Chapter

A Simulation Model of a Blockchain-Based Decentralized Patient Information Exchange System for Parkinson's Disease Patients

Armando de Jesús Plasencia Salgueiro and Arlety García García

Abstract

Parkinson's disease (PD) is a progressive disorder of slow progress of the nervous system produced by the absence of levels of dopamine, which can incite unrestrained instinctive movements of the body and psychological affections. For the development of a practical, low-cost, and general diagnosis system of the symptoms to support PD patients, the implementation of an IoT health monitoring system that uses smartphones for data collection is necessary. However, data can be processed in Cloud Computing (CC) for analysis and comparison, but to reduce the latency of retrieving data from sensitive applications, Fog Computing (FC) plays a vital role. Nevertheless, these technologies IoT, CC, and FC have several limitations and are vulnerable to security threats. Blockchain technology enhances IoT challenges in a network in terms of security and availability. This chapter implemented a Decentralized IoT Fog-based Solutions and Blockchain using Ethereum Smart Contract for the authentication system. The smart contract is programmed using Solidity to allow Things to communicate with each other automatically without intermediaries and to store data in a public/private blockchain. The validation of the system was simulated them using the simulations tools Cisco Packet Tracer, iFogSim, and Remix Ethereum. The obtained results proved the feasibility of the proposed system.

Keywords: Parkinson's disease patients, blockchain, fog computing, decentralized IoT, simulation

1. Introduction

Neurological and mental ailments regularly present with side effects that are mind boggling, abnormal, fluctuant in illness evolution, and show high fluctuation between patients. Current diagnostic and efficacy evaluation methods frequently rely on in-clinic visits and patients', caregivers', or clinicians' subjective evaluations. Most of the time, in-clinic evaluation methods are expensive, take a long time, and only allow for a limited number and quality of observations. They also frequently exhibit high inter- and intra-rater variability. In the early stages of the disease, when there is a lag between the onset of the pathological process and the onset of symptoms, the diagnostic process may be affected by the aforementioned issues with traditional methods of diagnosis [1].

Neurological and psychiatric illnesses typically last a long time and cause significant changes in symptoms over time. As a result, the primary challenges in evaluating distinct diseases during periodic in-clinic visits are recall and reporting biases. Sensor-based smart technologies are rapidly developing remote monitoring of patients in their daily lives, which may assist clinicians in facilitating early diagnosis and evaluating and adjusting interventions. Use of recently developed smart sensor technologies for patient monitoring has become increasingly popular [1].

Parkinson's disease is a degenerative disorder of slow progress of the nervous system caused by the lack of levels of dopamine, which can provoke uncontrolled involuntary movements of the body and psychological affections [2].

An incremental number of sensors, such as motion (acceleration and gyroscope), location (the Global Positioning System, or GPS), environment (barometer, temperature, and light), and health (heart rate) sensors, are included in the modern smartphones and wearables. Smartphones have the potential to replace in-clinic evaluations for a variety of valuations due to their extensive array of sensors, ability to collect ecological momentary assessments (EMA), and information about social interaction (such as social media, messaging, and phone calls). Digital biomarkers (DBs) are terms used to describe the health-related data gathered during clinical trials. In order to gain a deeper comprehension of particular diseases, DBs can provide information that is useful, objective, and ecologically valid. Additionally, DBs make it possible to conduct frequent assessments of larger target populations over longer time periods, which may provide an in-depth understanding of the variation in daily life between and within individuals due to disease [1].

A few commitments empowering the utilization of cell phones as a valuation device have been as of late presented. Commercial devices make up the first set. By displaying notifications about a user's heart rate, number of steps taken, and type of activity, these apps primarily aim to provide feedback on the user's daily activities. However, the majority of these devices do not support high-frequency data collection and only provide limited access to the raw data. Applications and platforms created by investigators are the second category. The primary goals of these mostly open-source platforms are to make it possible to share and reproduce data as well as collect data for investigation applications. However, these software packages are much of the time restricted by a frequently constricted concentration to a few explicit clinical signs or concerning protection perspectives. Additionally, these periodically updated platforms render some unstable for the rapidly expanding smartphone ecosystem [1].

An example is a System developed by [3]. It builds on the fact that the medication treats Parkinson's disease gait abnormalities. It works by putting a smartphone in the pocket without requiring any special skills—a common occurrence in our daily lives. The information about a person's gait can be continuously detected by a smartphone without the user's active participation. The system makes passive sensing possible in this way. The system has two ends: a smartphone end and a cloud-server end. The smartphone sends the raw gait data to the cloud server from the smartphone end. After that, it is the job of the cloud server to look at the data and send the results to the smartphone. The smartphone notifies the user of the next medication time or reminds them to take their medication according to a drug schedule set by the healthcare provider. The system assists patients in avoiding missed, anticipatory, or additional doses through this approach. Although several platforms can collect context-driven data, the trade-off between privacy, optimization, stability, and research-grade data quality is not finding an optimal solution [1].

This paper objects to the solution of the abovementioned problems, to propose a decentralized authentication system utilizing blockchain technology and fog computing. With the characteristics and features of blockchain technology such as smart contracts, it addresses authentication using a decentralized database and communication between fog devices (nodes). The proposed system achieves authentication and communication without a central authority typical for this technology.

The main contributions of this paper can be summarized as follows:

1. Suggesting a secure decentralized user authentication that utilizes blockchain technology, smart contract, and secure ledger.

2. The system proposes to handle authentication requests using the username, password, Ethereum address, user email, and data from a biometric sensor.

3. The system is scalable and developed under the conception of scale to multiple IoT devices.

4. Proposing a methodology for the simulation of secure decentralized fog/edge architecture for healthcare systems under an IoT conception.

The rest of this paper is organized as follows: Section 2 introduces authentication systems, fog computing, and blockchain technology in Smart Healthcare Systems. Section 3 discusses the related works.

Section 4 presents the proposed authentication system. Section 5 provides the implementation details of the proposed system, followed by details of the experimental setup using simulation for validation of the proposed system.

Section 6 concludes the paper along with future directions.

2. Methods

2.1 Authentication systems

Authentication has become increasingly important in the world, with individuals, corporations, and businesses making use of it to control access to data and information [4]. In Smart Healthcare Systems, Authentication has great significance for your implication in patient safety and the growing introduction of digital monitoring patients systems in General Health Systems.

Other authentication mechanisms, such as biometric authentication (inherent factors) and token-based authentication (possession factors), are increasingly becoming popular and used transversely in services. Basic authentication methods, such as username and password, are widely used across platforms and services [4].

Alphanumeric and special character usernames and passwords are used in knowledge-based authentication systems. These are widely used across platforms and services and are regarded as conventional. Because they are simple to remember and can be processed quickly, they are popular. Due to their widespread use, password-based authentication systems are vulnerable to a variety of attacks, including dictionary, shoulder surfing, brute force, and dictionary attacks. Additionally, password length, character count, and password strength are limitations. Some weaknesses in text-based passwords have been fixed by introducing graphical passwords. This technique is resistant to attacks, for example, word reference attacks on account of its huge secret key space [4].

It has long been known that biometric authentication is superior to password-based authentication, thereby reducing some of these vulnerabilities. Typically, it is divided into physiological (fingerprints, DNA, face) and behavioral (voice, signature, etc.) characteristics. On smartphones, laptops, and other smart devices, face recognition and fingerprints have largely been used to control user access and authenticate on social media, banking apps, and services. Additionally, biometric authentication comes with drawbacks and dangers. This is large because biometric data cannot be changed and are managed in a centralized database or module. The reuse of stolen biometric data, among other security risks, may result from the discovery of these biometric data [4]. Although these methods are popular, they are not widely used as the traditional username and password.

2.2 Cloud, edge, and fog computing in smart healthcare systems

2.2.1 Cloud-based solutions

A network, cloud servers, and a mobile device make up cloud-only medical architectures. The issue of high latency is exacerbated by these components' potentially large distances between them (**Figure 1**). A comparison of distributed, or fog, cloud architectures, and traditional cloud architectures has recently been included in a number of medical monitoring solutions. For real-time emergency situations such as fall detection and stroke mitigation, which both require immediate medical response times, cloud-only solutions have retrieval times that are too high. Frequently sending

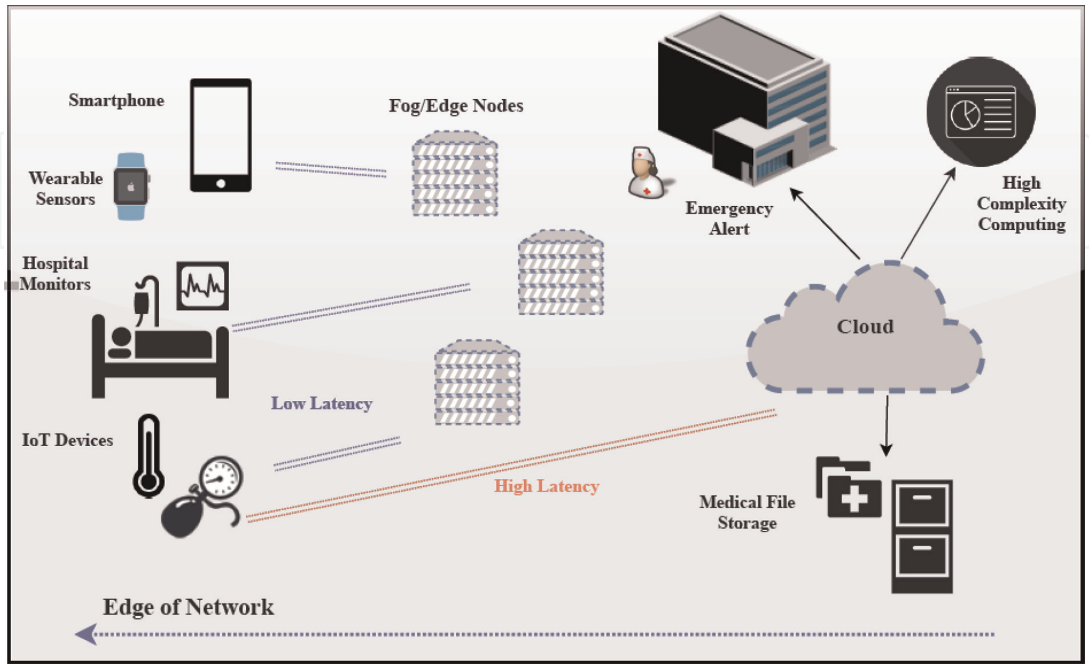


Figure 1.
General fog/edge architecture for healthcare systems [5].

