MINOR PROJECT REPORT

On

**Securing Cloud Transactions**

**using Blockchain**

By

Srejan Kumar Bera (19103011)

Vaibhav Chawla (19103027)

Akarsh Puri (19103028)

Under Supervision of

Dr. Taj Alam

Submitted in fulfillment of the

Degree of Bachelor of Technology

In

Computer Science & Engineering and Information Technology



Department of Computer Science & Engineering and Information Technology

Jaypee Institute of Information Technology, Noida-62, Uttar Pradesh

March 2022

**PROBLEM STATEMENT:**

Cloud computing has proven to be a key technology for delivering infrastructure and service at a low cost. The rapid growth in the use of this technology has been observed but the security issues related to it have still not been addressed completely.[1] Some of the major security issues related to cloud are data loss or leakage, data privacy and confidentiality and accidental exposure of credentials. Further some attacks involving malicious use also compromise the authenticity of the blockchain[6]. These attacks include On/Off attack, Collusion Attack, Sybil Attack, DoS attack and theft of data. Further, small scale cloud services cannot ensure proper security, therefore, in order to make them stand a chance in this industry an alternative to current scenarios must be formulated.

**HOW WE INTEND TO SOLVE THE PROBLEM?**

We intend to make use of Ethereum blockchain to prevent the malicious use of cloud services. Issues related to data leakage and credential exposure are implicitly handled by the blockchain further placing usage logs in the blockchain and checking them from the blockchain via APIs in the website, helps to check any malicious use. We are working towards making a dummy cloud with two services which have a maximum limit of users. One service is made for individual use and other for group use. The client asks for service from the server and based on the availability the server provides the service, basic entries like user, start time etc are placed into the blockchain. When the client exits the service the end time is noted. Based on these entries we can point out the malicious behaviour we observe through the logs. Further metrics of all the users will be provided to the admin with an option to blacklist/block any user in order to flag/block his/her transactions in future.

