MINOR PROJECT REPORT

On

Securing Cloud Transactions through 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, 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.

1. INTRODUCTION

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 behavior 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.

1.1 Various Attacks to be Tackled

1. Sybil Attack - A Sybil attack is a kind of security threat on an online system where one person tries to take over the network by creating multiple accounts, nodes or

computers. This can be as simple as one person creating multiple social media accounts. In really large-scale Sybil attacks, where the attackers manage to control the majority of the network computing power or hash rate, they can carry out a 51% attack. In such cases, they may change the ordering of transactions, and prevent transactions from being confirmed. They may even reverse transactions that they made while in control, which can lead to double spending. Blockchains manage sybil attacks by making use of different "consensus algorithms" to help defend against Sybil attacks, such as Proof of Work, Proof of Stake, and Delegated Proof of Stake. These consensus algorithms don't actually prevent Sybil attacks, they just make it very impractical for an attacker to successfully carry out a Sybil attack.

2. Dos Attack - A Distributed Denial-of-Service (DDoS) attack can occur in this blockchain. New transactions that are created correctly are always stored in memory pool, and only some transactions of a given size can enter in one block. A DDoS attack in blockchain can be accomplished by an attacker who creates an infinite number of correct transactions for a short time and delivers them to the connected nodes. The node receiving the transactions becomes overwhelmed with its own memory pool getting full and cannot store any transaction in own memory pool. As a result, it takes more time to deliver to other connected nodes after memory pool management. Victim nodes can lose their ability as a blockchain node. Also, the memory pool of all nodes filled with transactions that created by the DDoS attack node, which increases the time for a transaction created by a normal node to enter a block. Seriously, transaction can be remains in memory pool for a long period time, then removed from memory pool. Hence, it eventually cannot use the blockchain system. We define this style of DDoS attack as an overflood attack. Through the overflood attack, an attacker may not only damage a specific node, but also gradually destroy the entire blockchain system itself.

3. On/Off Attack - Trust helps to make decisions in unpredictable circumstances. Every security system depends on trust, in one form or another, among users of the system. A trust management scheme can used to help an automated decision-making

process for the access control strategy. Unintentional temporary errors are possible in system, the trust management solution must provide a scheme to allow nodes to recover trust. However, if a malicious node tries to disguise the malicious behaviors as temporary errors, that node may be given more opportunities to attack the system by disturbing the redemption scheme. Here the attacker behaves well and badly alternatively in system. Thus, a new trust management and redemption scheme that can discriminate between temporary errors and malicious behaviors and it is a new flexible trust management scheme that can well detects and defends against the On-off attacks.

1.2 Use cases

This app can be used by small-scale cloud services to avail the high-security provided by blockchain. Further, existing medium-scale cloud services can also use this app to analyze logs by stored in the blockchain as blockchain is one of the most secure ways to store data.

2. BLOCKCHAIN

Blockchain is a decentralized, distributed electronic database shared across a public or private network. Every transaction in a blockchain database is shared among a number of users, each one verifying that the database is accurate and preventing unauthorized transactions from being completed.

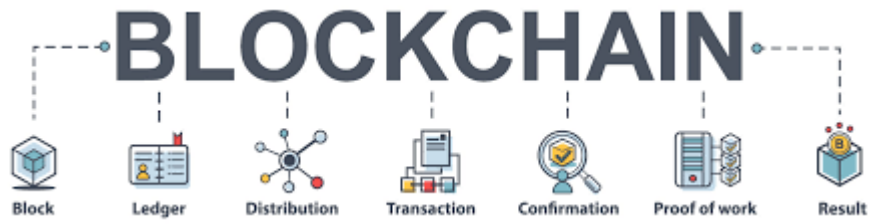
While the first version of blockchain was introduced by the Bitcoin protocol as a form of “peer to peer electronic cash,” the technology has implications far beyond financial transactions. In fact, if you have valuable data you need to protect, blockchain might be the key to guarding it against security threats and ensuring its integrity.