data to the cloud for computation results in increased power consumption and costs, which is especially true in today's world when sensors generate a lot of data. When compared with a distributed computing architecture with multiple computing nodes in various locations, the typical cloud service demonstrated high latency and low sustained performance. Additionally, cloud-based solutions lack a low-cost mobile environment, which is essential for many patient monitoring scenarios [5].

2.2.2 Edge and fog-based solutions

Data processing is moved closer to the network edge with edge and fog-based solutions, resulting in faster response times and improved energy efficiency. Rather than continually moving information to the cloud for figuring activities, which represents the energy costs, information can be mined and handled by on edge devices and servers nearer to the client. In situations involving health monitoring, low latency of edge and fog solutions enables prompt arrival of emergency medical assistance. Privacy and security remain major issues due to a large amount of data that is typically sent to cloud services, particularly in situations where a patient's medical data could be hacked. Improved privacy can be achieved by disseminating data across a fog rather than concentrating it in a single area of the network. Device usability is also important because accurate data transmission depends on these sensors being easy enough for untrained personnel to use. The particular needs that are addressed are [5]:

- Cost
- Low Latency
- High-Level Security/Privacy
- Location Awareness
- Energy Efficiency
- Usability

The fog is typically referred to as a decentralized distributed computing system in which various entities own multiple fog devices and organizations can participate from a variety of locations, including hospitals, schools, airports, and smart hubs. According to the investigators, fog computing is a virtualized environment that is tasked with the delivery, storage, and computing of resources in cloud computing centers. It is not entirely outside the network space. These are a variety of fog nodes with limited computing and storage capabilities. Fog computing is widely regarded as an extension of the cloud that is close to devices that collect data for resource-constrained IoT. These devices are referred to as fog nodes and have storage, a network connection, and computational power. They are situated in various geographical locations that have network connectivity. These fog nodes are located close to devices that collect data [4].

The main characteristics of fog computing are given below [4]:

- Adaptability: These are made up of a lot of network sensors and other fog devices that handle computing tasks and provide storage resources.

- Real-time communications: Real-time communication between fog nodes and cloud-based data is provided by fog computing.

- Physical distribution: Fog computing offers services and applications hosted in multiple locations that are decentralized.

- Less latency: The proximity of fog computing to the edge devices helps position responsiveness to host fog devices in multiple locations and reduces the amount of time spent computing information with the edge devices.

- Compatibility: Fog modules are made to work with a lot of different platforms and service providers.

- Cloud integration and web analytics: The fog's location between the cloud and the edge devices is crucial for data processing and computing near them.

- Heterogeneity: Edge devices and fog nodes are made by different companies and have different features; therefore, they need to be hosted in accordance with their features.

2.3 Blockchain technology

Blockchain is a distributed ledger (register) technology that achieves immutability, traceability, anonymity, security, transparency, and decentralization through the use of consensus algorithms and cryptographic methods. All nodes on the blockchain verify and record confirmed transactions with a timestamp. The distribution of the ledger among the blockchain network's members provides transparency. Security of the information kept in the record is ensured and cannot be obstructed. One of the most important aspects of blockchain is the smart contract. It is a short piece of code that is part of the blockchain and can be executed automatically if the conditions are met. Blockchain is known to be of three types: These include consortium, public, and private blockchains. A private blockchain is more trusted by participants because it is managed by a single organization and governs its activities, such as who can participate in the blockchain. Any entity can participate in a public blockchain, which does not require permission. Privacy, security, and performance are among its drawbacks due to its highly decentralized nature. A permissioned blockchain created by a group of different businesses is known as a consortium blockchain. Nodes are the entities in this kind of blockchain. Which organizations or entities participate in the blockchain is controlled by the consortium [4].

The Internet of Things (IoT) has been the primary beneficiary of blockchain technology's features, such as the decentralized characteristics that enable devices to connect and share sensitive data securely in an IoT environment. These applications include auctions, mutual authentication, and blockchain technology. Blockchain is able to provide authentication systems with a dependable computational platform and secure storage thanks to these characteristics. A consensus algorithm helps the devices in a blockchain network establish trust. This makes it possible for devices to keep cryptographic keys (both private and public keys) and a digital signature. This makes it easier for devices in the network to communicate and transact. These transactions are tamper-proof on the blockchain network because they cannot be changed [4].

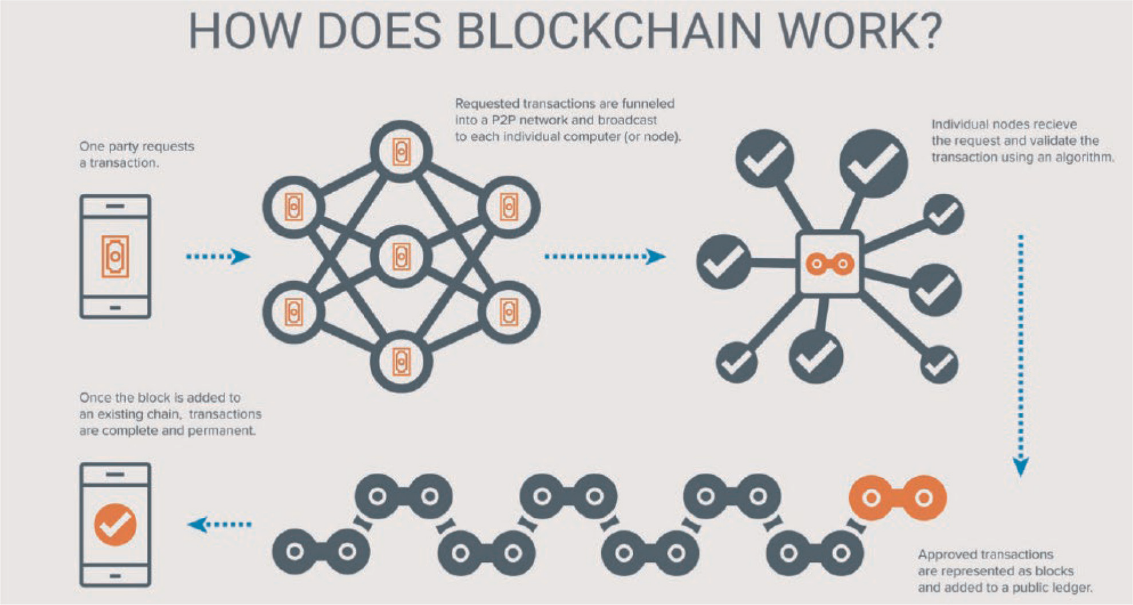

HOW DOES BLOCKCHAIN WORK?
One party requests a transaction.
Requested transactions are funneled into a P2P network and broadcast to each individual computer (or node).
Individual nodes recieve the request and validate the transaction using an algorithm.
Once the block is added to an existing chain, transactions are complete and permanent.
Approved transactions are represented as blocks and added to a public ledger.

Figure 2.
Functional diagram of Blockchain [6].

A blockchain is designed to resist data modification. It is "an open, distributed ledger that can efficiently and permanently record transactions between two parties." A blockchain is typically managed by a peer-to-peer network that collectively adheres to a protocol for inter-node communication and block validation in order to function as a distributed ledger. The data in any given block cannot be changed after it has been recorded without changing all subsequent blocks, which requires the majority of the network. Blockchains may be considered as secure by design and represent a distributed computing system with high Byzantine fault tolerance (any fault presenting different symptoms to different observers), despite the fact that their records can be changed. As a result, a blockchain has been advertised as providing decentralized consensus [6]. **Figure 2** represents the functional diagram of Blockchain.

3. Results

3.1 Related works

Fog computing and blockchain technology have been used in IoT, sales, cloud computing, and other systems, primarily in decentralized ways, in a number of works proposed for authentication. In order to comprehend the proposed scheme, a review of schemes including cloud computing, fog computing, blockchain technology, IoT, and authentication has been provided in this section.

3.2 JTrack

JTrack [1] was developed as an online server-side dashboard and an Android-based smartphone application. The main components of the JTrack application fall into the following categories: Human Activity Recognition (HAR), location data, sensor data, smartphone, and application usage monitoring, and both active (with user interaction) and passive (without user interaction) monitoring options are available for each component. The dashboard side is a web-based platform for study creation and management that incorporates DataLad infrastructures to make data management and sharing easier. Because JTrack is a modular open source with a high level of optimization, it is a practical solution for clinicians and researchers to collect, manage, and share digital biomarker data, particularly for Parkinson's disease patients.