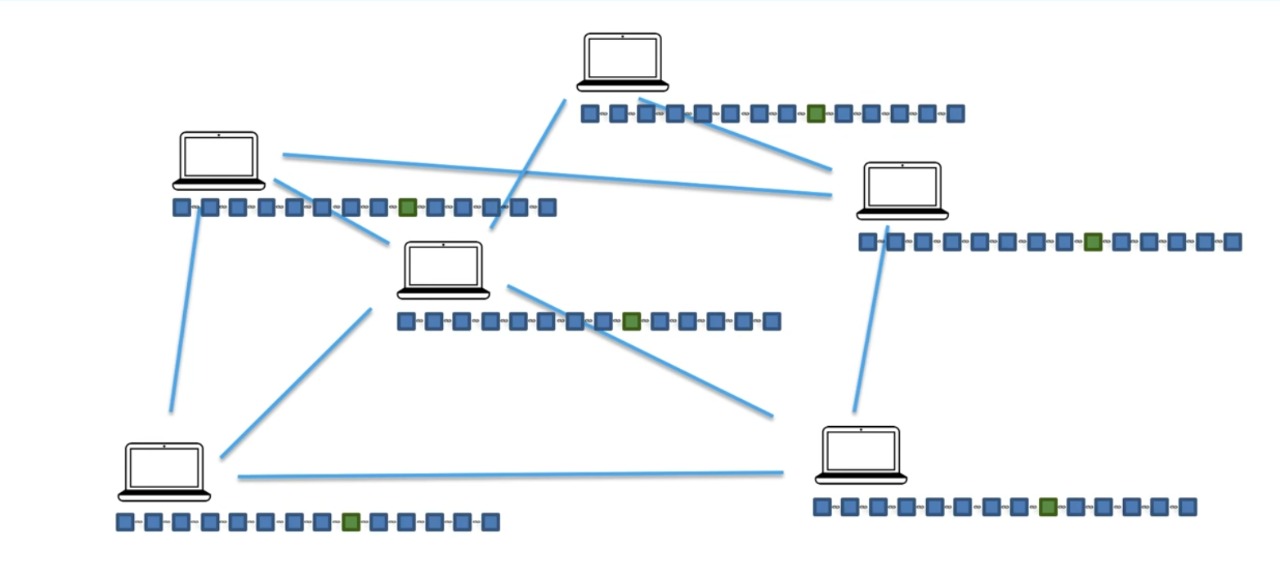
**FEASIBILITY**

**KEY CONCEPTS**

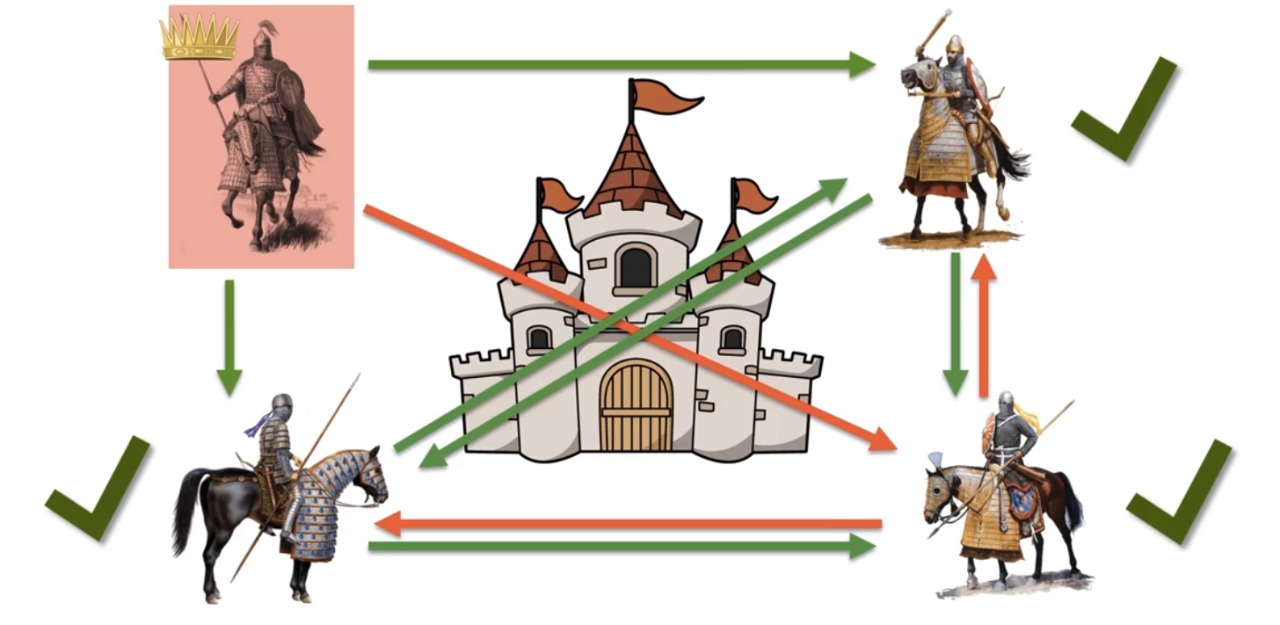
**1. Encryption -** Each block will use Keccak-256 encryption for encryption of stored data which includes the user's data and previous hash. The requirements fulfilled by this algorithm is that it is one-way, deterministic, avalanche effect and withstands collisions.

**2. Immutable Ledger -** Tampering of data in one block will change the hash of the tampered block and therefore it's link to the further block will be damaged thus ensuring that no data is overwritten. Further in the case if malicious code overwrites all the blocks in the chain by changing the previous hashes this has to be within the time blockchain verifies their data with other copies of it in distributed peer-to-peer networks. If all previous hashes are not changed then the blockchain can be disturbed which may lead to loss of data but this data is recovered by other copied blockchains in a distributed peer-to-peer network.

