Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

IoT-TrustChain: A Blockchain-Based Access Control Framework with Delegation and Event Awareness for Resource-Constrained IoT Environment

Hu Hongfei
Xi'an Eurasia University
Asad Ullah
Xi'an Eurasia University
Yazeed Yasin Ghadi
Al Ain University
Zia Ullah
University of Engineering and Technology Mardan
Bekarystankyzy Akbayan
Narxoz University
Hend Khalid Alkahtani
Hkalqahtani@pnu.edu.sa

Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University,

Article

Keywords: Internet of Things (IoT), blockchain, access control, attribute-based access control (ABAC), smart contracts, Ethereum, internet of things security, event-aware policies, dynamic delegation, resource-constrained internet of things

Posted Date: July 1st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9898452/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

IoT-TrustChain: A Blockchain-Based Access Control Framework with Delegation and Event Awareness for Resource-Constrained IoT Environment

Hu Hongfei[1], Asad Ullah[1], Yazeed Yasin Ghadi[2], Zia Ullah[3], Bekarystankyzy Akbayan[4], Hend Khalid Alkahtani[5*]

[1]School of Information Engineering, Xi'an Eurasia University, China.
[2]Department of Computer Science and Software Engineering, Al Ain University, UAE.
[3]Department of Computer Software Engineering, University of Engineering and Technology Mardan, 23200, Mardan, Pakistan.
[4]School of Digital Engineering, Narxoz University, Almaty, Kazakhstan.
[5*]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia.

*Corresponding author(s). E-mail(s): Hkalqahtani@pnu.edu.sa;

Abstract

The Internet of Things (IoT) continues to grow rapidly, bringing unprecedented challenges in access control due to its heterogeneity, large-scale deployments, and resource-constrained devices. Traditional access control systems like centralized Role-Based Access Control (RBAC) and static Attribute-Based Access Control (ABAC), do not meet real time, distributed, and contextual needs of the contemporary IoT environment. Although blockchain-based access control is a potentially promising alternative, existing solutions have serious flaws, such as high decision latency, weak delegation support, event responsiveness, and resource-inefficient resource utilization, especially in the limited IoT context. This study has highlighted some of the important shortcomings of existing models, such as the high latency of models under dynamic workloads, non-responsiveness to contextual policies, and wasteful use of gas because of poorly designed smart contracts. We suggest IoT-TrustChain a new blockchain-based access control system incorporating ABAC with event-based policy triggers, dynamic delegation,

and emergency override features. The architecture is resource-specifically tailored to support resource-constrained deployments in the IoT and is flexible and resource-efficient. In IoT-TrustChain have a set of smart contracts that we have created in Solidity and are running on a local Ethereum testbed using the Truffle framework. The experimental assessment in various operational conditions with benchmarks against the state-of-the-art models show that IoT-TrustChain operates with a reduction in the access decision latency rate of up to 43%, gas consumption rate of over 45%, and responsiveness of the policy update rate of 38%. In addition, the system is shown to have a latency of less than 200ms when subjected to composite event loads, which confirms that the system is useful in real-time and scalable IoT applications. The results indicate that IoT-TrustChain fills the gap existing between the concepts of decentralization, flexibility, and performance in the scope of the IoT access control, providing a holistic solution to the contemporary IoT ecosystems.

Keywords: Internet of Things (IoT), blockchain, access control, attribute-based access control (ABAC), smart contracts, Ethereum, internet of things security, event-aware policies, dynamic delegation, resource-constrained internet of things.

1 Introduction

The Internet of Things (IoT) [1] literally changes the face of contemporary technology, making the world hyperconnected whereby devices, sensors, and systems can communicate freely to exchange information [2] [3]. More than 20 billion devices are projected to be linked to the IoT ecosystem by 2025 [4]. The IoT has an enormous potential to enhance efficiency, convenience and decision-making across different industries [5], including smart home [6] and healthcare applications [7], as well as industrial automation [8]. Nevertheless, the massive use of inter-networked devices presents a plethora of serious security issues, especially in the area of access control. As machines continue to produce, distribute, and process data, it is more than ever essential to safeguard access to databases and information and make sure that they are secure, scalable, and efficient, while doing so, in a manner that remains sound and reliable [9].

Figure 1 demonstrates a conceptual map of the currently existing difficulties in access control to IoT systems. The core issue with securing IoT systems is the sheer number and diversity of devices, each possessing different characteristics and access requirements that are not uniform which creates a difficulty when securing them as a whole of one system [10]. The basic problem with securing IoT systems is that them are numerous and numerous with unique characteristics and access requirements that are not uniform in order to secure a whole of one system. Discretionary Access Control (DAC) (including its variations) [11], Identity-Based Access Control (IBAC) [12], and Mandatory Access Control (MAC) [13] are traditional access control models that are not best in the provision of IoT environments [14]. Both DAC and IBAC are based on the development of comprehensive lists of access control (ACLs) or fixed identities, neither of which can be applied in large-scale and dynamically operated IoT systems

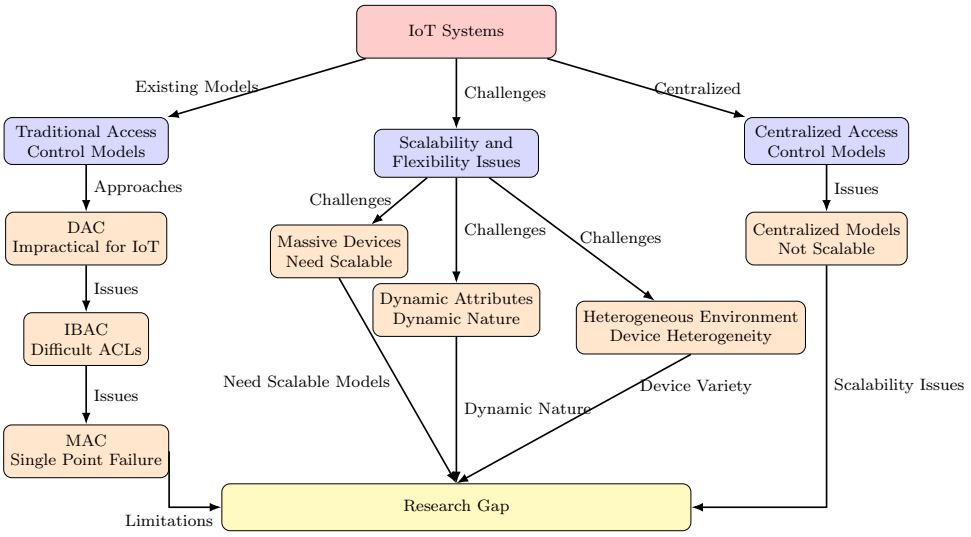


Fig. 1: Conceptual diagram stating the main limitations of traditional and centralized IoT access control models, which cause gaps in research on scalability, flexibility and trust.

[15]. In addition, these solutions do not offer the fine-grained, adaptable access management, which is required in IoT [16]. Although MAC has the ability to implement rigid access control, it is likely to be affected by a single point of failure, compromising the distributed character of the IoT [17]. These traditional approaches do not take into consideration the diversity of devices within heterogeneous settings which are administered by diverse organizations, policies as well as ownership frameworks [18]. To mitigate these constraints, Attribute-Based Access Control (ABAC) has come forth as a more flexible network to the IoT offering a viable solution to these constraints [9] [11]. In ABAC, unlike in static identities or roles, access policies depend on a collection of attribute (capabilities of devices, environmental conditions, and user roles) [19]. One can achieve more dynamic, context sensitive and fine-grained access control, because policies can be dynamically (depending on user and device attributes) adapted. Even though it has such benefits, ABAC does not necessarily focus on the issue of trust, transparency, and accountability in decentralized IoT systems [20] [21] [22]. Specifically, the absence of a single, untampered system to regulate and authenticate the dispensation of attributes and implement policies exposes the IoT systems to unwarranted access and malpractice offenses [2]. A promising solution to these issues is the blockchain technology, which is decentralized, immutable and transparent in nature, making it promising solution to these problems [23]. Blockchain could remove the need of centralized authorities and decrease the risk of single points of failure since they provide a public ledger through which transactions are recorded and checked by more than one party at the time of verification by multiple parties [24]. Blockchain consensus mechanisms make it impossible to change the data once it is stored, unless

the network agrees to it, hence creating an incorruptible audit trail of every access control event [25]. The nature of blockchain consensus mechanisms is that once data is captured, it cannot be changed without the consent of the network and generates an incorruptible audit trail of all events of access control. Blockchain can provide a secure, transparent, and distributed access control system in the IoT system when combined with ABAC [26]. Moreover, the capability of the blockchain to facilitate decentralized models of trust is particularly useful in the IoT where devices tend to be running on varying networks and different domains of ownerships as well [22] [27].

In the case of the ABAC combines with blockchain, the contemporary studies relating to the field are not high-level [28]. The current research is based on either single facets of access control, like the model of static authorization or centralized delegation and lacks the full capacity to take advantage of the benefits of blockchain to develop a comprehensive, flexible and scalable access control framework of the IoT to date [1]. Besides, although permission delegation models, like event-based and query-based delegation, are critical in real-time decision-making within the IoT, the existing models are still centralized or semi-centralized, which is vulnerable to failure and security breach [29] [30]. An urgent requirement is to have an integrated and holistic architecture that integrates ABAC with blockchain to offer dynamically changing access control and delegation of permissions to a decentralized and heterogeneous IoT setting [20].

1.1 Problem Statement

The intensive development of the IoT systems has resulted into the creation of a sophisticated and dynamic network of connected devices, all of which need secure, adaptable, and scalable access controls [4]. The conventional access control methods do not match the requirements of the IoT, especially since they are centralized and cannot be expanded in line with the amount of devices [18] [19] [9]. Moreover, available allowances delegation schemes are usually concentrated to form vulnerabilities and inefficiencies. The decentralized access control solution is in urgent need of scalability and flexibility demanded by IoT systems, a transparent yet immutable access control solution, and a decentralized access control solution, also referred to as decentralized access control solution or decentralized access control solution. The study will attempt to fill this gap by introducing a new blockchain-based access control model that combines both the dynamism of the Attribute-Based Access Control (ABAC) and the features of security, transparency, and decentralization of blockchain [19] [31] [32]. The solution will allow the provision of safe access control and permission delegation in IoT through the implementation of fine-grained access control and ensuring access is only provided to authorized devices and users on the basis of their attributes and request context. This paper will show how the blockchain can be applied to come up with a decentralized trust model to distribute the attributes in order to eliminate the risks posed by the centralized access control framework [33] [34]. Also, the suggested model will facilitate real-time and event-driven delegation of permissions, which ensures a powerful and scalable solution to the increasing security challenges of IoT systems. The following are the main contributions of this paper.