The main components of the JTrack platform are shown in **Figure 3**.

In [1], the authors proposed a solution with the aim of QR-Code Authentication to provide a secure way of activation. Additionally, the JTrack platform was developed in accordance with Google Developer Policies and GDPR. At no stage is any sensitive information, such as a person's name, phone number, contacts, or actual location, recorded. Using the MD5 stability checksum, all of the collected data were transferred using the Hypertext Transfer Protocol Secure (HTTPS) protocol.

Regarding patient privacy, all JTrack users receive clear explanations of what was recorded and why. During installation and activation, permission requests for each module must be approved. All members may likewise pause and leave a review whenever straightforwardly from the application. Additionally, clinicians can maximize control over the collected data with remote configuration and one-step recording without having to collect any identifying information [1].

Along with Firebase integration for performance and crash reports, automatic restarting is implemented to reduce data loss caused by crashes or reboots. The optional recorded data, which are not active by default, include information about the phone's manufacturer, model, and operating system version. This information can be used to analyze and deal with cross-sensor variability [1].

3.3 Continuous patient monitoring with a patient-centric agent

Internet of Things (IoT) applications in the modern healthcare system include devices, services, and wireless sensors that detect physiological signs with wearable or ingestible sensors that stream data to remote and often Cloud-based servers. Secure

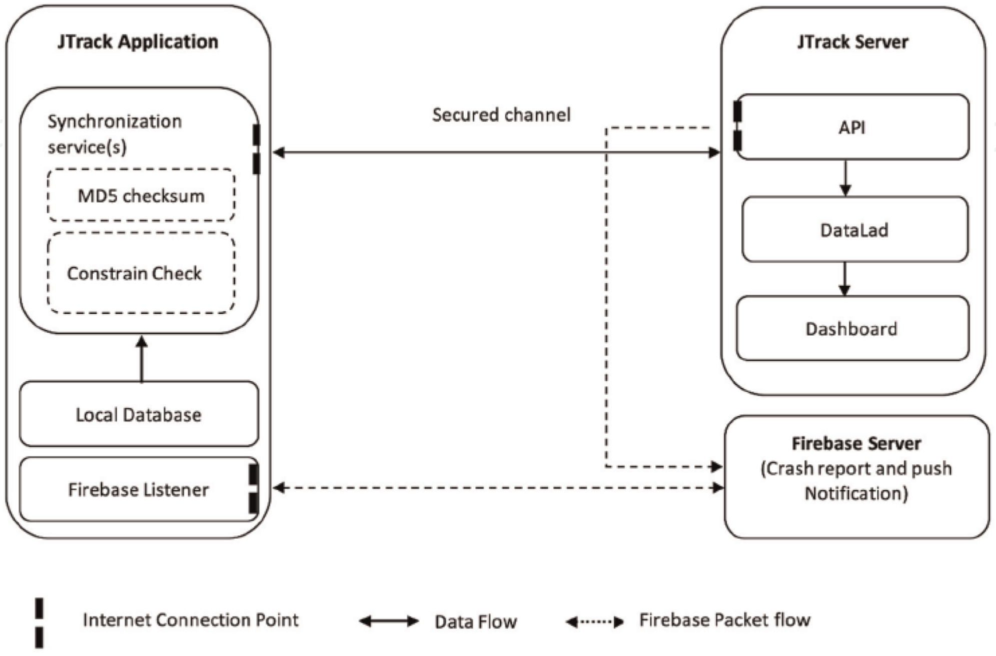


Figure 3.
JTrack platform overview [1].

continuous monitoring of patients' physiological signs has the potential to augment traditional medical practice [7].

The process of integrating demographic, health record, and geographic location data with physiological data obtained from wearable or implantable medical devices (IMDs) is known as remote patient monitoring (RPM) [7].

The aggregation and indexing of huge streams of continuous data while maintaining patient privacy are one of the challenges of designing remote patient monitoring systems that are effective, efficient, and secure. One's personal space, which also includes the ability to control data and set others' access levels, is referred to as privacy [7].

In addition to operational environments, insiders and outsiders pose threats to healthcare information's confidentiality, integrity, and accessibility. By gaining unauthorized access to confidential data, insiders such as healthcare professionals and support staff, service providers and outsiders such as hackers pose a threat to the security of health information [7].

Unauthorized actions have the potential to alter patient information and even result in death. Patient and medical professional trust in the system can be damaged by privacy breach [7].

Health information systems can also be threatened by resource misuses such as personal use of systems and software disruption caused by viruses, worms, and malware. Threats to the confidentiality and integrity of patient data include communication infiltration, interception, embedded malicious code, and repudiation. The security of health information can also be compromised by accident, technical infrastructure failure, and operational errors [7].

These threats have not yet been addressed by existing RPM architectures, as described in the following section. As a result, RPM software and devices require architectures that provide increased attack protection [7].

Efforts to ensure privacy in RPM have been made in recent years; however, most approaches focus on a single link in the architecture that chains data from patient sensors to healthcare professionals through intermediary devices and servers [7]

An effective and efficient RPM needs to address issues of rapid storage at appropriate security levels, user authentication, access control, mobility management, and sustainability of patient health data [7].

In order to ensure that appropriate levels of the trade-off between effectiveness and privacy can be established for rapid, secure data storage and access, user authentication, role-based access, and sustainability, an advanced End-to-End eHealthcare architecture that addresses RPM healthcare data management issues has been developed. A Patient-Centric Agent for End-to-End data stream coordination and a Blockchain component for distributed data storage are key architectural features [7].

In [7], it was suggested that the inclusion of a Patient-Centric Agent (PCA) can decrease RPM challenges. The End-to-End data flow is inaccurate for the Patient-Centric Agent (PCA). The level of storage, security, and access required at any given time is determined by the PCA. The patient sensors and devices, Blockchain nodes, and healthcare service provider devices are all coordinated by the PCA. If a stream of data should be stored in a Blockchain, the PCA manages the process and determines whether it should. The PCA executes on a machine with mass memory limit and high handling power.

There are two levels to the proposed architecture. The data streaming and storage solution is provided by the lower tier, while the Healthcare Control Unit (HCU) manages the primary healthcare provider. **Figure 4** depicts six systems that make up

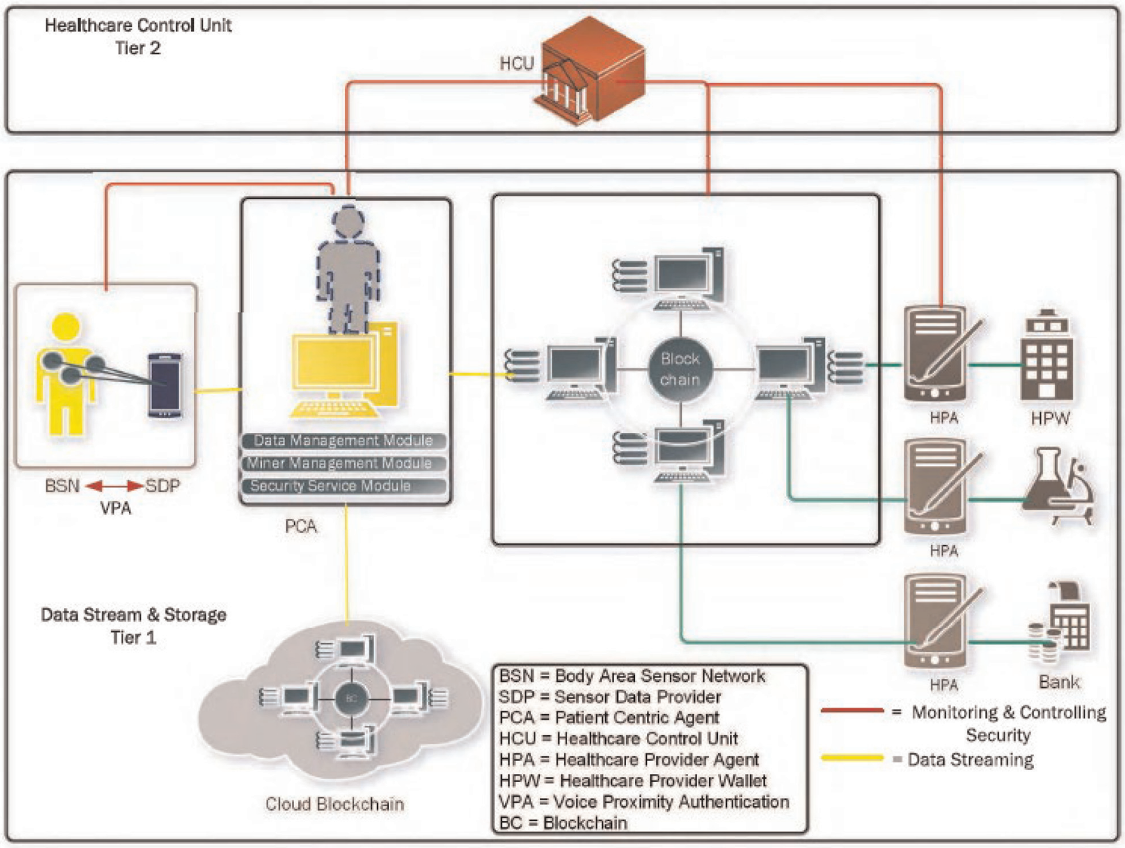

Healthcare Control Unit
Tier 2
HCU
Block chain
HPA
HPW
HPA
HPA
Bank
Data Management Module
Miner Management Module
Security Service Module
PCA
BSN
SDP
VPA
Data Stream & Storage
Tier 1
BC
Cloud Blockchain
BSN = Body Area Sensor Network
SDP = Sensor Data Provider
PCA = Patient Centric Agent
HCU = Healthcare Control Unit
HPA = Healthcare Provider Agent
HPW = Healthcare Provider Wallet
VPA = Voice Proximity Authentication
BC = Blockchain
= Monitoring & Controlling Security
= Data Streaming

Figure 4.
The tier-based remote patient monitoring architecture [7].

the lower tier. Patient Centric Agent (PCA), Blockchain, Healthcare Provider Agent (HPA), and Healthcare Provider's Wallet (HPW) are all examples of Body Area Sensor Network (BSN). In **Figure 4**, a Sensor Data Provider, such as a smartphone, connects BSN to the Patient-Centric Agent (PCA) [7].

