Scientific Research Publishing

Journal of Signal and Information Processing, 2022, 13, 1-23
https://www.scirp.org/journal/jsip
ISSN Online: 2159-4481
ISSN Print: 2159-4465

Design and Evaluation of a Distributed Security Framework for the Internet of Things

Kelechi G. Eze, Cajetan M. Akujuobi

The Center of Excellence for Communication Systems Technology Research (CECSTR), Department of Electrical Engineering, Prairie View A & M University, Prairie View, USA
Email: kelechigodwin9@gmail.com, cmakujuobi@pvamu.edu

How to cite this paper: Eze, K.G. and Akujuobi, C.M. (2022) Design and Evaluation of a Distributed Security Framework for the Internet of Things. *Journal of Signal and Information Processing*, **13**, 1-23.
https://doi.org/10.4236/jsip.2022.131001

Received: November 3, 2021
Accepted: February 25, 2022
Published: February 28, 2022

Abstract

The adopters of IoT face challenges with the surging Internet-based attacks on their IoT assets and inefficiencies within the technology. Unfortunately, IoT is overly distributed, still evolving and facing implementation and security challenges. Given the above scenario, we argue that the IoT network should always be decentralized design, and security should be built by design. The paper is the design and construction of a decentralized IoT security framework, with the goal of making emerging IoT systems more resilient to attacks and supporting complex communication and resource sharing. The framework improves efficiency and scalability in IoT, exposes vulnerable subsystems and components as possible weak links to system compromise, and meets the requirements of a heterogeneous computing environment. Other features of the framework including efficient resource sharing, fault tolerance, and distributed storage support the Internet of Things. We discuss the design requirements and carry out the implementation of Proof of Concept and evaluation of our framework. Two underlying technologies: the actor model and the blockchain were used for the implementation. Our reason for choosing the actor model and blockchain is to compare its suitability for IoT integration in parallel. Hence, evaluation of the system is performed based on computational and memory efficiency, security, and scalability. We conclude from the evaluations that the actor-based implementation has better scalability than the block-chain-based implementation. Also, the blockchain seems to be computationally more intensive than the actors and less suitable for IoT systems.

Keywords

Actor Model, Blockchain, Security Framework, Internet of Things

1. Introduction

Internet of Things (IoT) system or subsystem is a network of interconnected

physical or virtual objects that communicate to perform or enhance essential functions through sensing and communication over the Internet. IoT is growing in terms of number and use, with about 13% in 2014 to about 30% today [1] [2]. Accordingly, there are growing attacks on IoT networks which are concerning due to the share number of connected IoT devices and the large-scale impact of such attacks. The overall IoT devices worldwide are estimated to be about over 40 billion in 2023 [2] and this upward growth trend is expected to be there beyond 2023.

An IoT system or subsystem consists of four main essential layers of components: the sensing layer, the edge layer or gateway, the storage (local or cloud) and the analytics and visualization, as shown in **Figure 1**. These layers constitute different technology stacks and communication protocols with varying security requirements. As a network of heterogeneous devices, there are various possible design and configuration choices; however, there is increased attention towards integrating blockchain and IoT into a single network to take advantage of security and resiliency in blockchain protocol. This way, the limitations of client-server IoT networks like the single point of failure can be circumvented. However, due to inherent inefficiencies within the blockchain protocol, we also explore the actor model integration with the IoT for its security, distributed and lightweight nature.

We argue that a security framework built using distributed technology with security controls to meet the computational and communication requirements of IoT will support emerging and next-generation IoT. Our focus in this paper is

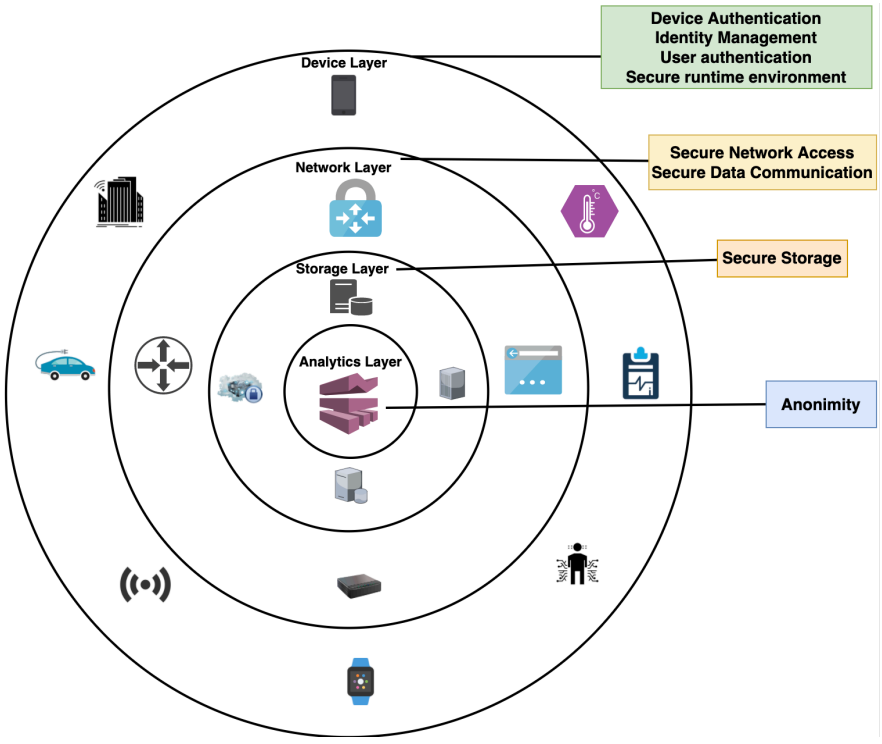

Figure 1. Layers of the IoT ecosystems.

the design and implementation of such a framework that ensures the best security and operational efficiency in IoT which aligns with the top requirements for IoT.

The framework supports efficient security mechanisms that meet standard security policies of Confidentiality, Integrity, and Availability (CIA) as well as other specialized IoT security requirements, as shown in **Figure 1**. Our solution framework takes into account several security requirements of IoT [3] to mitigate the weakness or vulnerabilities in IoT. The device layer supports authentication (device and user), identity management, and, a secure runtime environment. The network layer provides secure network access and secure data communication. The storage layer ensures secure storage, and the analytics layer preserves data privacy through anonymity or lightweight homomorphic encryption as the case may be. These are the primary security requirements in IoT [3], extending the standard policy of Confidentiality, Integrity, and Availability (CIA) [3] [4]. Our framework takes the computational and communication requirements of IoT into consideration, it allows IoT devices in a heterogeneous computing environment to share resources and communicate in a distributed fashion, thereby boosting latency, efficiency, and robustness of the network to failure of any kind. A solution like this has a low risk of attack, and there is no failure of the entire system due to threat or attack incidents.

IoT System components, devices, gateways, protocols, cloud, apps, and Application Programming Interfaces (APIs) as factors that contribute to system vulnerabilities are the first considerations. Another set of concerns is the system configurations (design) where given system components are selected and integrated optimally to reduce the likelihood of system-wide compromise. An optimally modeled and designed IoT solution should, therefore, take the considerations mentioned above into account to survive unpredictable attackers' behaviors to exploit vulnerabilities. In as much as some vulnerabilities cannot be eliminated, such as wireless jamming [5], the impact of such vulnerabilities can be controlled to reduce risks of such weaknesses to a bare minimum using an optimal model and design of the framework. Our goal in this paper is to design and evaluate such a framework. A novel design approach is presented for the framework under consideration. Furthermore, the topic of IoT and its security requirements is extensive and substantial work has been done around this topic. We recognize past works relevant to our focus in this paper.

