IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 22, 2021, accepted January 22, 2022, date of publication January 31, 2022, date of current version February 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3147795

Novel and Secure Blockchain Framework for Health Applications in IoT

MARAH R. BATAINEH, WAIL MARDINI, YASER M. KHAMAYSEH, (Member, IEEE), AND MUNEER MASADEH BANI YASSEIN, (Member, IEEE)
Faculty of Computer and Information Technology, Jordan University of Science and Technology, Irbid 22110, Jordan
Corresponding author: Wail Mardini (mardini@just.edu.jo)

ABSTRACT Internet of Things (IoT) has grown increasingly in the past decade. This growth brings up several challenging issues for a successful continuous operation of IoT applications. Some of these challenges that need to be taken care of are resource constraints, central server overload, and the risk of illegal use of private data. On the other side, Blockchain technology is increasingly popular and has gained huge success in cryptocurrencies. It offers numerous vital qualities such as a technique for consensus, peer communications, confidence-building without a trustworthy third party, and a transaction controlled by conditions and functions using the intelligent contract technique. Blockchain is an excellent candidate to establish a decentralized, autonomous IoT system addressing the above issues. This study proposes an IoT-Blockchain integration architecture using an Ethereum Blockchain infrastructure within a rich-thin client IoT approach to address the challenges created by the limited IoT resources while implementing the Blockchain mining technique in IoT systems. The architecture depends on load distribution between the resources. Limited resource devices are the thin-clients, while the higher resource devices are assigned as rich-clients. Both clients can access the blockchain and collect the data, but rich-client can only execute the mining process. In addition, we implement a healthcare system based on the proposed architecture in which surgical process management is carried out. We also prove our solution's efficiency by testing and comparing the architecture against other well-known IoT-based blockchain architectures. The obtained results show that proposed blockchain-IoT architecture is suited for many IoT applications while avoiding difficulties created by IoT devices' limitations.

INDEX TERMS Blockchain, Ethereum, healthcare, Internet of Things (IoT).

I. INTRODUCTION

The twenty-first century is geared up by rapid development in many technological fields affecting almost every aspect of our lives. Nowadays, all life activities are associated with or managed by technology. There are many examples of technology gathering data and embedding it in different forms to be used in different applications. It can be seen while doing our daily personal and professional activities. For example, smart devices (such as phones and tablets) can help calculate the distance, the number of steps, the heartbeat and blood pressure, and the number of calories burned. This example and many other scenarios where sensors and communication devices are involved lead to creating a new concept called the Internet of Things (IoT). In IoT, the devices usually interact, share data, and even make decisions on our behalves. IoT term first mentioned in (1999) by Kevin Ashton [1]. Since then, the term blossomed to be a significant section in the network world. IoT provided a platform for many applications in many life fields. IoT facilities the creation of smart homes or even smart cities. The huge evaluation in IoT causes a vast number of devices connected to the network and the big data collected. As a result, IoT now suffers from many challenges and issues that the researcher tries to solve.

The rapid development and usage of IoT introduced many challenges that should be tackled. IoT devices are usually equipped with low computing power, low storage capacity, and limited network bandwidth. Moreover, IoT uses centralized system architecture. The centralized system architecture causes constraints in cost and capacity, server failures that affect the whole system, and security issues [2].

The associate editor coordinating the review of this manuscript and approving it for publication was M. Sabarimalai Manikandan.

Regardless of the security issues in IoT at the network layer, physical interface, and unpredictable attacks.

To make life activities part of the modern age, IoT integrates and improves manual systems, obtaining data quantities that can provide knowledge at highly thought levels. This knowledge encourages the creation of intelligent technologies, such as optimizing people's management and quality of life through the digitization of public resources. Recently cloud computing systems have helped provide the IoT with the required capabilities to examine and process data and transform it into actions and knowledge. This exponential IoT development has created new possibilities for the community, such as accessing and exchanging information. In these programs, the open data model is the foundation. However, one of the most noticeable drawbacks of these programs is the lack of trust. Centralized architectures, such as the one used in cloud computing, have contributed to IoT growth. However, they serve as black boxes concerning data privacy; network users do not clearly view when and how the information they generate will be used.

It has proved invaluable to incorporate promising developments such as IoT and cloud computing. Also, we understand that Blockchain has a massive ability of revolutionizes IoT. By offering a trustworthy sharing service where knowledge is accurate and traceable, Blockchain will enrich the IoT. Data origins should still be known, and data can be maintained static over time and improve its security. If many users safely exchange IoT knowledge, this incorporation will constitute an important and safe revolution. Using Blockchain, accurate and secure information can be delivered to the IoT. Thus, Blockchain technology addresses challenges relating to the IoT model in terms of scalability, safety, and reliability.

In (2008), Satoshi Nakamoto invented Bitcoin [3]. Bitcoin is a cryptocurrency in which the currency transactions use a distributed ledger called Blockchain. Blockchain has improved its efficiency in the Bitcoin enterprise. In late (2016) developers studied the idea of integrating Blockchain with IoT. They figured that Blockchain provides advantages that help to solve IoT issues.

The best way to demonstrate Blockchain is by discussing how it was implemented and used in Bitcoin. Blockchain in Bitcoin was used as database storage in many computers. This Blockchain was storing every transaction that has been made ever. The main difference between ordinary database and Bitcoin Blockchain was the computers was distributed in many geographical locations controlled by uniquely separated people. Bitcoin network was made up of millions of computers known as nodes in Blockchain terms. Blockchain manner was categorized decentralized, as it shows. However, there are private, centralized Blockchains, in which one device control the stored data and the nodes makes the network. Blockchain's most important property is that it is irreversible. Each node in the Blockchain store a record of all transactions in the network. If any node contains wrong data, it can correct itself by using thousands of other nodes as a reference. Also, if any node user tries to change the stored data, all other nodes would cross-reference each other and detect the wrong data. This mechanism helps to construct an exact and straightforward sequence of events. So it is possible to Blockchain to store many kinds of data rather than transactions such as legal contracts, state identifications, or product inventory of a business. To modify the data in the Blockchain, it has to take the approval of the majority of the computing power of the decentralized network. This ensures that any modifications will serve the best for the majority.

Blockchain technology has proved its efficiency in digital currency transactions. For that, researchers adopt the idea of Blockchain integration with IoT. [4] work demonstrate the use of Blockchain in digitalizing governments services in the optimized transform. Some of Blockchain's features make it an appealing technology to solve IoT privacy and security challenges:

1) Decentralization: the absence of central control ensures scalability and robustness through the use of resources from all participating nodes, and the removal of traffic flows from many to one. This feature also eliminates delay and overcomes the issue of a single failing stage.
2) Anonymity: the inherent anonymity offered is well-suited for most IoT usage cases in which the user's identity must be kept secret.
3) Security: With multiple and heterogeneous networks, Blockchain recognizes a stable network over untrustworthy parties that is desirable in IoT.

It is not easy to implement the Blockchain in IoT, and it results in the following flaws:

1) Processing power and time: Various devices create IoT networks with distinct computational capacities; not all of them would be able to execute the same encryption algorithms at the desired level. Mining is computer-intensive, and it will not be practical to handle any of the IoT nodes. Besides, block mining takes time, although low latency is advantageous in most IoT applications.
2) Storage: The Blockchain ledger needs to be stored within the nodes themselves, and as time passes, it can grow in size. This is beyond the capability of a wide range of smart devices such as sensors with relatively little storage capacity.
3) Overhead traffic: The underlying Blockchain protocols produce substantial overhead traffic that could be unacceptable for IoT devices with bandwidth limitations.
4) Scalability: Blockchain scales poorly as the number of network nodes grows. However, IoT networks can involve a significant number of nodes.

In summary, Blockchain technology was suitable to solve security issues in IoT. Blockchain is a decentralized system that depends on the anonymity of the users—moreover, it grantee data integrity. However, adopting the Blockchain in IoT is not that easy. As we explained previously, IoT devices suffer from low computing power that will not be capable of running encryption algorithms rapidly. Also, the Blockchain ledger must be stored in each node, which needs a high storage capacity. Furthermore, Blockchain protocols cause overhead traffic that the limited bandwidth of the IoT devices cannot handle. Moreover, one of the Blockchain problems is that Blockchain is poorly scaling with the large number of nodes in the IoT network.