Because blockchain can streamline and cut out third-party middlemen, it can provide a faster and cheaper way to share critical and confidential business data or personal information. It also creates an indisputable digital trail of transactions, making it

possible for you to audit that trail so that you can know exactly what has been happening on your network.

In other words, blockchain ensures that you are basing business decisions on accurate, reliable data by allowing you to create a verifiable digital record of every financial transaction, process, task, contract and more.

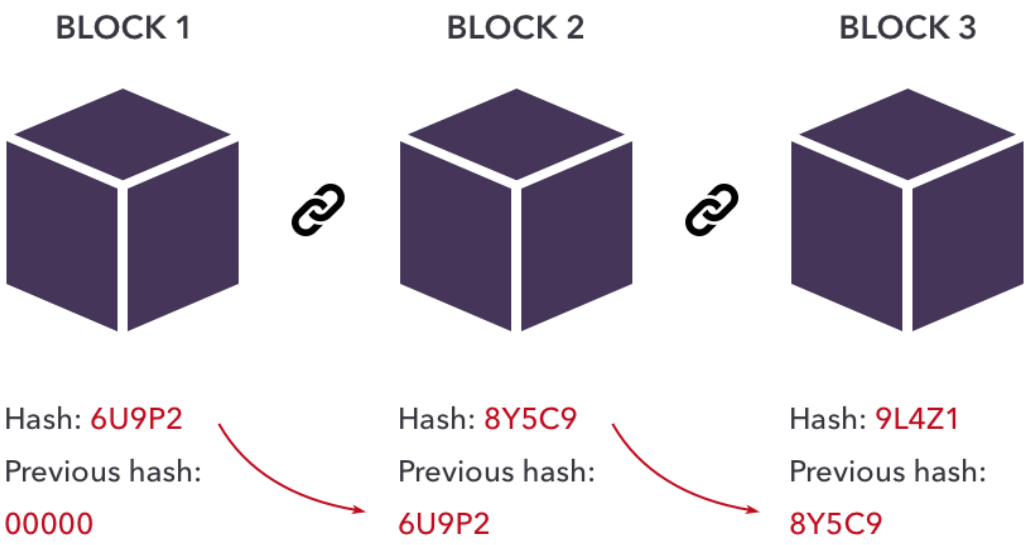
Yet blockchain is more than just a transactional database for critical data. Once data is committed onto a blockchain, it's permanent and nearly impossible to manipulate or hack. As such, businesses that adopt blockchain can operate more leanly and efficiently with greater trust in the security of their data. This is why organizations are turning to technology to solve a wide range of problems, including quality assurance, accounting, contract management, supply chain management, data protection and much more. Regardless of your industry or business type, blockchain has potential to help cut costs, improve customer service or boost overall efficiency.