In their work on embedded security framework for IoT, S. Barba *et al.* [3], carried out literature analysis and survey on embedded security as needed by the IoT, stressing the need for device-level security to be inbuilt in IoT. As a review, the work does not contain any implementation of the security framework, but bridges the gap in IoT security requirements and paved the way for the formal discussion of security need and security framework for IoT while taking into account computational, energy consumption, and memory impacts on the device. In their paper, Raman *et al.* [4] proposed a security framework to mitigate threats in IoT in a multi-layer sensor to the cloud ecosystem, taking into account

proper design and configuration. The work treated IoT security framework as a specification of security for a generic IoT sensor to cloud ecosystem with no real implementation of framework. The Industrial Internet of Things (IIoT) working group has put out a security framework for IIoT [6]. It provided a guide on best practices for existing and new implementations of IIoT solutions and tried to build a consensus among the heterogeneous IoT ecosystem [6]. The Industrial Internet of Things Security Framework is a technical guideline to implementing security in IoT encompassing design, architecture and security and hence does not involve a real implementation. [7] proposed a framework, IoT-HarPSecA that could be employed during the early phase of IoT design featuring three functions, namely security requirement gathering, best practice guideline for secure IoT development, and recommendation for suitable cryptographic algorithms for constrained devices. IoT-HarPSecA improves on existing security framework by automating security requirement elicitation and security recommendation as a guideline for best practice implementation. Though the tool is a real implementation that aids secure implementation of an IoT system, thus helping organizations improve security of their IoT products, there was no proof of concept implementation of a real IoT system to demonstrate the functionality of the tool developed. George *et al.* [8] leveraged the graph algorithm to model vulnerable relations in the IIoT to act as a security framework for risk assessment in the IIoT network and proposed risk mitigation strategies for improving the overall security of the system. An access control security framework for IoT services that is based on the publish-subscribe communication pattern was proposed by Duan *et al.* [9]. The proposed framework improves on the existing access control scheme to handle data confidentiality and service privacy issues. The work features a bi-directional policy matching strategy and a fully homomorphic encryption scheme for privacy and data confidentiality in large-scale IoT. A distributed prototype of access control secure publish/subscribe system was implemented and performance evaluation is carried out on metrics of latency and throughput. A software-defined Internet of Things (SD-IoT) framework was presented in Yin *et al.* [10] for detecting DDoS attacks using a cosine similarity algorithm, where the framework consists of an integration pool of SD-IoT controllers and SD-IoT switches with IoT gateway and IoT devices.

Kim *et al.* [11] [12] [13] were focused on authorization, authentication, and trust where distributed authorization, authentication, and trust were extensively analyzed, and a toolkit for the construction and deployment of this distributed authorization was proposed. Authorization and authentication are performed with local authorization entities that work on the principle of the actor model, hence scalable and lightweight to meet the needs of resource-constrained IoT devices. The work included security analysis (using security properties, threat model, and formal analysis) and scalability analysis. In [14], Sheron *et al.* worked on the design of a decentralized and scalable security framework for end-to-end security provision in IoT. The work used concepts from blockchain technology to achieve scalable features and security while considering optimization techniques such as hash pruning

to reduce computational complexity. A distributed security framework for the emerging IoT is presented by Medhane *et al.* [15] using blockchain technology, SDN, and edge cloud to fulfill the data confidentiality requirements of the IoT., Pacheco, *et al.* [16] presented an IoT security framework for smart infrastructures such as smart homes and intelligent buildings that features the security of the device layer, the network layer, the service layer, and the application layer. The study also performed security analysis for threat detection, behavioral analysis, and monitoring. Motivated by the rise in the attack on medical IoT devices, Alsubaei *et al.* [17] presented a modular web-based risk assessment framework for the Internet of Medical Things (IoMT-SAF) to measure security posture in the medical field as well as ranking. The framework takes factors such as stakeholders, solution type, architecture as input, processes them, and outputs recommendations of potential security issues with its attack categories. Casola *et al.* [18] proposed a three-step approach to automated threat modeling and risk assessment in IoT. The method performs a system model to identify the main assets to protect, followed by threat modeling to identify relevant threats and finally risk analysis and security controls identification.

Our approach extends on the existing literature by taking into account a design approach and technologies that guarantee a low risk of failure in IoT system in addition to having the ability to support various security mechanisms in place to mitigate emerging threats on IoT. Also, the framework is efficient, supporting various communication patterns such as publish and subscribe and resource sharing in a heterogeneous computing IoT environment. We have made use of real IoT devices and technologies to prototype the proposed framework.

1.1. Paper Contribution

We develop a decentralized IoT framework for IoT by extending the actor model and the blockchain. The framework is to be applicable to a wide variety of IoT settings that are energy-efficient, robust, secure, and scalable.

1) It supports distributed computing to meet the resource sharing needs of a heterogeneous computing environment like the IoT.

2) The framework takes into consideration essential design considerations, system configuration, and attacker behaviors to reduce the risk of failure due to attacks in complex IoT systems.

To show which method between actor model and blockchain is more aligned to IoT requirements, we carried out a comparative analysis of the two against their computational overhead, security, and scalability based on our framework.

1.2. Paper Organization

We have organized the remainder of the work as follows: Section 2 covers the discussion of the core technologies involved in the framework design and development and the framework architecture. Section 3 discusses the IoT system

design considerations that meet the security and operational requirements of IoT. Section 4 entails the discussion of the framework and evaluations. Section 5 summarizes and concludes the paper with some focus on future directions.

2. Blockchain and Actor Model

In modeling and designing IoT system, we argue that the distributed approach is the best for its resilience to faults and attacks, hence we extend the actor model and the blokchain to achieve the same. This solves the problems that come with centralized request/reply model, to bring in advantages such as privacy, efficiency, scalability, and security. Also the problem with single point of failure is solved with a system whose components can be connected and disconnected without disturbing the function of the overal system [19] [20] [21] [22]. Technologies like the blockchain and actor model have shown potentials in solving the problems enumerated above. These technologies eliminate the single point of failures in IoT [22] [23]. **Figure 2** is a schema of a decentralized IoT architecture using blockchain and the actor model.

We represent the IoT system as a network of components with the functionality to carry out sensing, communication, computation, security and storage. We refer to components in the actor model as actorsystems and refer to component in blockchain as the nodes.

2.1. Blockchain Approach

Blockchain is a technology with numerous application potentials within the IoT space and beyond. Blockchain is an implementation of decentralized ledger technology that can be defined as a peer-2-peer network of distrusted nodes with the ability communicate and transfer digital assets in a cryptographically protected (tamper-proof) fashion [24] [25] [26] by design. The blockchain is designed to be fault-tolerant and provides security against data corruption to all participating nodes using cascaded or recursive style encryption suitable for IoT security requirements [27] [28]. Ethereum and Hyperledger are two popular open-source

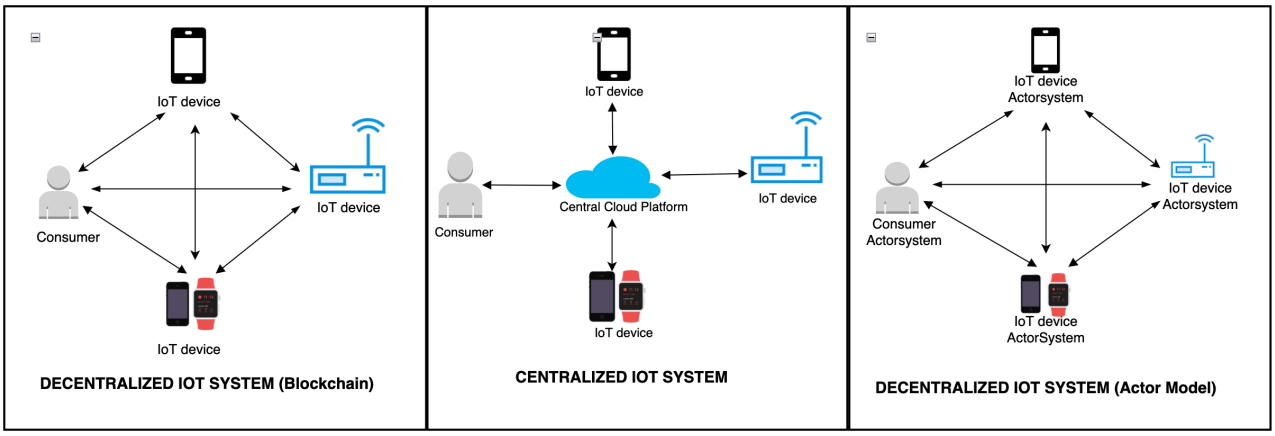

Figure 2. Architectures of IoT systems.

implementations of blockchain that have served as a suitable prototyping platform for implementing various security mechanisms for the IoT, such as access control, authentication, and authorization. Many other blockchain platforms work on the same basic principle of using a consensus engine or layer for validating transactions or communication among peer nodes and storing transactions in blocks linked cryptographical to create the notion of chains [29]. We present the blockchain as a potential secure platform for IoT security and highlight the challenges of applying the blockchain to realizing a secure IoT. The blockchain architecture in Figure 3 has been generalized to fit into the various blockchain solutions appropriate for IoT.