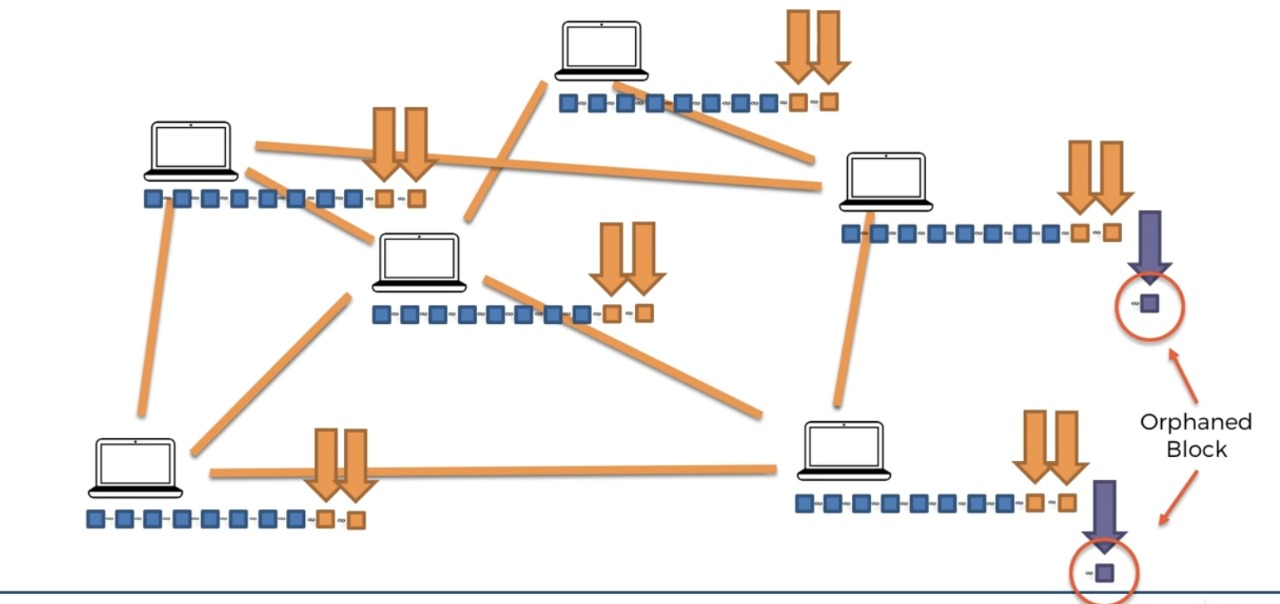
**3. Distributed P2P Network -** Each blockchain is copied in distributed P2P which prevents the loss of data and malicious change in the data of a block of the blockchain. Therefore, for the attacker to succeed he/she will have to alter data in all copies of the blockchain in its peers within a short time. This time is that when each instance verifies it to others.



**4. Byzantine Fault Tolerance –** Blockchains are decentralized ledgers which, by definition, are not controlled by a central authority. Due to the value stored in these ledgers, bad actors have huge economic incentives to try and cause faults. That said, Byzantine Fault Tolerance, and thus a solution to the Byzantine Generals’ Problem for blockchains is much needed [1].



**5. Proof of Work** - Proof of Work (PoW) is the **consensus algorithm** in a blockchain network. The algorithm is used to confirm the transaction and creates a new block to the chain. In this algorithm, **minors** (a group of people) compete against each other to complete the transaction on the network. The process of competing against each other is called **mining**. As soon as miners successfully created a valid block, he gets **rewarded**. The most famous application of Proof of Work (PoW) is Bitcoin.



**TOOLS:**

Visual Studio Code

Node Package Manager

MongoDB

Ethereum (Solidity)

**LIBRARIES USED:**

Express – It is a framework used for creating a server and serving requests to frontend.

Nodemailer – It is used for confirmation mail for authentication.

Passport – This library is used for setting up Google Authentication.