The Healthcare Control Unit, the Cloud, and the Blockchain network are all connected to PCA. Medical services Supplier Specialist interfaces Blockchain, Medical care Control Unit, and Medical services Wallet at the medical services supplier end. The functional view of the architecture is shown in **Figure 5**, and the architecture is explained by the communication links that connect the various segments below. The architecture is built to handle a lot of patients at once [7].

3.4 Fog-enabled blockchain-based authentication system

3.4.1 System architecture

In [4], the blockchain-based components of the decentralized authentication system are described. **Figure 6** illustrates the system's architecture. The components of the proposed system are outlined below:

- Ethereum Smart Contract:

  This authentication system's contract is used to handle user registration and authentication. The agreement would require data such as the email, password,

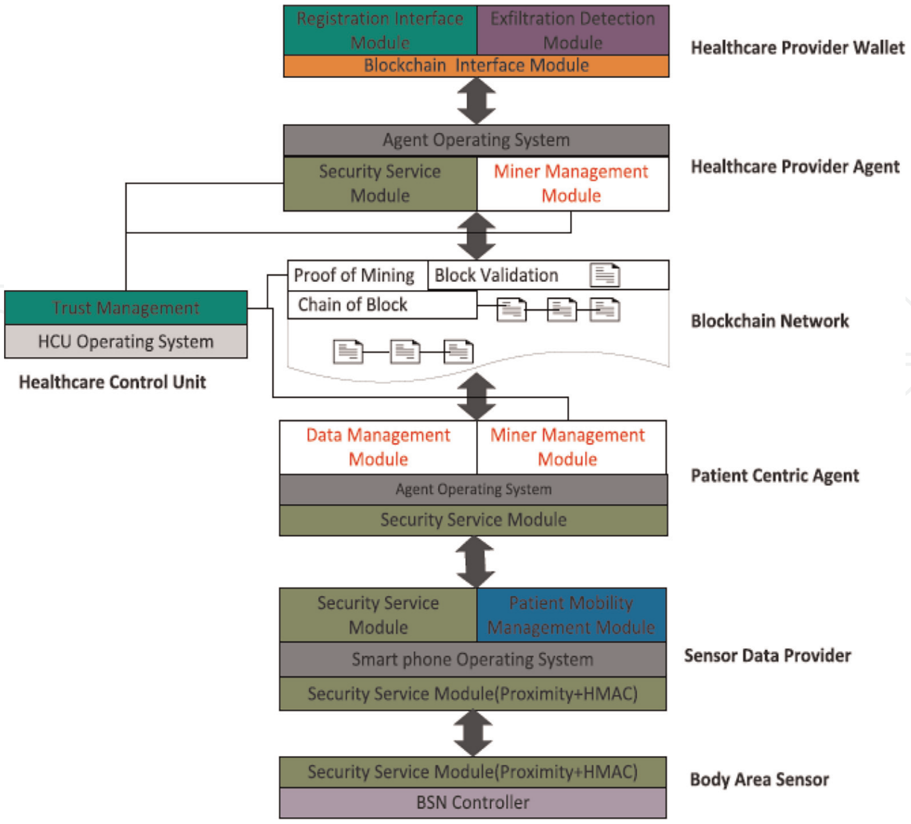


Figure 5.
Conceptual view of the tier-based health monitoring architecture [7].

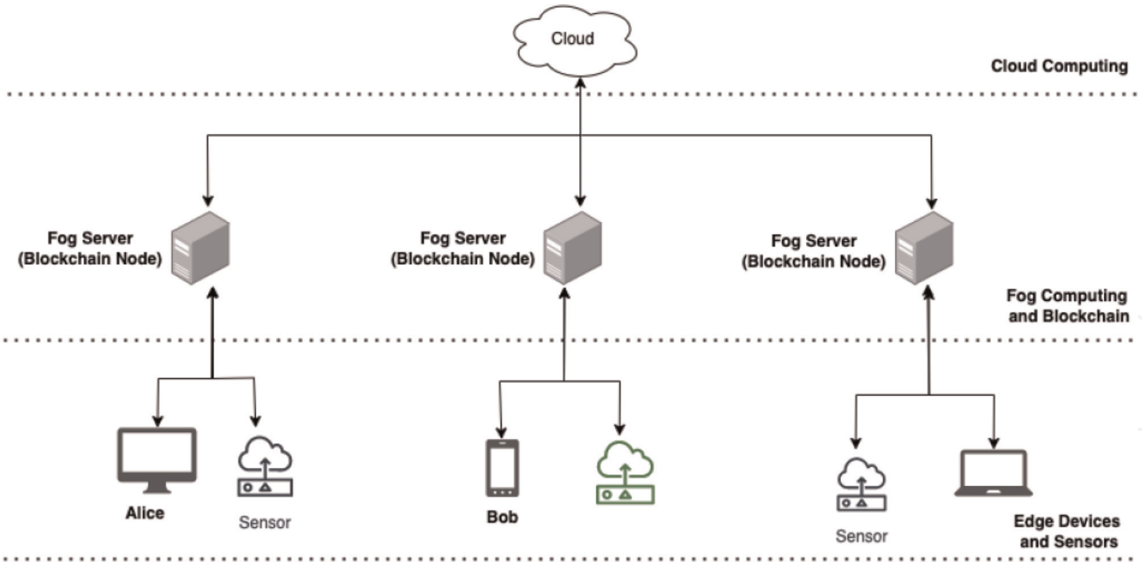

Cloud
Cloud Computing
Fog Server (Blockchain Node)
Fog Server (Blockchain Node)
Fog Server (Blockchain Node)
Fog Computing and Blockchain
Alice
Sensor
Bob
Sensor
Edge Devices and Sensors

Figure 6.
The fog-enabled blockchain-based authentication system's system architecture [4].

and the UserEthAdr to enlist clients upon enrollment and to validate clients in the ensuing collaboration with the framework [4].

- Fog Node:

Devices that serve as blockchain nodes and servers are known as fog nodes. Every node has a duplicate of the BlockC, LDG, and SmContract. When a User registration or authentication transaction takes place on a node, the BlockC information there is updated. To host or be a part of the BlockCN, the fog device or fog server must meet sufficient requirements [4].

- Edge Devices:

  During registration and authentication, the user's end devices are mapped to nodes. The BlockC cannot be hosted on these devices due to a lack of resources [4].

- Cloud:

  IoT data are stored, hosted, and computed in the cloud, which is a large storage unit. Data generated by IoT or edge devices must be processed and analyzed by this cloud server [4].

3.5 Automated decentralized IoT-based blockchain using ethereum smart contract for healthcare

In Ethereum, smart contracts are used to automate participant interactions and the execution of data from Things or any other type of data. Nodes use it to test, debug, verify, and test transactions. Consensus algorithms are at the heart of the blockchain, ensuring its security and integrity [8].

The proposed blockchain of Things architecture in the work [8] consists of five layers and is based on Ethereum smart contract including Things, gateway, Fog, Cloud, and Application layer (see **Figure 7**), demonstrated as follows:

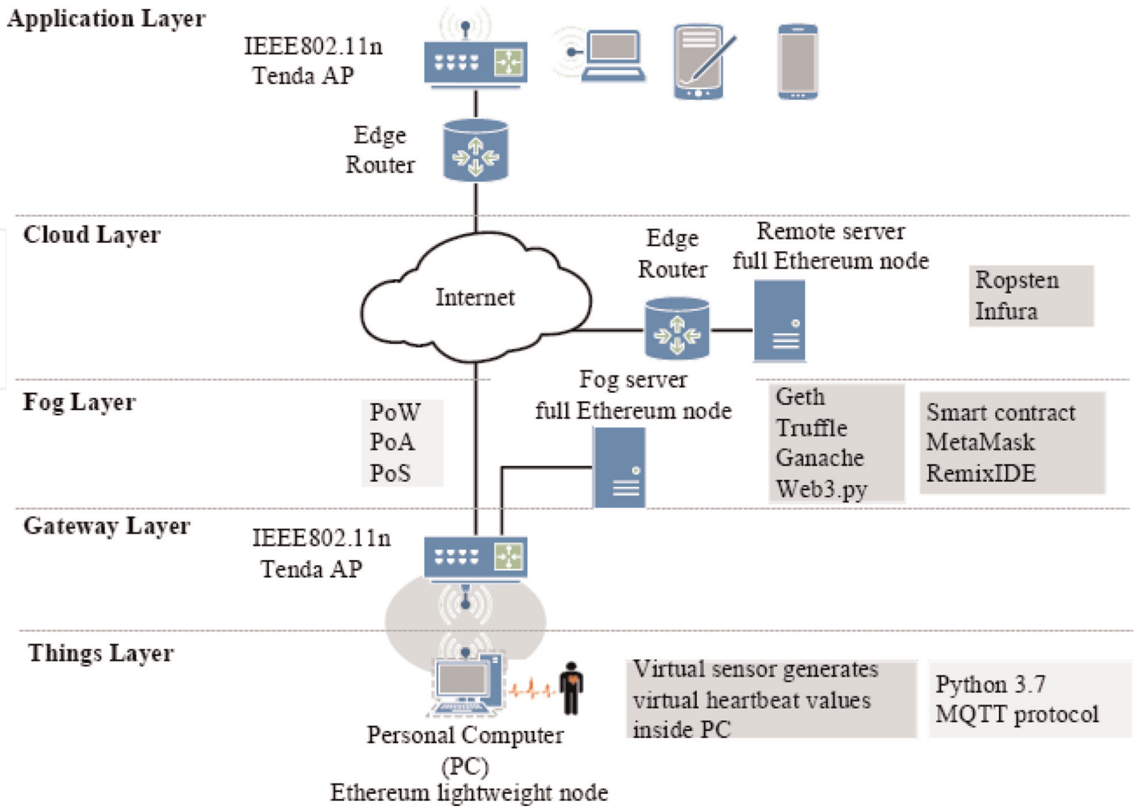


Figure 7.
Proposed IoT-based blockchain architecture for healthcare network [8].

Things layer: it consists of a virtual sensor programmed in python version 3.7 programming language [8].

Gateway layer: To transfer data to higher layers, such as the cloud and fog layers, the virtual sensor is connected to an Access Point (AP) as a gateway device with WiFi connectivity (IEEE802.11n Tenda AP) [8].

Fog layer: It stores IoT data on a private blockchain by collecting it from the Things layer's virtual sensor. This layer is a 24-hour power-on with enough storage space to store a local copy of the entire blockchain [8].

Cloud layer: This layer is built on top of a public testnet blockchain that was made for testing and mining. It uses Ether, which has no value, which can be bought from faucets that look like the mainnet network [8].

Application layer: Things' data can be monitored by doctors, patients, and their families. However, because blockchain data are immutable and unchangeable, they are unable to alter or delete them. Additionally, physicians can quickly respond to an emergency by monitoring data in real time even if it is not processed in blockchain [8].

4. Proposed system

The proposed decentralized authentication system, which is required for user authentication, is described in detail in this section. Some of the assumptions that are taken into consideration when developing the proposed methodology are outlined before the proposed system is presented in detail. The blockchain-based authentication systems described above frequently employ these conventions, such as [4]:

- There are mobile and immobile devices connected via multiple networks in the fog computing environment.
- The blockchain technology is accessible to registered user devices.
- To be able to serve as a node or a server and host the blockchain, the fog device must meet certain requirements.
- The registration and authentication of users ought to be carried out by the smart contracts.
- Nodes should not have to rely on other nodes to do their jobs.

Conceived of a smart contract, ledger, and Ethereum blockchain-based decentralized authentication system. Fog nodes will function as blockchain nodes in this system and host a decentralized digital ledger in which each fog node possesses a copy of the smart contract.

5. Experimental setup

An evaluation based on several experiments is presented to validate the proposed system. The proposed authentication system is implemented through some simulators such as Cisco Packet Tracer [9], iFogSim [10], and Ethereum smart contracts utilizing

Solidity [11]. This smart contract is tested, and the simulation is run through Remix Ethereum. This IDE offers various features such as the creation, deployment, testing, and debugging of smart contracts. The network layout of the system and simulations are executed in Ethereum Remix IDE and Cisco Packet Tracer.

Data from the metrics are collected such as the time needed to send packets from user devices to the fog nodes using Cisco Packet Tracer, the energy consumed, the cost of execution in the cloud, the total time required for module migration using iFogSim, and the transaction cost, execution cost, and miner fees are recorded for both registration and authentication requests through the blockchain network.

Simulations are run through Cisco Packet Tracer to replicate the fog network with nodes and determine the time needed to send packets from user devices to the fog nodes.

5.1 Cisco packet tracer simulation

The network is replicated in Cisco Packet Tracer [9]. Fog servers (nodes) and user devices are used to run simulations and tests on a variety of packets and protocols (HTTPS, SSH, SMTP, ICMP), and multiple requests are made on both wired and wireless networks to find out how long it takes to send authentication packets on the network. For the purpose of this experiment, these packets were chosen because they are frequently used for secure communications over computer networks [4].

Figure 8 represents the developed configuration of the proposed network scheme in Cisco Packet Tracer. The development was taken as a reference to the configuration given by [12], also the conceptions developed in [13]. **Table 1** shows the relation of components of the proposed network.

The results for simulation showed that all the packets are delivered in 40.049 s in a wired and wireless network, considering delivering SMTP packets, SSH packets, and HTTPS and ICMP. The time required to authenticate various packets using the proposed method is shown in the obtained results. This demonstration can quickly and effectively handle authentication requests, scale to multiple devices in a fog

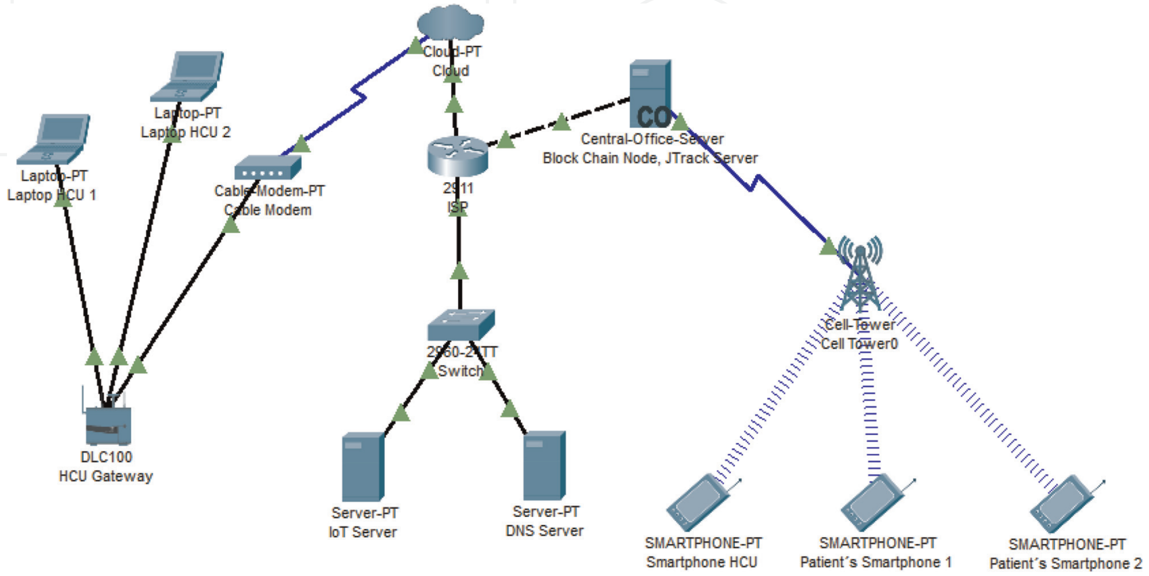


Figure 8.
Configuration of proposed network scheme in packet tracer.

No	Devices	Function
1	Router (2911)/ISP	Used to connect a cellular network to HCU
2	Cable modem	Used to HCU gateway to cloud
3	HCU gateway	Used for smart devices registration
4	IoT server	Used to control smart devices registered on it
5	DNS server	Used to access smart devices by the domain name
6	Central office server (Fog Server)	Used to connect a cell tower to a router and vice versa (JTrack Server). Block Chain Node
7	Cell tower	Used to connect the smartphone to the internet
8	Smartphone	Used to monitor Parkinson's Disease Patients
9	Laptop	Connect to the HCU gateway to access the Patient's smartphones

Table 1.
Devices used for the simulation.

computing environment, and is suitable for use on edge devices. The used version of Cisco Packet Tracer was 8.1.1.

5.2 Simulation results in iFogsim

iFogSim simulation toolkit is developed upon the fundamental framework of CloudSim. CloudSim is one of the wildly adopted simulators to model Cloud computing environments. A customized Fog computing environment with a large number of Fog nodes and Internet of Things (IoT) devices (such as sensors and actuators) can be simulated with the help of iFogSim. However, the classes in iFogSim are annotated in such a way that users without prior knowledge of CloudSim can quickly and easily define the policies for Fog computing's infrastructure, service placement, and resource allocation. When simulating any application scenario in the Fog computing environment, iFogSim employs the Sense-Process-Actuate and distributed dataflow models. It makes it easier to evaluate end-to-end latency, network congestion, power consumption, operational costs, and customer satisfaction with QoS [14].

iFogSim2 is a simulator [15], which is an extension of the iFogsim simulator and addresses distributed cluster formation among Edge/Fog nodes of various hierarchical levels, microservice orchestration, and service migration for various mobility models of IoT devices.