Generally, blockchain is transactional state machine with an initial state s_{t-1} and current state s_t. Where S is the set of states and T is the set of transactions, then:

$$\left(s_{t-1}, s_t\right) \in S \mid t \in \mathbb{R} \tag{1}$$

$$T_t \in T \mid t \in \mathbb{R} \tag{2}$$

Denoting the blockchain state transition function as τ and σ for storing and maintaining the states, then:

$$s_{t+1} \Rightarrow \tau\left(s_t, \sigma, T_t\right) \tag{3}$$

Transactions are packaged into blocks and chained using cryptographic hash functions as a way of secure proof by reference. We denote the set of blocks as B:

$$B_k \in B \mid k \in \mathbb{N} \tag{4}$$

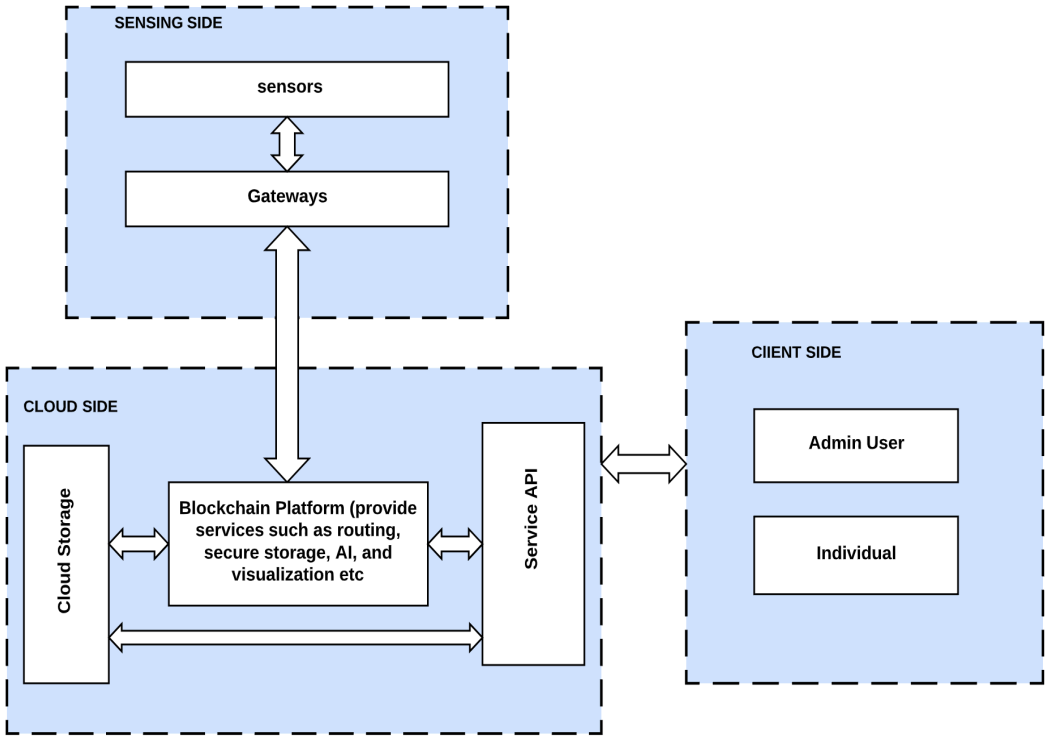

Figure 3. Blockchain model for IoT.

where k is the block number. For most blockchain $k=0$ or B_0 is called the genesis block.

$$B \equiv \left(T_0, T_1, \cdots, T_n\right) \tag{5}$$

where $n+1$ is the number of transaction contained in a block.

$$\forall B_k \in B, s_{t+1} \Rightarrow \Pi\left(s_t, \sigma, B_k\right) \tag{6}$$

where Π is the state transition function at T_n transaction or the block level, therefore:

$$\Pi\left(s_t, \sigma, B_k\right) \equiv \tau\left(s_t, \sigma, T_m \mid m = \sum_{i=0}^{n} i\right) \tag{7}$$

$$\Pi\left(s_{t+1}, \sigma, B_k\right) \equiv \Omega\left(B_k \tau\left(\tau\left(s_t, \sigma, T_0\right), T_1\right), \cdots\right), \tag{8}$$

where $t+1=T_n$ and Ω is the block finalization state transition function.

The above gives the basic building block of blockchains. However, blockchains have evolved and are pretty complex. We used Ethereum as a prototyping platform for the work. Ethereum supports smart contracts, thus making it possible for us to extend it to IoT. We modelled transactions (communication) and computations using smart contracts. Components of the system are modelled as classes while the functionalities of the components are modelled as functions. Nodes are then networked to realize the IoT system.

The model has been abstracted and simplified into three main parts: the sensing part, the user part and the cloud part that can be adapted to support numerous sensors and edge devices gateways (IoT devices). In the context of IoT, every node in the blockchain model is either an IoT device (a thing) or traditional computing device connected to the blockchain, forming a distributed secure network capable of sensing, collecting, processing, and storing data in a secure manner. As a heterogeneous network of systems, the things would be made up of many constrained devices combined with traditional computing devices such as servers, desktops, and gateways, working together to form a blockchain Integrated IoT system.

2.2. Actor Model Approach

The Actor Model is proposed by Bishop Hewitt together with Steiger in 1973 as a model of computation that handles a distributed computation task only by sending messages [29]. An actor is the unit of computation in an actorsystem. Actors communicate through message exchange. Actors, therefore, can send messages and react to a message. Primarily, an actor can react to a message: 1) create new actors; 2) send messages to other actors; or 3) determine future behavior for future messages. Actors can be terminated as well, thus having a lifecycle that can be controlled, a property that can be used to efficiently manage resources in actorsystems. The actor model considers developing IoT applications at higher abstractions to increase robustness in IoT systems.

The actor model lends itself well to the development of distributed systems for

solving the problem of complexity in designing IoT systems that meet the system design requirements of scalability, mobility, security, resource sharing, and data integrity [31] [32]. There are several implementations of the actor model using different programming languages to make actor style programming seamless such as Akka, C++ Actor Frame (CAF), Thespian, Pykka, cloud. The actor model solves the problem of scalability, transparency, and inconsistency in distributed communication environments and multi-agent systems [30] [31].

Among the actor implementations, some examples projects leverage the underlying principles of the actor model, such as asynchronous atomic callbacks (AAC) and discrete events (DE) semantics for developing an IoT framework that addresses the challenges of security and heterogeneity. There are also parameterized actors that communicate with one another under an actor runtime environment with time-stamped, discrete event semantics [12]. Accessors is a type of parameterized actors that are used for modeling and architecting IoT [12]. Actors are so powerful that they can be adapted to provide several services like authorization, scalability, and heterogeneity in IoT environments [12] [13]. In addition to security, as mentioned earlier in IoT requirements, the framework for IoT should provide guarantees for privacy, automated mutual authentication, intermittent connectivity, energy efficiency, and dynamic entity registration [13].

We extend the actor model to provide the communication, computation and security of the framework. The IoT devices (e.g., raspberry pi) are represented as an actorsystem with a set of primitives and behaviors. We represent a set of atom At to contain atomic values:

$$\forall a_t \in At, a_t \in \mathbb{N} \vee a_t \in \{\text{true}, \text{false}\} \tag{9}$$

And a set X of variables. B_n is a set of n-ary operations (behaviors) on At with cardinality m and P_n is set of actor primitives (operations of arity a) with cardinality n, where:

$$p_i \in P \,|\, 1 \le i \le n \tag{10}$$

$$b_j \in B \,|\, 1 \le j \le m \tag{11}$$

We then define G_n as a generic set of cardinality $(m+n)$ such that:

$$B_n \subset G_n \wedge P_n \subseteq G_n \tag{12}$$

where $B_n - G_n$ is possibility an empty set but $B_n - P_n \neq \varnothing$. Primitives as well as behavior are of the form method of an object or component of the actorsystem that enables it to carry out communication, computation and security functions. The classes encapsulating these primitives and behaviors are the actual components of the system.

3. Design Considerations of the Framework

Internet of Things (IoT) is a vast technology space that is currently being given increased attention among industrial and academic researchers. IoT systems are

still very fragmented, comprising millions of devices of different types (things) that work across tones of communication protocols to collect and transmit data for analysis on a cloud infrastructure [32] over an open, untrusted and hostile cyberspace [31]. Also, most IoT solutions are poorly architected, having a high risk of failure from attacks resulting from vulnerable components and poor design. The problem of IoT design, system complexity was considered in [30] as the authors propose Calvin, a new IoT application development approach that seamlessly approaches IoT application development using the actor paradigm. Although, in practice, most current IoT solutions follow the client-server model, it is prone to failures due to deficient fault tolerance. Motivated by the facts above about IoT, we take into account the following considerations in the design and implementation of the framework [33] [34] [35].

1) **Standards and Protocols:** Standards are a critical consideration in IoT framework design as many technologies from different manufacturers or vendors, including devices, protocols, and cloud, would be put together to deliver various solutions. Standards are essential to make different IoT devices and protocols work together efficiently, ensuring interoperability and security. The Standard making body, Open Connectivity Foundation (OCF) releases standard specifications to be followed by manufacturers.