- To develop and deploy a decentralized, blockchain-based access control model that requires no centralized authority, and, thus, breaks the constraints of the traditional access control architectures (including DAC, IBAC, and MAC) in large-scale IoT systems. The model will see the access control policies transparent, immutable and scalable with the use of blockchain distributed ledger to handle access rights and permissions of diverse IoT devices.
- To suggest a new type of permission delegation system that is embedded in the blockchain-based model of access control that allows both query-based and event-based delegation. It aims to make IoT systems have real-time and context-aware access control through the delegation of permissions, based on events (e.g., fire, traffic accident) or queries (e.g., medical records requests), which will enhance the responsiveness and flexibility of the system.
- To review and improve the concept of using blockchain technology to enhance the integration of the Attribute-Based Access Control (ABAC) to enable fine-grained and context-sensitive access control. Because it will be aimed to make IoT systems more secure, this goal will consist of making sure that only authorized devices or users that have valid attributes can be allowed access, that attribute distribution is reliably and transparently logged to the blockchain, and therefore makes IoT access control systems both more scalable and secure.

1.2 Organization of the Paper

The rest of this paper is divided into the following.

In section 3, provides a review of all the literature that exists about access control mechanisms in IoT systems. This entails the comprehensive analysis of conventional access control models, the Attribute-Based Access Control (ABAC), and applying blockchain technology to IoT security. Section 1 states the proposed access control model, which relies on blockchain. This section explain how the model combines ABAC with blockchain technology to offer secure, scalable and flexible access control in the IoT environment. The important aspects of the model, such as the mechanisms of delegating and the role of blockchain in providing transparency and trust are also discussed. The system architecture of the proposed model is presented in Section 4. Describes the communication between the IoT devices, attribute authorities, and blockchain nodes. This part also describes the procedures of attributes assignment, performing access control and delegation of permissions in the system. A detailed analysis of the scalability, security and efficiency of our proposed model has been provided in Section 5 according to the results of massive simulations. The outcomes point out that the transaction latency, throughput, and gas consumption are significantly better than that of the conventional and modern IoT access control solutions, which proves the feasibility and the advantage of our solution.

Lastly, Section 6 summarizes the article by giving a conclusion of the important findings and contributions of the research. We also present the possible opportunities of future work, especially in the regard of the changing IoT systems and security issues.

2 Preliminaries

2.1 Blockchain Technology Overview

A blockchain [24] is mathematically defined as a sequence of blocks $X_1, X_2, \ldots, X_n$ where each block X_i contains a header H_i and a body T_i. The header H_i stores:

$$H_i = \big(\text{Prev_hash}_i,\ \text{Timestamp}_i,\ \text{MerkleRoot}_i\big). \tag{1}$$

Here, Prev_hash_i is the cryptographic hash of the previous block header:

$$\text{Prev_hash}_i = \text{Hash}(H_{i-1}). \tag{2}$$

Interplanetary File System (IPFS) [23], Storj [32], and BigChainDB [35] are among

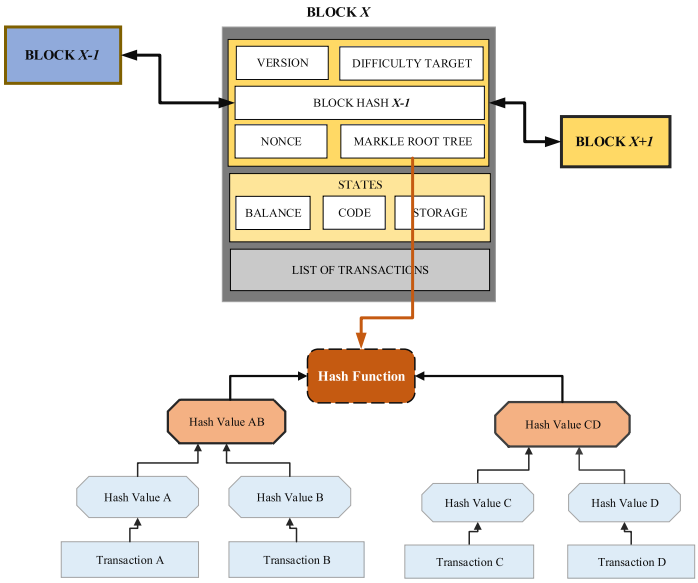


Fig. 2: Structure of the Blockchain Network

the decentralized storage technologies that are currently popular. The Merkle root [31] of block i is computed recursively over the hashes of transactions:

$$\text{MerkleRoot}_i = \text{Hash}\left(\text{Hash}(t_1 \| t_2) \| \text{Hash}(t_3 \| t_4)\right), \tag{3}$$

where $\|$ denotes concatenation and t_j represents a transaction.

Any modification in a transaction t_j propagates changes to the Merkle root, thereby altering Prev_hash_{i+1} and invalidating subsequent blocks. This cryptographic linking ensures tamper-evidence and immutability, crucial for maintaining a trustworthy distributed ledger.

2.2 Address and Transaction

Every blockchain entity (e.g. an IoT device) generates a private key $sk \in \mathbb{Z}_n$ uniformly selected at random. Its corresponding public key pk is calculated using elliptic curve cryptography (ECC):

$$pk = sk \cdot G, \tag{4}$$

where G is the base point of the elliptic curve. The blockchain address $Addr$ is derived as:

$$Addr = \text{Base58Check}\big(\text{Hash}(pk)\big). \tag{5}$$

Given only $Addr$ or pk, it is computationally infeasible to recover sk due to the discrete logarithm problem of the elliptic curve (ECDLP).

In our proposed framework, attribute authorities issue attribute transactions τ as:

$$\tau : AA \rightarrow (Addr, \alpha), \tag{6}$$

where AA is the address of the attribute authority, $Addr$ is the device address, and α denotes a given attribute.

3 Related Work

IoT access control has become a significant field of research due to the unique challenges associated with decentralized environments and resource-limited devices. Conventional authentication mechanisms have been extensively investigated to ensure secure communications [36] [20] [27]. Zhonghua et al. [6] introduced an identity-based authentication scheme wherein each IoT device is granted a virtual IPv6 address serving as a unique identifier, thus acting as a cryptographic certificate. Similarly, Bello et al. [28] proposed a lightweight authentication protocol tailored for wireless sensor networks, relying on credentials issued by local cluster heads to enable future authentication. Zhang et al. also incorporated attribute-based encryption (ABE) tools, which allow the incorporation of attribute-based access control and data secrets at the same time [20]. However, most ABAC and capability-based solutions currently rely on a trusted third-party to manage the key and enforce policy [24], which introduces single points of failures and trust bottlenecks in the implementation of the solutions to the problem at hand [37].

He et al. [19] exploited identity-based elliptic curve cryptography (IBE-ECC) and one-time passwords to develop a lightweight end-to-end authentication system, which minimized storage space and cryptographic overhead, thus improving its application to the IoT setup. The resulting improvements factor notwithstanding, identity-based schemes can be computationally and communication-intensive, which is not a feature of highly resource-constrained IoT-based devices. In order to address these constraints, distributed capability-based access control models have risen. In [15] suggested a model whereby capabilities-in the form of cryptographically secure tokens- are delegated between smart devices so that fine-grained authorization can be achieved without use of centralized parties. Another approach based on a capability-based approach

was introduced by Bakhtiary et al. [4] and focuses on minimum initial privileges and dynamic delegation of further privileges when needed.

Although OAuth-based centralized solutions have been extensively used in industries [29], their high computational and communication requirements make them unfeasible to directly use in the context of IoT. A centralized model can be the CALIPSO project described in the paper by Castiglione, et al. [38], which assigns access control to a potent IoT-OAS server. In [39] the actual implementation of OAuth logic on small devices is not viable and this encourages the development of decentralized solutions.

Role-Based Access Control (RBAC) [11] has always been the preferred method of controlling permissions based on a set of pre-defined roles, which may not be flexible in the dynamic IoT environment where devices may need fine-grained and context-sensitive access control. Attribute-Based Access Control (ABAC) [28] has come out as a more granular substitute, whereby access policies are specified in terms of attributes, and not in terms of fixed-point roles.

Qiu et al. [41] presented a wide review of access control systems in the IoT, and concluded that the traditional centralized system does not suit the needs of the resource-constrained and distributed world. ChainAnchor is an identity management system and access control with attributes, which proposed by Yang et al. [40] is based on blockchain and anonymous identities that can be verified. In [42] proposed a completely decentralized access control system, based on smart contracts to define and enforce policies on the blockchain itself. Similarly, with FairAccess by FairAccess [43] and Xu et al. [44], blockchain is used to store policies safely and support fine-grained authorization using a novel type of transactions.

3.1 Blockchain-based Access Control in the Internet of Things

The blockchain has been more and more viewed as a potential pillar of access control in the Internet of Things (IoT) to overcome the shortcomings of the traditional centralized designs, such as the single points of failure, scalability, and inter-heterogeneity.

Yang et al. [40] suggested ChainAnchor framework, which uses the blockchain to provide anonymous but verifiable identities and fine-grained access control to prevent risks to privacy and promote accountability. They use a permissioned blockchain in their system, which enables authorized participants to provision access policy without revealing sensitive identities. On the same note, Zhonghua et al. [6] proposed FairAccess, which is a decentralized authorization system in which blockchain transactions are coded with access permissions. This method is transparent and auditable and can potentially cause a performance overhead with an extensive blockchain writes that can be an issue with resource-constrained IoT devices. The article authored by Zhonghua et al. [6] introduced a decentralized architecture that utilizes smart contracts to establish and implement access policies. Smart contracts can provide programmable fine-grained control but are computationally expensive and might not be appropriate on lightweight IoT nodes. Similarly, Jakhar et al. [12] suggested blockchain-based access control systems with access policies posted on-chain by resource owners. This

Table 1: Comparative analysis of state-of-the-art access control approaches in IoT.