The new iFogSim2 simulator components are loosely coupled to support various simulation scenarios. As a result, the components (Mobility, Clustering, and Microservices) can be used solely for simulation or integrated for more complex scenarios [15].

In the Healthcare solution of Parkinson's Disease, body-connected IoT devices perceive the health context of the users through a Client application module. Generally, the IoT devices are usually connected to smartphones, but in this particular case, the sensors are inside the same smartphone.

For the corresponding application, the smartphones serve as the Application gateway node. These nodes prepare the sensed data from IoT devices. The application's data analysis and event management operations are carried out in upper-level Fog computational nodes unless the Application gateway node's resource availability meets the requirements. Application gateway nodes select appropriate computational nodes in the second scenario to deploy additional application modules and initiate actuators based on the results of those modules [14].

In the IoT-enabled Parkinson's Disease Patient monitoring healthcare solution using iFogSim was simulated the necessary Fog environment. The application model for the IoT-enabled healthcare solutions is represented in **Figure 9**.

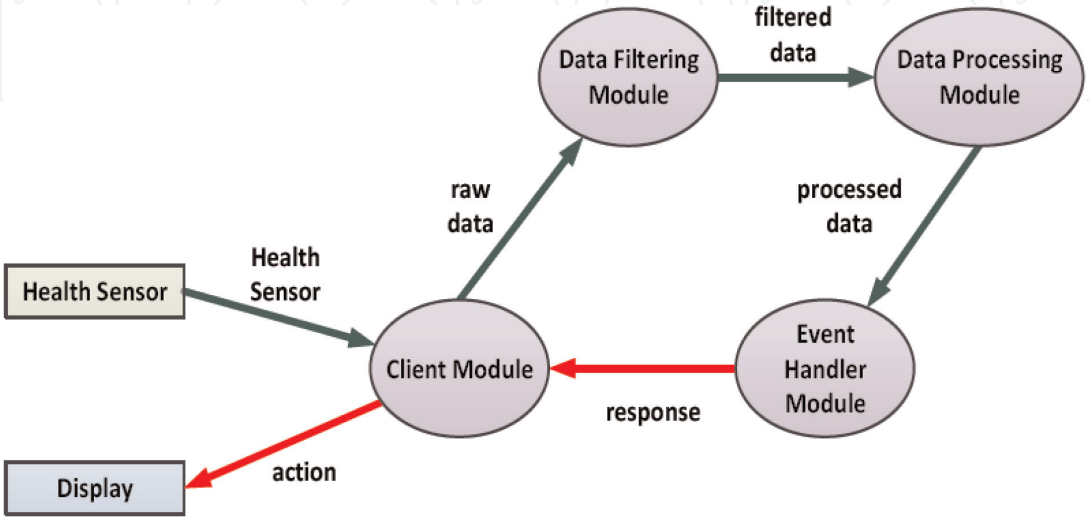
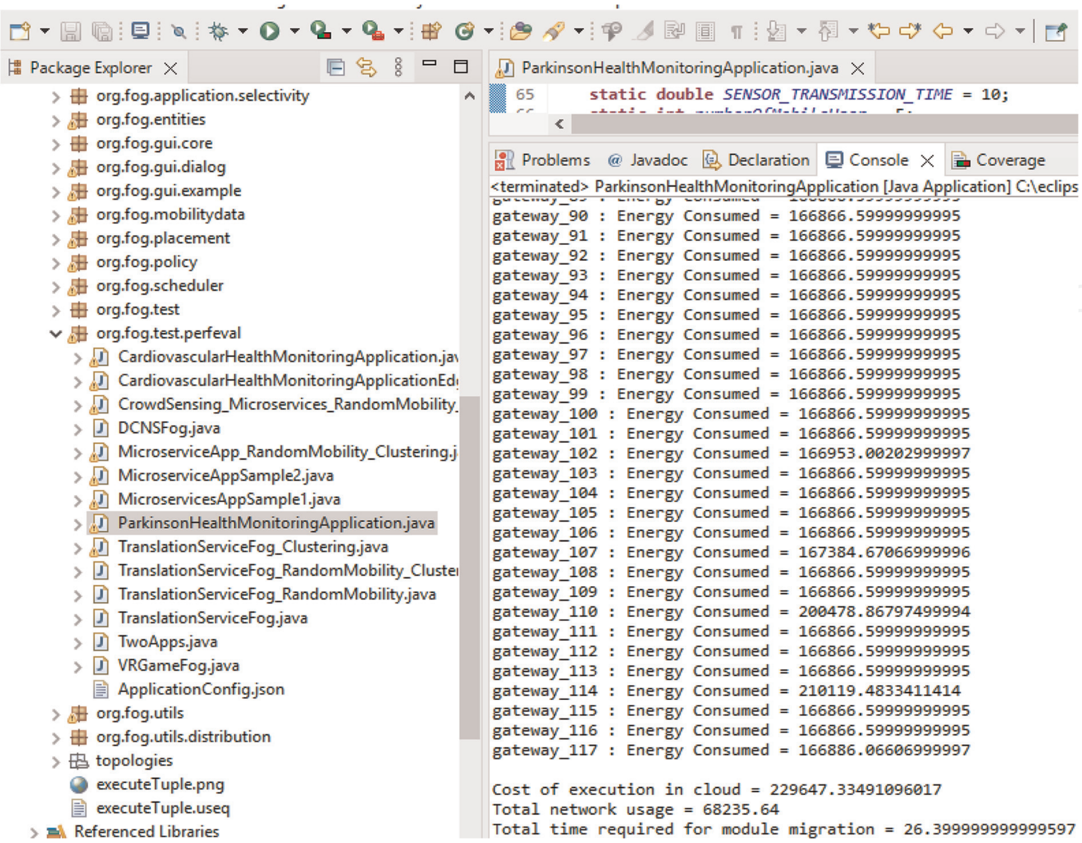


Figure 9.
Application model for IoT-enabled healthcare case study [14].

The system and the application along with the required guidelines to model them in iFogSim are an n-tire hierarchical Fog environment. As the rank of Fog levels goes higher, the number of Fog devices residing at that level gets lower. Fog devices form Clusters among themselves and can be mobile. IoT devices (smartphones) are connected to lower-level Fog devices [14]. The sensing frequencies of IoT devices are different. The application model consists of four modules with a sequential unidirectional data flow. The requirements of the application modules are different, and each application module can request additional resources from the host Fog devices to process data within the QoS-defined deadline. The results of the simulation are shown in **Figure 10**.

5.3 Simulation smart contract through remix Ethereum

Ethereum is a decentralized, open-source blockchain with smart contract functionality. The Ethereum network is a well-established blockchain network that allows people to actively develop new and innovative applications and products that directly tie to Ethereum and use its native cryptocurrency, ether. This allows people to use a blockchain network that already has nodes and avoid the need to set up self-hosted nodes for testing. However, some limitations include long transaction times and high gas prices [16]. For the simulation of blockchain in IoT communications, Ethereum was the single simulator. This meant that Ethereum was used in hashing and encoding. In smart devices, Ethereum was used as the base network for storing data via smart contracts and generating hashes [16]. For all these properties, Ethereum was selected for the simulation.



```
org.fog.application.selectivity
org.fog.entities
org.fog.gui.core
org.fog.gui.dialog
org.fog.gui.example
org.fog.mobilitydata
org.fog.placement
org.fog.policy
org.fog.scheduler
org.fog.test
org.fog.test.perfeval
  CardiovascularHealthMonitoringApplication.ja
  CardiovascularHealthMonitoringApplicationEd
  CrowdSensing_Microservices_RandomMobility_
  DCNSFog.java
  MicroserviceApp_RandomMobility_Clustering.j
  MicroserviceAppSample2.java
  MicroservicesAppSample1.java
  ParkinsonHealthMonitoringApplication.java
  TranslationServiceFog_Clustering.java
  TranslationServiceFog_RandomMobility_Cluste
  TranslationServiceFog_RandomMobility.java
  TranslationServiceFog.java
  TwoApps.java
  VRGameFog.java
  ApplicationConfig.json
org.fog.utils
org.fog.utils.distribution
topologies
executeTuple.png
executeTuple.useq
Referenced Libraries
```

```
65      static double SENSOR_TRANSMISSION_TIME = 10;
```

```
<terminated> ParkinsonHealthMonitoringApplication [Java Application] C:\eclips
gateway_90 : Energy Consumed = 166866.59999999995
gateway_91 : Energy Consumed = 166866.59999999995
gateway_92 : Energy Consumed = 166866.59999999995
gateway_93 : Energy Consumed = 166866.59999999995
gateway_94 : Energy Consumed = 166866.59999999995
gateway_95 : Energy Consumed = 166866.59999999995
gateway_96 : Energy Consumed = 166866.59999999995
gateway_97 : Energy Consumed = 166866.59999999995
gateway_98 : Energy Consumed = 166866.59999999995
gateway_99 : Energy Consumed = 166866.59999999995
gateway_100 : Energy Consumed = 166866.59999999995
gateway_101 : Energy Consumed = 166866.59999999995
gateway_102 : Energy Consumed = 166953.00202999997
gateway_103 : Energy Consumed = 166866.59999999995
gateway_104 : Energy Consumed = 166866.59999999995
gateway_105 : Energy Consumed = 166866.59999999995
gateway_106 : Energy Consumed = 166866.59999999995
gateway_107 : Energy Consumed = 167384.67066999996
gateway_108 : Energy Consumed = 166866.59999999995
gateway_109 : Energy Consumed = 166866.59999999995
gateway_110 : Energy Consumed = 200478.86797499994
gateway_111 : Energy Consumed = 166866.59999999995
gateway_112 : Energy Consumed = 166866.59999999995
gateway_113 : Energy Consumed = 166866.59999999995
gateway_114 : Energy Consumed = 210119.4833411414
gateway_115 : Energy Consumed = 166866.59999999995
gateway_116 : Energy Consumed = 166866.59999999995
gateway_117 : Energy Consumed = 166886.06606999997

Cost of execution in cloud = 229647.33491096017
Total network usage = 68235.64
Total time required for module migration = 26.399999999999597
```

Figure 10.
Results of simulation IoT-enabled Parkinson's Disease Patients monitoring healthcare solution using iFogSim.