Also, it uses an open-source platform IoTivity to create a middleware that allows devices to interoperate in the IoT ecosystem. Protocols, on the other hand, are rules followed by IoT devices to be able to talk to and exchange data with one another. Specialized protocols have been developed to support the requirements of IoT in terms of bandwidth, range of communication and power consumption). Existing protocols such as IP and HTTP are still used in IoT. Choosing the right protocol suite for IoT system design is a vital as these protocols as well as specialized IoT prorocols (**Figure 4**) must conform to industrial standards and be supported by the IoT devices. **Figure 5** shows leading protocols used in Industrial Internet of Things (IoT) [36]. Further analysis of these protocols is beyond the scope of this paper.

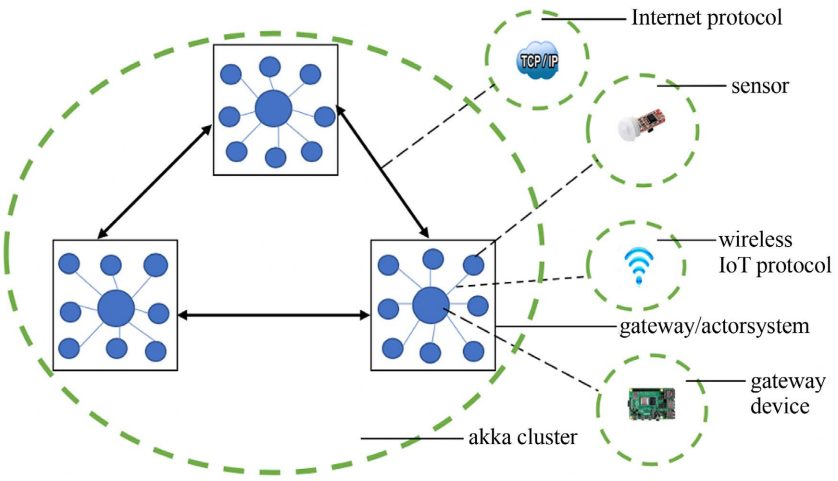

Figure 4. Simplified architecture of the actor based IoT system.

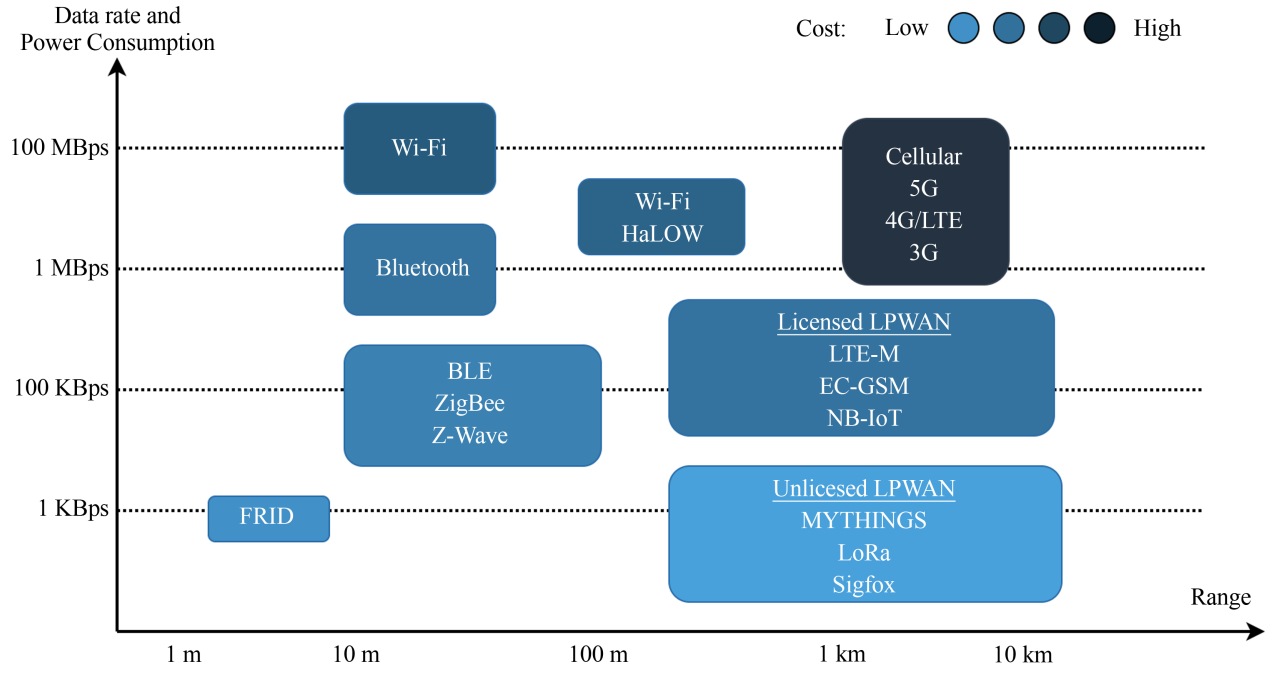

Figure 5. Different protocol standards in IoT [36].

2) **Latency and Throughput:** Latency and throughput are two figures of merit that are very important in IoT networks that a designer must consider. Latency in an IoT network is the time delay (seconds) between a sending node and a receiving node where a node is a thing on the network. Latency depends on the processing time and the travel time of a signal at a node. Throughput on the other hand measured in (bps) is the amount of data delivered over a network from point-to-point per second, *i.e.*, work done by the things in the network [35]. As IoT solution would be working in so many real-time application, very low latency are essential design consideration.

3) **Security:** Security is a critical consideration to be addressed in the IoT system design because IoT works over malicious environments. Security by design in IoT is achieved by first implementing threat modeling. Threat modeling is a proactive way of ensuring a sound security posture in IoT systems design by identifying and enumerating potential threats to prioritize mitigation mechanisms. Appropriate security mechanisms are then implemented according to the security requirement.

4) **Peer-To-Peer (P2P) Networks vs. Client-Servers:** How does your system communicate? P2P network is a type of architecture for distributed systems where machines on the network (peers) are both producers and consumers of resources and are equally privileged. In contrast, the traditional client-server network is a type of architecture where communication is to and from a central server. Due to the inherent nature of IoT and its requirement, a decentralized network topology is considered in this paper for fault-tolerance, low-latency, and increased network efficiency.

5) **Logging and Monitoring for Intrusion Detection:** Logging is a method of

collecting information about your system and having a service to manage these logs for easy debugging. Monitoring is a process of using tools to gain visibility into the health of your system. Monitoring tools use system logs to measure and display important system metrics. Monitoring is usually hooked up to Intrusion Detection System (IDS) to raise an alert if something malicious happens in your system. Our framework has logging as an essential feature.

6) **Publish and Subscribe Pattern:** Publish and Subscribe is a very important paradigm in a distributed system design. This paradigm consists of four entities: the publishers, the subscribers, the topics, and the messages. The publishers publish data to topics (a form of a database) that subscribers listen to receive the message. The framework supports publish and subscribe communication pattern that serves well in many IoT scenarios.

7) **Proxies for DDoS Prevention:** We consider proxy as an essential design consideration in this framework for the role they play in DDoS prevention, which has become widespread in IoT. There are two types of proxy: forward proxy and reverse proxy. A forward proxy is a server located between the client and the server or peers and communicates with the server on behalf of the client. Forward proxy conceals the identity of the client. In this way, clients can access restricted resources on the network. On the other hand, reverse proxy acts on behalf of the server such that a client communicates with a reverse proxy instead of the server. We consider any communicating peer or pair as a client-server pair or peers in our framework (which might be one to one or many to one). Reverse proxy conceals the identity of the server and could be used to filter out requests, caching, logging, and a load balancer. Reverse proxy as a load balancer can serve as a strategy for mitigating against DoS or DDoS attacks by distributing the request loads to prevent the servers from getting overwhelmed.

8) **Secure Storage:** IoT system design must consider a reliable way of storing and retrieving data. Questions such as where the data is written (disk or memory). How is the data recovered? Does your data store a centralized database or a distributed storage? If distributed, is your data consistent across the storage, what is the format of the data, is the data encrypted in storage, etc. Depending on the security need for your IoT application, you can decide the security needs of your data storage.