Considering the features mentioned above and the constraints of Blockchain networks, it became clear that is developing an integration between Blockchain and IoT to meet all these requirements is a challenging mission and needs hard work. This study solves some of the problems that other integration methods suffer, such as the limitation in resources at the IoT layer and the difficulty in implementation at the blockchain layer. In this study, we studied the work of [5] and developed his integration architecture to solve the limitations of resources in the integration process more effectively. In our work, we decided to test the integration output on a vital life field, healthcare. As known, healthcare applications using IoT have been developed over the years since IoT consider an important element in technology evaluation. Recently in the growth of IoT new challenges appeared that affect the effectiveness of IoT applications in healthcare management, so the integration between IoT and Blockchain provided solutions to security, privacy, and resource limitations problems.

II. BLOCKCHAIN AND IoT INTEGRATION

The problems of IoT systems include fragmentation of IoT systems, low interoperability, IoT software resource limitations, privacy vulnerabilities, and protection. With enhanced interoperability, safety, and stability, blockchain technology will complement IoT systems. Besides, Blockchain can increase IoT systems' stability and scalability. Briefly, we call blockchain integration with IoT BCoT. In comparison to current IoT schemes, BCoT has potential advantages.

Blockchain would improve the IoT device's interoperability. The interoperability of IoT systems can be significantly enhanced by blockchain transformation and storage of IoT data. This method transforms, processes, extracts, compresses, and eventually preserves heterogeneous forms of IoT data in blockchains. Besides, interoperability can also easily be accomplished across many forms of decentralized networks, as blockchains are built above the universal web access Peer-to-peer network.

Blockchain could enhance IoT systems' stability. On the one side, IoT information can be protected via blockchains as it is stored as cryptographically encrypted and signed blockchain transactions. In addition, the integration of IoT systems with blockchain technology can help improve the protection of IoT systems by upgrading the IoT device configuration automatically to correct insecure infringements, which improves the security of the system.

IoT data traceability and trustworthiness will be guaranteed with the integration with Blockchain. Blockchain data will anywhere at every time be found and checked. In the meanwhile, all past transactions in the blockchains can be traced. For example, Qinghua's work has developed a blockchain-based traceability framework that offers traceable services to manufacturers and retailers. This way, goods can be tested and confirmed for consistency and originality. Moreover, blockchains' immutability also guarantees the authenticity of IoT records, as transactions stored in blockchains could hardly be changed or falsified.

Blockchain also provides to the IoT systems' autonomous communications. IoT machines or subsystems may be automatically dealt with by Blockchain technology.

A. INTEGRATION ARCHITECTURES

In the Integration between Blockchain and IoT, too many parameters may be in consideration to configure the architectural structure. The interactions of IoT, such as correspondence between the IoT underlying infrastructure, are a factor to be taken into account. When Blockchain is integrated, it must be determined where these connections can occur: a hybrid architecture of IoT and Blockchain, or within the IoT only or a blockchain only. Fog Computing, therefore, had modernized IoT with the integration by incorporation of a new layer between cloud computers and IoT applications. These alternatives and their advantages and drawbacks are listed below:

a) Blockchain as a database: in terms of reliability and security, this solution may be the quickest because it can operate offline. IoT systems must connect, typically requiring processes for exploration and routing. In the Blockchain, only part of IoT data is stored, while without the use of Blockchain, there would be IoT interactions. This method can be helpful in situations where stable IoT data was used in low latency IoT interactions.

b) Blockchain is combined in all layers: all the interactions flow via Blockchain with this method that makes an unchanging history of interactions. This way, all the communications selected are traceable as their information can be verified in the Blockchain, and the flexibility of IoT devices can be improved. This strategy can be leveraged through IoT apps that plan to exchange or rent, such as Slock. It is for the provision of its services. However, tracking all of the transactions in Blockchain will entail raising the bandwidth and records, one of the famous blockchain challenges. On the other hand, the Blockchain can also hold all the IoT information related to these transactions.

c) Hybrid approach: eventually the hybrid architecture in which only part of the communications and data are exchanged directly between IoT devices in Blockchain. One of the difficulties with this strategy is to determine which connections the Blockchain can go through and to determine in time. The best way to combine both technologies will be to orchestrate this solution perfectly as it takes advantage of Blockchain and the advantages of IoT interactions in practice. In this method, fog computing and sometimes cloud computing may be used to supplement blockchain and IoT constraints.

In many studies, there are many designed architectures to integrate IoT and Blockchain, all these architecture rolling around the combination method in a certain topology. We can classify some of them in names, hierarchical architecture, general architecture, modular architecture, and hybrid architecture. Each of them serves the purpose of the build application. The centralization of architecture is induced by the existence of hierarchy formation. Supernodes and the leader of all nodes would most certainly be the highest node. In hierarchical architecture, new elements or applications could be constructed over the underlying blockchain system, which is sufficient to create IoT networks [6]–[8].

The general architecture for IoT Blockchain integration typically incorporates various blockchain systems at the connectivity layer. Also, the database layer uses both public and private distributed ledgers. This architecture could be an impressive construct for using blockchain technology in complicated IoT applications [9]. Another suitable architecture for complex IoT systems is a hybrid architecture [10]. Hybrid architecture may be a combination of more than one architecture and using different technologies such as cloud computing [11] or artificial intelligence [12], [13] technologies in the implementation of the IoT Blockchain application.

One of the significant architectures proposed by [14] in IoT Blockchain integration is modular architecture. A modular architecture is a layer-based architecture in which each layer is isolated from other layers. In modular architecture, it is easy to replace or incorporate a new module without affecting the remainder of the structure. The physical layer of IoT comprises numerous connected devices with communication, processing, and data storage capabilities. The key role of the connectivity layer is routing management, as an auto organization is needed because there are no global internet protocols on physical devices (IPs). This layer also includes other service modules, such as management of a network, information security, and message handler. The IoT blockchain module layer includes all the modules organizing traditional resources, including access control, consensus, and peer-to-peer networking, to include different functionality of blockchain technology.

Finally, In [5] used a based rich-thin client architecture in IoT and Blockchain integration. Sun developed a solution for electric vehicle battery refueling using the rich-thin client. the rich client was the battery refueling station that swap empty battery with full battery by simple smart contract compares the batteries information in the station with vehicle battery information, then take the decision of battery swapping by certain conditions. the thin clients were the vehicles themselves. The thin client was a simple IoT device that contains the vehicle battery information and interacts with the rich client by sending the battery information and receiving the new battery information. this architecture reduced the complexity of the system in involving all devices in the blockchain by integrating only the station devices to the blockchain. Also, it ensures the privacy and security of the battery information using the provided securing techniques by Blockchain.

B. BLOCKCHAIN SCHEMES FOR IoT

Blockchain was described as a transformative technology that will have a major effect on many sectors. In this segment, we are focusing on the most common and most appropriate IoT domains. The number of schemes is so high and in continuous transition that we cannot evaluate them all. Bitcoin was the first blockchain network for cryptocurrencies. It provides a scheme for quick, cheap, and secure monetary transactions, which can be implemented as a protected payment method in applications. In the field of IoT, self-sufficient devices can make micro-payments with Bitcoins that function primarily as wallets. In general, apps that restrict the use of Blockchain to micropayments are attached to the currency that can be a detriment because coin deflation can adversely affect the software. As demonstrated by smart contracts, the combination of Blockchain with IoT is a popular approach. Bitcoin has a scripting language that enables those terms to be defined when transacting.

As stated, Ethereum [15] is one of the platforms which in recent times had a major effect. Ethereum was one of the founders of smart contracts, including blockchains. Ethereum can be defined both as an embedded programming language (Solidity) blockchain and as an internationally controlled, consensus-based virtual machine (Ethereum Virtual Machine EVM). Including smart contracts pushes the Blockchain away from currencies and makes this platform easy to integrate through different sectors. This makes Ethereum the most popular forum for creating apps along with its involved and wide community. Ethereum is used or compliant in most IoT programs. A smart contract is the easiest way to describe devices that are able to publish actions and policies that lead to changes.