References	Technique / Technology Deployed	Key Findings / Contributions	Strengths and Weaknesses
Zhonghua et al. [6]	Identity-based authentication using virtual IPv6 addresses	Provided lightweight device identity and authentication via cryptographic IPv6 mapping	Strength: Simple identity management; Weakness: Relies on centralized authority, high communication overhead
Bello et al. [28]	Lightweight cluster-head-based authentication protocol	Enabled local authentication through cluster-issued credentials	Strength: Low device overhead; Weakness: Semi-centralized, vulnerable if cluster head compromised
Saidi et al. [32]	One-time passwords (OTP) with IBE-ECC	Reduced storage and cryptographic costs for IoT authentication	Strength: Storage efficiency; Weakness: Requires trusted key issuer, limited scalability
He et al. [19]	Capability-based distributed access tokens	Delegated secure capabilities among devices for fine-grained authorization	Strength: Decentralized and flexible; Weakness: Capability distribution introduces trust dependencies
Bakhtiary et al. [4]	Dynamic capability delegation	Allowed progressive authorization via minimum initial privileges	Strength: Granular and adaptive; Weakness: Requires secure initial delegation channel
CALIPSO [38]	OAuth-based centralized IoT-OAS server	Delegated access decisions to powerful servers	Strength: Industrially mature; Weakness: High overhead, unsuitable for constrained IoT
Kaur et al. [11], Li et al. [21]	Role-Based (RBAC) and Attribute-Based Access Control (ABAC)	Defined access via roles or attributes for IoT systems	Strength: Fine-grained control with ABAC; Weakness: Dependent on trusted policy/key authorities
Zhang et al. [38]	ABAC integrated with ABE	Provided confidentiality alongside attribute-based access control	Strength: Data security with policy enforcement; Weakness: High encryption overhead, limited IoT suitability
Yang et al. [40] (ChainAnchor)	Blockchain-based ABAC with permissioned blockchain	Enabled anonymous yet verifiable access policies	Strength: Privacy + decentralization; Weakness: Performance cost due to consensus
Xu et al. [2] (BlendCAC)	Blockchain + capability tokens via smart contracts	Achieved fine-grained authorization stored on blockchain	Strength: Secure and auditable; Weakness: Transaction latency and scalability limits
Merlec et al. [8] (BlAsT)	Blockchain-enhanced STR (Signed Tree Root)	Efficient policy validation through cryptographic structures	Strength: Transparency and key integrity; Weakness: Requires off-chain identity providers
Waheed et al. [26]	Lightweight blockchain without PoW	Reduced cost in smart home IoT environments	Strength: Low computational overhead; Weakness: Assumes trusted local domain, limited generalization

increases the level of transparency and eliminates the reliance on central powers but creates scale issues since blockchain storage and computability is limited.

Merlec et al. [8] introduced this further with BlAsT that combines blockchain to manage transparent keys, and Signed Tree Root (STR) structure to provide security policies with an efficient validation process. Although BlAsT has got innovative cryptographic structures, it still needs off-chain identity providers, which make it dependent on trust. Xiao et al. [43] proposed BlendCAC, which is a composite of blockchain and capability-based access control that handles the management of capability tokens through smart contracts. The given model is efficient in distributing access management, but can be limited by latency and throughput issues of the public blockchains.

Waheed et al. [26] suggested a lightweight blockchain architecture of IoT systems of a smart home without proof-of-work to lower the cost of computation. Nevertheless, the model is limited to more open and adversarial IoT environments due to their assumptions of trusted local environments. Although blockchain has major benefits, such as immutability, distributed consensus, and cryptographic security, there are still a few challenges to its successful implementation in the development of IoT access control. Poor transaction cost, latency, and computation needs are incompatible with limited resources common to IoT devices. In addition, several of the existing solutions are heavily user-centered in their approach to access requests and ignore event-centered authorization cases that are important in IoT implementations, like changing the response dynamically in case of emergency or system failure [45]. More literature has touched upon the assignation of smart contracts to dynamic and programmable access control. In Usman et al. [35], the authors recommended a system in which resource owners set resource access policies and put them on-chain, which can be used to perform decentralized authorization checks. In [21], the design of BlAsT, a blockchain-enhanced key management transparent architecture, was developed, in which a Signed Tree Root (STR) structure of cryptographic proofs is deposited to the blockchain, and provides independent verification of integrity and compliance with the policy by the clients.

A number of gaps that are critical to the existing research remain. To begin with, several blockchain-based schemes presuppose implicit trust between the devices within the localized IoT environments, which ignores some situations when the devices can be compromised or misconfigured, leaving vulnerabilities despite distributed infrastructures. Secondly, although smart contracts can be used to automatize policies, their computational requirements may continue to be a challenge to the IoT devices with strict resource limitations. Besides, the majority of the currently available schemes place strong emphasis on user-requested access and do not support event-driven authorization, which is instrumental in an IoT scenario like emergency response where access permission should be dynamically issued in response to a specific context.

4 Proposed Methodology

The suggested IoT-TrustChain framework will be organized into four logically related steps, which will provide safe, adaptable, and effective access control of the IoT environment (see Figure 3). All of these stages are implemented by consuming special smart contracts that are deployed on a consortium blockchain, thus offering transparency, auditability, and low latency decision-making. These contracts in aggregate manage attribute-based access control (ABAC) enforcement, context-based delegation and trust management to achieve dynamic IoT situations. The selection of Ethereum blockchain was motivated by the fact that it is significantly more flexible in the experimentation, benchmarking, and comparative analysis of studies across studies. Such flexibility will ensure that the Ethereum will be a trustworthy and viable framework against the state of the art framework, as well as generalizations of the results would be obtained to larger access control frameworks in Internet of Things.

4.1 Stage 1: Device Onboarding and Registration

IoT devices are securely onboarded through a registration process in which each device submits a signed request. These credentials are verified by the smart contract of the name of the device registry and the metadata of the device is stored on the blockchain. This will ensure that only legitimate devices are used in the network.

4.2 Stage 2: Policy Definition and Enforcement

Attribute-Based Access Control (ABAC) policies are used to formulate access policies. The *PolicyManager* contract compares these policies with contextual characteristics wrt the user roles, resource sensitivity and environmental settings. This provides fine grained and dynamic access control enforcement.

4.3 Stage 3: Event-Driven Delegation

The framework allows access rights to be delegated temporarily in case of dynamic IoT-based scenarios (in the case of an emergency). DelegationManager contract uses event-driven rules which enables quick adaptation without affecting the integrity of the system.

4.4 Stage 4: Access Validation and Monitoring

Both ABAC policies and delegation rules are used to validate access requests. The final decision is stored on-chain to be audited, and anomalies are monitored by the modules and revoked in the case of necessity. This will guide a constant accountability and trust management. Figure 3 is a representation of the overall workflow of the proposed IoT-

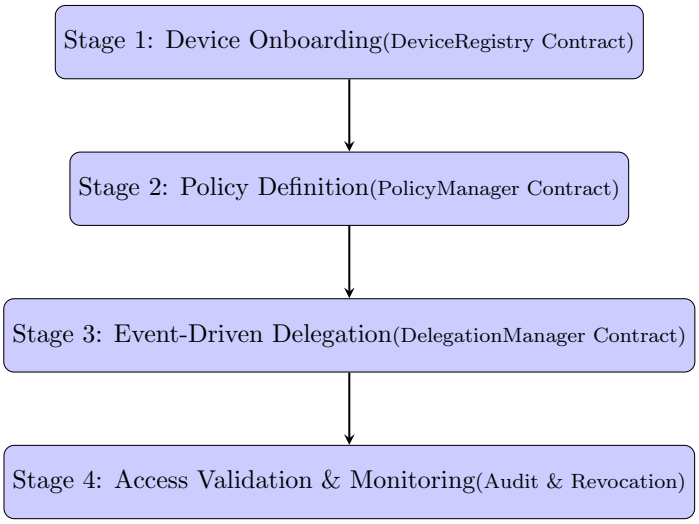


Fig. 3: Proposed IoT-TrustChain workflow consisting of four sequential stages.

TrustChain framework. As illustrated, the methodology follows four steps, which are sequential. In the first step, the device is onboarded into the blockchain network with the help of the *DeviceRegistry* contract. Second, access policies (fine-grained ones) are defined and implemented through the *PolicyManager*. Third the contextual and event based delegation is facilitated by using the *DelegationManager* and thus dynamic adaptability is facilitated in real time IoT environment. Lastly the stage of Access validation and monitoring validates the user request, audit actions and revocation when necessary. This is a logical and modular workflow that would guarantee scalable and resilient system and provide adaptive and secure access control in the dynamic circumstances of the IoT.

4.5 System Overview

The proposed architecture consists of the following main components: IoT Devices, User Devices, an Application Manager, an IoT Manager, a Blockchain Manager, and a consortium Blockchain Network. Each component plays a distinct role in enabling secure and dynamic access control. The structure of the control components and their interactions are shown in Figure 5. The given architecture includes the following key

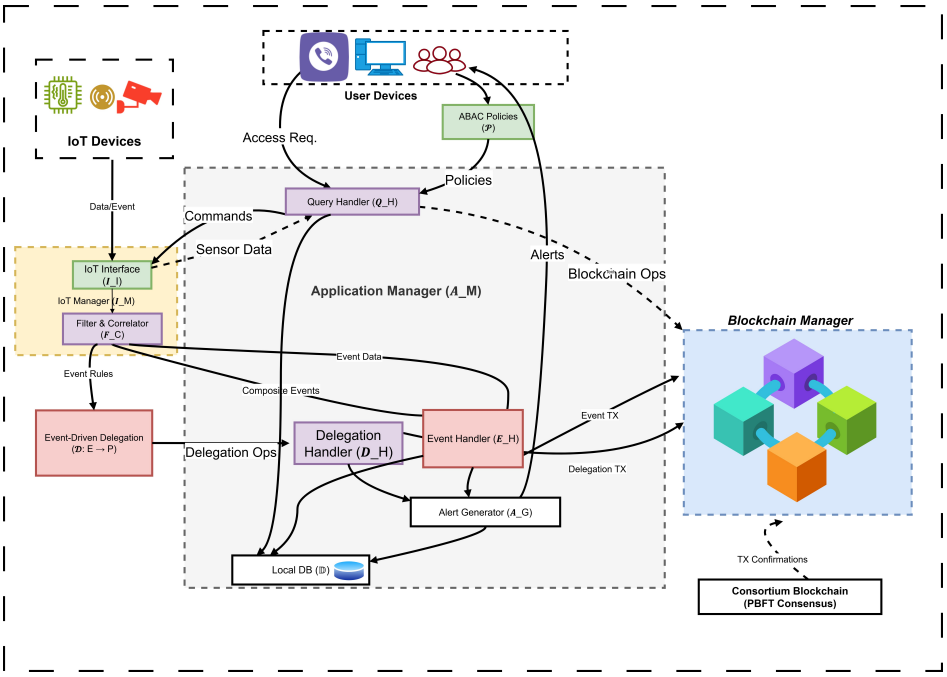


Fig. 4: Proposed system architecture integrating IoT device monitoring, dynamic event-driven delegation, and blockchain-based trust management.

elements: IoT Devices, User Devices, an Application Manager, an IoT Manager, a Blockchain Manager and a consortium Blockchain Network. All the elements have a

specific contribution to the dynamic and secure access control. The control components structure and interactions are depicted in Figure 5.