2.1 Critical Characteristics of Blockchain Architecture

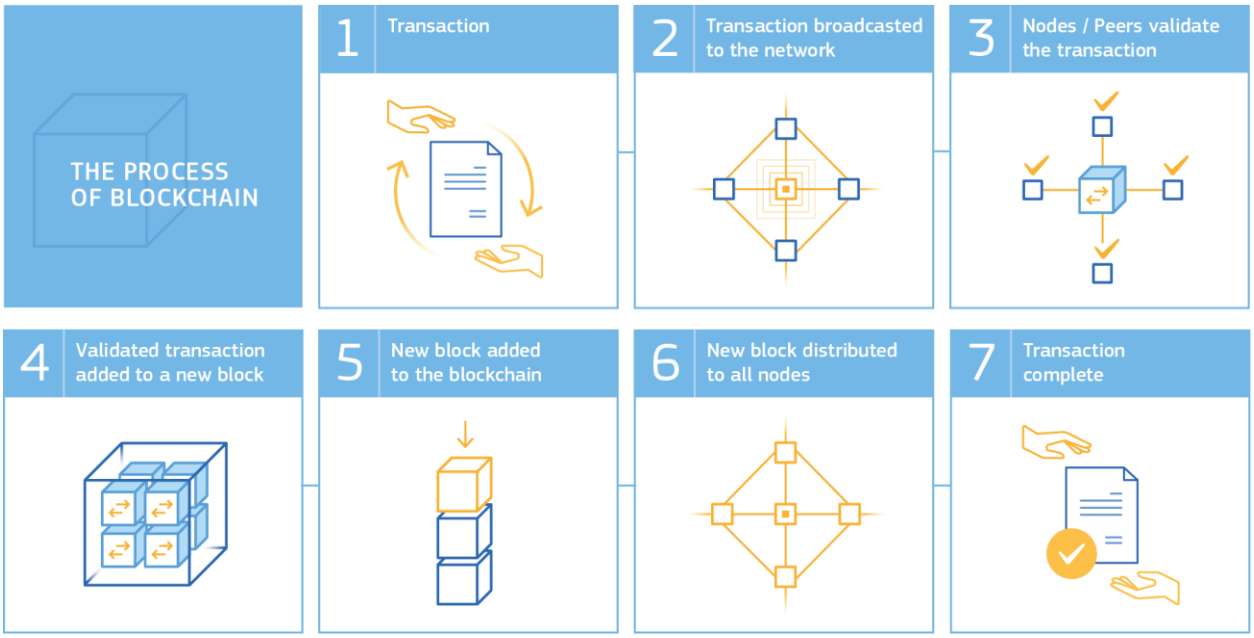
1. Cryptography: Blockchain transactions are authenticated and accurate because of computations and cryptographic evidence between the parties involved
2. Immutability: Any blockchain documents cannot be changed or deleted
3. Provenance: Every transaction can be tracked in the blockchain ledger
4. Decentralization: The entire distributed database may be accessible by all members of the blockchain network. A consensus algorithm allows control of the system, as shown in the core process
5. Anonymity: Maintains anonymity, especially in a blockchain public system
6. Transparency: It means being unable to manipulate the blockchain network. It does not happen as it takes immense computational resources to erase the blockchain network.

7. Consensus: A collection of commands and organizations to carry out blockchain processes.



2.2 Decentralized Ledger

2.3 Processes of blockchain



3. TOOLS & TECHNOLOGIES

3.1. Language

- HTML
- CSS
- Javascript
- Solidity

3.2. Libraries

- React: React.js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components. React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple
- Near-sdk: NEAR Protocol is a decentralized development platform where developers can host serverless applications and smart contracts that easily connect to "open finance" networks and benefit from an ecosystem of "open web" components. Near sdk is used to connect near blockchain to web app.

- Bcrypt - The bcrypt hashing function allows us to build a password security platform that scales with computation power and always hashes every password with a salt
- cookie-session - A user session can be stored in two main ways with cookies: on the server or on the client. This module stores the session data on the client within a cookie, while a module like express-session stores only a session identifier on the client within a cookie and stores the session data on the server, typically in a database.
- Cors - CORS is a node.js package for providing a Connect/Express middleware that can be used to enable CORS with various options. CORS stands for Cross-Origin Resource Sharing. It allows us to relax the security applied to an API. This is done by bypassing the Access-Control-Allow-Origin headers, which specify which origins can access the API.
- Express - Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications.
- passport-google-oauth20 - Passport strategy for authenticating with Google using the OAuth 2.0 API. This module lets you authenticate using Google in your Node.js applications. By plugging into Passport, Google authentication can be easily and unobtrusively integrated into any application or framework that supports Connect-style middleware, including Express.
- Jsonwebtoken: This library helps in generating a token and verifying it using a secret key. JSON Web Tokens (JWT) are an RFC 7519 open industry standard for representing claims between two parties
- Nodemailer: Nodemailer is a single module with zero dependencies for Node.js, designed for sending emails.
- Morgan: Morgan is a HTTP request logger middleware for Node.js. It simplifies the process of logging requests to your application.
- Nodemon: Nodemon is a utility that will monitor for any changes in your source and automatically restart your server. Perfect for development.
- Passport: Passport is authentication middleware for Node.js. Extremely flexible and modular, Passport can be unobtrusively dropped into any Express-based web application.
- Axios: Axios is a Javascript library used to make HTTP requests from node.js or XMLHttpRequests from the browser and it supports the Promise API that is native to JS ES6. It can be used to intercept HTTP requests and responses