Another effective Blockchain framework is Hyperledger [16]. Hyperledger is a framework for open source applications, including Hyperledger Fabric, a blockchain that is stripped of permissions and without cryptocurrencies onto which commercial projects such as IBM Blockchain are based. Hyperledger is an open-source framework. It allows for consensus and participation of various components. In the Blockchain, a distributed application with general languages can be created. IoT devices can provide blockchain data through the IBM Watson IoT Platform to monitor devices, analyze and filter data. By selling it as an application, Bluemix IBM's network enables integration with blockchain technologies. This platform aims to speed up development and testing software and also has produced a variety of use cases.

Multichain utilizes a financial adviser, asset management, authorization, transactions, etc., which expands the essence of the original bitcoin program by adding a modern feature. It also includes an online command tool to communicate with a network and multiple consumers who are able to interact with the Node.js, Java, C#, and Ruby networks using JSON-RPC. Multichain is a Bitcoin Central fork, compiling the source code for 64-bit. Multichain Blockchain is used in [17] work on three nodes, one of them an Arduino board, to prove the IoT–blockchain implementation definition.

Litecoin [18], is functionally similar to Bitcoin; however, due to reduced time of block-generation (from 10 min to 2,5) and scrypt-based proofs of work, advanced control key derivation feature based on a password, it has more fast transaction confirmation times and improved storage performance. This means that Litecoin nodes are less computer-based, so it is more fitting for IoT.

Lisk [19] provides a forum for Blockchain with the use of decentralized blockchain implementations and a number of cryptocurrencies for sub-blockchain or sidechain (e.g., Bitcoin, Ethereum, etc.). Known as the javascript developers' blockchain framework, Lisk also provides support in designing and implementing decentralized apps within the platform for end-users to be used directly to build an ecosystem of Blockchain interoperable resources. The built applications will use LSK or build customized tokens. Lisk uses Delegated Stakeholder Proof. Lisk works with Chain of Things to see how blockchain technologies will create secure IoT systems effectively.

Quorum [20] is an Ethereum blockchain network that is intended to promote transaction and contract protection in the financial services industry. It facilitates several mechanisms of agreement and protects data security through encryption and segmentation. ZeroCash technology has been recently integrated to mask any recognizable transaction information. The network Chronicled [21] used the Quorum platform to establish protected relations between physical and Blockchain properties.

Smart contracts exist on most networks allowing application functionality outside transactions in cryptocurrencies. For a blockchain implementation, cryptocurrency is a dualism amongst blockchains. The required infrastructure can be created for a blockchain implementation on an existing network with a cryptocurrency like Ethereum. You can use AWS (Amazon Web Services) or Google Cloud Network as the implementation of your own cloud infrastructure. However, distribution is the key to trust in the case of Blockchain.

III. METHODOLOGY

This study proposes an Enhanced Rich-Thin-Client architecture (ERTCA) to address the limitations of IoT devices. The proposed architecture has two main contributions as the following:

a) To avoid the crash of overloaded IoT nodes, we provided a solution that aims to fit the resources as can as possible. The solution is represented by giving each node enough tasks to suites the device resources.

b) To enhance the connectivity between IoT and Blockchain platform, we propose to connect the Blockchain at a major node with high capabilities, and other poor nodes can take the benefits of Blockchain without overloading on them. Nodes classification depends on two factors, the computational power of the device and storage capability.

This contribution solves some of the problems that other integration methods suffer, such as the limitation in resources at the IoT layer, and the difficulty in implementation at the blockchain layer. in this contribution, we studied the work of [5] and developed his integration architecture to solve the limitations of resources in the integration process more effectively.

A. ERTCA

We supply Ethereum Blockchain with a rich-thin-clients IoT solution. The ERTCA design would accomplish the research goals between the limited IoT and the centralized problem of architecture. Thin clients, responsible for user interface and information collection, are considered IoT devices with limited resources; the rich clients, thin clients, and complete Blockchain nodes may be seen as devices with resources equivalent to personal computers or equal to them. As our basic Blockchain system, we use our own Ethereum private Blockchain network. To create relatively complicated connections between different IoT devices, human users, and devices, Ethereum allows smart contracts and generates a new block easier than Bitcoin. We used the original consensus system: PoW and Ethereum account, as the original encryption method.

Our work differs from [5] work in that we used another level of the thin client to collect data without interacting with the system and it's not attached to the rich client. the second level thin-client is only responsible to send the needed data from the real world. the second level thin clients can be sensors or existing databases or humans. the purpose of that enhancement is to reduce the complexity in each layer of the architecture; because the load on IoT devices be more distributed. Moreover, ERTCA rich clients' smart contracts are more complex; because is process unexpected data by the application scenario conditions to produce knowledge and decisions.

1) SYSTEM ARCHITECTURE AND TOPOLOGY

The Enhanced Rich-Thin-Clients Architecture (ERTCA) was designed differently from the hierarchical architecture in which all architecture's clients have some duplicated tasks, while the levels in a hierarchical architecture process distinct tasks.

The Rich and thin clients each support a graphical user interface to be easy to use by the users, and this interface requires Blockchain APIs via the Blockchain interface provided by the rich client, business principles definition, and IoT data collection (elective in rich clients). The main difference between a rich client and a thin client is that only the rich client has a full Blockchain node that stores all Blockchain transaction records; moreover, it can add

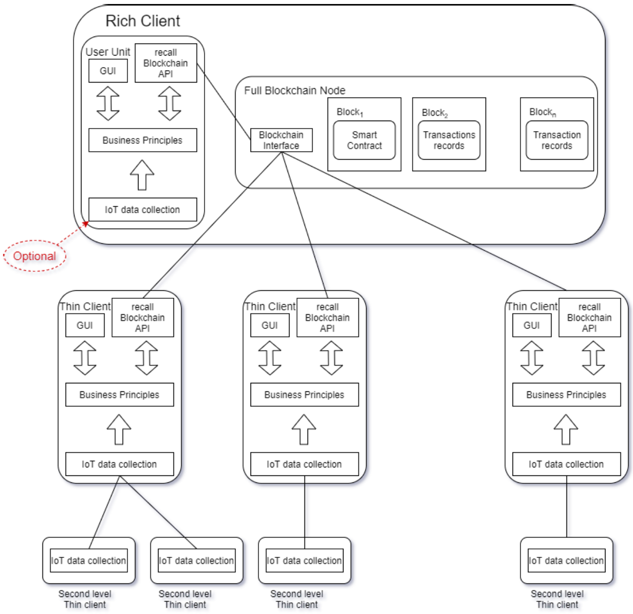


FIGURE 1. The Enhanced Rich-Thin-Clients Architecture (ERTCA).

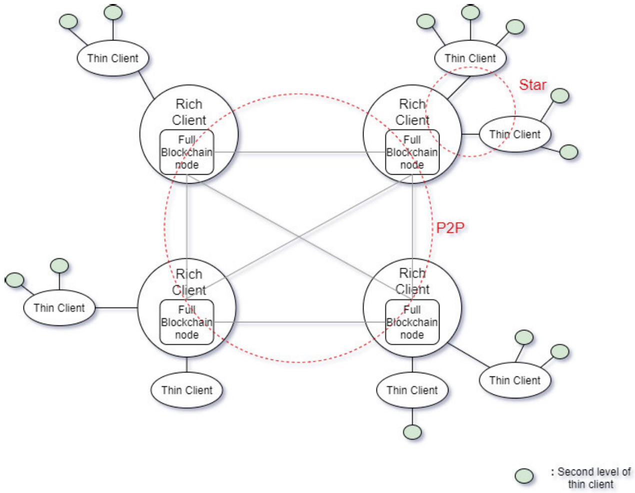


FIGURE 2. The system topology.

transactions to a Blockchain; because it is a mining node. The second layer of the thin client only collects IoT data without any interactions between users or Blockchain. Figure (1) shows the ERTCA.

Each rich client includes an Ethereum Blockchain node, which can work with another Ethereum Blockchain node and execute a PoW consensus algorithm. Much like public Ethereum Blockchain nodes, the rich clients are a P2P network. The thin clients linked to a certain rich client create a topology, which is a star topology. the advantage of using star topology in Blockchain is that any failure in the rich client other rich clients can handle it and save the attached thin clients since it is a decentralized network, rather than other rich clients will refuse any illegitimate transaction done by the failed rich client. As our network should be close to the Ethereum BC public performing network, we should overcome the problem between small IoT computer resources and centralized infrastructure problems since only rich clients with higher resources execute mining and consensus algorithms. In figure (2), the full image of the architecture and the created topology by the clients.