4. Proposed Framework and Evaluations

The framework architecture is the same for both the blockchain and actor model but with different configuration to reflect the underlying technology. For a Proof of Concept PoC, The proposed framework is set up using raspberry pis embedded with sensor and a MacBook computer to carry out the functionality as shown in **Figure 4** and **Figure 6**. As a very flexible and configurable framework as shown in **Figure 6** you can choose a suitable protocol at the perception layer for sensing and relaying sensor data to the gateway and decide on the messaging pattern to use in your specific design (peer-to-peer, or publish and subscribe model,

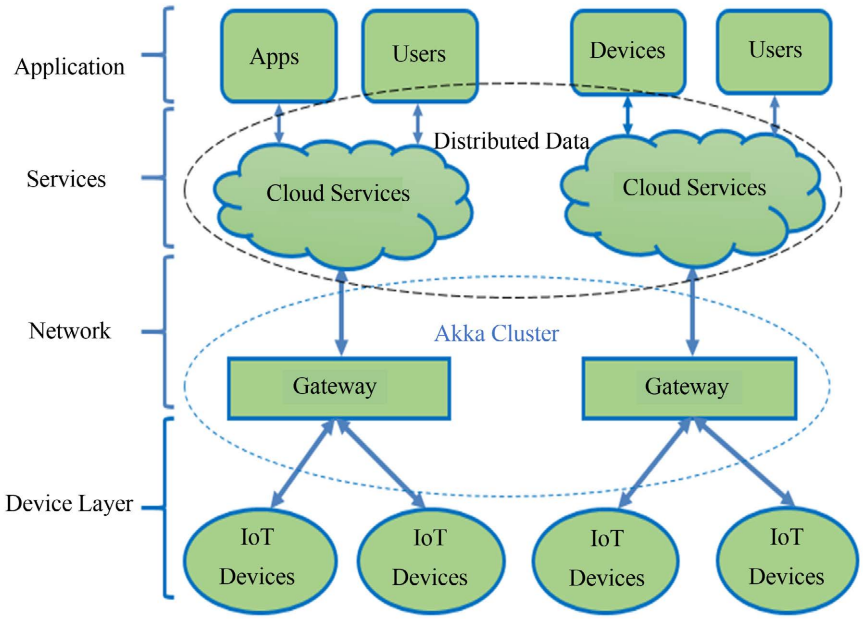

Figure 6. Basic components of the proposed framework.

etc.). Device layer security and secure protocol are therefore ensured at this layer. Devices at this layer are standalone sensors and or sensors embedded in devices. At the gateway level and beyond, processing communication and storage is done employing actors in a distributed fashion and supports standardized protocols TCP/IP which uses Transport Layer Security (SSL/TLS) in an optimal design as defaults to reduce overheads.

The framework evaluation was performed based on the computational power and the memory consumption of the two platform technologies discussed for achieving the security requirements of IoT. We consider the overhead impact of the model on our proposed solution and make a comparison on the energy efficiency at the gateway (cluster head) level, as this is where most of the security mechanisms reside. Other metrics used for comparison between blockchain and the actor model are security features and scalability.

4.1. Computational Overhead and Resource Efficiency

Blockchain is designed to be a distributed secure storage, communication, and computing system. It uses cryptography extensively in a way to make its data store resistant to modification. Also, computationally intensive operations are involved in validating transactions at the consensus layer in most blockchains. The actor model, on the other hand, is a model of computation for communication, processing, and storage using lightweight actors. An actor system within a machine creates an actor or pool of actors to perform various tasks such as sending messages, reacting to messages (processing or storing messages). Applying the proposed framework, we set up an actor model-based peer-to-peer communication environment using a gateway device in addition to other heterogeneous machines like PCs (Mac and Dell). Network configuration for the setup is shown in **Figure 7**.

(a)

```
akka {
  actor {
    # provider=remote is possible, but prefer cluster
    provider = cluster
    allow-java-serialization = on
  }
  remote {
    artery {
      ssl.config-ssl-engine {
        key-store = "/Users/kelechieze/KEYSTORE/mykeystore.jks"
        trust-store = "/Users/kelechieze/KEYSTORE/mytruststore.jks"
        //hostname-verification=on
        key-store-password = "securexxx"//${?SSL_KEY_STORE_PASSWORD}
        key-password = "securexxx"//${?SSL_KEY_PASSWORD}
        trust-store-password = "securexxx"//${?SSL_TRUST_STORE_PASSWORD}

        protocol = "TLSv1.2"

        enabled-algorithms = [TLS_DHE_RSA_WITH_AES_128_GCM_SHA256]
        #hostname-verification=on
      }
      transport = tls-tcp # See Selecting a transport below
      canonical.hostname = "192.168.1.103"
      #canonical.hostname = "10.124.14.15"
      #canonical.hostname = "192.168.1.89"
      canonical.port = 1212
    }
  }
}
```

(b)

Figure 7. Setup configuration for the framework. (a) Blockchain; (b) Actor model.

We measured the computational overhead on the gateway devices during regular peer to peer communication, as shown in Figure 8 and Figure 9. The same set of machines was used to set up a peer-to-peer communication with Ethereum blockchain, and the computational overhead on the gateway devices are shown in Figure 10 and Figure 11. From the data, it is clear that the blockchain network has more overhead than the proposed framework except for the single spike. The single spike on the energy consumption occurred during TCP handshake for both gateways on the proposed framework.

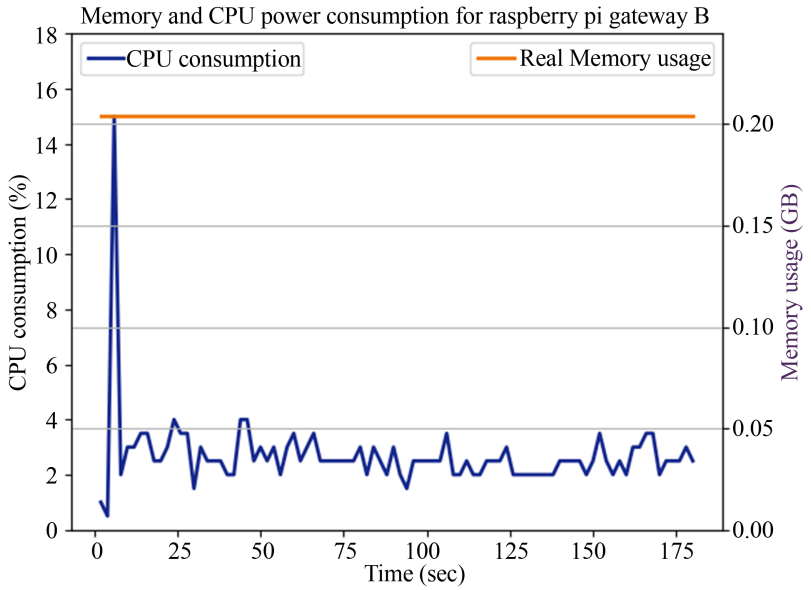

Figure 8. Computational overhead (CPU and memory consumption) for gateway devices on the proposed framework.

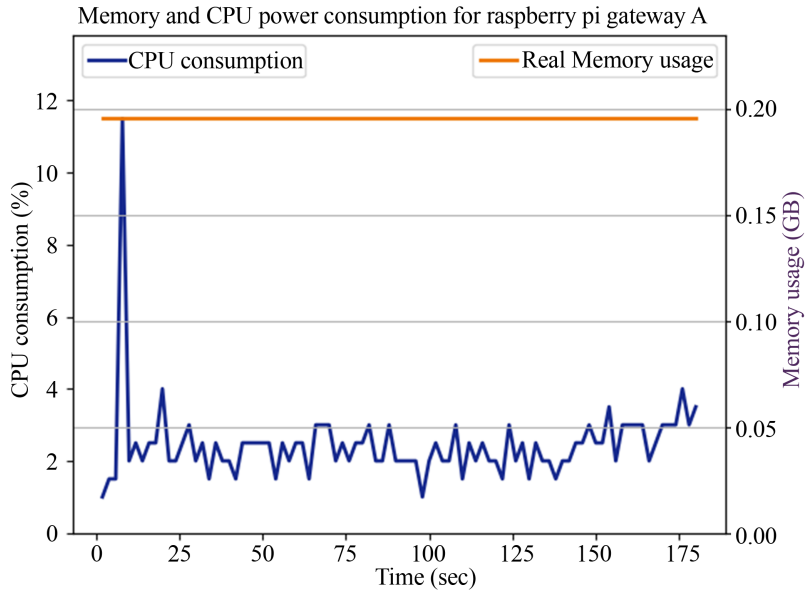

Figure 9. Computational overhead (CPU and memory consumption) for gateway devices on the proposed framework.

