ARCHITECTURE DOCUMENT

Team:

Pratik Patel -101339642

Priya Minhas -101339644

Natasha Rupani -101339597

Shivam Rattan -101339648

Vaibhav Madaan -101339653

Team – UnPerish

COVID-19 Vaccine Supply Chain solution

Design Patterns for Blockchain – BCDV 1011

Professor: Dave Mckay & Paul Chafe

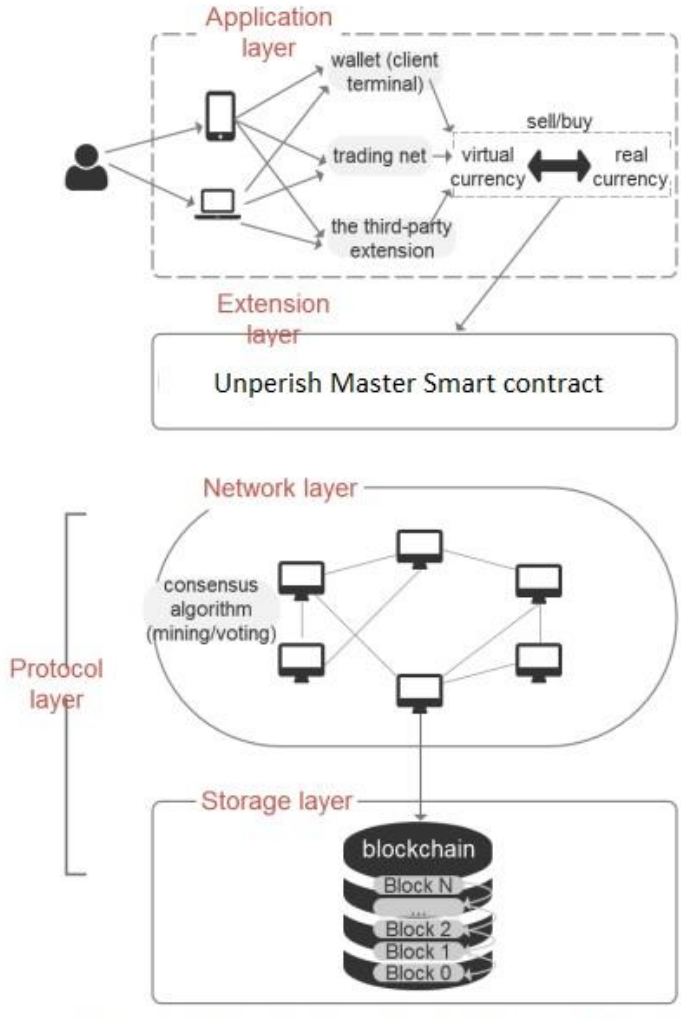
Table of Contents

Architecture of Unperish Supply chain

Solution Goal: A Blockchain solution for solving supply chain pain points of COVID 19 vaccine, this solution enables:

1. transparent tracking of all raw material, intermediate products and final finished goods throughout the supply chain,
2. building implicit trust between interacting parties since all records are un-mutable once recorded
3. reducing entry and exit barriers for participating parties

Solution Architecture diagram:



Third party solutions that can be built over the base solution

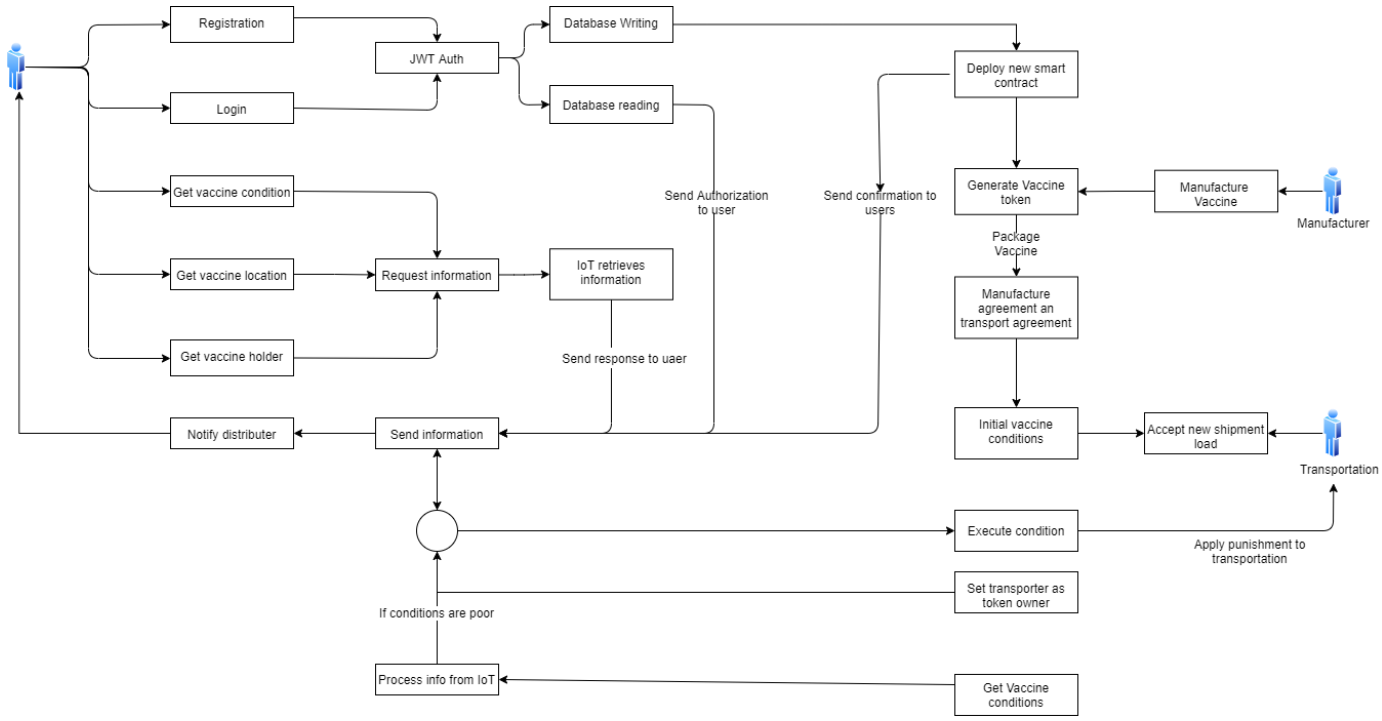
Store ERC721 tokens on Ethereum blockchain