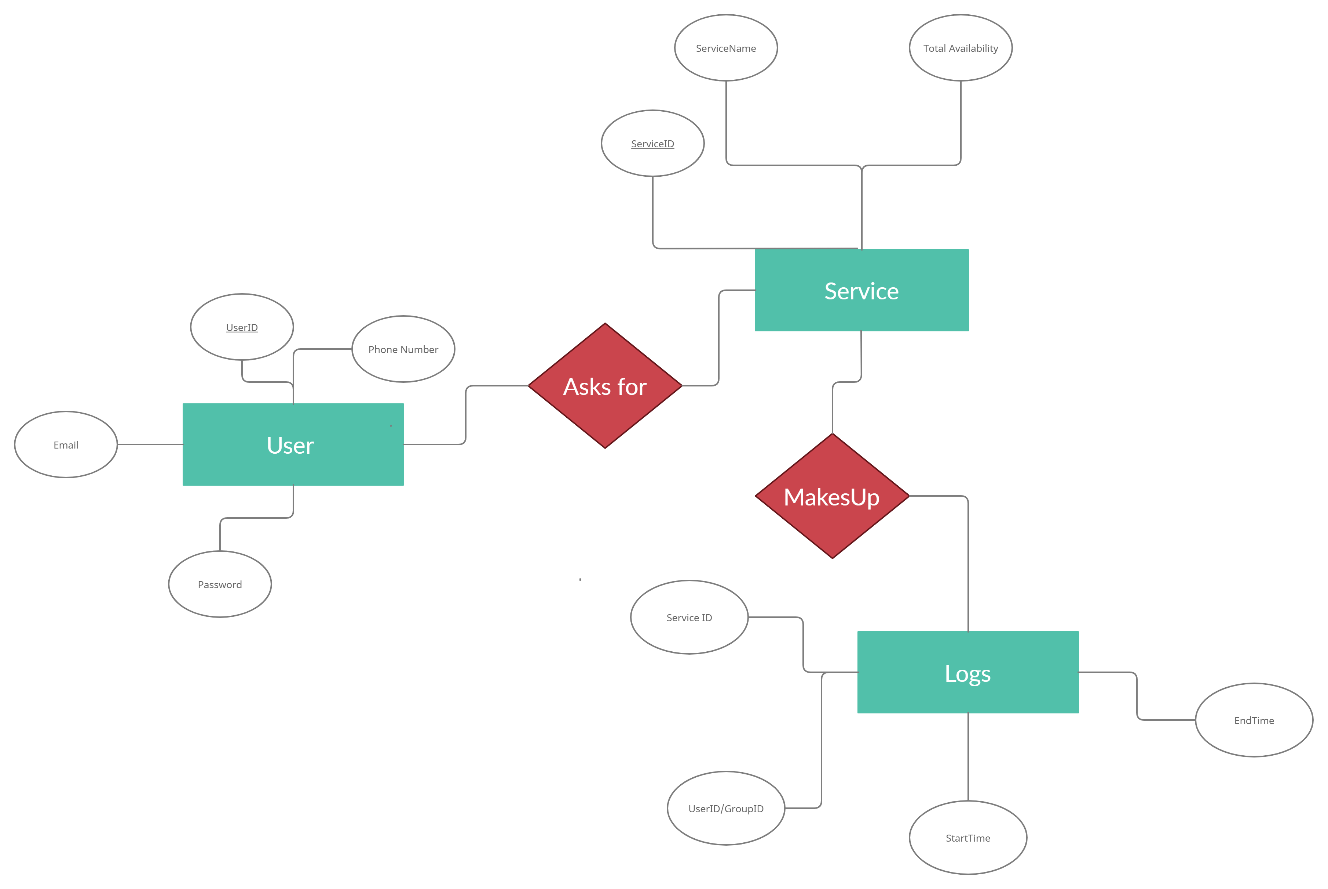
Mongoose – Mongoose is an Object Data Modelling (ODM) library for MongoDB and Node.js

React – This library is used for building composable user interfaces.

