Universal Vaccination Passport Using Blockchain

Github repo: https://github.com/sjsucmpe272SP22/Universal-Vaccination-Passport
Hosted Application URL: https://shubhadasanjaypait.wixsite.com/my-site-5
Shubhada S. Paithankar (shubhadapaithankar@sjsu.edu),
Shri Ram V N(shrivatson.ramaratnamgiridhar@sjsu.edu),
Nikhil Konduru (nikhil.konduru@sjsu.edu),
Arun Satvik Mallampalli (venkataarunsatvik.mallampalli@sjsu.edu)

Abstract- **Healthcare facilities process millions of patients every day all around the world. These institutions are the backbone of the healthcare system in any country. Each prescription and purchase made is considered an event in our approach. Healthcare and pharmaceutical facilities will generate millions of events every day. We designed a data pipeline leveraging Apache Kafka to process and transform this data. These events are interpreted in real-time. We build a distributed ledger of medical records for patients by consuming these events. An event is only consumed if it passes certain checks. We enforced that by writing smart contracts on top of our ledger. Our simple web application displays vaccine passports.**

The benefit of using DLT (Distributed Ledger Technology) is the elimination of middlemen in maintaining a global ledger of medical records. Also, we have total automation in building vaccine passports with no one authority having an absolute monopoly over the records.

I. Introduction

In the Covid-19 pandemic, almost every country in the world conducted campaigns worldwide to

vaccinate its population. As a result, many countries have been asking for proof of vaccination for travel. However, there is no single reliable source of health records available. Thus, It isn't easy to build a vaccine passport accepted everywhere. We intend to solve this problem by leveraging the blockchain's distributed ledger technology (DLT).

II. Previous Work

Even before the rise of COVID-19, the healthcare and life sciences industries faced significant issues, including interoperability, privacy, and supply chain traceability.

As the pandemic continues, healthcare and the life sciences face new challenges, including adapting supply chains to deliver protective equipment and rapidly developing treatments, tests, and vaccines. Meanwhile, healthcare professionals are grappling with

managing consent and keeping individual health data secure as they look to leverage health data to safely re-open for business.

Blockchain has already demonstrated its value in healthcare and the life sciences by enabling trust and collaboration and will continue to be at the forefront of addressing ever more challenges.

III. Use Case-business value

1. Our Portal generates vaccination passports for anyone in the world.
2. The portal is built on a blockchain where tamper-proof vaccination records are stored and maintained.
3. It can be used for verifications anywhere in the world.

IV. Persona

1. Governments All Over the World
2. WHO (World Health Organizations)
3. Private Medical Facilities

V. The flow of Project

1. Vaccination Sites will stream vaccination events to Lamda Function in the cloud (JSON formatted event)
2. Lambda function will call the POST endpoint */vaccination/record* with the event data in the POST JSON body.
3. Backend code will use the web3 library to call smart contract `createRecord` to write transactions on the private blockchain (Ganache)
4. Use the vaccination portal to enter your unique Government ID (passport No, SSN, TIN, Aaadhar Num)
5. Frontend calls our backend API GET endpoint `/vaccination/record/{id}` to verify and fetch a valid vaccination record.
6. Backend code will use the web3 library to use the smart contract `getRecord` to fetch vaccination records from the private blockchain.

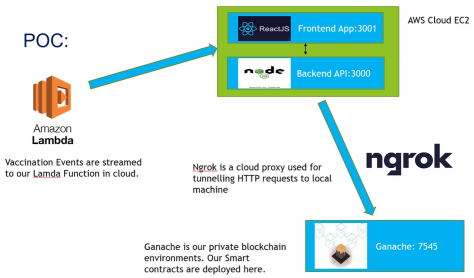


Figure **1**: POC Diagram

VI. Architecture Diagram

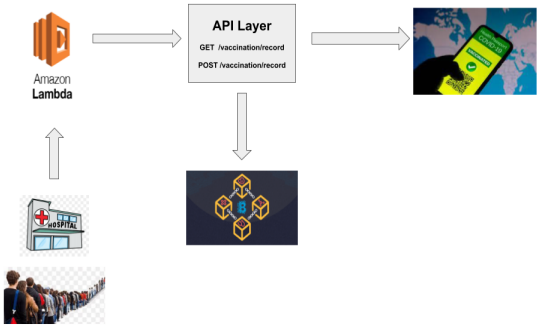


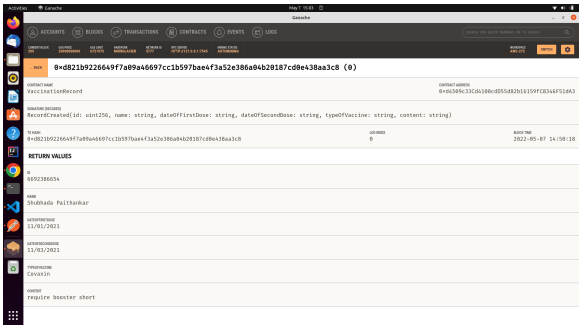
Figure 2**:** Architectural Flow Diagram

1. Client-Side Apps will stream vaccination records to the Lambda Function in the AWS cloud. Lambda Function scales with the rate of load.
2. Lambda Function will hit our API layer, GET and POST APIs.
3. The APIs will call Smart Contracts and write transactions to the blockchain in the cloud.