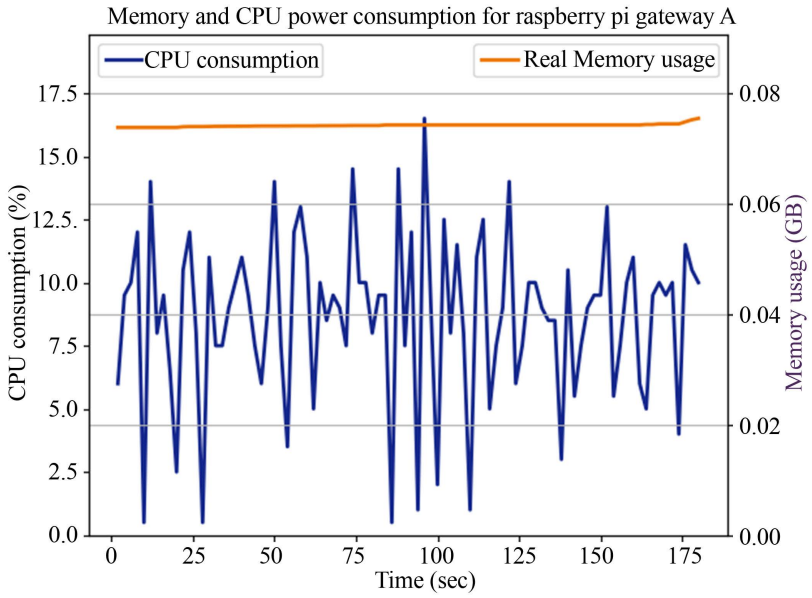

Figure 10. Computational overhead (CPU and memory consumption) for gateway devices on proposed framework using blockchain.

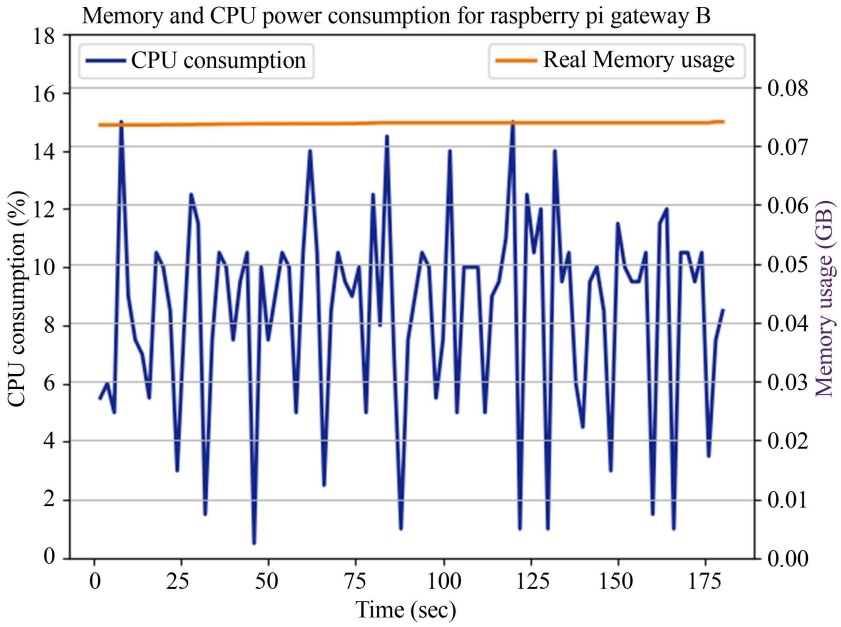

Figure 11. Computational overhead (CPU and memory consumption) for gateway devices on proposed framework using blockchain.

4.2. Security

The blockchain and the actor framework approach security differently. The blockchain is a configurable or programmable application platform with inbuilt security. At the same time, the actor model is implemented using toolkits that allow you to develop actor-based distributed applications the way you want. Blockchain security is based entirely on cryptography and uses distributed trust where no single authority validates transactions [13].

Therefore, the blockchain imposes security by default (design), whereas the actor model toolkit (Akka) allows you to configure security as per your application needs. The Actors model supports authentication, authorization, and encryption for processing, storage, and communication. These are significant features of the proposed framework that makes it very attractive in a decentralized IoT and cloud environments. Algorithm 1 and Algorithm 2 show authentication in an actor system and blockchain, respectively.

4.3. Scalability

Scalability is one of the top goals for embracing distributed peer-to-peer architecture such as the blockchain and the actor model over the traditional client-server model. This is very important to support the explosion trend in heterogeneous IoT devices connected to the Internet. A framework to support the challenge faced by this explosion and widespread adoption of the IoT must, therefore, be scalable. We have proposed such a framework on top of the actor model

Require: $username, password, authenticatingActor$
Ensure: $success, failure$
  $authenticate \leftarrow authservice(username, password)$
  if $authenticate$ is called **then**
    $worker \leftarrow createActor()$
    $token \leftarrow generateToken()$
    send token to authenticatingActor
  end if
  if $token$ is valid **then**
    success!, can do work
  else
    failure!, notifies requesting actor
  end if

Algorithm 1. Authentication with actors.

Require: $sender'skeys, receiver'skeys$
Ensure: $authorized, denied$
  $transaction \leftarrow createTransaction()$
  $signedTransaction \leftarrow signTransaction(transaction, key)$
  $cipherInstance \leftarrow getInstance(\text{“}AES\text{”})$
  $aescipher.init(Cipher.ENCRYPT\_MODE, key)$
  if $aesCipher.doFinal(transaction)$ is invoked **then**
    generates encryptedTransaction
    send encryptedTxHash to blockchain network
    receiver receives TxHash from the blockchain
  end if
  if $receiver$ address matched the intended recipient **then**
    $cipherInstance \leftarrow getInstance(\text{“}AES\text{”})$
    $aes.cipher.init(Cipher.DECRYPT\_MODE, key$
    $aesCipher.doFinal(byteCiphertext)$
  else
    access denied
  end if

Algorithm 2. Authentication with blockchain.

using the akka implementation, and comparison with the blockchain framework is hereby presented. The fundamental unit of computation in the actor model is an actor that incorporates storage, processing, and communication. Actors are created from actorsystems and can send messages (communicate) with actors within actorsystems in the same machine or remote machines. Using the right networking configuration, IoT can be scaled up and out terrifically. As scalability is a critical requirement of IoT, the actor framework supports scaling out (or horizontal scaling) using remoting/cluster features as well as scaling up (or vertical scaling) using actor pools. This gives room for various performance tuning options in a distributed and heterogeneous IoT environment using the proposed framework.

In the evaluation performed using a remoting/cluster set up, an actor manages communication, processing and storage need of every sensor that communicates with the raspberry pie gateway. The number of actors is equivalent to the number of IoT devices (sensors) that communicates through the gateway. It was found that the raspberry pie (gateway) scales very well as the number of messages from sensors increases by adding more actors to handle the increased messages as shown in **Figure 12** and **Figure 13**.

Conversely, the blockchain has issues with the ability to scale, which is currently an open research issue with various solution approaches [37] [38] while the actor model shine for modeling and developing distributed systems and services that scales really well [39]. These issues include the limited block size and the current consensus mechanism popular in blockchain [40] [41]. Also, other factors that influence the blocksize are the input size (operational complexity) and the gas limit of a given transaction. From an experiment conducted, it is determined that input size (bytes), blocksize (byte), and gas (Wei) obeys a linear relationship in that increase in one quantity leads to a rise in the others which soon riches maximum limits and limits scalability as shown in **Figure 14**. Other scalability issues observed are the storage as there is a lot of wasted computational power and storage overhead in the blockchain. Solutions existing on the blockchain research space for

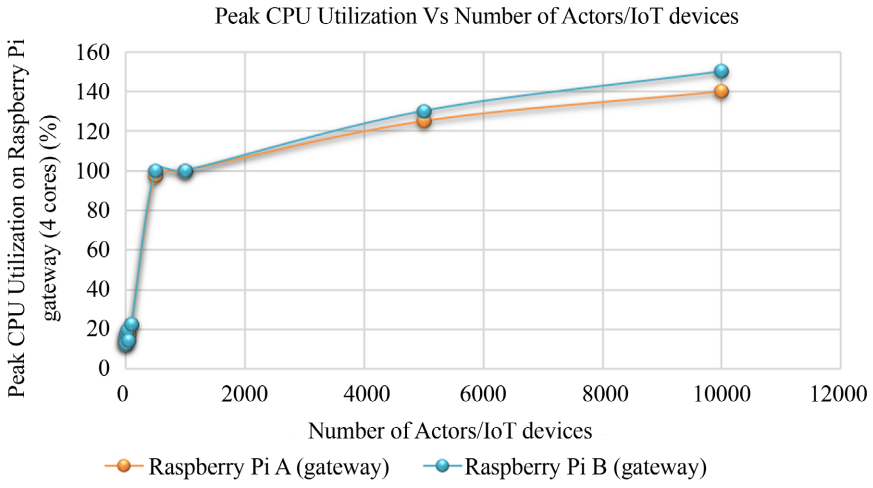

Figure 12. Vertical scaling with actors.