Our model combines attribute-based policies and event-based delegation techniques to provide fine-grained access control, and at the same time provide auditability and trust without employing central authorities. Figure 4 Figure 5, and Figure 6 depicts the proposed architecture, integrating IoT devices, event-driven policy management, and blockchain for secure, dynamic access control. The IoT devices produce events and data, which are processed by the IoT Manager using its interface and Filter & Correlator modules to detect notable conditions. The Application Manager facilitates access decisions by involving the Query Handler of user request, the Delegation Handler of dynamic delegation functionality and the Event Handler of event-driven triggers, where all important activities are logged in the local database. The Alert Generator sends out critical alerts to user devices when it is required. The Blockchain Manager logs, securely to a consortium blockchain with PBFT consensus, event and delegation transactions and transports them to ensure transparency and tamper-resistance. This architecture supports real time, fine level policy adjusting, which is effective in the methods of coping with the challenge of scalability, heterogeneity and trust of an IoT environment.

4.6 IoT Devices

IoT devices in our system are typically resource-constrained nodes capable of sensing, data collection, or actuation. Each IoT device is uniquely identified and possesses a cryptographic key pair (pk_d, sk_d), where pk_d is the device public key and sk_d its private key. IoT devices are assumed to store or control resources (such as sensor data or physical services) whose access is subject to authorization. Our architecture accounts for the potential compromise of IoT devices, ensuring that sensitive operations such as transaction validation, policy enforcement, and trust management are handled by more capable system components. This design approach minimizes computational overhead on resource-constrained IoT devices while maintaining robust security.

4.7 User Devices

The user devices, denoted by U and will contain smartphones, tablets, and personal computers and will be the main point of contact between the end users and the IoT ecosystem and represented in Figure ref:system-architecture. These devices have much greater computing power and memory than resource-constrained IoT nodes; therefore, enabling them to perform cryptographic primitives which are necessary to achieve secure communication and blockchain transactions, including encryption, digital signatures, and hash calculations. The main roles of user devices in the proposed framework are the registration of IoT devices on the blockchain network, the definition, and management of fine-grained access policies $\mathcal{P}$, denoted by icons, to the resources under their ownership, denoted by R, and the generation of access requests to resources owned by other network nodes. Furthermore, user devices facilitate dynamic permission delegation, whereby a user $u_i \in U$ may delegate a specific set of privileges $\delta \subseteq \mathcal{P}$ to another user u_j or device d_k, contingent upon predefined conditions or triggered

events E, satisfying the logical implication:

$$E \implies \delta(u_j, R)$$

To maintain lightweight and secure mutual authentication between user devices and IoT nodes, the architecture employs efficient mechanisms such as the bubbles-of-trust model, renowned for its low communication latency and resilience against diverse attack vectors, thus ensuring that security operations impose minimal computational burden on constrained devices.

4.8 Application Manager

The Application Manager, which is also referred to as the Application Manager, abbreviated as the symbol, and denoted as $\mathcal{AM}$,is a key middleware layer that facilitates interactions between the network applications and the base access control infrastructure, which is blockchain-based. It summarizes the fundamental logic needed to handle user requests, dynamic delegation, event-based triggers and policy enforcement (see Figure 6 detail architecture). Formally, $\mathcal{AM}$ comprises four tightly integrated modules:

1. **Query Handler** ($\mathcal{QH}$): Receives and processes user-originated queries, including resource access requests and event subscriptions. Each request qi is logged and evaluated against current policy rules $\mathcal{P}$.
2. **Delegation Handler** ($\mathcal{DH}$): Orchestrates the delegation and revocation of permissions. For any delegation request $\delta(ui \rightarrow uj, Rk)$, the module constructs a blockchain transaction Tdelegate and forwards it to the blockchain manager for submission.
3. **Event Handler** ($\mathcal{EH}$): Continuously monitors real-time events $E = \{e1, e2, \ldots, en\}$ from the IoT environment. Using predefined rules and temporal correlation, it synthesizes high-level composite events E^* that reflect meaningful environmental conditions.
4. **Alert/Warning Generator** ($\mathcal{AG}$): Generates alerts, warnings, and notifications for subscribed users upon detection of critical events, i.e., $E^{\Rightarrow}$Notify(Us), where $Us \subseteq U$ is the set of subscribed users.

The Application Manager maintains a local database $\mathbb{D}$ consisting of policy states, active subscriptions, and delegation mappings, enabling responsive decision making and reducing unnecessary interactions with the blockchain layer.

4.9 IoT Manager

The *IoT Manager*, denoted as $\mathcal{I}_M$, is tasked with interfacing between the blockchain-enabled access control system and the underlying physical IoT infrastructure (see Figure 4). It is made up of two core modules:

1. **IoT Interface** ($\mathcal{I}_I$): Enables communication in both directions with the IoT devices $D = \{d1, d2, \ldots, dn\}$ to allow real-time command transmission and data acquisition. This module is used to make sure there is smooth integration of heterogeneous IoT nodes into the system.

2. **Data Filter and Event Correlator** ($\mathcal{F}_C$): Implements signal-processing techniques to filter noisy sensor data and applies correlation logic, $\phi(e_i, e_j) \mapsto \hat{E}$, to detect complex conditions. For example, simultaneous temperature-spike (e_1) and smoke-detection (e_2) events can be combined into a fire alert ($\hat{E} = \text{fire}$).

The capacity to draw meaningful composite events on the fly is vital to providing a way of context-sensitive and dynamic access control decision-making, in which access policies can vary depending on changing environmental conditions. Figure 5 shows the internal detail architecture of our proposed blockchain-based IoT access control framework. The system will be split into three main parts: the Application Manager $\mathcal{AM}$, the Blockchain Manager, and the IoT Manager $\mathcal{IM}$. The Application Manager orchestrates core logic via dedicated modules, including the Query Handler $\mathcal{QH}$ for managing access requests, the Delegation Handler $\mathcal{DH}$ for handling dynamic delegation operations, the Event Handler $\mathcal{EH}$ for detecting and processing events, and the Alert Generator $\mathcal{AG}$ responsible for issuing notifications and warnings. Meanwhile, the IoT Manager integrates raw device-level operations through the IoT Interface $\mathcal{II}$ and processes contextual information using the Filter & Correlator $\mathcal{FC}$. The Blockchain Manager provides safety in recording of transactions, consensus and storage of policy that cannot be changed. The architecture mediates data flows between the user devices and the IoT devices and allows enforcing policies, delegation in real-time based on events, and recording secure transactions on the blockchain. All these interactions create a context-sensitive, fine-grained access control mechanism that can change in response to dynamic IoT environments, as also confirmed in our simulation experiments [46–49].

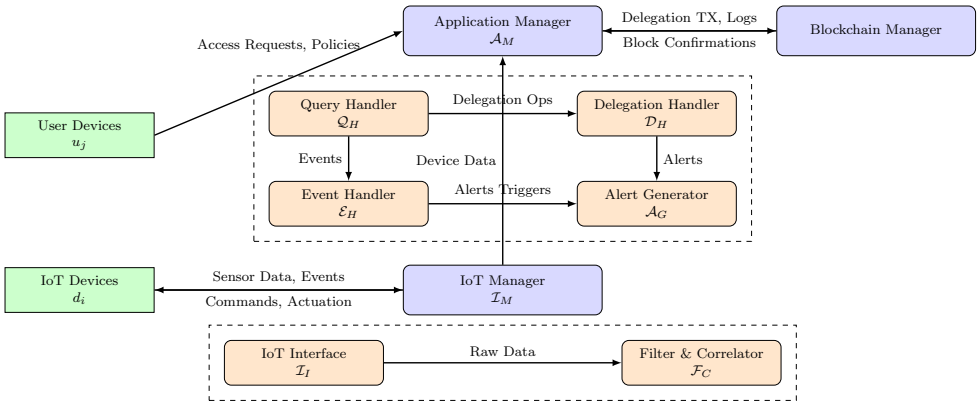


Fig. 5: Control layer building in the proposed blockchain-based access system in IoT. The arrows imply user requests, policy enforcement, event detection, blockchain transactions, and feedback of the system.

4.10 Blockchain Manager

The Blockchain Manager connects the operations of the application-layer with the consortium blockchain (see Figure 5). It carries out the following activities:

- onboard IoT devices to the blockchain using smart contracts, which encode device-specific access policies.
- Processes permission delegation, activation and revocation transactions, and these changes are recorded in-chain permanently.
- Prepares and commits transactions to the blockchain of both user-based operations and event-based updates to access control policies.

4.11 Consortium Blockchain Network

Our architecture has a consortium blockchain network that is operated by multiple Attribute Authorities (AAs). The model also offers security and efficiency in that the number of participants is reduced to known and authorized parties as opposed to the public blockchains. The blockchain network:

- Maintains an immutable ledger of device registrations, attribute assignments, access policies, and permission delegation transactions.
- Operates a consensus protocol, such as Practical Byzantine Fault Tolerance (PBFT), capable of tolerating up to $\lfloor \frac{n-1}{3} \rfloor$ malicious nodes among n total authorities.
- Provides decentralized trust and auditability, ensuring no single entity can compromise the system's integrity.

4.12 Attribute-based Access Control

To support dynamic IoT environments, our proposed architecture will rely on the Attributese-Based Access Control (ABAC) to give the access control a fine-grained, context-aware access control. ABAC is a more adaptable substitute of non-flexible Role-Based Access Control (RBAC) frameworks, where licenses are specified according to specified roles. ABAC policy may be stipulated as a rational requirement on a set of attributes characterizing users, devices and environment around.

Formally, an access policy $\mathcal{P}$ is defined as a Boolean function:

$$\mathcal{P} : \mathcal{A} \rightarrow \{0, 1\}$$

where $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ denotes a vector of attributes. For example:

$$\mathcal{P}(\vec{a}) = a_1 \wedge (a_2 \vee a_3)$$

Here, a_i represents conditions such as:

- User identity or role (e.g., *first responder*)
- Device type (e.g., *traffic light controller*)
- Contextual factors (e.g., time of day, emergency status)

A user or device is granted access to a resource only if the attribute vector $\vec{a}$ satisfies $\mathcal{P}$, ensuring precise and dynamic access decisions. In our architecture, these policies

are encoded and enforced using smart contracts on the blockchain. This guarantees immutability and auditability of policy definitions and enforcement, mitigating the risk of unauthorized policy changes and enhancing trust in distributed IoT ecosystems [7, 14, 16].