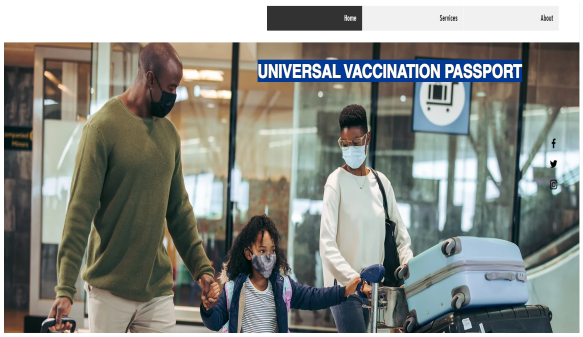
Our Smart Contract is as follows:

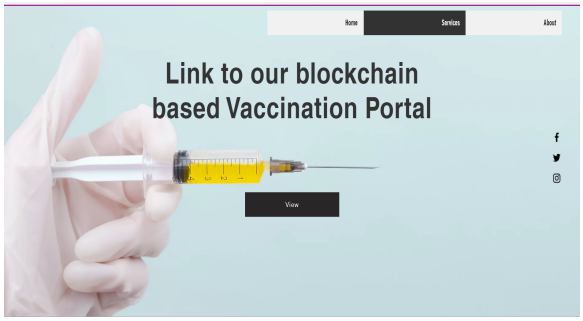
```
pragma solidity^0.5.0;
pragma experimental ABIEncoderV2;
contract VaccinationRecord{
uint public recordCount=0;
struct VaccinationRecord{
uint256 id;
string name;
string dateOfFirstDose;
string dateOfSecondDose;
string typeOfVaccine;
string content;
}
mapping(uint256=>VaccinationRecord)public
vaccinationRecordMappings;
event RecordCreated(
uint256 id,
string name,
string dateOfFirstDose,
string dateOfSecondDose,
string typeOfVaccine,
string content
);
event RecordFetched(
uint256 id,
string name,
string dateOfFirstDose,
string dateOfSecondDose,
string typeOfVaccine,
string content
);
constructor()public {
createRecord(1,"name","dateOfFirstDose","da
teOfSecondDose","typeOfVaccine","content");
}
function createRecord(uint256 id,string
memory name,string memory
dateOfFirstDose,string memory
dateOfSecondDose,string memory
typeOfVaccine,string memory content)public {
recordCount++;
vaccinationRecordMappings[id]=VaccinationR
ecord(id,name,dateOfFirstDose,dateOfSecon
dDose,typeOfVaccine,content);
emit
RecordCreated(id,name,dateOfFirstDose,date
OfSecondDose,typeOfVaccine,content);
}
function getRecord(uint256 id)public
returns(VaccinationRecord memory record){
VaccinationRecord memory
fetchedRecord=vaccinationRecordMappings[i
d];

if(fetchedRecord.id>0){
emit
RecordFetched(fetchedRecord.id,fetchedRec
ord.name,fetchedRecord.dateOfFirstDose,fetc
hedRecord.dateOfSecondDose,fetchedRecor
d.typeOfVaccine,fetchedRecord.content);
return fetchedRecord;
}
VaccinationRecord memory emptyRecord;
return emptyRecord;
}
}
```

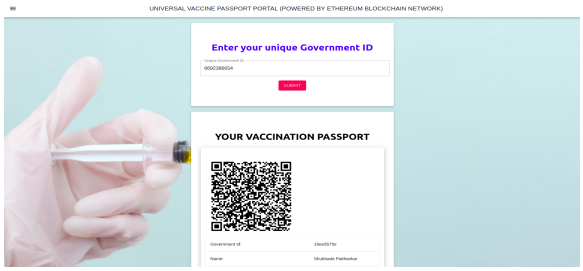
4. We are using Ganache and rock to simulate one node blockchain for testing for the Demo.