2) PRIVACY AND SECURITY

To distinguish every client and protect transactions, we use the Ethereum account system. Per client is assigned a specific Ethereum account in order to recognize them exclusively within our scheme. Any client receives his own private and public key, including his Ethereum account. Moreover, since accounts are not specifically linked to actual personal records, this may give users anonymity.

IoT devices (Thin clients) are the most sensitive elements experiencing threats in an IoT application. In the situation of an IoT device has been attacked, the hacker probably will do one of the following three options:

1) Thieve the account used on the thin-client (referred to as IoT device), which is somehow hard because of the Ethereum account mechanism; that provides a username (named as public key) and a password for each account, rather than an unknown private key stored in the rich-client devices for each thin-client. the private key is used to add and remove the thin-client account from the Blockchain system. Also, The public key is generated from the private key using the Elliptic Curve Digital Signature Algorithm. Unfortunately, by way of centralized architecture and decentralized architecture, that issue cannot be resolved if it happens, except removing the account from the chain.
2) Attack some devices with this device. This issue could pose serious problems in a centralized architecture since an IoT device can target main servers, but in a decentralized structure, it would not be.
3) Fake an account or use this device to submit false info. Since only rich clients can create a legitimate account, our system cannot verify the false account. Moreover, due to Blockchain authentication mechanisms, illegitimate transactions or accounts are rejected when blocks are created.

Moreover, if a hacker attacks any rich client, other rich clients will refuse any illegitimate request sent by the hacked rich client; since each rich client is just one node in the Blockchain.

B. AN APPLICATION SCENARIO

ERTCA is a proposed architecture that integrates IoT and Blockchain at any application, smart homes, smart city, smart manufacturing, or health care. In our work, we tented to a vital field nowadays, healthcare. In order to study the effectiveness of the proposed architecture, we are considering the following scenario inspired by the work of Cheng in [22] and adapted

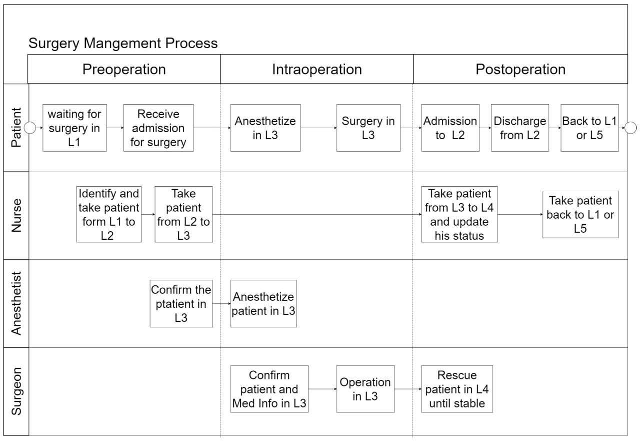

Surgery Mangement Process
Preoperation
Intraoperation
Postoperation
Patient
Nurse
Anesthetist
Surgeon
waiting for surgery in L1
Receive admission for surgery
Anesthetize in L3
Surgery in L3
Admission to L2
Discharge from L2
Back to L1 or L5
Identify and take patient form L1 to L2
Take patient from L2 to L3
Take patient from L3 to L4 and update his status
Take patient back to L1 or L5
Confirm the ptatient in L3
Anesthetize patient in L3
Confirm patient and Med Info in L3
Operation in L3
Rescue patient in L4 until stable

FIGURE 3. Surgical process workflow.

to the protocol of the Jordanian hospitals. We consider a big hospital, incredibly busy, and dysfunctional institution that treats hundreds of cases every day. Hundreds of doctors and other employees are continuously commuting, patients and community hospitals are being monitored around the hospital, and clinical supplies for urgent use are usually recalled. While information systems are used to facilitate healthcare workers' work, the movement of patients and assets is not reliably monitored in real-time. So, in this complex and unstable scenario, detecting and reacting to urgent conditions becomes difficult. A large amount of knowledge sharing, teamwork, situational recognition, and rapid responses is particularly necessary for surgery. It is very difficult to fulfill these criteria if appropriate infrastructure is not used to monitor patient flow, medical equipment and alert staff to unexpected circumstances. Failure in such areas will jeopardize patient safety, reduce surgical performance and raise medical costs.

As it can be seen in figure (3), the workflow of the scenario can be divided into three stages; the stage before the operation (preoperation), the stage during the operation (intraoperative), and the stage after the operation (postoperative). Also, the process members can be defined into four groups, patients, nurses, anesthesia doctors, and surgeons. The surgical procedure involves a set of locations to proceed: the ward, the operating suite, the operating room, the recovery room, and the ICU, named in the workflow as L1, L2, L3, L4, and L5, respectively. During the surgery progress, there might be many emerging incidents that could render vital operating instruments missing from the operating room, leading to patient loss, among other factors, if not attended to it. It is hard to track the patient's and instrument's positions in real-time without IoT technology, and it can be very difficult to locate them in an emergency. Another possible difficulty can arise when a patient is checked manually, and medical errors can lead to a human error.

Finally, the appropriate technologies could enable critical circumstances to be detected in advance, preventing certain potentially serious situations. For instance, a nurse can wrongly transfer a patient to the incorrect Operating room, thereby delaying his/her surgery. Furthermore, where there is evidence that a patient is in danger of an accidental heart attack during an operation, a further medical device may be necessary for immediate use because if it is not sent to the correct Operating room on time, severe complications may arise. There are also some obstacles to operations in this complex and often unpredictable setting.

C. IMPLEMENTATION OF ERTCA BASED SURGERY MANAGEMENT SYSTEM

We developed a private Ethereum network as an underlying Blockchain system for the application. The Blockchain of Ethereum is a transaction-based state machine. In computational theory, a state machine refers to a sequence of inputs that can transform into a new state-dependent on those inputs. We start with a ‘‘genesis state’’ with the State machine of Ethereum. This is like a clean canvas before any network transfers have occurred. This state of genesis is transformed into a final state as transactions are completed. This ultimate state reflects Ethereum's actual state at any moment. Millions of transactions take place in Ethereum State. To make a transaction legitimate, a validation method known as mining must be carried out. A mathematical algorithm is regarded as a ‘‘proof of work’’ to verify each block by the rich client. As we mentioned before, we used an Ethereum account mechanism, Ethereum's public ‘‘shared state’’ has several small artifacts (‘‘accounts’’), which can communicate via message transmission. There are related states and 20-byte addresses of each account. In Ethereum, two kinds of accounts are available: Externally owned accounts (thin clients) managed via private keys and have no related code, and contract accounts (rich clients) are managed by and connected with their contract code. In creating our private Ethereum network, we used Ganache, a tool provided by Truffle suit, to develop DApp. Ganache provided ten Ethereum accounts with 100 ether each.

To create the application, we identify the elements of the system. Our system contains the surgery process and its states, and each belongs to the sub-systems. In figure (4) shows the state machine of our system. As shown, there is a Patient Assistant System (PAS), a Nurse Assistant System (NAS), an Anesthesia Assistant System (AAS), and a Surgeon Assistant System (SAS). All of these systems are controlled by Surgery Management System (SMS). Also, each has its sequence of events.