and enables client-side protection against CSRF. It also has the ability to cancel requests.
- react-jwt: Small library for decoding json web tokens (JWT) for react

3.3. Frameworks

- react-bootstrap: This framework is used to design the user interface of the client side

4. 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.

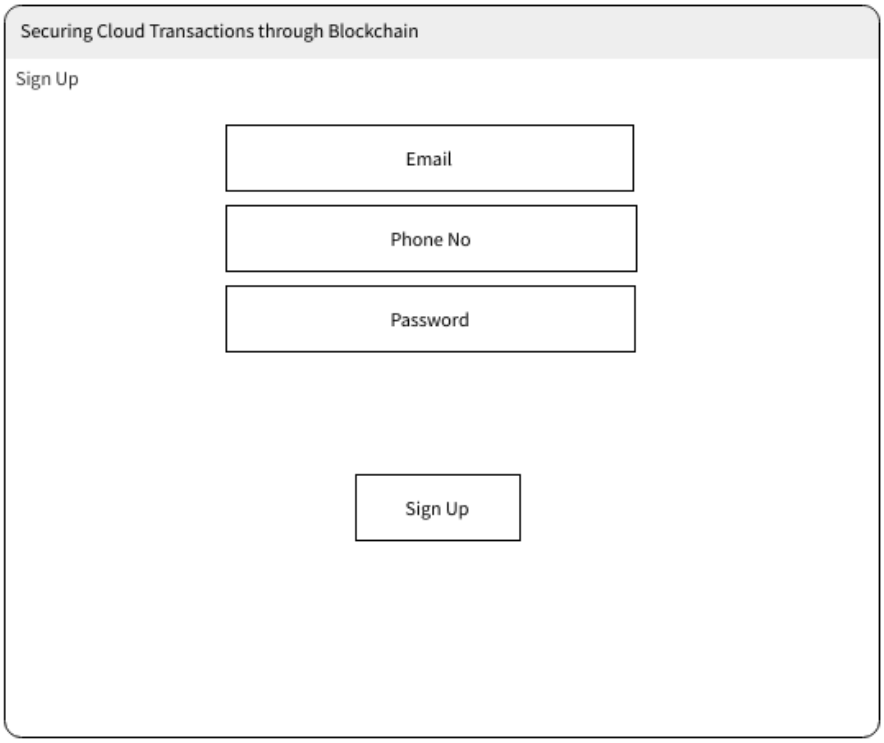
3. Distributed P2P Network - Each blockchain is copied in distributed P2P which prevents the loss of data and malicious change in the data of a block of the blockchain. Therefore, for the attacker to succeed he/she will have to alter data in all copies of the blockchain in its peers within a short time.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