4.13 Event-Driven Permission Delegation

To provide fine-grained, context-aware access control, our proposed architecture will use the Attribute-Based Access Control (ABAC) to support dynamic IoT environments. ABAC is a more flexible replacement of inflexible Role-Based Access Control (RBAC) systems, in which permissions are defined in relation to defined roles. The ABAC policy can be defined as a logical condition on a collection of attributes describing users, devices and the surrounding environment.

Let $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$ denote the set of significant events monitored in the IoT environment. We define a delegation mapping:

$$\mathcal{D} : \mathcal{E} \to \mathcal{P}$$

such that for an event e_j, a corresponding access policy $\mathcal{P}_{e_j}$ is activated or updated. For instance:

$$\mathcal{D}(e_{\text{accident}}) = \mathcal{P}_{\text{emergency}}$$

Here, e_{accident} could represent a composite event detected by sensors indicating a traffic accident, triggering a policy $\mathcal{P}_{\text{emergency}}$ that grants temporary control rights to emergency responders over traffic management systems. Any change that is based on the events is recorded as a transaction on the blockchain, making delegations transparent, verifiable, and impossible to alter. The design facilitates not only timely and secure administration of access privileges to time-driven-iot applications, but it also corresponds with our general aspiration of trustless, decentralized access control in highly dynamic worlds.

The given architecture provides a lot of improvements to the current solutions in the field of the IoT access control. It is the only system that combines support of user-driven queries and event-driven operations allowing the system to behave dynamically in response to explicit user requests and contextual changes that the system senses in the environment. In contrast to rigid and role-based models, our framework uses the fine-grained and attribute-based access policy, which takes into consideration many factors including user identities, device types, and contextual parameters, and, therefore, in line with the heterogeneous and changing nature of IoT ecosystems. Furthermore, through the use of blockchain technology, the system attains decentralized and secure permission delegation, without the need to have centralized authorities which tend to have single points of failure and thus trust bottlenecks in traditional systems. The blockchain is used to document all the access control decisions and changes in policy, which remains immutable, transpired, and resilient to alteration. All these innovations together provide a strong, adaptive and resilient architecture that faces the most important problems of the scaling issues, the management of trust and the ability to ensure the adaptability in real time in the contemporary IoT landscape.

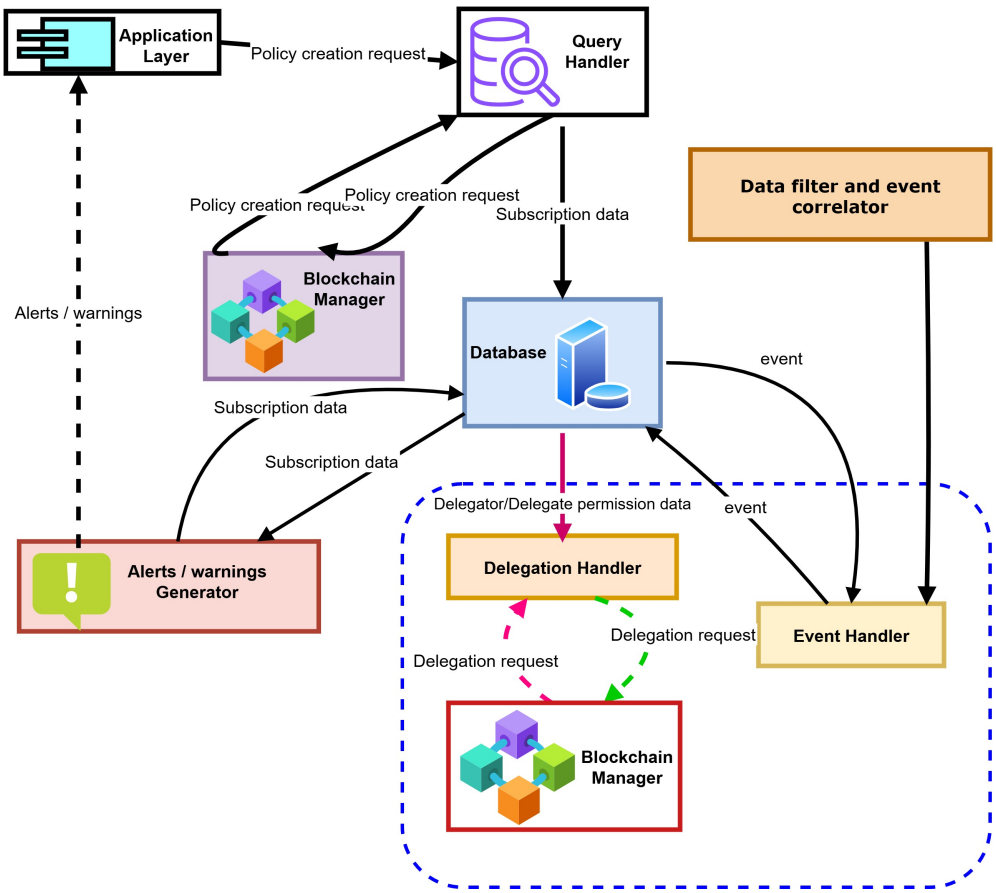


Fig. 6: Architecture of the proposed **IoT-TrustChain** framework showing the interaction between the Application Layer, policy creation, event filtering, delegation, and blockchain-based access control. Events and delegation requests are processed dynamically to ensure secure, context-aware access in IoT environments.

Figure 7 presents the event-driven delegation mechanism within the IoT-TrustChain framework. The model processes real-time IoT data, filters significant events, and triggers permission delegation dynamically using blockchain for registration and trust validation. Incoming data from IoT devices is collected by the IoT Interface (I_I), which forwards it to the Data Filter and Event Correlator (F_C). This component detects significant or composite events from real-time data streams and communicates them to the Event-Driven Delegation Engine. The delegation engine is then communicated with the Application Manager to start dynamic delegation depending on the nature of the event. At the same time, the Blockchain Layer guarantees the safety of registering and verifying devices and participating entities, which adds credibility and reliability to the process.

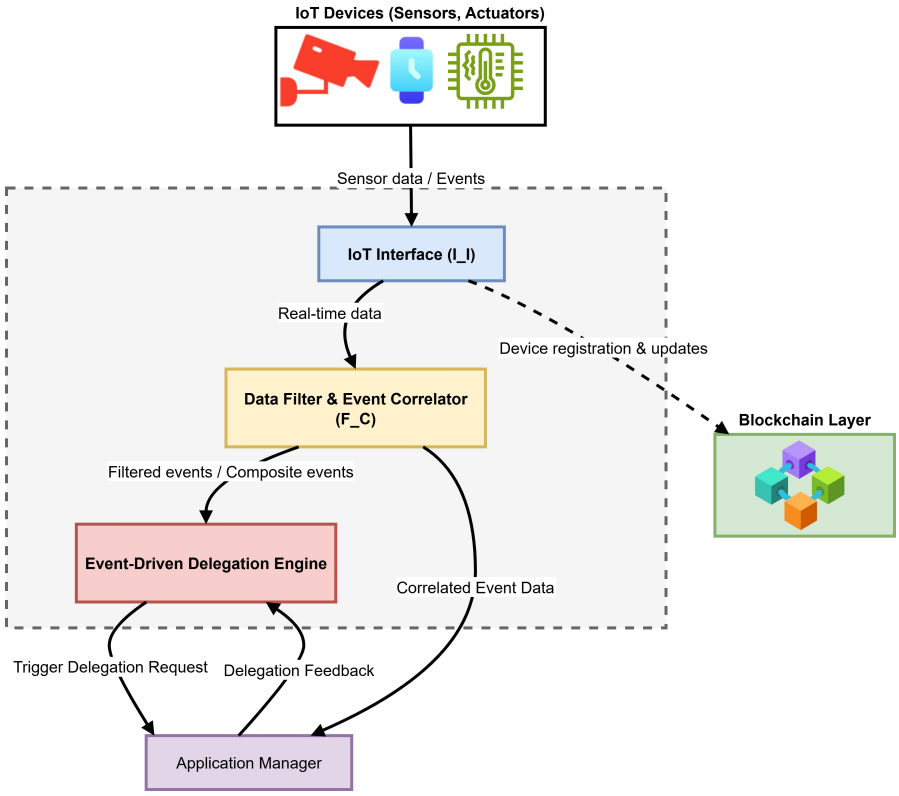


Fig. 7: Event-driven delegation workflow in the IoT-TrustChain architecture.

4.14 Smart Contracts Design

Algorithm 1 (DeviceRegistry). Handles IoT device onboarding. The function `registerDevice` stores device identities, types, and owners on-chain, preventing duplicate registrations. `getDevice` allows lookup of device records. **Algorithm 2 (ABACPolicyManager).** Manages creation and retrieval of Attribute-Based Access Control (ABAC) policies. `createPolicy` defines a policy tied to resource and conditions, generating a unique policy ID. `getPolicy` retrieves stored policy rules. **Algorithm 3 (EventDrivenDelegation).** Implements dynamic delegation of permissions. `delegateAccess` grants access rights to another user or device and logs the delegation on-chain. `revokeDelegation` deactivates a delegation, enforcing revocation when conditions change.

Algorithm 4 (AccessControlManager). Performs authorization decisions. `requestAccess` verifies whether access is allowed based on either active delegations or evaluation of ABAC policies, and emits blockchain events for auditability.

Algorithm 1 DeviceRegistry

```
Input: deviceAddr — address of the IoT device, deviceType — type of the device
Output: Confirmation of registration ("Device registered") or rejection ("Device already registered")
Start: Upon receiving a device registration request
End: After the device is registered or registration is rejected
procedure REGISTERDEVICE(deviceAddr, deviceType)
    if deviceAddr ∉ devices then
        devices[deviceAddr] ← (owner = requester, deviceType)
        emit DeviceRegistered(deviceAddr, owner, deviceType)
        return "Device registered"
    else
        return "Device already registered"
    end if
end procedure
procedure GETDEVICE(deviceAddr)
    if deviceAddr ∈ devices then
        return devices[deviceAddr]
    else
        return "Device not found"
    end if
end procedure
```

All these smart contracts cover utility cases of IoT. As an example, in the smart healthcare, the makes sure that only authenticated medical sensors (e.g. heart monitors) are onboarded. The ABACPolicyManager in the context of smart homes and Industry 4.0 imposes fined-grained policies, like limiting the manageability of energy to the utility providers during times of peak demand. EventDrivenDelegation contract would be useful in emergency response systems, which can temporarily hand over roadside units access to the ambulances to control traffic lights. Lastly, the AccessControlManager is an important part of the so-called smart cities, as it allows access to the traffic cameras by the relevant authorities and prohibits their misuse by third parties. The following scenarios underscore the application of the suggested design to real-world IoT applications.