Figure 11 depicts the Ethereum-enabled fog nodes of the proposed system architecture. Five main participants with Internet access to Ethereum smart contracts make up the architecture: fog nodes, users, administrators, and cloud servers that store IoT data. Although IoT devices have a unique Ethereum address and public and private

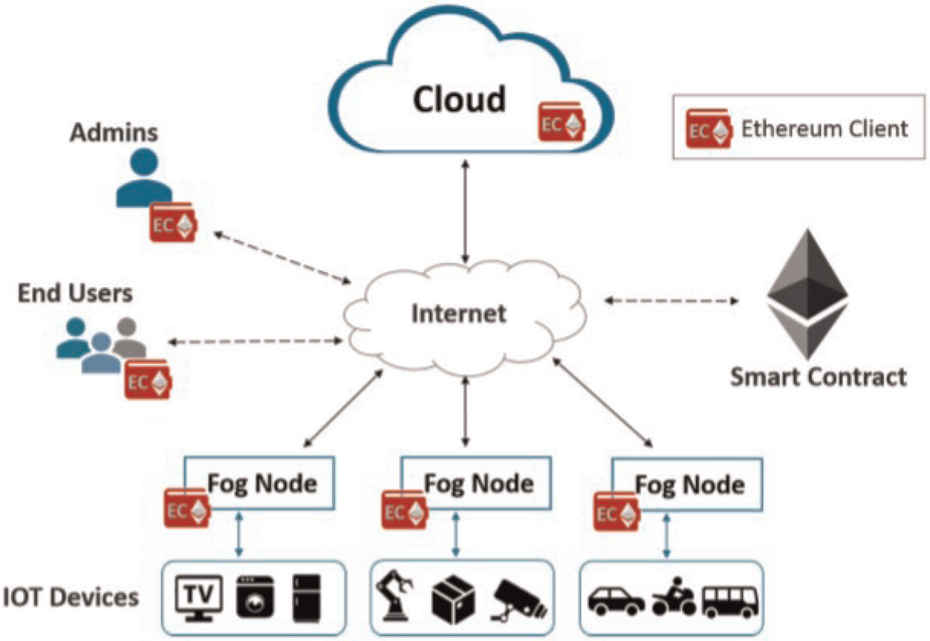


Figure 11.
System architecture with fog nodes that can use Ethereum [17].

keys, they do not interface with the smart contract or have connectivity. In the case of fog and cloud nodes, each participant has a unique Ethereum Address (EA) and interacts directly with the smart contract via an Ethereum client or a front-end application or wallet for administrators and end users [17].

Ethereum Remix IDE is chosen for this experiment for its features, which allows for the development, administering, and deployment of smart contracts in a virtual blockchain environment.

The simulation model is displayed in **Figure 12**. In step 1, User A presents data and sends a registration request to the blockchain-enabled fog node via the edge device. User A is registered as a new user in step 3, while user data are stored on a distributed ledger by the blockchain-enabled fog node in step 2. User B presents data and sends an authentication request to the blockchain-enabled fog node through the edge device in step (4). At the following stage (5), the blockchain-empowered fog node affirms that Client B's information is legitimate and exists in a distributed ledger. The user is then verified or denied. The exchange cost, execution cost, and miner fees are recorded for both registration and authentication demands through the blockchain network [4].

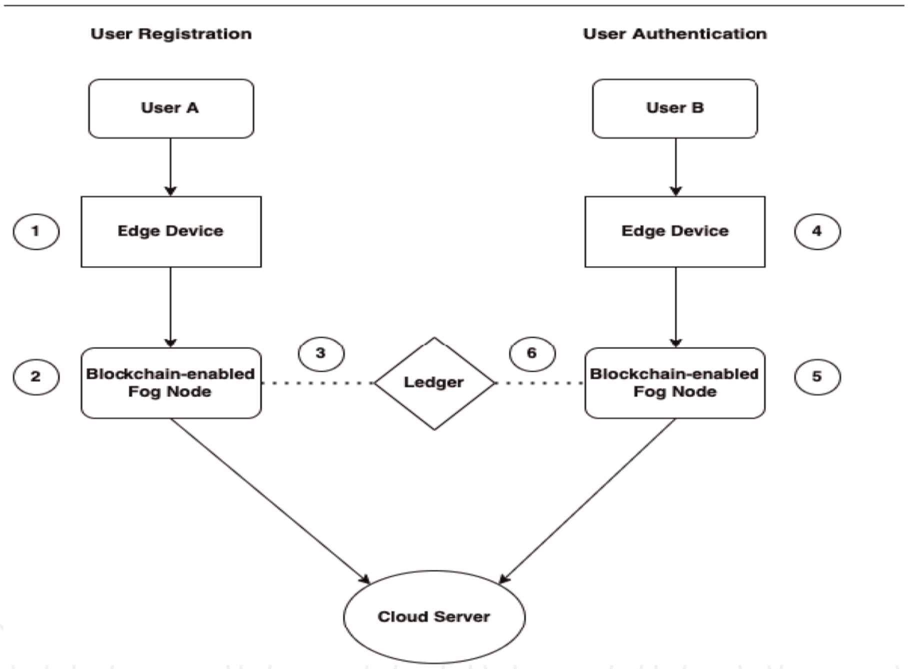


Figure 12.
The simulation model [4].

Figure 13 shows the sequence diagram of exchange messages showing a successful authentication of the end user to the IoT Device.

In order to carry out the authentication scheme, the system entities interact in two primary ways; namely: interactions between the on- and off-chains, as shown in **Figure 13**. A sequence diagram for a secure session connection between an end user and an IoT device following successful authentication is depicted in **Figure 13**. The admin first creates the smart contract, registers the IoT devices, and maps them to a fog node using the *addDeviceFogMapping* function in the on-chain activity. The Ethereum Address (EA) is unique to each fog node and IoT device. Using the *addUserDeviceMapping* function, the administrator can also grant access permissions to specific IoT devices to end users. On Github, the entire smart contract logic, lists, rules, and code for the authentication registry are also made available to the public [18].

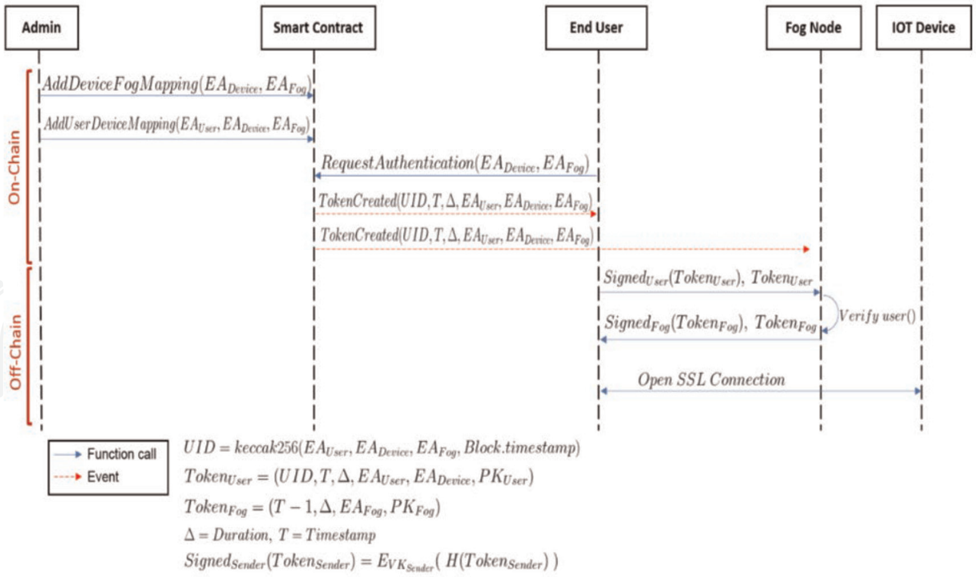
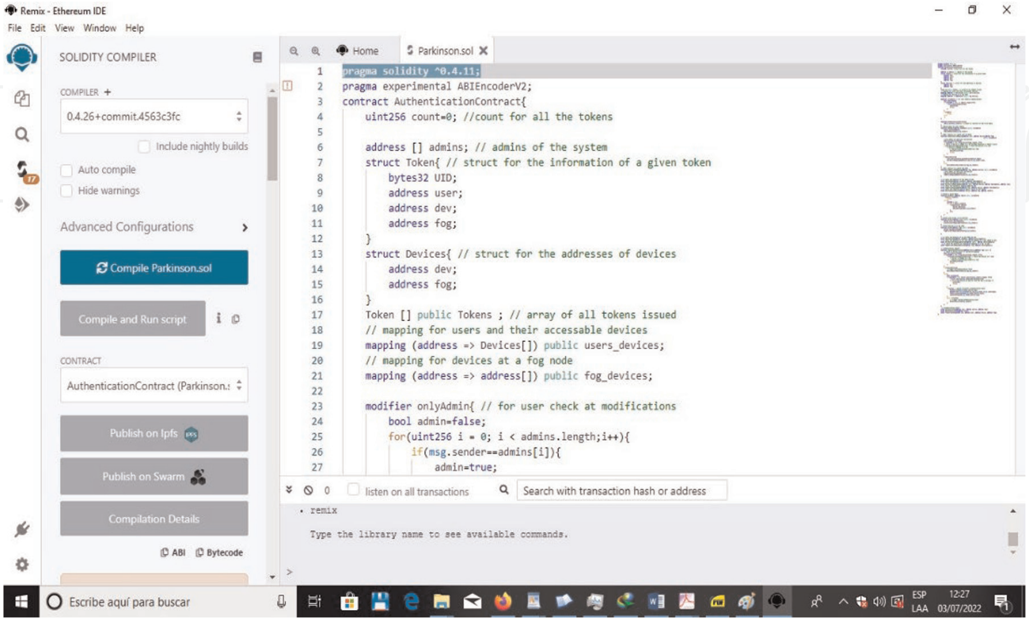