The following systems are the thin clients in the architecture as they are controlled by the rich client, which is the SMS:

1) PAS: this system is concerned with creating a record for the patient as the surgery is required. It arranges for the surgery and initializes patient records with basic information (name, age, gender, …etc.), next state confirming the surgery's admission.
2) NAS: this system is the beginning of the medical journey. It identifies the patient's records and his/her

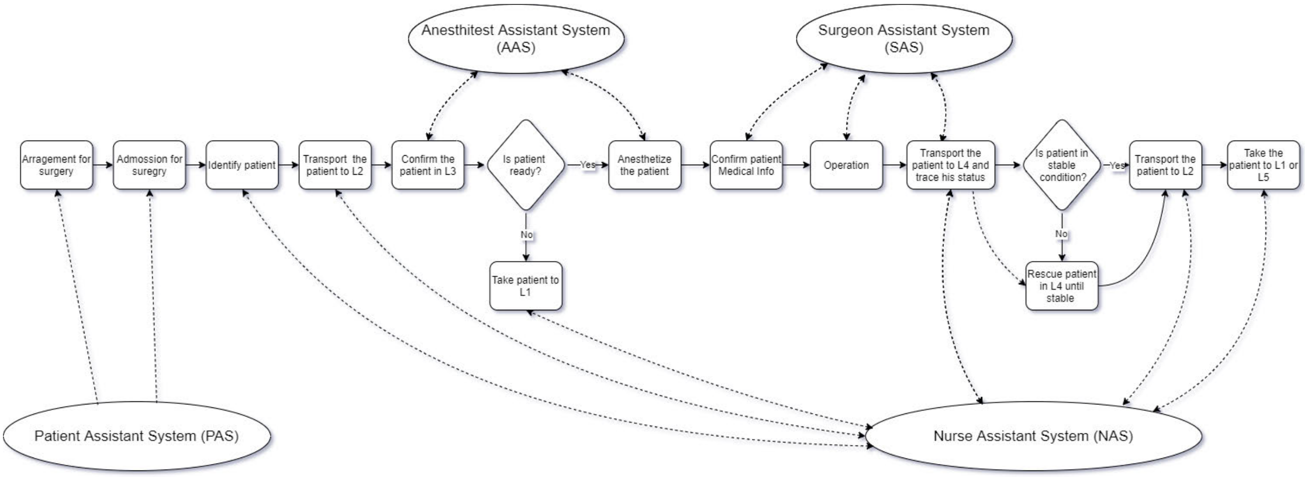

Anesthitest Assistant System (AAS)
Surgeon Assistant System (SAS)
Arragement for surgery
Admossion for suregry
Identify patient
Transport the patient to L2
Confirm the patient in L3
Is patient ready?
Yes
No
Take patient to L1
Anesthetize the patient
Confirm patient Medical Info
Operation
Transport the patient to L4 and trace his status
Is patient in stable condition?
Yes
No
Rescue patient in L4 until stable
Transport the patient to L2
Take the patient to L1 or L5
Patient Assistant System (PAS)
Nurse Assistant System (NAS)

FIGURE 4. The event-flow of the application state machine.

medical information (heart rate, blood pressure, oxygen saturation, . . . etc.). It is also responsible for transferring the patient in the surgical location set based on each event. Moreover, updating the medical information for the patient as needed.

3) AAS: in this system, it requires to confirm the patient and his ability to proceed in the surgery based on his medical status and some standards (most known is the fasting hours, allergies), next state will depend on the previous, it may send back the patient, or anesthetize to proceed the surgery.
4) SAS: this system is responsible for operating and tracking the patient status after the surgery.

NAS, AAS, and SAS thin clients have second-level thin clients collecting medical data, medical devices (pulse oximeter, sphygmomanometer, etc.).

The rich client and the thin client has been specified, and each has an Ethereum account. The next step was deploying the applicable rules, the smart contract. We implemented smart contracts to manage the state machine of the SMS. We used the Truffle library to connect the smart contract with the Ethereum account. Our smart contract was written with the solidity programming language.

Finally, we used ReactJS to create the graphical user interface, as it combines JS, HTML, and CSS in the same code. ReactJS provides many libraries in connecting with Blockchain. We used the Web3.js library to connect each client interface with its account. Also, the Web3.js library provides functions to connect the rich client interface with the smart contract. To interact with the accounts, we use MetaMask, as it provides an interface to interact with the ganache account from the user interface. In figure (5) shows the system implementation architecture.

D. SMART CONTRACT IMPLEMENTATION TO SMS

The smart contract was written using solidity programming language to maintain the surgery management process. This implementation simulates the surgery process that reflects the real-life process. The contract can be divided into three sections:

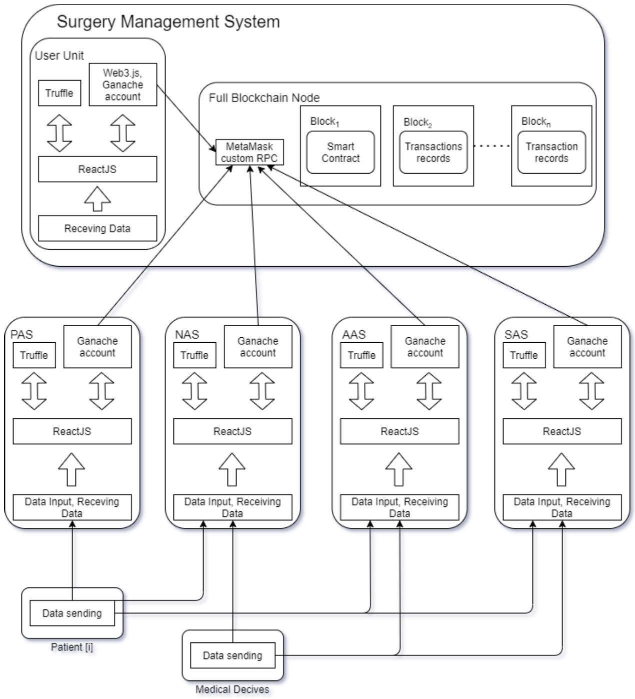


FIGURE 5. The ERTCA and SMS full architecture.

1) Database, which contains a dynamic array of objects to store the patients' data. These data are id, name, age, gender, surgery admission, patient consent, patient location, systolic blood pressure, diastolic blood pressure, heart rate, oxygen saturation, food fasting hours, water fasting hours, surgery duration, anesthesia confirmation, anesthetized, operation confirmation, and operated, patient stability.

2) The event handler, that call surgery process functions at each event, which are: Patient record creation, surgery admission confirmation, patient consent confirmation, patient current location, patient medical information update, anesthesia confirmation, anesthetized patient, operation confirmation, patient surgery operated, patient stability after surgery.
3) Functions, which contain all the functions needed to proceed with the event. Which are:
    a) Patient creation: add the patient to the array with the basic information: id, name, age, gender.
    b) Admission confirmation: when the surgery admission has been confirmed the status of the patient's surgery admission changed to true.
    c) Consent confirmation: when the patient (or his guardian if he is under the legal age or not conscious or mentally unstable, or the doctor in emergency cases) confirms his consent to any medical procedures, the status of the patient's surgery admission changed to true.
    d) Set the current location, change the location of the patient if he transferred, mentioned the surgical process locations in the scenario are, the ward, the operating suite, the operating room, the recovery room, and the ICU.
    e) Set the patient medical information, this function is responsible to set the data in the database and updating them when changed.
    f) Anesthesia confirmation function, this function process the data to give the doctor an indication of the proper decision in non-emergency conditions, but it can not be taken definitely since the doctor is the decision-maker after all. This function depends on medical conditions, to confirm the anesthetize: the food fasting hours should be minimum of 8 hours, the water fasting hours should be minimum of 2 hours, the mean blood pressure depending on a medical equation ((systolic blood pressure + 2 * diastolic blood pressure) / 3) should be more than or equal 60 and less than or equal 120, the oxygen saturation should be more or equal to 90%, the heart rate should be more than or equal to 60 and less than or equal to 100, finally, the patient consent must be true.
    g) Anesthetized, this function sets the patient status if got anesthetized.
    h) Operation confirmation, this function also processes the data to give the doctor an indication of the proper decision, which is confirmed if the patient is anesthetized.
    i) Operated, this function sets the patient status if the surgery is done.
    j) Patient stability, this function also processes the data to give the doctor an indication of the proper decision, it decides where the status of the patient is stable or not after surgery by certain conditions and then decides the after surgery location whether the ward or ICU. To decide if the patient is stable or not, the mean blood pressure should be more than 50 and less than 130, the heart rate more than 50, and less than 120, the oxygen saturation more than 88, and the surgery duration less than 4 hours.