5 Simulation Results

In this section, the performance assessment of the proposed blockchain-based IoT access control architecture is provided in full detail, which justifies the effectiveness of the architecture in facilitating a fine-grained, decentralized, and context-aware security control. The simulations will explain how our inputs, such as attribute-based policy, event-based delegation, and blockchain auditability allow access control in IoT to be scaled and efficient.

Algorithm 2 ABACPolicyManager

Input: *resource* — the resource to protect, *condition* — policy conditions for access
Output: *policyId* — unique identifier for the created policy, or policy details upon retrieval
Start: Upon receiving a policy creation or query request
End: After policy is created or retrieved
procedure CREATEPOLICY(*resource*, *condition*)
    policyId ← hash(resource, condition, timestamp)
    policies[policyId] ← (resource, condition)
    emit PolicyCreated(policyId, resource, condition)
    return policyId
end procedure
procedure GETPOLICY(*policyId*)
    if *policyId* ∈ policies **then**
        return policies[policyId]
    else
        return "Policy does not exist"
    end if
end procedure

5.1 Experimental Environment

All simulations and experiments were conducted on a workstation equipped with an Intel Core i7-11700 CPU at 3.6 GHz, 32 GB RAM, running Ubuntu 22.04 LTS. The blockchain layer was implemented using a private Ethereum-based consortium network simulated via `Ganache` [48], while smart contracts [6] were developed in `Solidity` (v0.8.x) [50] and deployed using the `Truffle` framework [49]. Interactions between the blockchain and application layer were integrated via `Web3.js`. IoT data generation and event simulations were implemented in `Python` using the `SimPy` discrete-event simulation library [51].

Figure 8 displays the deployment environment, which is configured in the Remix IDE, the popular web-based platform of Ethereum smart contracts development. In this configuration, the Injected Provider - MetaMask environment [47] is chosen that allows to connect with the Ethereum blockchain without complications with the help of the MetaMask browser extension. This will enable safe signing of transactions and management of accounts through the user browser. The address of the active account (partially shown as `0x829...ac9Ea`) is chosen to be deployed, which means that the user is integrated into the Ethereum network through MetaMask. Moreover, Gas Limit is specifically fixed to a value of custom gas amount of `3,000,000` gas units. This will provide enough gas to run a complex contract logic especially the one that will be used in areas of access control delegation, policy modification and event handling in our proposed IoT system.

Algorithm 3 EventDrivenDelegation

Input:

- *to* — the entity to delegate access to
- *resource* — the target resource
- *permissions* — access rights to grant

Output: *delegationId* — unique identifier for the delegation, or status of revocation
Start: Upon receiving a delegation or revocation request
End: After delegation is granted or revoked
procedure DELEGATEACCESS(*to*, *resource*, *permissions*)
    delegationId ← hash(msg.sender, to, resource, permissions, timestamp)
    delegations[delegationId] ← (from = msg.sender, to, resource, permissions, active = true)
    emit DelegationGranted(delegationId, msg.sender, to, resource, permissions)
    return delegationId
end procedure
procedure REVOKEDELEGATION(*delegationId*)
    if delegations[delegationId].from == msg.sender **then**
        delegations[delegationId].active ← false
        emit DelegationRevoked(delegationId)
        return "Revocation successful"
    else
        return "Not authorized to revoke"
    end if
end procedure

5.2 Performance Metrics

The system was tested based on the following metrics, and mathematical expressions were given to make them clear and replicable:

- **Transaction Latency (L_{tx})**: Defined as the time taken for a transaction t to be confirmed on the blockchain, measured as the difference between the transaction submission time $T_{sub}(t)$ and the confirmation time $T_{conf}(t)$:

$$L_{tx}(t) = T_{conf}(t) - T_{sub}(t)$$

- **Access Decision Time (T_{acc})**: Represents the total time to process an access request, which includes policy evaluation time (T_{pol}) and event-driven delegation check time (T_{del}):

$$T_{acc} = T_{pol} + T_{del}$$

- **Blockchain Throughput (TPS)**: Throughput is defined as the number of successfully processed transactions N_{tx} within a measurement window of duration

Algorithm 4 AccessControlManager

Input:

- *deviceAddr* — address of the IoT device requesting access
- *resource* — target resource to access
- *policyId* — identifier of the ABAC policy

Output: Access decision: "Granted" or "Denied"
Start: Upon receiving an access request
End: After access is granted or denied
procedure REQUESTACCESS(*deviceAddr*, *resource*, *policyId*)
    (owner, deviceType) ← getDevice(deviceAddr)
    delegationId ← hash(owner, msg.sender, resource, "full")
    if delegations[delegationId].active **then**
        emit AccessGranted(msg.sender, resource)
        return "Granted"
    end if
    (policyResource, condition) ← getPolicy(policyId)
    if condition satisfied **then**
        emit AccessGranted(msg.sender, resource)
        return "Granted"
    else
        emit AccessDenied(msg.sender, resource)
        return "Denied"
    end if
end procedure

Δt:

$$TPS = \frac{N_{tx}}{\Delta t}$$

- **Gas Consumption (G_{avg})**: Refers to the cost of computation of smart contracts. Gas consumption of a given number of transactions, a set of n transactions is calculated as the average gas used per transaction; the average gas g_i is denoted as:

$$G_{avg} = \frac{1}{n} \sum_{i=1}^{n} g_i$$

These equations allow the strict evaluation of the efficiency and the feasibility of the offered framework in actual IoT situations.

5.3 Transaction Latency and Throughput

Figure 9 of the appendix indicates the average time of confirming a change in transaction with the rate at which the transaction was sent. It has been revealed that the proposed consortium blockchain architecture can maintain a steady and predictable operation under the diverse load conditions. Especially, the system achieves an average

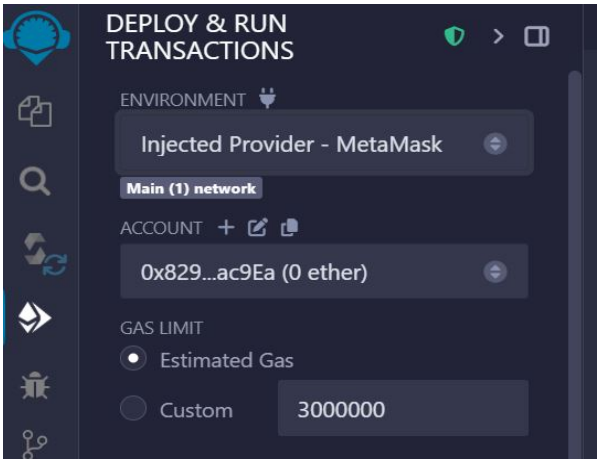


Fig. 8: Smart contract deployment environment using Remix IDE with MetaMask injected provider.

transaction latency of approximately 2.3 seconds with throughput of approximately 30 transactions per second (TPS). This latency profile is much lower than the allowed limit in an IoT environment where devices periodically make access control events but do not have hard time constraints. Besides, the latency increases significantly with an increase in the transaction rate up to a point of 50 TPS. Such scaling trend attests to the power of the architecture because the consensus mechanism, along with smart contract execution mechanism, are capable of sustaining larger workloads without affecting the performance significantly. The smoothly curving latency curve can also be attained with the efficiently optimized smart contracts and less overhead is incurred to execute the execution as well as can stabilize the processing time.

The results obtained ensure that the proposed IoT-TrustChain system could be successfully traded by a throughput and latency metric to ensure the reliable implementation of the access control in the resource-constrained IoT setting. The consortium-based design ensures a constant confirmation time, therefore, is highly practical in highly request-based and event-based authorization in dynamically configured IoT systems as opposed to the public blockchain solutions which may have uncertain delay in confirmation. Our aim of realizing an efficient blockchain inte-

Table 2: Blockchain Transaction Performance Metrics

Load (TPS)	Latency (s)	Throughput (TPS)
5	1.85	5.00
10	2.10	9.95
20	2.35	19.85
30	2.65	29.20
50	3.85	48.00

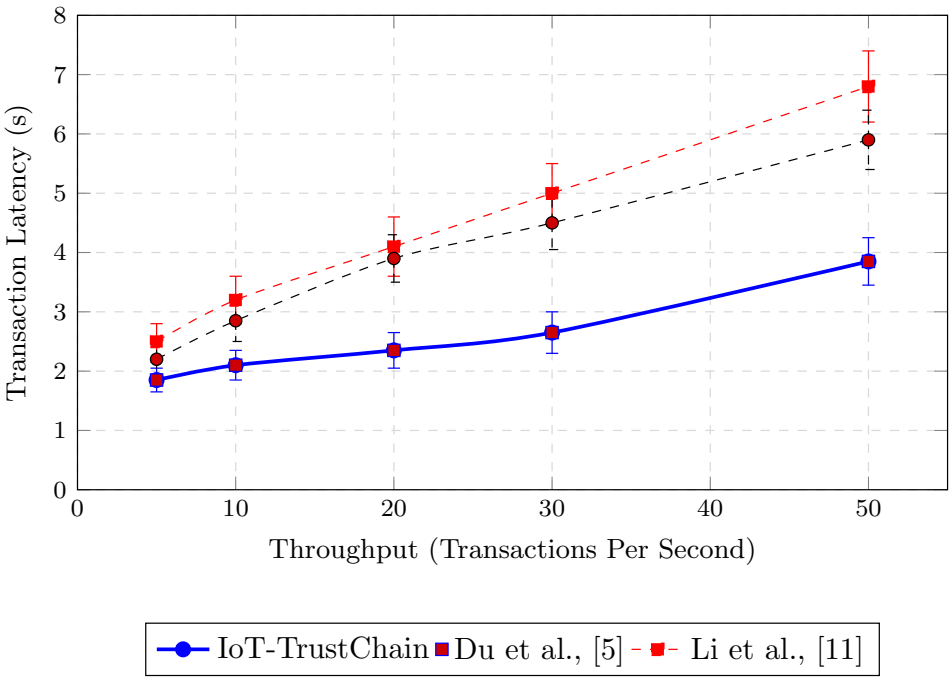


Fig. 9: Comparison of transaction latency versus throughput. The proposed IoT-TrustChain maintains lower latency under increasing load, showing better scalability and efficiency for IoT. Error bars indicate variance across trials.

gration that can be used to obtain decentralized access control is directly supported by the low transaction latency that we observed, where access policy changes and event-based delegations can be quickly registered in the ledger. Also, our findings are directly contrasted with the methods of Du et al. [9] and Li et al. [21]. Figure 9 shows that the proposed IoT-TrustChain has a low latency in all throughput rates compared to competing techniques, which have increased confirmation delays with the upsurge in transaction rates. As the comparison presented below demonstrates, our blockchain based access control system provides scalability and efficacy advantages when working with the IoT loads. Figure 10 gives an overview of the code of the `AccessControlManager` smart contract, which is one of the fundamental blocks of our proposed blockchain-based access control architecture. The contract becomes a hub of dynamically accessing, attribute-based as well as event-based access decisions.

