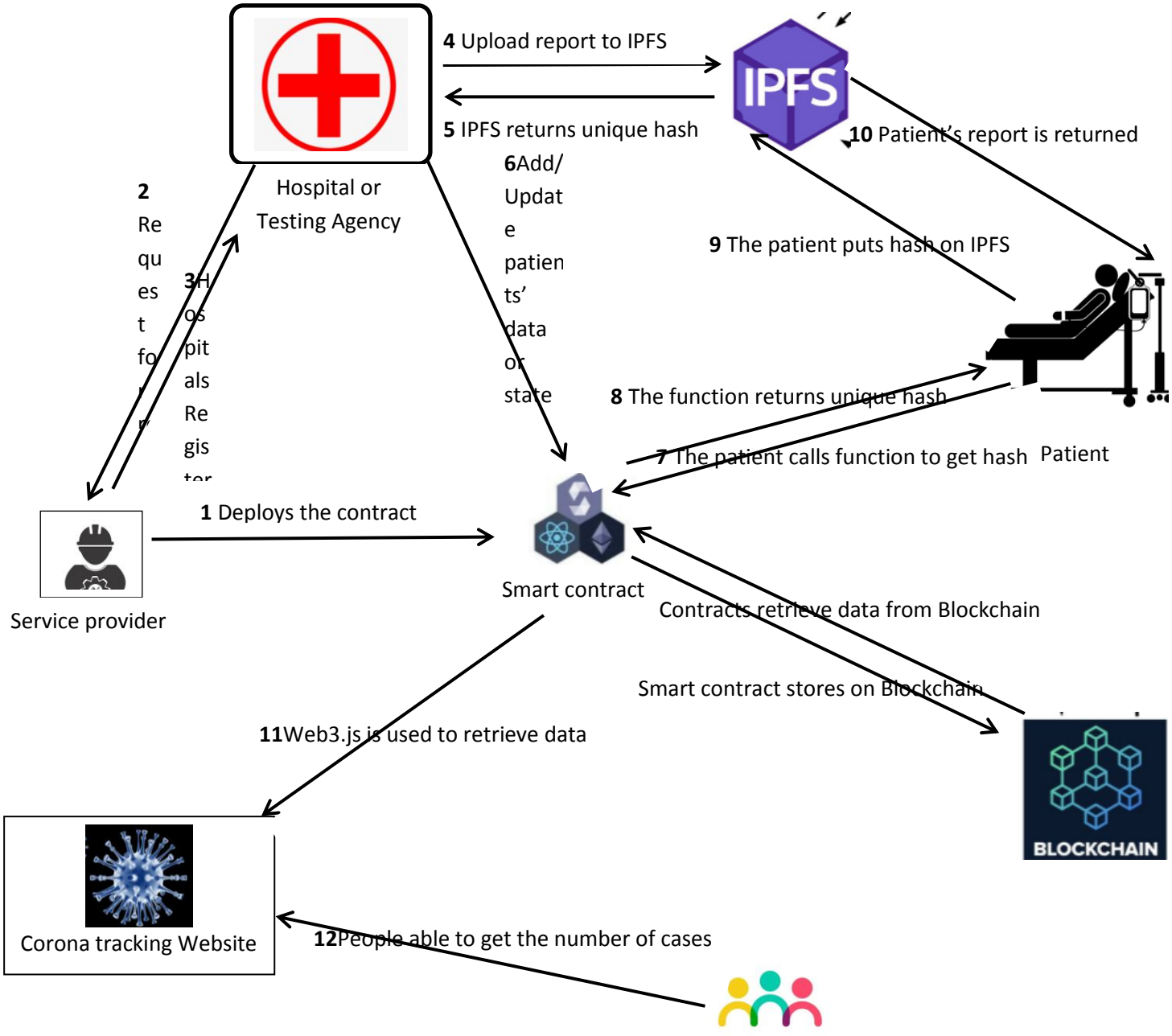
- **Consensus Algorithm:** A voting mechanism where all the nodes connected to a network vote on the validity of a block only then the block is confirmed over the blockchain. The consensus algorithms that can be used for this project are Practical Byzantine Fault Tolerance, Proof of Work and Proof of stake.

- **Hashing Algorithm:** It plays a crucial role in the blockchain process and also in the integrity of the transaction and confidentiality of data. It transforms and maps an arbitrary length of input data value to a unique fixed-length value. The algorithm should be one-way function and collision-free. Some majorly used hashing functions are SHA-256 and Keccak.

## 4. WORK

- **Contract Design:**A blockchain based decentralized database with ethereum and IPFS as main technologies. The smart contract will be coded as desired and will be deployed by the EVM. This work will have 4 groups of users involved government service provider [who will manage the smooth functioning of the smart contract and also register the different testing agencies on the smart contract], hospitals or testing agencies [who will be able to upload patients' report and update the patients' state.], patients [who can get their report], general people [will be using the web front-end of it which will show us the count of cases active, deceased or recovered.

- **Flow chart:** In this project initially the government service provider will deploy the smart contract on the EVM. Then all the hospital will request for registering. Once registered the hospital needs to put the patient's public key mapped with his aadhar no. and this aadhar no. will be mapped with patients' data in which there will be an IPFS hash which will be returned from the IPFS server once the hospital uploads the patients' report. The patients can only access their IPFS hash with their public key and they can view their report. At a later stage, the hospital can change the patient's data's state from active to recovered or deceased as desired.The general people will be provided with a website that will have a web3 connection directly to the smart contract and provide the no. of patients' on the basis of their state [active, recovered and deceased].



## 5. CONCLUSION

Data immutability and privacy is achieved by this system and now all the data is stored in a decentralized database so no one hack data from a decentralized database. The tracker directly accesses the data and gives the count of patients in each category. The data can only be updated by the testing agencies or doctors.

## 6. FURTHER RESEARCH

IPFS can be made private so that the data is stored in all the peers of India only by having the nodes in a swarm.

The testing report generators can be made IoT devices and it can directly carry out the state changes and updating of patients' reports by directly transact on a smart contract.

## 7. REFERENCES


I. Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," www.bitcoin.org

II. Gavin Wood, "Solidity: readthedocs," https://solidity.readthedocs.io/

III. Buterin, V. (2013), "Ethereum White Paper,"https://github.com/ethereum/wiki/wiki/White-Paper

IV. Juan Benet and Protocol Labs, "IPFS: readthedocs," https://github.com/ipfs