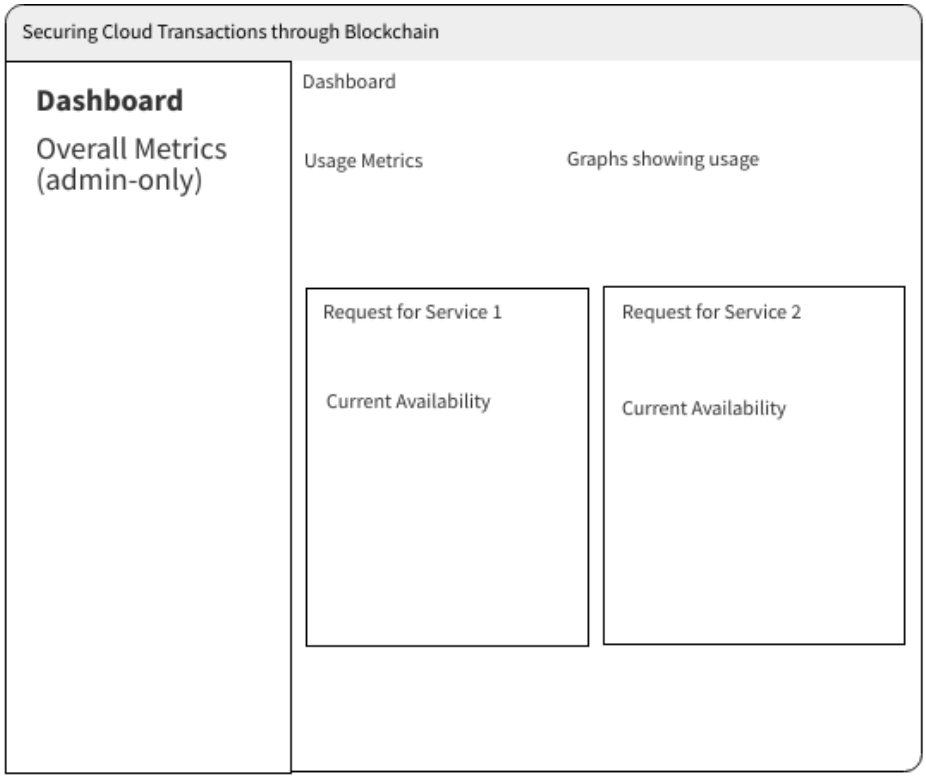
4. Byzantine Fault Tolerance – Blockchains are decentralized ledgers which, by definition, are not controlled by a central authority. Due to the value stored in these ledgers, bad actors have huge economic incentives to try and cause faults. That said, Byzantine Fault Tolerance, and thus a solution to the Byzantine Generals' Problem

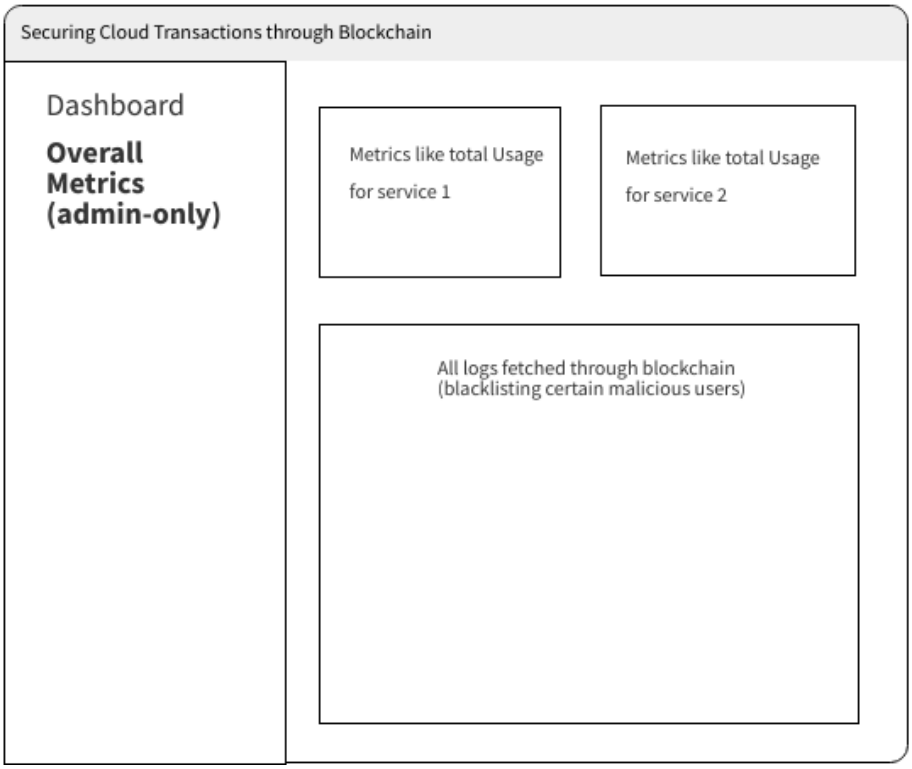
for blockchains is much needed [1].