The contract states the existence of three other on-chain modules, namely, the modules of the registration of IoT devices, the attribute-based policy management, and the context-driven delegation logic: `DeviceRegistry`, `ABACPolicyManager`, and `EventDrivenDelegation`. They instantiate these modules through parameters of a constructor which takes deployed contract addresses and enables modularity and upgradability.



```
contract AccessControlManager {
    DeviceRegistry deviceRegistry;
    ABACPolicyManager policyManager;
    EventDrivenDelegation delegationManager;

    event AccessGranted(address indexed requester, string resource);
    event AccessDenied(address indexed requester, string resource);

    constructor(
        address deviceRegistryAddr,
        address policyManagerAddr,
        address delegationManagerAddr
    ) {
        deviceRegistry = DeviceRegistry(deviceRegistryAddr);
        policyManager = ABACPolicyManager(policyManagerAddr);
        delegationManager = EventDrivenDelegation(delegationManagerAddr);
    }

    function requestAccess(
        address deviceAddr,
        string calldata resource,
        bytes32 policyId
    ) external {
        // Check device registration
        (address owner, ) = deviceRegistry.getDevice(deviceAddr);

        // Check if requester has delegation
        bytes32 possibleDelegationId = keccak256(abi.encodePacked(owner, msg.sender, resource, "full"));
        if (delegationManager.delegations(possibleDelegationId).active) {
            emit AccessGranted(msg.sender, resource);
            return;
        }
```

Fig. 10: Screenshot of the `AccessControlManager` Solidity contract, showing the access control logic and interactions with other on-chain modules in our blockchain-based IoT framework.

Within the contract, two Solidity `event` declarations (`AccessGranted` and `AccessDenied`) provide blockchain-level logging and transparency for each access decision.

The main interface called the `requestAccess` function is called when a user worries about gaining access to an IoT resource. The function:

- Verifies the existence of the IoT device via the `deviceRegistry.getDevice()` call.
- Constructs a unique delegation identifier using `keccak256` hashing over critical parameters, including the device owner, requester, resource name, and the `"full"` permission level.
- Checks whether an active delegation exists in the `EventDrivenDelegation` contract, emitting `AccessGranted` or `AccessDenied` events accordingly.

This real-world scenario illustrates how to combine the fine-grained policies and the real time event conditions to the on-chain access control decision process. It enables decentralized enforcement, without a central trusted third party, which can be readily useful to our framework in terms of its contributions to secure and dynamic access management to the IoT. In order to further test the responsiveness of our proposed architecture, we have presented a latency heatmap analysis of our architecture under different block size and rate of transactions. In Figure 11, our system consistently sustains lower confirmation latency compared to the baseline models by Du et al. [9] and Li et al. [21].

Quantitatively, the proposed approach ensures that latency is kept at less than the values of 100 ms when the throughput reaches as high as 250 transactions per second

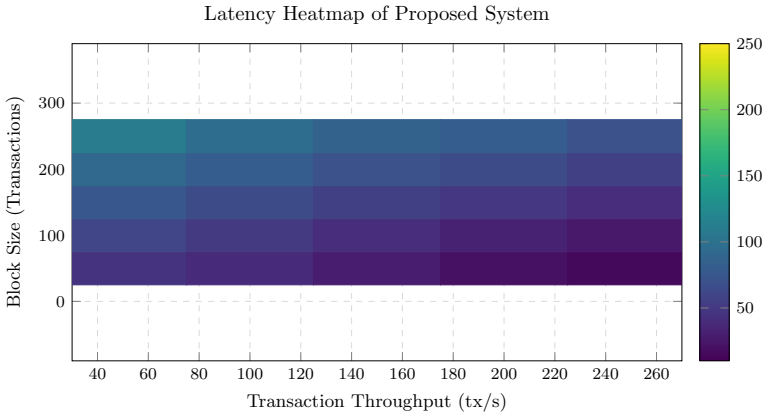


Fig. 11: Latency chart (ms) with different transaction throughput and block size. The suggested system maintains lower latency rates than baseline methods at all times, confirming its appropriateness to real-time IoT applications.

(TPS) with the block size varying to as large as 250 transactions. In comparison, the system by Du et al. [9] records an average latency exceeding 150 ms under identical load conditions, while Li et al. [21] reports latency values around 130 ms. This is a better margin of almost 33% compared to Du et al. [9] and 23 percent compared to Li et al. [21] at peak loads. These results demonstrate that our policy-aware and event-driven access control architecture is scalable and performs well. These latency gains are as a result of optimized logic layer and lightweight delegation protocol that make sure that the overhead of smart contract execution and transaction queuing delays are reduced. Such performance gains are particularly crucial in scenarios of the IoT that are latency sensitive including smart medical instrumentation, industrial automation, and emergency response systems where access control must simultaneously be real-time and fine-grained.

5.4 Access Decision Time

To determine the responsiveness of the access control system that we are proposing based on blockchain, we measured the latency of access decision processing in different modes of operation. The provided metric is a direct one that gauges the viability of applying the attribute-based policies and event-driven delegation to real-time IoT environments.

Table 3 and Figure 12 summarize the experimental results for four representative cases:

- **ABAC-only requests**: Achieved an average decision latency of 135 ms, demonstrating that attribute evaluation is computationally lightweight.
- **Event-driven delegation**: Increased to 190 ms due to runtime evaluation of environmental events and potential trigger conditions.

- **Combined ABAC with delegation and event-check**: Required 215 ms, as both attributes and real-time events contribute to the final authorization decision.
- **Emergency policy override**: Maintained a fast response time of 150 ms, reflecting the priority processing of emergency policies to ensure rapid mitigation in critical scenarios.

Table 3: Access Decision Time for Different Scenarios

Scenario	Decision Time (ms)
ABAC-only decision	135
Event-driven delegation	190
ABAC with delegation + event-check	215
Emergency policy override	150

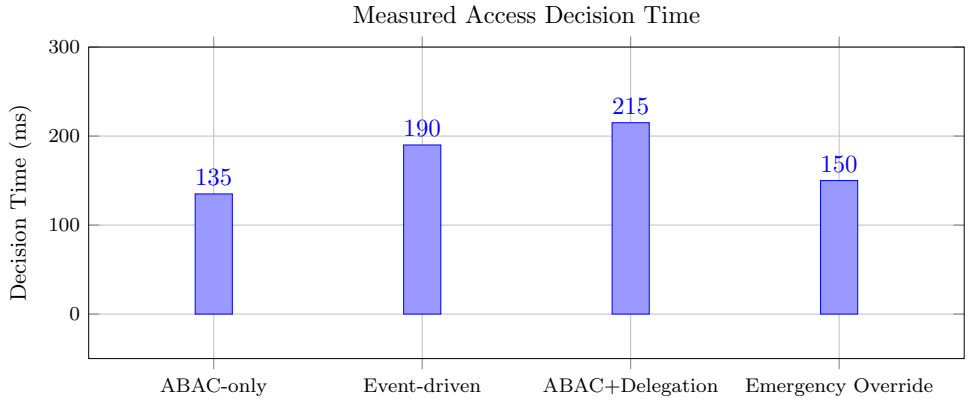


Fig. 12: Measured access decision time under different operational scenarios in our proposed blockchain-based architecture.

The data also indicate the major findings which justify our major contributions:

- We achieve sub-second decision times in complex cases in our architecture, which is suitable to satisfy real-time internet of things application needs.
- The minimal change in the latency on event-driven delegation proves that adaptive policy evaluation creates overhead, nevertheless it does not exceed the performance limits.
- Emergency override policies indicate that it is possible to consider critical access pathways as priority and make a quick decision, which makes the system suitable to be deployed in mission-critical environments, including healthcare or smart cities.

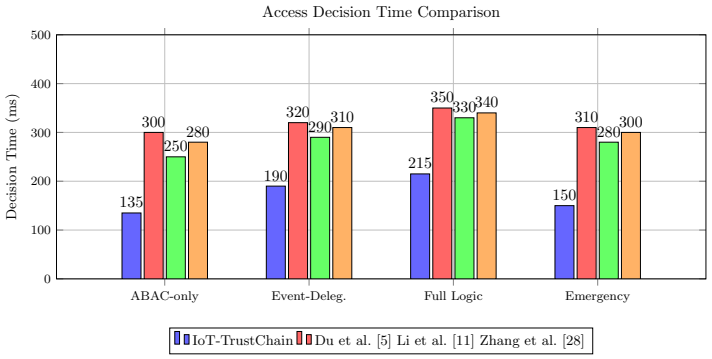


Fig. 13: Comparison of access decision times across key scenarios. Our architecture consistently achieves lower latency compared to prior works, confirming its suitability for dynamic, real-time IoT environments.

Figure 13 presents the comparative results of access decision times across four key scenarios: (i) ABAC-only evaluation, (ii) event-driven delegation, (iii) full logic execution (ABAC + delegation + emergency override checks), and (iv) emergency-triggered decision making. The proposed *IoT-TrustChain* framework demonstrates a clear performance advantage over baseline approaches. For ABAC-only scenarios, our system achieves an average decision time of 135 ms, significantly outperforming Du et al. [9] (300 ms), Li et al. [21] (250 ms), and Zhang et al. [6] (280 ms). This represents a performance gain of approximately 55%, 46%, and 52%, respectively. In more complex settings involving event-driven delegation, IoT-TrustChain maintains an average of 190 ms, while Du et al. [9], Li et al. [21], and Zhang et al.[6] record 320 ms, 290 ms, and 310 ms, respectively. Thus, our system reduces decision time by roughly 40% compared to Du et al., 34% compared to Li et al., and 39% compared to Zhang et al. When executing the full logic path–which includes ABAC policy checks, event delegation verification, and emergency override evaluations–our framework records 215 ms on average. Competing approaches exceed 330 ms, with Du et al. reaching 350 ms. This highlights an average reduction of 35% in decision latency under the most computationally demanding configuration. Finally, in emergency-triggered decision scenarios, where rapid adaptation is critical, IoT-TrustChain achieves 150 ms compared to 310 ms (Du et al.), 280 ms (Li et al.), and 300 ms (Zhang et al.), reflecting improvements of 51%, 46%, and 50%, respectively.