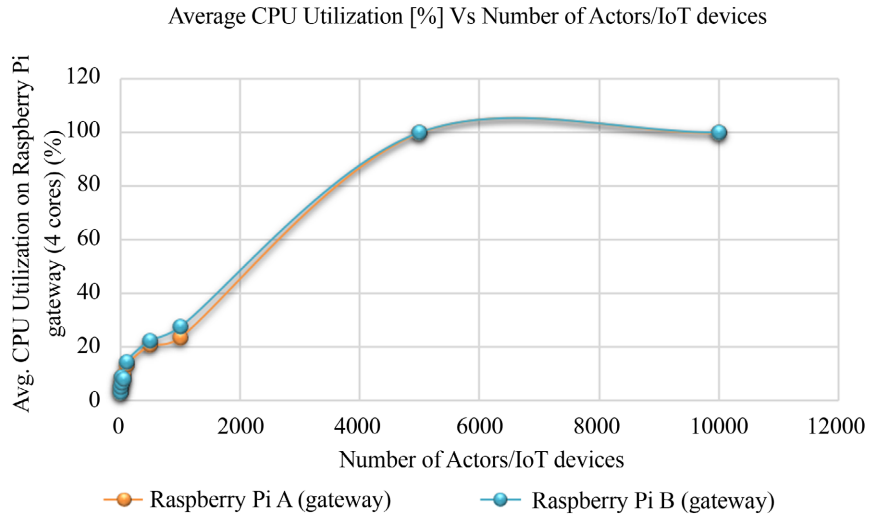

Figure 13. Vertical scaling with actors.

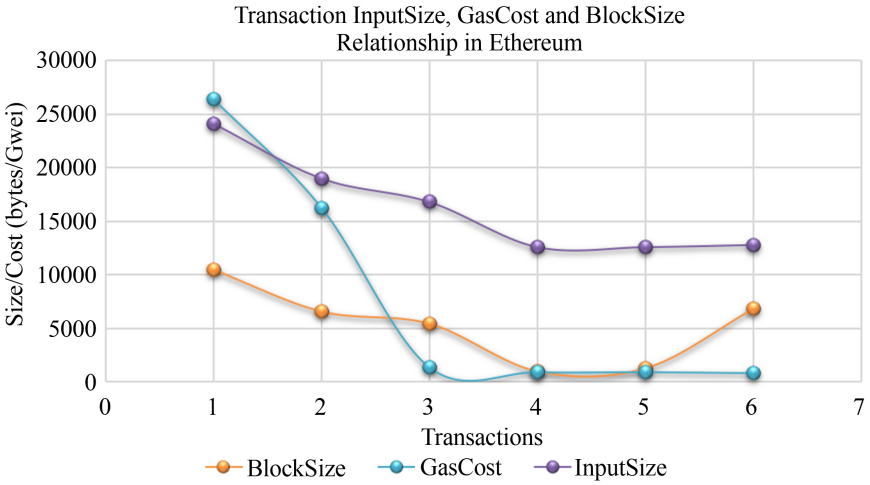

Figure 14. Graph of the relationship of input size, gas cost, and block size.

solving the scalability issues in the blockchain technology have its problems limiting its applicability [40]. Examples of these solutions are Lightning networks, where the transaction is sped up by reusing network channels for transaction and sharding, which is breaking down the blockchain into independent units called shards, having its state and history to reduce the load on the original blockchain.

5. Conclusions

The Internet of Things (IoT) environment that is made up of a large number of heterogeneous nodes needs a security framework that provides security and meets the scalability and efficiency requirements of IoT. In this work, security by design approach and a distributed framework is proposed. Focused on security and efficiency we chose the blockchain and actor model as the underlying technology for our framework and carried out a comparative analysis. Design consideration

for achieving an optimal design of the security framework for the IoT includes a choice of standards and protocols, latency and throughput of the machine to machine communication, the security of the framework, communication pattern (p2p versus client-server), monitoring unit for IDS, publish and subscribe pattern, proxies and secure storage. Evaluation of the framework was done against blockchain (Ethereum) using three figures of merits, computational and memory overhead, security, and scalability on a distributed network sample experimental setup consisting of 2 gateways (raspberry pies) and 2 PCs (MacBook and Dell). The computational and energy performance of the actor-based framework was better than the blockchain-based framework in both memory and computational power. The framework supports authorization, authentication, and data encryption using various cryptographic primitives and allows implementation as per need. Similarly, blockchain is secure and distributed; however, the security of the blockchain is by design, thereby limiting the flexibility of its configuration to suit a non-crypto currency environment such as a heterogeneous IoT environment. The framework supports horizontal scaling using remoting and cluster and vertical scaling using the actor pool (measuring about 400 bytes per actor).

It is obvious that the mechanism of operation of the blockchain and the actor model is different, relying on various protocols for member interactions; also both are distributed systems with different computational overhead and resource efficiency as well as security. There are many other features supported by the framework that are beyond the scope of this paper such as publish and subscribe communication pattern, distributed computing (resources sharing), distributed storage under the umbrella of the security supported by the framework. Other future directions and the analysis of the IoT protocols based on their security strengths.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Liu, X., Zhao, M., Li, S., Zhang, F. and Trappe, W. (2017) A Security Framework for the Internet of Things in the Future Internet Architecture. *Future Internet*, **9**, Article No. 27. https://doi.org/10.3390/fi9030027

[2] Dahlqvist, F., Patel, M., Rajko, A. and Shulman, J. (2019, July 22) Growing Opportunities in the Internet of Things. https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/growing-opportunities-in-the-internet-of-things

[3] Babar, S., Stango, A., Prasad, N., Sen, J. and Prasad, R. (2011) Proposed Embedded Security Framework for Internet of Things (IoT). *Proceedings of the International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology* (*Wireless VITAE*), Chennai, 28 February-3 March 2011, 1-5. https://doi.org/10.1109/WIRELESSVITAE.2011.5940923

[4] Rahman, A.F.A, Daud, M. and Mohamadr, M.Z. (2016) Securing Sensor to Cloud Ecosystem Using Internet of Things (IoT) Security Framework. *Proceedings of the In-*

ternational Conference on Internet of things and Cloud Computing, Cambridge, 22-23 March 2016, Article No. 79. https://doi.org/10.1145/2896387.2906198

[5] Mohsin, M., Sardar, M.U., Hasan, O. and Anwar, Z. (2017) IoT Risk Analyzer: A Probabilistic Model Checking Based Framework for Formal Risk Analytics of the Internet of Things. *IEEE Access*, **5**, 5494-5505. https://doi.org/10.1109/ACCESS.2017.2696031

[6] Inustrial Internet Consortium (2016) Industrial Internet of Things Volume G4: Security Framework. https://www.iiconsortium.org/pdf/IIC_PUB_G4_V1.00_PB-3.pdf

[7] Samaila, M.G., Sequeiros, J.B.F., Simões, T., Freire, M.M. and Inácio, P.R.M. (2020) IoT-HarPSecA: A Framework and Roadmap for Secure Design and Development of Devices and Applications in the IoT Space. *IEEE Access*, **8**, 16462-16494. https://doi.org/10.1109/ACCESS.2020.2965925

[8] George, G. and Thampi, S.M. (2018) A Graph-Based Security Framework for Securing Industrial IoT Networks from Vulnerability Exploitations. *IEEE Access*, **8**, 43586-43601. https://doi.org/10.1109/ACCESS.2018.2863244

[9] Duan, L., Sun, C., Zhang, Y., Ni, W. and Chen, J. (2019) A Comprehensive Security Framework for Publish/Subscribe-Based IoT Services Communication. *IEEE Access*, **7**, 25989-26001. https://doi.org/10.1109/ACCESS.2019.2899076

[10] Yin, D., Zhang, L. and Yang, K. (2018) A DDoS Attack Detection and Mitigation with Software-Defined Internet of Things Framework. *IEEE Access*, **6**, 24694-24705. https://doi.org/10.1109/ACCESS.2018.2831284

[11] Kim, H., Kang, E., Lee, E.A. and Broman, D. (2017) A Toolkit for Construction of Authorization Service Infrastructure for the Internet of Things. *Proceedings of the IEEE/ACM* 2*nd International Conference on Internet-of-Things Design and Implementation* (*IoTDI*), Pittsburgh, 18-21 April 2017, 147-158. https://doi.org/10.1145/3054977.3054980

[12] Kim, H., Wasicek, A., Mehne, B. and Lee, E.A. (2016) A Secure Network Architecture for the Internet of Things Based on Local Authorization Entities. *Proceedings of the IEEE* 4*th International Conference on Future Internet of Things and Cloud* (*FiCloud*), Vienna, 22-24 August 2016, 114-122. https://doi.org/10.1109/FiCloud.2016.24

[13] Kim, H. and Lee, E.A. (2017) Authentication and Authorization for the Internet of Things. *IT Professional*, **19**, 27-33. https://doi.org/10.1109/MITP.2017.3680960