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.

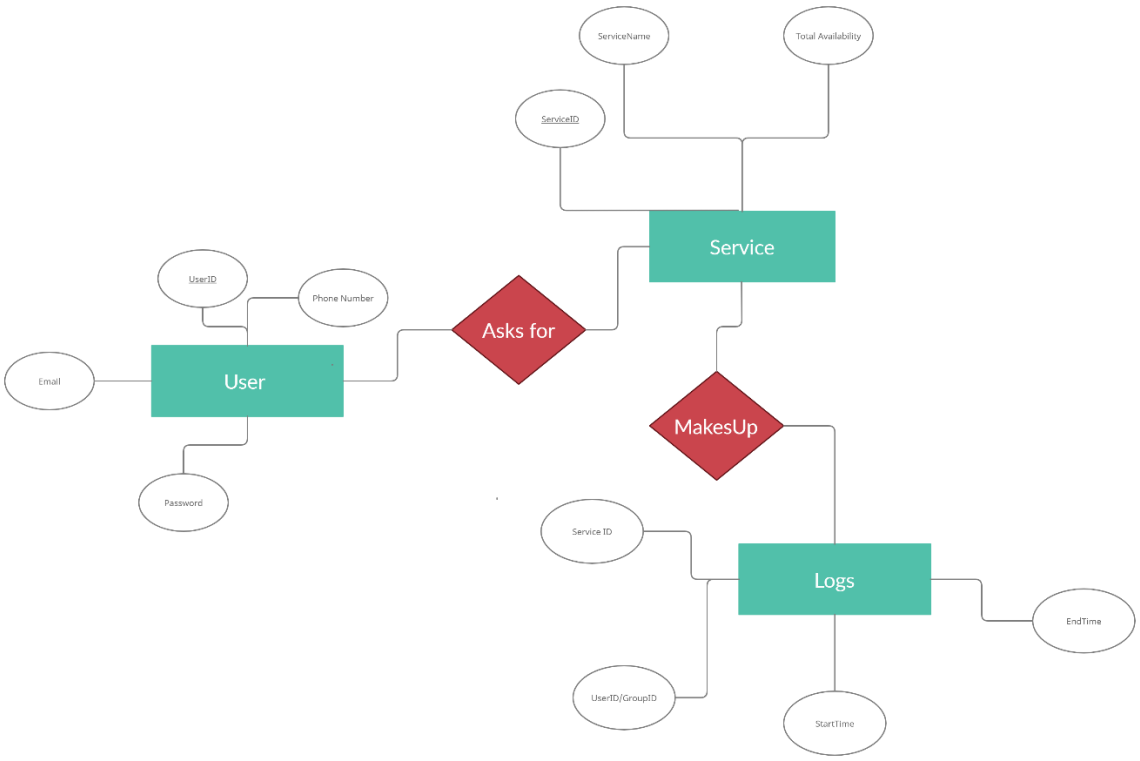
5. APPLICATION WIREFRAME




Securing Cloud Transactions through Blockchain
Dashboard
Overall Metrics (admin-only)
Dashboard
Usage Metrics
Graphs showing usage
Request for Service 1
Current Availability
Request for Service 2
Current Availability


Securing Cloud Transactions through Blockchain
Dashboard
Overall Metrics (admin-only)
Metrics like total Usage for service 1
Metrics like total Usage for service 2
All logs fetched through blockchain (blacklisting certain malicious users)

6. DATABASE SCHEMA



7. 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

[7] T. P. Abayomi-Zannu et al 2019 J. Phys.: Conf. Ser. 1378 032104