Unperish Supply chain solution architecture

Component Technologies

1. IoT devices/sensors:
   IoT Devices and sensors will be used to capture the real-world statistics of the vaccine throughout the supply chain. Technologies like RFIDs, motion sensors, GPS trackers, temperature sensors and gyroscopic sensors will be used to gather storage and movement information
2. Ethereum Blockchain:
   ERC721 tokens will represent vaccine on the blockchain. All movements will be tracked as transactions on the chain and will be accessible by all to view and verify. Smart contract of supply chain will control the Token creation and creating appropriate transactions on the blockchain as the vaccine completes its journey through the supply chain.
3. Webpage front end:
   Basic webapp which interfaces with the blockchain. Distributors can check the status of the vaccines on the blockchain.
4. Middleware:
   Rest APIs will be created and exposed to interact with the blockchain so that all future third party integrations are seamless.
5. Backend:
   Database is used to store other relevant information of the supply chain transactions for tracking. This is done to reduce costs of storing data on Ethereum blockchain.

Unperish Process Flow



Data Definition

Below section will give detail information on how data definition would be managed for out application.

Smart Contract

a. Data Storage:

Vaccine struct
uint256 id
string manufacturer
bytes6 serialno
int16 thermal
bytes26 location (Byte array to store latitude and longitude in the form
<lat>;<long> Example: -89.12345678;-168.12345678)
address requestingDistributor

mapping(uint256 => Vaccine) private vaccines (Maps token ids to vaccine info)

uint256 private nextId (Next token id to be minted)

b. Functions
   i. addManufacturer: Add address as manufacturer role
   Inputs: (address manufacturer) public onlyAdmin()
   ii. addTransporter: Add address as transporter role
   Inputs: (address transporter) public onlyAdmin()
   iii. addDistributor: Add address as distributor role
   Inputs: (address distributor) public onlyAdmin()
   iv. makeVaccine: Mints a new Vaccine token
   Inputs: (string memory _manufacturer, bytes6 _serialno, int16 _thermal, bytes26 _location) public onlyManufacturer()
   v. orderVaccine: Set requesting Distributor in Vaccine token
   Inputs: (uint256 _id) public onlyDistributor()
   vi. thermalMonitor: Update current thermal and location info
   Inputs: (uint256 _id, int16 _thermal, bytes26 _location) public onlyTransporter() onlyOwner(_id)
   vii. transferVaccine: Transfer Vaccine
   Inputs: (uint256 _id, address _to, int16, _thermal, bytes26 _location) public onlyOwner(_id)

viii. getVaccineTrackingInfo: Get current vaccine info. Returns thermal, location and current owner
Inputs: (uint256 _id) public view onlyRequestingDistributor(_id) returns (int16, bytes26, address)

Database:

- name (string)
- password (string)
- address(string)
- registration date(date)

Backend

1. Endpoints

ADMINISTRATOR

EndPoint	Description
/admin/addManufacturer	This endpoint calls addManufacturerfrom smart contract
/admin/addTransporter	This endpoint calls addTransporterfrom smart contract
/admin/addDistributor	This endpoint calls addDistributorfrom smart contract

DISTRIBUTOR

EndPoint	Description
/distributors/index	get list of registered distributors in db
/distributors/new	register new distributor
/distributors/login	distributor login
/distibutors/vaccineInfo/:tokenId	Get current info of vaccine. This endpoint callsgetVaccineTrackingInfo from smart contract
/distributors/orderVaccine/:tokenId	This endpoint calls orderVaccine from smart contract

MANUFACTURER

EndPoint	Description
/manufacturers/transferVaccine	This endpoint calls transferVaccine from smart contract
/manufacturers/makeVaccine	This endpoint calls makeVaccine smart contract

TRANSPORTER

EndPoint	Description
/transporters/transferVaccine	This endpoint calls transferVaccine from smart contract

ORACLE (IoT)

EndPoint	Description
/transporters/thermalMonitor	This endpoint calls thermalMonitor from smart contract

Tech Stack

Smart Contract

- Solidity - smart contract programming language
- Truffle - dApp environment
- Ethereumjs-util - utility functions for Ethereum
- Truffle-assertions - additional assertions for truffle
- Bignumber.js - library to handle big numbers

Backend

- Express.js - web application framework
- MongoDB - NoSQL database
- Mongoose - object data modeling (ODM) library for MongoDB and Node.js
- Async - library to perform asynchronous operations
- Express validator - middleware to validate data
- Bcryptjs - library to perform cryptography
- JWT.IO - JSON Web Tokens to allow, decode, verify, and generate JWT
- Jest - library for tests
- Web3.js - interact with smart contracts
- Dotenv - loads environment variables from a .env file

Frontend

- Bootstrap- design system
- ReactJS - frontend library
- Axios - HTTP requests