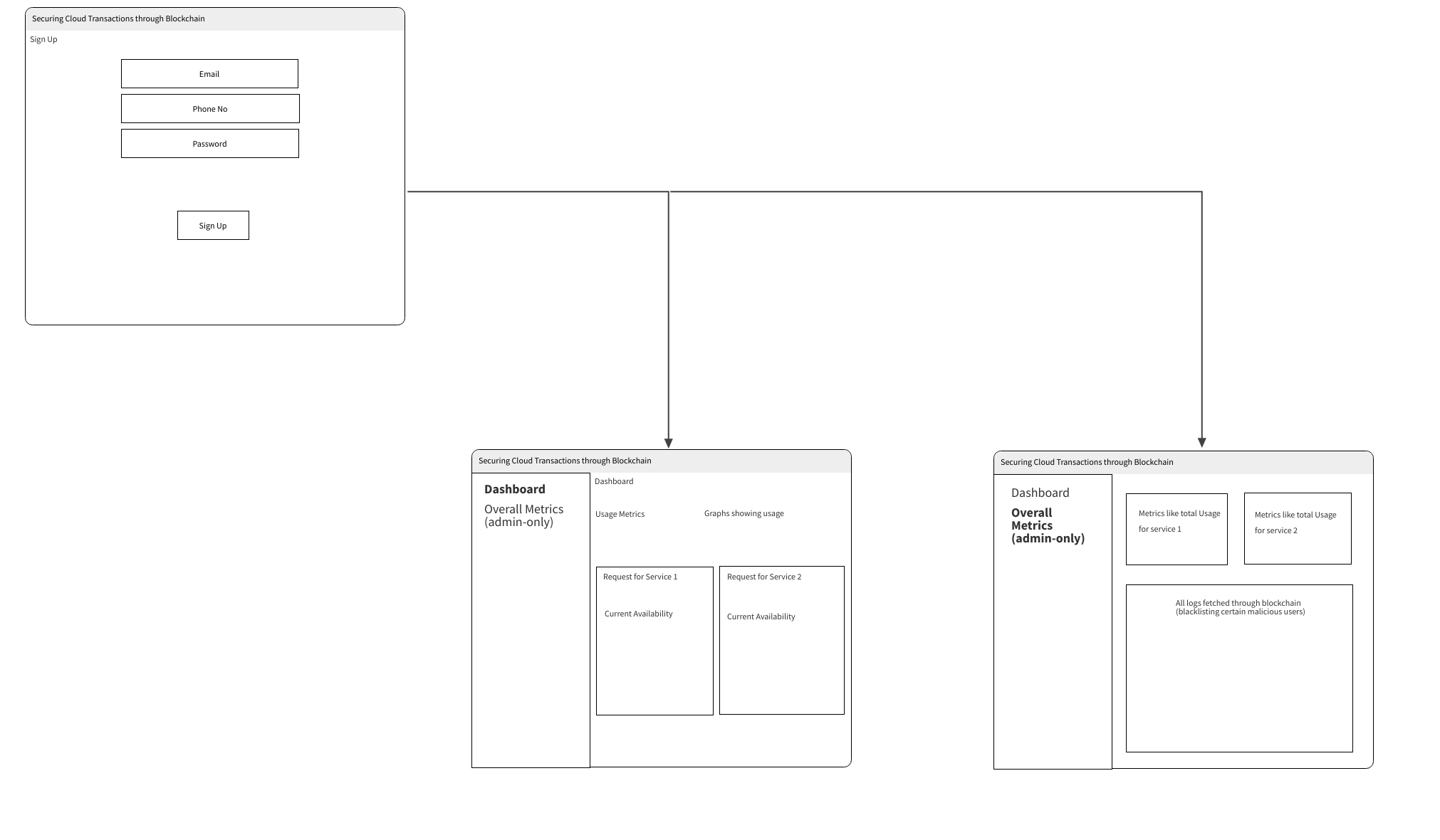
Axios – This is used for sending api request to backend and fetching data from backend.