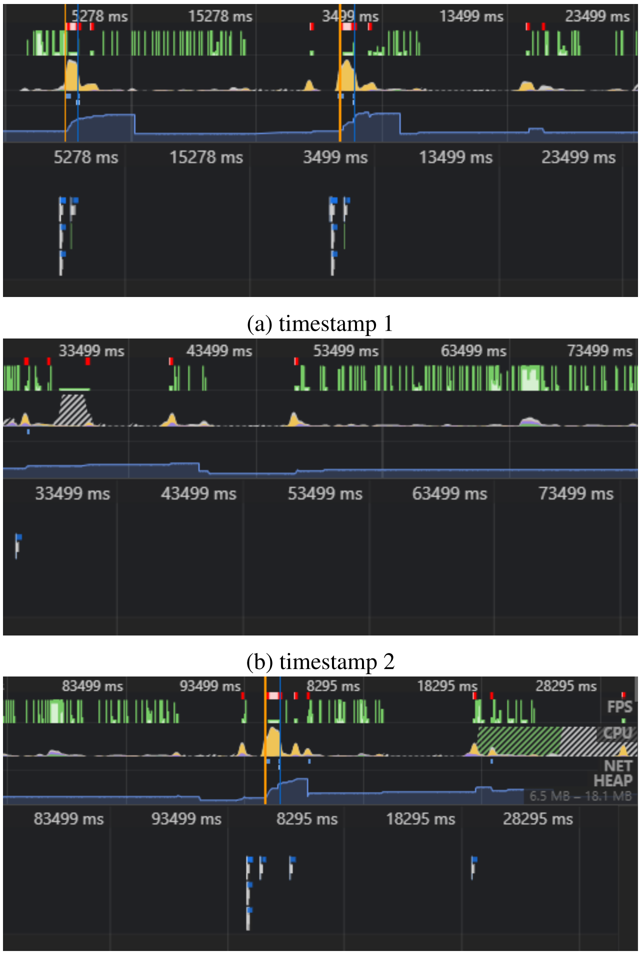
However patients' information is restricted by the global Electronic Health Record (EHR) standards. The phrase ''standards'' has a wide definition, and when referring to health information technology, it may appear that it refers to computer hardware technical criteria. The government's standards, on the other hand, apply to the software that these systems utilize. Our system is integrated with those standards, which is work with other systems and can be integrated with existing databases. Certification is a method of enforcing rules. Hospitals, doctors, and other qualifying professionals must utilize certified EHRs to obtain meaningful use incentive payments. After completing testing for functionality, dependability, security, and standard compliance, EHRs are certified. The data in blockchain as we mentioned are reliable, secure, and hard to manipulate.

IV. RESULTS AND DISCUSSIONS

In this chapter, we evaluate the main components in the system the proposed architecture, and the customized Blockchain. First, ERTCA performance is assessed and compared with the hierarchical architecture. Secondly, the efficiency of the customized private Ethereum-based Blockchain is evaluated. To achieve these experiments we build the system by a personal computer with a Core i7 CPU processor and 8 GB RAM. The rich client used all the resources of the PC, but the thin client was simulated by the settings of x6 slowdown in the CPU and using a slow 3G network. the result of each experiment is discussed in sections 1 and 2.

A. ERTCA EVALUATION

The main goal of this work is to establish an architecture that can handle the integration between IoT and Blockchain. As mentioned before, the integration suffers from the limited resource of IoT devices which can not handle the energy-intensive Blockchain networking and mining operations. The proposed ERTCA framework provides a solution to distribute the load in a way that IoT devices can tolerate. To evaluate the performance of the architecture, we compare its performance against a hierarchical architecture. In the hierarchical architecture, each system in this work will be considered as a layer, and each layer will have its own miners and smart contract to perform the needed function. In another word, there is no difference between a rich client and a thin client; because all the client was a Blockchain full node with their own smart contract definition. As shown in Figure (6 (a),(b),(c)) and Figure (7 (a),(b),(c)), the CPU and network performance at each system for the full cycle of

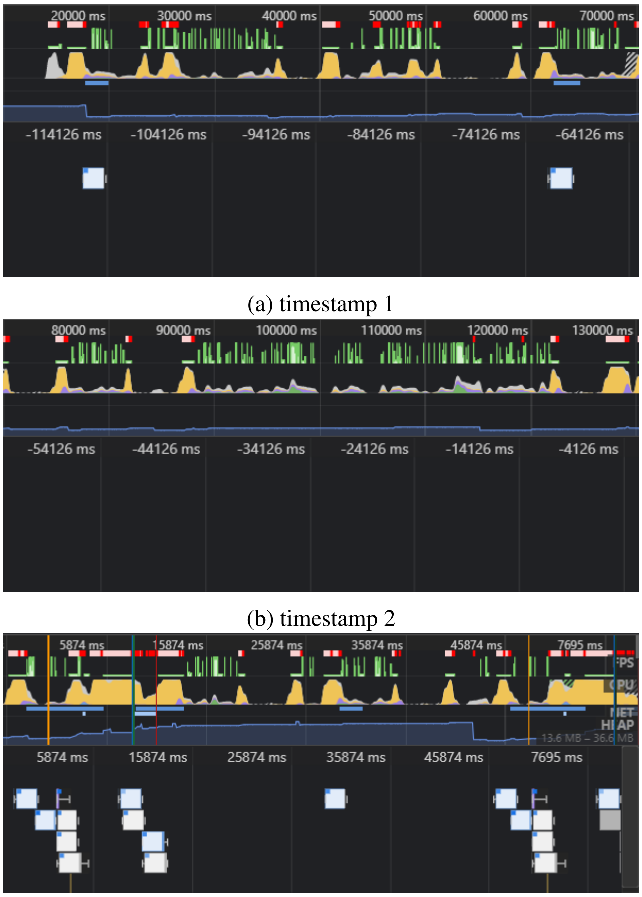


(a) timestamp 1

(b) timestamp 2

(c) timestamp 2

FIGURE 6. **ERTCA CPU and network performance timeline.**



(a) timestamp 1

(b) timestamp 2

(c) timestamp 2

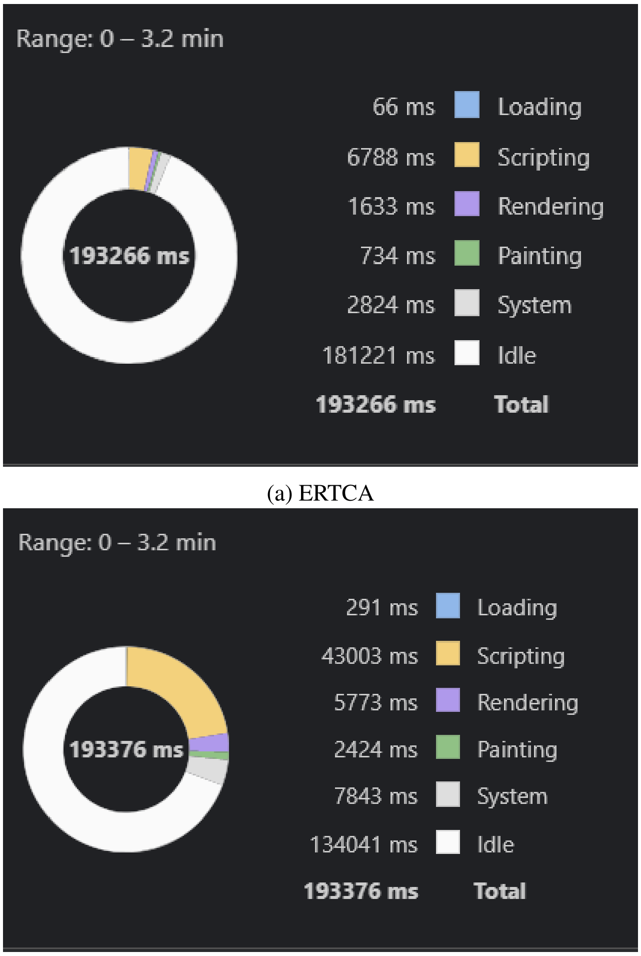
FIGURE 7. **Hierarchical architecture CPU and network performance timeline.**

one patient, as shown the hierarchical architecture clients has high peaks in CPU and network performance in all systems when ERTCA peaks were in rich clients devices performance. Moreover, Figure (8 (a),(b)) shows the summary performance of the process for each architecture, which can be noticed in the difference in system consumed time at the same range for each architecture, where the ERTCA system (CPU, GPU, and network) consumed time was 2824 ms and the hierarchical architecture system consumed time was 7843 ms, the difference was almost three times. Besides the values of the other metrics that the differences were almost four times, six times, three and half times, and three times, respectively. Figure (9) shows the analyzed data of the CPU time metrics where the difference can be seen clearly that ERTCA is more efficient than hierarchical architecture in our IoT-Blockchain application.