Admin
Smart Contract
End User
Fog Node
IOT Device
On-Chain
Off-Chain
$AddDeviceFogMapping(EA_{Device}, EA_{Fog})$
$AddUserDeviceMapping(EA_{User}, EA_{Device}, EA_{Fog})$
$RequestAuthentication(EA_{Device}, EA_{Fog})$
$TokenCreated(UID, T, \Delta, EA_{User}, EA_{Device}, EA_{Fog})$
$TokenCreated(UID, T, \Delta, EA_{User}, EA_{Device}, EA_{Fog})$
$Signed_{User}(Token_{User}),\ Token_{User}$
$Verify\ user()$
$Signed_{Fog}(Token_{Fog}),\ Token_{Fog}$
Open SSL Connection
Function call
Event
$UID = keccak256(EA_{User}, EA_{Device}, EA_{Fog}, Block.timestamp)$
$Token_{User} = (UID, T, \Delta, EA_{User}, EA_{Device}, PK_{User})$
$Token_{Fog} = (T-1, \Delta, EA_{Fog}, PK_{Fog})$
$\Delta = Duration,\ T = Timestamp$
$Signed_{Sender}(Token_{Sender}) = E_{VK_{Sender}}(\ H(Token_{Sender})\ )$

Figure 13.
Sequence diagram of exchange messages showing a successful authentication of the end user to IoT Device [17].

The end user first sends an authentication request to the smart contract using the request Authentication function, specifying the EA of the IoT device, before attempting to access that device. The user's saved list of authorized IoT devices will be reviewed by the smart contract. A rejected request event will be generated in the event that the user is not authorized to access that device. Otherwise, the smart contract will issue an acceptance event and an access token if the user has permission to do so, $TokenCreated = (UID, T, \Delta, EA_{User}, EA_{Device}, EA_{Fog})$. By definition, the event is broadcasted to all users and fog nodes [17].

The rest of the exchange messages showing a successful authentication of the end-user to the IoT Device are detailed and described in [17]. A view of the implementation of the given solidity software code given in the available Github is shown in **Figure 14**.



COMPILER
0.4.26+commit.4563c3fc

□ Auto compile
□ Hide warnings

CONTRACT
AuthenticationContract (Parkinson.s

```
pragma solidity ^0.4.11;
pragma experimental ABIEncoderV2;
contract AuthenticationContract{
    uint256 count=0; //count for all the tokens

    address [] admins; // admins of the system
    struct Token{ // struct for the information of a given token
        bytes32 UID;
        address user;
        address dev;
        address fog;
    }
    struct Devices{ // struct for the addresses of devices
        address dev;
        address fog;
    }
    Token [] public Tokens ; // array of all tokens issued
    // mapping for users and their accessable devices
    mapping (address => Devices[]) public users_devices;
    // mapping for devices at a fog node
    mapping (address => address[]) public fog_devices;

    modifier onlyAdmin{ // for user check at modifications
        bool admin=false;
        for(uint256 i = 0; i < admins.length;i++){
            if(msg.sender==admins[i]){
                admin=true;
```

```
• remix

  Type the library name to see available commands.

>
```

Figure 14.
Implementation of the simulation smart contract through the IDE Remix Ethereum.

6. Conclusion

In this chapter, a decentralized, Fog Computing, user authentication system was proposed for the development of a practical, low-cost, and general diagnosis system of the symptoms of PD patients using an Ethereum smart contract. It aimed to address user authentication in a decentralized environment and address fog computing security problems inherited from IoT and Cloud/Fog computing. It provided a solution for immutability and scale-ability problems in fog computing.

The system was simulated for validation and design using the simulation tools Cisco Packet Tracer, iFogSim, and Remix Ethereum. The obtained results proved the feasibility of the proposed system.

In future work, it is necessary to execute the process of building a fully functional system prototype involving real-based presented IoT devices connected to fog nodes equipped with Ethereum client to be connected with the real public Ethereum network, which is hosting the smart contract code. This makes it possible to complete a real diagnosis system of symptoms to support PD patients, necessary for the implementation of a real IoT health monitoring system that uses smartphones for data collection and machine learning algorithms for data processing.

Conflict of interest

"The authors declare no conflict of interest."

Author details

Armando de Jesús Plasencia Salgueiro[1*] and Arlety García García[2]

1 Nacional Center of Animals for Laboratory (CENPALAB), La Habana, Cuba

2 Youth Island University "Jesús Montané Oropesa", Nueva Gerona, Cuba

*Address all correspondence to: aplasencia278@gmail.com

References

[1] Sahandi Far M et al. JTrack: A digital biomarker platform for remote monitoring of daily-life behaviour in health and disease. Frontiers in Public Health. 2021;**9**(763621):11

[2] de Jesús A, Salgueiro P, Shichkina Y, García AG, Rodríguez LG. Parkinson's disease classification and medication adherence monitoring using smartphone-based gait assessment and deep reinforcement learning algorithm. Procedia Computer Science. 2021;**186**: 546-554. DOI: 10.1016/j. procs.2021.04.175

[3] Zhang H, Xu C, Li H, Rathore AS, Song C, Yan Z, et al. PDMove: Towards passive medication adherence monitoring of Parkinson's disease using smartphone-based gait assessment. Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies. 2019:23. DOI: 10.1145/ 3351281

[4] Umoren O et al. Securing fog computing with a decentralised user authentication approach based on blockchain. Sensors. 2022; **22**(3956):21

[5] Hartmann M et al. Edge computing in smart health care systems: Review, challenges, and research directions. Transactions on Emerging Telecommunications Technologies. 2022;**33**(3710):28

[6] E. LLC. Lab Design Guide For Artificial Intelligence (AI), Internet of Things (IoT), Autonomous Vehicles, AR/VR, Blockchain and Industry 4.0 Labs. Dubai, United Arab Emirates: EdNex; 2022

[7] Uddin A et al. Continuous patient monitoring with a patient centric agent: A block architecture. IEEEAccess. 2018; **6**(32700):27

[8] Al-Joboury IM et al. Automated Decentralized IoT Based Blockchain Using Ethereum Smart Contract for Healthcare. Baghdad: Springer Nature Switzerland; 2021

[9] Jesin A. Packet Tracer Network Simulator. Birmingham: Packt Publishing; 2014

[10] Awaisi KS, Abbas A, Khan SU, Mahmud R, Buyya R. Simulating Fog Computing Applications using iFogSim Toolkit. In: Mobile Edge Computing. Cham: Springer; 2021. pp. 565-590

[11] Mukhopadhyay M. Ethereum Smart Contract Development. Birmingham - Mumbai: Packt Publishing; 2018

[12] Thera D. davidthera/iot-simulation-with-cisco-packet-tracer, 28 Jun 2020. [Online]. Available from: h ttps://github.com/davidthera/iot-simula tion-with-cisco-packet-tracer. [Last access: 30 May 2022]

[13] Thera D. Internet of things simulation using CISCO packet tracer. [Master of Science in Computer Engineering thesis], İzmir Institute of Technology. 2020

[14] Mahmud R, Buyya R. Modeling and simulation of fog and edge computing environments using iFogSim Toolkit 433 Wiley STM. In: Srirama B, editor. Fog and Edge Computing: Principles and Paradigms. Chapter 17 Introduction to Fog and Edge Computing. 2019. DOI: 10.1002/9781119525080.ch17

[15] Mahmud R et al. iFogSim2: An extended iFogSim simulator for

mobility, clustering, and microservice management in edge and fog computing environments. 2021. p. 17. arXiv: 2109.05636v2

[16] Zheng J et al. An in-depth review on blockchain simulators for IoT environments. Future Internet. 2022; **14**(182):22

[17] Almadhoun R, Kadadha M, Alhemeiri M, Alshehhi M, Salah K. A User Authentication Scheme of IoT Devices using Blockchain-Enabled Fog Nodes. In: 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), Aqaba, Jordan. 2018. pp. 1-8. DOI: 10.1109/AICCSA.2018.8612856

[18] Kadadha M. Authentication at a scale. 28 Apr 2018. [Online]. Available from: https://github.com/mkadadha/AuthenticationAtAScale1.git. [Accessed: 30 Apr 2022]