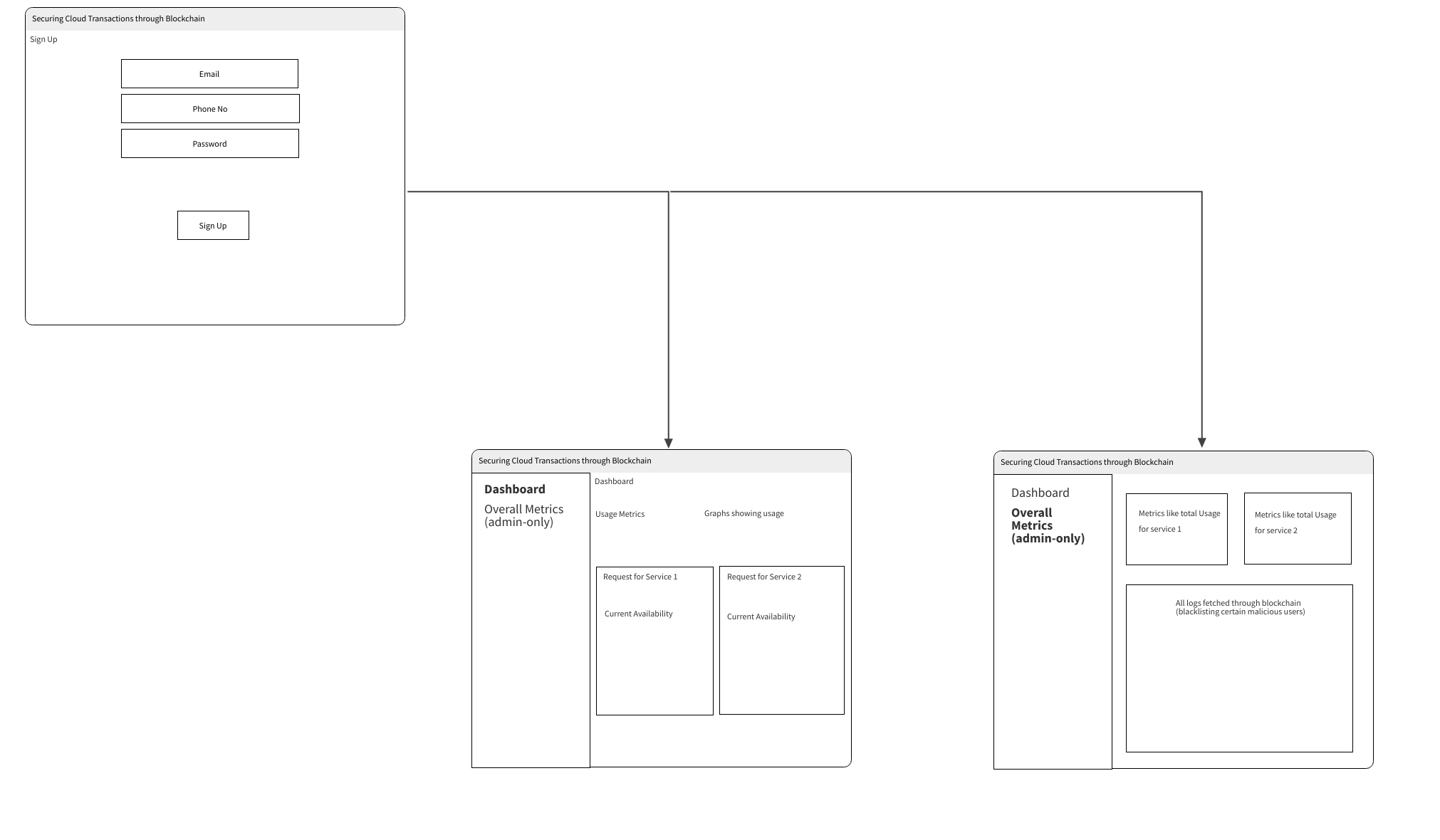
BootStrap – This framework is used for frontend CSS Designing.