[14] Sheron, P.S.F., Sridhar, K.P., Baskar, S. and Shakeel, P.M. (2019) A Decentralized Scalable Security Framework for End-to-End Authentication of Future IoT Communication. *Transactions on Emerging Telecommunication Technology*, **31**, Article No. e3815. https://doi.org/10.1002/ett.3815

[15] Medhane, D.V., Sangaiah, A.K., Hossain, M.S., Muhammad, G. and Wang, J. (2020) Blockchain-Enabled Distributed Security Framework for Next Generation IoT: An Edge-Cloud and Software Defined Network Integrated Approach. *IEEE Internet of Things Journal*, **7**, 6143-6149. https://doi.org/10.1109/JIOT.2020.2977196

[16] Pacheco, J. and Hariri, S. (2016) IoT Security Framework for Smart Cyber Infrastructures. *Proceeding of the IEEE* 1*st International Workshops on Foundations and Applications of Self* Systems* (*FAS*W*), Augsburg, 12-16 September 2016, 242-247. https://doi.org/10.1109/FAS-W.2016.58

[17] Alsubaei, F., Abuhussein, A., Shandilya, V. and Shiva, S. (2019) IoMT-SAF: Internet of Medical Things Security Assessment Framework. *Internet of Things*, **8**, Article No. 100123. https://doi.org/10.1016/j.iot.2019.100123

[18] Casola, V., De Benedictis, A., Rak, M. and Villano, U. (2019) Toward the Automation

of Threat Modeling and Risk Assessment in IoT Systems. *Internet of Things*, **7**, Article No. 100056. https://doi.org/10.1016/j.iot.2019.100056

[19] Sharma, P.K., Rathore, S., Jeong, Y. and Park, J.H. (2018) SoftEdgeNet: SDN Based Energy-Efficient Distributed Network Architecture for Edge Computing. *IEEE Communications Magazine*, **56**, 104-111. https://doi.org/10.1109/MCOM.2018.1700822

[20] Tian, F. (2017) A Supply Chain Traceability System for Food Safety Based on HACCP, Blockchain Internet of Things. *Proceedings of the International Conference on Service Systems and Service Management*, Dalian, 16-18 June 2017, 1-6.

[21] Nair, G.R. and Sebastian, S. (2017) BlockChain Technology Centralised Ledger to Distributed Ledger. *International Research Journal for engineering and Technology*, **4**, 2823-2827.

[22] Puthal, D., Malik, N., Mohanty, S.P., Kougianos, E. and Yang, C. (2018) The Blockchain as a Decentralized Security Framework [Future Directions]. *IEEE Consumer Electronics Magazine*, **7**, 18-21. https://doi.org/10.1109/MCE.2017.2776459

[23] Wei, L., Liu, S., Wu, J. and Long, C. (2019) Enabling Distributed and Trusted IoT Systems with Blockchain Technology. *IEEE Blockchain Technical Briefs*, January 2019.

[24] Okada, H., Yamasaki, S. and Bracamonte, V. (2017) Proposed Classification of Blockchains Based on Authority and Incentive Dimensions. *Proceedings of the IEEE 19th International Conference on Advanced Communication Technology* (*ICACT*), Pyeong-Chang, 19-22 February 2017, 593-597. https://doi.org/10.23919/ICACT.2017.7890159

[25] Castellanos, J.A.F., Coll-Mayor, D. and Notholt, J.A. (2017) Cryptocurrency as Guarantees of Origin: Simulating a Green Certificate Market with the Ethereum Blockchain. *Proceedings of the* 5*th IEEE International Conf. on Smart Energy Grid Engineering*, Oshawa, 14-17 August 2017, 367-372. https://doi.org/10.1109/SEGE.2017.8052827

[26] Dorri, A., Kanhere, S.S. and Jurdak, R. (2017) Towards an Optimized BlockChain for IoT. *Proceedings of the IEEE/ACM* 2*nd International Conference on Internet-of-Things Design and Implementation* (*IoTDI*), Pittsburgh, 18-21 April 2017, 173-178. https://doi.org/10.1145/3054977.3055003

[27] Almadhoun, R., Kadadha, M., Alhemeiri, M., Alshehhi, M. and Salah, K. (2018) A User Authentication Scheme of IoT Devices Using Blockchain-Enabled Fog Nodes. *Proceedings of the IEEE/ACS* 15*th International Conference on Computer Systems and Applications* (*AICCSA*), Aqaba, 28 October-1 November 2018, 1-8. https://doi.org/10.1109/AICCSA.2018.8612856

[28] Samaniego, M. and Deters, R. (2016) Hosting Virtual IoT Resources on Edge-Hosts with Blockchain. *Proceedings of the IEEE International Conference on Computer and Information Technology* (*CIT*), Nadi, 8-10 December 2016, 116-119. https://doi.org/10.1109/CIT.2016.71

[29] Pahl, C., El Ioini, N. and Helmer, S. (2018) A Decision Framework for Blockchain Platforms for IoT and Edge Computing. *Proceedings of the International Conference on Internet of Things*, *Big Data and Security*, Funchal, 19-21 March 2018, 105-113. https://doi.org/10.5220/0006688601050113

[30] Hewitt, C., Bishop, P. and Steiger, R. (1973) A Universal Modular ACTOR Formalism for Artificial Intelligence. *Proceedings of the* 3*rd International Joint Conference on Artificial Intelligence*, Stanford, 20-23 August 1973, 235-245.

[31] Persson, P. and Angelsmark, O. (2015) Calvin-Merging Cloud and IoT. *Procedia Computer Science*, **52**, 210-217. https://doi.org/10.1016/j.procs.2015.05.059

[32] Brooks, C., Jerad, C., Kim, H., Lee, E.A., Lohstroh, M., Nouvelletz, V., Osyk, B. and Weber, M. (2018) A Component Architecture for the Internet of Things. *Proceedings of the IEEE*, **106**, 1527-1542. https://doi.org/10.1109/JPROC.2018.2812598

[33] Al-Twajre, B.A. (2019) Performance Analysis of Messages Queue in the Different Actor System Implementation. *Proceedings of the* 11*th International Scientific and Practical Conference on Electronics and Information Technologies* (*ELIT*), Lviv, 16-18 September 2019, 127-131. https://doi.org/10.1109/ELIT.2019.8892329

[34] Wikipedia (n.d.) Actor Model. https://en.wikipedia.org/wiki/Actor_model

[35] Grochowski, E., Ronen, R., Shen, J. and Wang, H. (2004) Best of Both Latency and Throughput. *Proceedings IEEE International Conference on Computer Design*: *VLSI in Computers and Processors*, San Jose, 11-13 October 2004, 236-243. https://doi.org/10.1109/ICCD.2004.1347928

[36] Behrtech (n.d.) 6 Leading Types of IoT Wireless Tech and Their Best Use Cases. https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/

[37] Eze, K.G., Akujuobi, C.M., Sadiku, M.N.O., Chouikha, M. and Alam, S. (2019) Internet of Things and Blockchain Integration: Use Cases and Implementation Challenges. *Proceedings of the International Conference on Business Information Systems*, Seville, 26-28 June 2019, 287-298. https://doi.org/10.1007/978-3-030-36691-9_25

[38] Dang, H., Dinh, T.T., Loghin, D., Chang, E., Lin, Q. and Ooi, B.C. (2019) Towards Scaling Blockchain Systems via Sharding. *Proceedings of the ACM International Conference on Management of Data* (*SIGMOD*'19), Amsterdam, 30 June-5 July 2019, 123-140. https://doi.org/10.1145/3299869.3319889

[39] Abdelmoamen, A. and Jamali, N. (2018) A Model for Representing Mobile Distributed Sensing-Based Services. Proceeding of the IEEE International Conference on Services Computing (SCC), California, 2-7 July 2018, 282-286. https://doi.org/10.1109/SCC.2018.00049

[40] Chauhan, A., Malviya, O.P., Verma, M. and Mor, T.S. (2018) Blockchain and Scalability. *Proceedings of the IEEE International Conference on Software Quality*, *Reliability and Security Companion* (*QRS-C*), Lisbon, 16-20 July 2018, 122-128. https://doi.org/10.1109/QRS-C.2018.00034

[41] Rosales, R., Guibene, W. and Garcia, G.F. (2018) Actor-Oriented Design Patterns for Performance Modeling of Wireless Communications in Cyber-physical Systems. *Proceedings of the* 14*th ACM International Symposium on QoS and Security for Wireless and Mobile Networks* (*Q2SWinet*'18), Montreal, 28 October-2 November 2018, 29-38. https://doi.org/10.1145/3267129.3267136