B. PRIVATE ETHEREUM-BASED BLOCKCHAIN EVALUATION

One of the main elements of the system is the Blockchain infrastructure. In order to evaluate the performance of the proposed private Ethereum-based Blockchain solution, we used Ganache Test Net, a testing blockchain environment maintained by Ethereum. This environment is available to perform tests in an emulated environment, which contains similar characteristics to the main Public Ethereum network. An important advantage of evaluating the solution in Ganache is that no financial investments are required, as Ganache provides a faucet to request Ethers to this testing network. It is important to notice that we did not perform an evaluation through the main Public Ethereum network due to the financial costs associated with it. As we know, Ethereum is a financial system with a cryptocurrency called ether. Ethereum uses the term gas, which is a unit that is paid as an amount of ether depending on the selected price by GEWI (one Nano of an Ether), to perform transactions. Depending on ether price (on 17th of May, 2021), the full process for each patient will cost 65.35 USD, which is considered a costly process.

Table (1) presents a qualitative discussion about different Ethereum options that can be used by the proposed solution. In this evaluation, Ganache had a mining time lower than 1 minute. However, in the main Public Ethereum network, mining time is higher than 5 minutes its may take a very



Range: 0 – 3.2 min

66 ms	Loading
6788 ms	Scripting
1633 ms	Rendering
734 ms	Painting
2824 ms	System
181221 ms	Idle
193266 ms	**Total**

(a) ERTCA

Range: 0 – 3.2 min

291 ms	Loading
43003 ms	Scripting
5773 ms	Rendering
2424 ms	Painting
7843 ms	System
134041 ms	Idle
193376 ms	**Total**

(b) Hierarchical

FIGURE 8. Hierarchical architecture and ERTCA performance summary analysis.

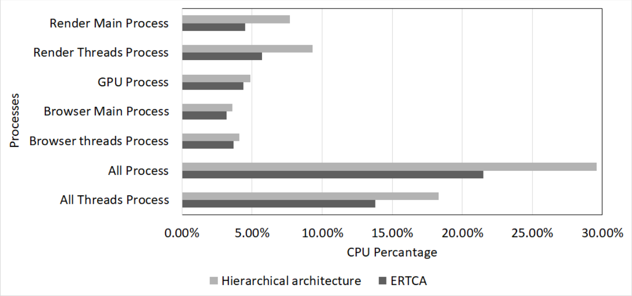


FIGURE 9. CPU execution time parentage of processes.

TABLE 1. A qualitative discussion about different Ethereum options for the proposed solution.

	Ethereum (Main)	Ganache
Mining Time	>5 minutes	<1 minute
Mining Difficulty	high	medium
Financial costs	yes (Ether)	no

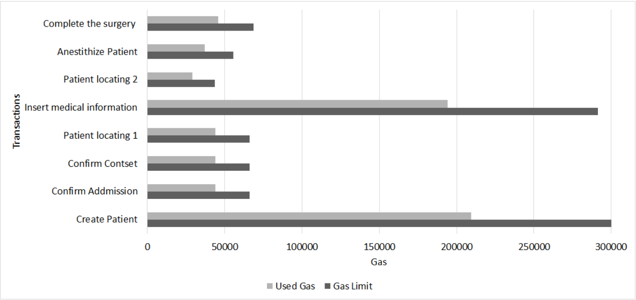


FIGURE 10. The needed time to execute smart contract functions by miners node.

long time or less depending on the network speed at the transaction computing time. It cannot be specified because of the high difficulty present in main Public Ethereum due to the dynamic difficulty increase over time, especially due to the high computing power of the miners.

One of the major elements in Ethereum is the smart contract. To evaluate the smart contract of our application on the created private Ethereum network, we used the Truffle test. The full cycle of the surgery management process for a patient consumes 5 seconds on mining and adding blocks to the Blockchain. Figure (10) shows each function performing time (in ms).

To evaluate the smart contract of our application on the implemented application is done by analyzing the amount of used gas by miners. The gas measures the amount of effort or activity to be taken. Any transaction or contract conducted in the Ethereum platform costs a particular quantity of gas and requires higher computer resources than actions using lesser computer resources. The reason gas is significant is that it helps guarantee that the transactions that are sent to the network pay an appropriate cost. By ensuring that a transaction pays with each operation it does, we ensure that the network does not become stuck with a great deal of work that is not worthwhile for anybody. Gas expenses occur in operations of the EVM, but the gas itself is also gwei measured. Every transaction stipulates the gas price that it will pay in gwei for each gas unit so that the market is able to determine the link between gwei pricing and computer cost (as measured in gas). Total gas utilized, together with the price paid for gas, leads to the total charge paid by the transaction. In our evaluation, we let the default settings of gas price as 20000000000 gwei and gas limit as 6721975. Table (2) shows the generated gas limit for each transaction according to the required computational power and the gas used by miners. Figure (11) shows the relationship between each transaction gas limit and the gas used by miners, which can be noticed in the efficiency of our Blockchain miners in executing the functions under the needed computational power.

C. DISCUSSION

The results indicate that the integration between IoT and Blockchain can be optimized by designing efficient

TABLE 2. The gas limit for each transaction and the used gas.

Transaction	Gas Limit	Used Gas
Create Patient	314079	209386
Confirm Admission	66178	44119
Confirm Patient Consent	66231	44154
Locating Patient 1	66034	44023
Insert Medical Information	291456	194304
Locating Patient 2	43534	29023
Anaesthetize Patient	55563	37042
Operate	68533	45689

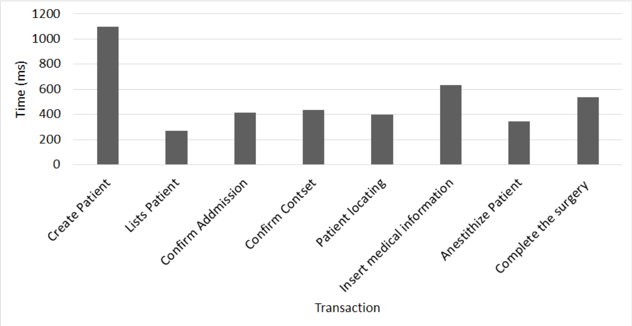


FIGURE 11. The relationship between each transaction gas limit and the used gas.

architecture. Our architecture provides an optimization solution for the lack of resources in most IoT applications by distributing the tasks according to the needed load depending on each device's capabilities, which can be seen in the performance results at the devices level and the used gas on the blockchain level, where all CPU and GPU processes 5% better than the hierarchical architecture, also the used gas in all transactions was less than the limit. Contrary to the hypothesized association that the integration between IoT and Blockchain can reduce the performance due to the low capabilities of IoT devices and networks, the result can show that integration can merge the advantages of Blockchain technology with IoT limited resources to provide a decentralized system with reliable data and ensure privacy and security.

These results should be taken into account when considering how to provide architecture support for the limitation in resources in merging Blockchain to your IoT application. ERTCA can be implemented in IoT applications that contain too many attached devices and stakeholders which each of them provides a certain service. For example, ERTCA can be implemented in a smart school application, where the thin clients can be students, and the rich clients can be an educational managing system for teachers with a smart contract to simulate the education functions and cases, for the second-level thin clients can be educational dump devices such as projectors, tablets, and smartboards.

Due to the lack of experiments on the scalability of the system, the results cannot confirm the scalability of the system. However, hypothetically the system can be scalable on the thin clients level; because of the simplicity of the tasks and the low latency. But at the rich clients level, it may suffer at some point due to the needed computational power and the storage for the Blockchain transactions record. Further research is needed to establish an architecture to handle the scalability of rich clients.

V. CONCLUSION AND FUTURE WORK

Health care is a recurring and important topic to society, as several advances, new techniques, activities, and methods are constantly emerging. A plethora of systems and applications for health care and health activities monitoring are also currently available. However, it is important to create methods to promote end-user adoption and usage, in special for a collaborative approach that can help to prevent healthcare management problems. Thus, this research presents a solution for a collaborative healthcare management system using IoT and Blockchain integration architecture, ERTCA, exemplifying the usability of this technology in order to improve the performance, resource utilization, privacy of data, and security of the system. In addition, we presented some benefits of using blockchain in private infrastructure or blockchain public networks. Also, it was observed that the financial costs on the main Ethereum Public network are higher when the number of transactions is also high. However, when choosing to use a private Blockchain system, the performance of infrastructure, privacy, and security must be considered. Finally, we can conclude that Blockchain and IoT can be integrated in a manner that could help to create a collaborative health monitoring system, as it makes the system safe by providing data immutability, ensuring that business logic is preserved and the possibility of gamification by completing managing health activities.