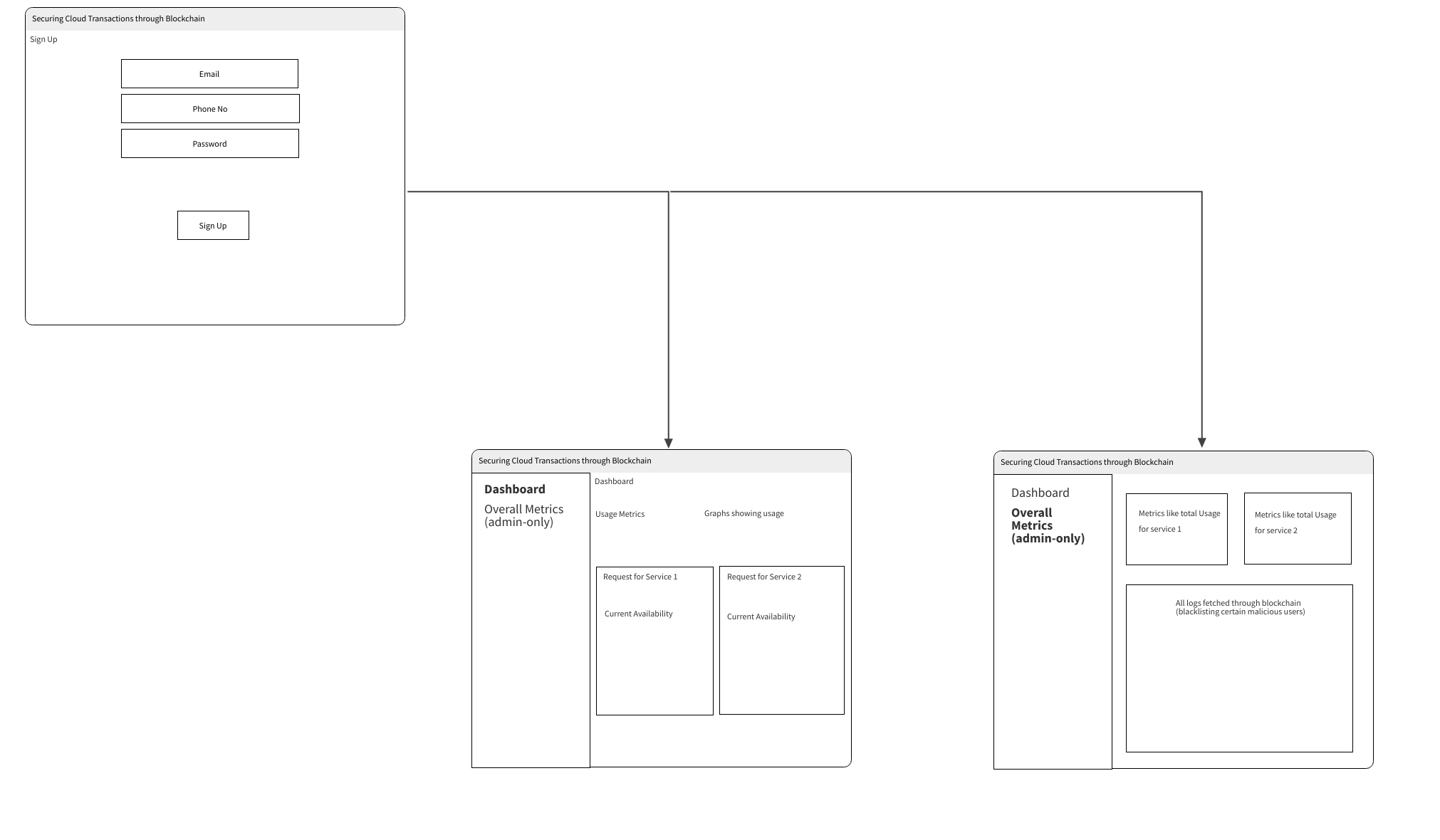
**SCHEMA**



**APPLICATION WIREFRAME**







**REFERENCES:**

**[1]** The Byzantine Generals Problem LESLIE LAMPORT, ROBERT SHOSTAK, and MARSHALL PEASE SRI International

**[2]** DOLEV, D. The Byzantine generals strike again. J. Algorithms 3, 1 (Jan. 1982).

**[3]** PEASE, M., SHOSTAK, R., AND LAMPORT, L. Reaching agreement in the presence of faults. J. ACM 27, 2 (Apr. 1980), 228-234.

**[4]** Gupta, Ashok & Siddiqui, Shams & Alam, Shadab & Shuaib, Mohammed. (2019). Cloud Computing Security using Blockchain. 6. 791-794.

**[5]** https://www.geeksforgeeks.org/benefits-and-applications-of-blockchain-in-cloud-computing/

**[6]** ISC2 Cloud Security Report 2021