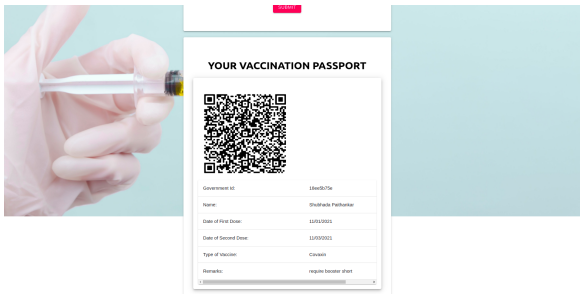


5. Our Vaccination Portal will use the same API layer to generate a Vaccination Passport with a QR code.









VII. Deployment Diagram

The deployment diagram is as follows

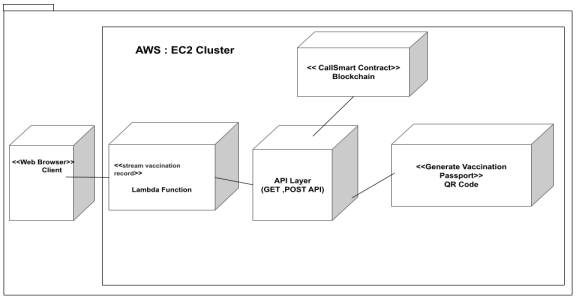


Figure 3: Deployment Process

VIII. Tools used in project

1. Frontend: Reactjs
2. Backend API: Nodejs
3. Blockchain Environment: Ganache(to create and deploy smart contracts on one node blockchain) and Truffle (manage and compile smart contracts).
4. Lambda: Consume vaccination events.

IX. Brief Information About Blockchain

What is Blockchain?

Blockchain is a term used to describe DLT or Distributed Ledger Technology. Blockchain is used to create a storage system for data in a distributed and immutable manner.

What are some of the standard terms in Blockchain Technology?

(i) Distributed Ledger:

A database that contains all the blocks in the blockchain is called a ledger. This ledger is maintained in a distributed manner on all

the nodes. Hence it is called a distributed ledger.

(ii) Peers (nodes or hosts):

The distributed ledger is replicated and maintained by multiple nodes or hosts. The ledger is made immutable by using cryptographic algorithms. Blockchain networks are peer-to-peer (P2P) networks. There is no centralized authority.

(iii) DLT:

DLT is an acronym for Distributed Ledger Technology.

(iv) Smart Contracts:

Smart Contracts are programs stored on the blockchain. These programs have rules (logic) to accept or deny a transaction. If the rules specified in the smart contract accept a transaction, it is appended as a new block in the blockchain.

(v) Consensus Protocol:

The consensus protocol gives a specific method for verifying whether a transaction is valid or not. It provides a way of reviewing and confirming what data should be added to a blockchain's record. Because blockchain networks typically don't have a centralized authority dictating who is right or wrong, nodes on a blockchain all must agree on the state of the network, following the predefined rules.[9]

Advantages of Blockchain Technology

(i) Immutability or Data Integrity:

Data written to a blockchain can not be changed. Blockchain is an immutable chain of cryptographically encrypted blocks. Let's make a comparison with traditional relational databases like MySQL and Oracle. Applications write transactions to these relational databases. However, a hacker or an ill-minded administrator can access and modify data. A blockchain-based system will report an error if data is changed on the ledger. In the blockchain-based system, data is stored in a distributed manner on multiple nodes. Hence, a valid state can be recovered from any one of the nodes.

(ii) Distribution: Blockchain achieves trust by replicating data on several nodes. Suppose one of the peers goes into an invalid state. Other peers can filter it out. As long as the majority of the nodes agree on data being in a valid state, data can be trusted.

Types of Blockchain Networks

(1) Permissioned blockchain:

Permissioned like Hyperledger Fabric is more suitable for IoT use cases [1]. There is no extra cost associated with writing new transactions. These blockchains are maintained by a group of authorities rather than the public.

(2) Permissionless blockchain:

Public blockchains like Ethereum and Bitcoin are called permissionless blockchains. These are not suitable for storing or streaming a large amount of data or executing complex computation due to the inherent transaction cost of storing data on-chain and constraints of computation

complexity. It should store only the representation of data or metadata rather than the raw data itself.[1]

Reference:

[1]. IBM Blockchain Renaltu Usecase
https://www.ibm.com/case-studies/renault/

[2]. Microsoft Blog on fighting piracy with Blockchain
https://markets.businessinsider.com/news/currencies/microsoft-msft-ethereum-blockchain-fight-piracy-digital-tech-public-ledger-2021-8

[3]. Ganache private Blockchain use case and deployment
https://trufflesuite.com/docs/ganache/

[4]. Truffle is widely considered the most popular tool for blockchain application development.
https://trufflesuite.com/docs/truffle/