There are quite a few directions in the future work for this research. First, we intend to expand the tests and evaluate our solution with others IoT-Blockchain integration architecture such as modular architecture. To ensure the completeness of this work, comparing ERTCA with different integration architectures would highlight the advantages and indicate the disadvantages to help in the improvement of ERTCA. On the other hand, we intend to study the scalability of ERTCA and the security of such architecture by performing different experiments and extensive evaluations. Moreover, we intend to evaluate the Smart Contracts in different Blockchains than Ethereum, especially in Blockchains with different consensus algorithms, such as Hyperledger Fabric.

On other hand, we intend to test ERTCA on different application scenarios rather than health care to blow the soul to the architecture. But as we know how is health care is a vital field for us nowadays, we also would like to expand the system to include medicines, medical consultation data, and other activities regarding the management healthcare process.

REFERENCES

[1] K. Ashton, "That 'Internet of Things' thing," *RFID J.*, vol. 22, no. 7, pp. 97–114, 2009.

[2] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018.

[3] S. Nakamoto and A. Bitcoin. (Apr. 2008). *A Peer-to-Peer Electronic Cash System*. Bitcoin. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] R. Sujatha, C. Navaneethan, R. Kaluri, and S. Prasanna, "Optimized digital transformation in government services with blockchain," in *Blockchain Technology and Applications*. Boca Raton, FL, USA: Auerbach Publications, 2020, pp. 79–100.

[5] H. Sun, S. Hua, E. Zhou, B. Pi, J. Sun, and K. Yamashita, "Using ethereum blockchain in Internet of Things: A solution for electric vehicle battery refueling," in *Blockchain—ICBC* (Lecture Notes in Computer Science), vol. 10974, S. Chen, H. Wang, and L. J. Zhang, Eds. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-94478-4_1.

[6] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized BlockChain for IoT," in *Proc. 2nd Int. Conf. Internet Things Design Implement.*, Apr. 2017, pp. 173–178.

[7] L. Tseng, L. Wong, S. Otoum, M. Aloqaily, and J. B. Othman, "Blockchain for managing heterogeneous Internet of Things: A perspective architecture," *IEEE Netw.*, vol. 34, no. 1, pp. 16–23, Jan. 2020.

[8] Y. E. Oktian, S.-G. Lee, and H. J. Lee, "Hierarchical multi-blockchain architecture for scalable Internet of Things environment," *Electronics*, vol. 9, no. 6, p. 1050, Jun. 2020.

[9] K. Biswas and V. Muthukkumarasamy, "Securing smart cities using blockchain technology," in *Proc. IEEE 18th Int. Conf. High Perform. Comput. Communications; IEEE 14th Int. Conf. Smart City; IEEE 2nd Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2016, pp. 1392–1393.

[10] G. Sagirlar, B. Carminati, E. Ferrari, J. D. Sheehan, and E. Ragnoli, "Hybrid-IoT: Hybrid blockchain architecture for Internet of Things–PoW sub-blockchains," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2018, pp. 1007–1016.

[11] D. V. Medhane, A. K. Sangaiah, M. S. Hossain, G. Muhammad, and J. Wang, "Blockchain-enabled distributed security framework for next-generation IoT: An edge cloud and software-defined network-integrated approach," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6143–6149, Jul. 2020.

[12] H. Tenhunen and T. Westerlund, "Artificial intelligence at the edge in the blockchain of things," in *Proc. Int. Conf. Wireless Mobile Commun. Healthcare*, vol. 320. Dublin, Ireland, Nov. 2019, p. 267.

[13] R. Kaluri, D. S. Rajput, Q. Xin, K. Lakshmanna, S. Bhattacharya, T. R. Gadekallu, and P. K. R. Maddikunta, "Roughsets-based approach for predicting battery life in IoT," 2021, *arXiv:2102.06026*.

[14] L. Hang and D.-H. Kim, "Design and implementation of an integrated IoT blockchain platform for sensing data integrity," *Sensors*, vol. 19, no. 10, p. 2228, May 2019.

[15] V. Buterin. (Dec. 2020). *Ethereum Whitepaper*. [Online]. Available: https://ethereum.org/en/whitepaper/

[16] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, and S. Muralidharan, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, Apr. 2018, pp. 1–15.

[17] M. Samaniego and R. Deters, "Internet of smart Things–IoST: Using blockchain and CLIPS to make things autonomous," in *Proc. IEEE Int. Conf. Cognit. Comput. (ICCC)*, Jun. 2017, pp. 9–16.

[18] Litecoin Wiki Contributors, Litecoin Wiki. (May 8, 2019). *Main Page*. Accessed: Jan. 1, 2022. [Online]. Available: https://litecoin.info/index.php?title=Main_Page&oldid=98

[19] (2017). *The Lisk Protocol*. [Online]. Available: https://lisk.io/documentation/lisk-sdk/index.html

[20] (2016). *Quorum Whitepaper*. [Online]. Available: https://github.com/ConsenSys/quorum-docs/blob/master/QuorumWhitepaperv0.1.pdf

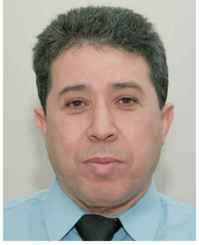
[21] *Chronicled: MediLedger Network*. Accessed: Jan. 1, 2022. [Online]. Available: https://www.chronicled.com/chronicled-resources

[22] B. Cheng, M. Wang, S. Zhao, Z. Zhai, D. Zhu, and J. Chen, "Situation-aware dynamic service coordination in an IoT environment," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2082–2095, Aug. 2017.

MARAH R. BATAINEH received the B.S. degree in computer science from Yarmouk University, in 2016, and the M.S. degree in computer science from the Jordan University of Science and Technology, in 2021. From 2017 to 2019, she was a Teaching and Research Assistance at the Jordan University of Science and Technology. Her research interests include the applications of the IoT, blockchain, artificial intelligence, and semantic web.

WAIL MARDINI received the master's degree from the University of New Brunswick, Canada, in 2001, and the Ph.D. degree in computer science from the University of Ottawa, Canada, in 2006. He has been working with the Jordan University of Science and Technology (JUST), Jordan, since 2006. He was the CS Department Chair during the academic years 2010–2011, 2011–2012, and 2019–2020, and the Faculty Vice Dean during the academic year 2012–2013. He is currently a Professor of computer science at JUST. He has many publications in the area of network survivability, wireless and wireless sensor networks, and optical-wireless networks. He is currently working on wireless mesh networks, wireless sensor networks, optical network survivability, WiMax technology, scheduling in parallel computing, and intrusion detection in database techniques.

YASER M. KHAMAYSEH (Member, IEEE) received the bachelor's degree from Yarmouk University, Irbid, Jordan, in 1998, the master's degree from the University of New Brunswick, Canada, in 2001, and the Ph.D. degree from the University of Alberta, Canada, in 2007, all in computer science. He joined the Jordan University of Science and Technology, in 2007, where he is currently an Associate Professor of computer science. He has more than 15 years of experience in research and teaching in the field of data communication and computer networks in different institutions, he worked at TR Laboratories for testing high speed network devices. He has more than 80 published articles in international journals and conferences. His research interests include simulation and modeling, wireless networks, performance evaluation, security, next generation internet, and evolutionary computation. He has received several awards.



MUNEER MASADEH BANI YASSEIN (Member, IEEE) received the Ph.D. degree in computer science from the University of Glasgow, U.K., in 2007. He served as the Chairperson of the Department of Computer Science, from 2008 to 2010, and the Vice Dean of the Faculty of Computer and Information Technology, from 2010 to 2012, from 2013 to 2014, and since 2018. He is currently a Professor of computer science with the Department of Computer Science, Jordan University of Science and Technology (JUST), Jordan. He is also conducting research in mobile *ad-hoc* networks, wireless sensor networks, cloud computing, simulation and modeling, and the Internet of Things. He has published over 170 technical papers in well reputed international journals and conferences. He is a member of the technical programs of several journals and conferences.