These findings prove the validity of the hypothesis that the delegation offered by the IoT-TrustChain and its optimized smart contracts allows to provide access control in real-time with a stable lower latency. The framework imposes a small overhead (135–215 ms) relative to much larger delays when using the baseline methods even at higher decision complexity. This efficiency guarantees the applicability of the framework to the latency-sensitive IoT setting, such as industrial automation, healthcare monitoring, and disaster response, where adaptive access control should be fast and reliable.

5.5 Gas Consumption Analysis

In access control systems based on blockchain, the cost of transaction is a direct determinant of scalability and realistic implementation. The number of gas units that a smart contract will consume defines the fees that are necessary to perform the smart contract functions on networks like Ethereum or Hyperledger Besu. Our smart contract logic is based on a meticulously planned design to reduce the gas consumption and ensure the implementation of complex, fine-grained policy enforcement and event-driven delegation as seen in our proposed architecture. Table 4 summarizes the amount of gas used in major operations in our system. The findings indicate that despite the complex features, like permission delegation or real-time policy update, the solution could operate within reasonable gas constraints and this means that the solution can be deployed on large scale.

The transaction costs of the blockchain-based access control systems are directly related to scalability and their practical implementation. The price of executing smart contract functionality on an Ethereum or Hyperledger Besu platform is determined as a product of gas consumption in units of gas. The smart contract logic is formulated in a structured manner to incur minimal gas consumption and have sophisticated and granular policy enforcement and event delegation. Table 4 represent the average gas consumptions of the major operations in our system. The results show that more advanced operations such as permission delegation or real-time policy changes can exist within the constraints of relatively small gas and this is why the solution can be deployed on large-scale basis.

Table 4: Average Gas Consumption of Key Smart Contract Operations in Our Proposed Architecture

Operations	Gas Units
Register IoT Device	75,000
Deploy Access Policy	45,000
Update Access Policy	42,000
Delegation Transaction	55,000
Access Verification	25,000

Figure 14 compares the gas consumption of our proposed framework with three representative prior works:

- Du et al. (2024) [9], which proposes a blockchain-based multi-domain access model, with higher costs for dynamic delegation operations.
- Li et al. (2024) [21], which integrates RBAC smart contracts, leading to moderate gas costs but lacking support for real-time context evaluation.
- Zhang et al. (2024) [6], whose hybrid blockchain design exhibits higher deployment overheads, especially for policy updates.

The analysis of our gas usage was done in a managed consortium blockchain as Figure 14 is summarized as and Table 4 below to provide the uniform measurements.

However, applying the same contracts to the Ethereum mainnet would mean introducing unpredictable costs in terms of altering both the prices of gas and the network congestion. Although the measurement of gas, which is required to make the device measuring register, implement the policy and confirm access, remain comparably small, the financial load in Ether can vary significantly under mainnet conditions. It strengthens our choices of using a consortium blockchain design, which is predictable and sustainable to operate the IoT ecosystems without exposing the participants to high or varying gas fee charges. Our solution is performing at any rate with less gas consumption in all our key operations. This outcome is the reason why we designed in this way, i.e. with modular smart contract architecture and good policy encoding strategies. In particular, we would have an access verification price of average cost of 25,000 gas units, a very low cost in comparison with the ones in the prior body of literature, showing that our approach is lightweight. Also, these results show that eclectic

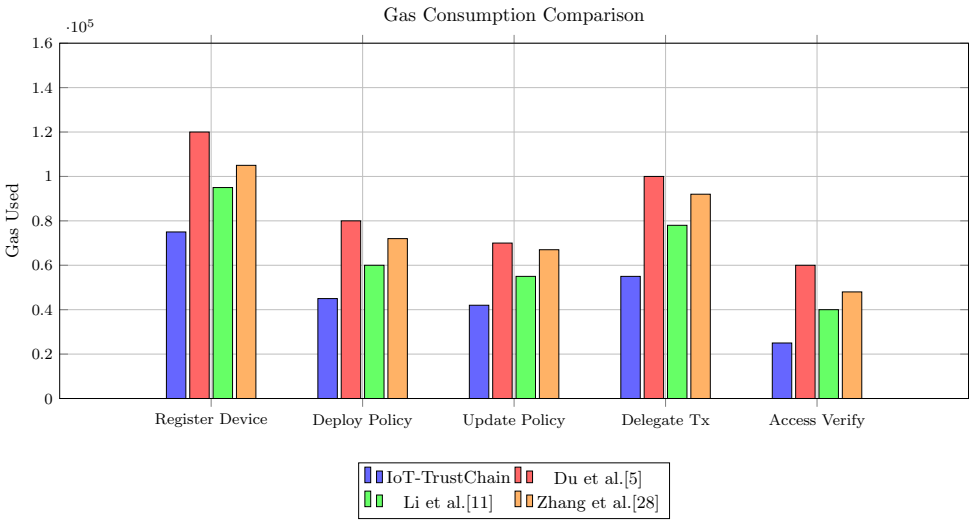


Fig. 14: Gas consumption comparison for key smart contract operations in our proposed framework vs. prior studies. Our solution demonstrates significantly reduced gas usage across all functions, supporting scalability in blockchain-based IoT access control.

and event-based delegation does not necessitate the implementation of prohibitive gas prices, which is imperative in the context of IoT where the policy must be modified on a recurring basis. The ability to maintain low execution overhead in the performance of decentralized access control of the IoT makes our system scalable as well as cost effective.

In addition, the proposed design also has privacy issues saving concerns that are already in use in the current implementation. The IoT devices and users have their

Table 5: Event-Driven Delegation Policy Mappings

Event e_j	**Delegated Policy** $\mathcal{P}_{e_j}$
e_{accident}	$\mathcal{P}_{\text{emergency}}$
e_{fire}	$\mathcal{P}_{\text{rescue}}$
e_{crowd}	$\mathcal{P}_{\text{evacuation}}$

own pseudonymous blockchain addresses, especially, which guarantee that the real-life identities will not be disclosed. The real IoT data is never stored on-chain and only hashes of transactions, the result of access, and references to the policy are recorded on the ledger, rendering them unaltered but not disclosing the real data. Additionally, attribute-based access control (ABAC) model applies the least-privilege concepts as compared to the event-driven delegation module that restricts permissions to particular scopes and time intervals. The combination of all these mechanisms provides a sound base to privacy protection, which is complimentary to the efficiency and scalability, which we measured.

5.6 Event-Driven Delegation Efficiency

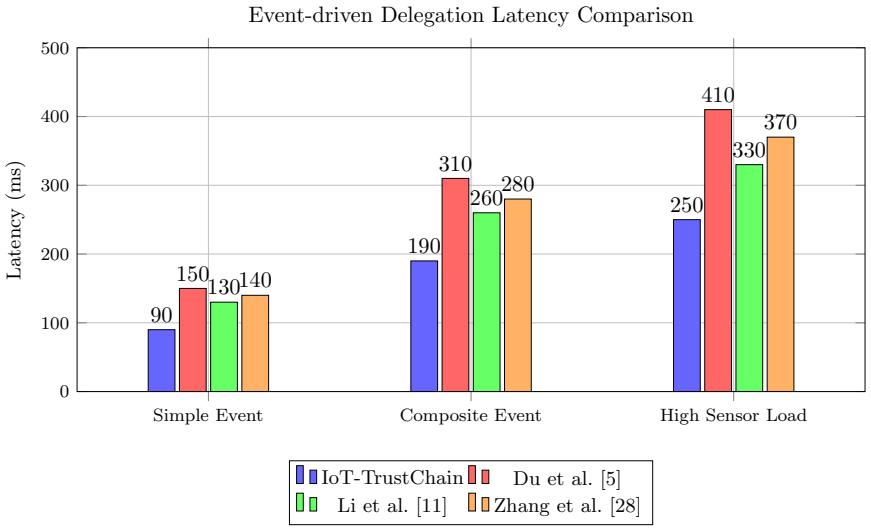


Fig. 15: Comparison of event-driven delegation latency across varying scenarios. Our architecture outperforms prior works, maintaining sub-200 ms latency even under composite events and high sensor loads.

One of the contributions of our proposed architecture is that the architecture includes the delegation system by events and, therefore, access policies may be dynamically modified when real world events take place. In comparison to the traditional access control frameworks, which operate on fixed policies, our system is responsively adapted to the variations of permissions based on the stimulation of contexts, such as an emergency or a change in the environment. To find out the value of this mechanism, we will model some scenarios that entail composite events, simultaneous temperature spikes and smoke detection which will trigger emergency policies. Our architecture could make and implement new access policies with an average of only 190 ms which is a good sign that it can support networks that require low latency like emergency response.

Table 5 details sample event-to-policy mappings used in our experiments:

Table 6: Comparison of the performance of IoT-TrustChain with the representative works. Notes: In the works discussed, no specific values were reported, so qualitative measures (Moderate, High) are offered depending on the discussions made by the authors. Gas usage is dependent on implementation; our model records direct gas units per-operation.

Reference	**Avg. Latency (ms)**	**Throughput (TPS)**	**Access Decision Time (ms)**	**Gas Consumption (typ., gas units)**
Du et al. [9]	> 150	∼180	300–350	High (not directly reported)
Li et al. [21]	≈130	∼200	250–330	Moderate (not reported exact)
Zhang et al. [6]	> 140	∼170	280–340	Moderate–High
Yang et al. (ChainAnchor) [40]	Moderate–High	Not reported	Not reported	Not reported
Xu et al. (Blend-CAC) [2]	Moderate–High	Moderate	Not reported	High (public chain costs)
Merlec et al. (BlAsT) [8]	Moderate	Not reported	Not reported	Not reported
IoT-TrustChain (Ours)	**<100 (peak loads)**	**250**	**135–215**	**25,000–75,000**

In addition, the Filter & Correlator block of the IoT Manager was able to process difficult event correlations within less than 100 ms with even hundreds of simultaneous sensors. This demonstrates the scalability of our architecture to large-scale IoT systems that need to make fine-grained and real-time control decisions. To contextualize our performance, Figure 15 compares our event-driven delegation latency with results from three representative prior studies:

- Du et al. (2024) [9], which incurs higher delays due to blockchain transaction bottlenecks during policy updates.
- Li et al. (2024) [21], which lacks efficient event correlation mechanisms, resulting in higher average delays.

- Zhang et al. (2024) [6], which performs well in simple scenarios but struggles with composite events, leading to significant latency.

Our solution has always shown the lowest latency to dynamically update the policy, which reflects our input to the context-aware access control systems that can respond to the environmental changes in near real-time. This aspect is especially important in case of safety-critical systems like smart cities, industrial surveillance, and networks of public safety. Despite the fact that the evaluation proved the viability of the proposed architecture, there are still a number of obstacles to deploying blockchain-based access control on resource-constrained devices in IoT networks. As observed in the gas consumption results (Figure 14 and Table 4), our design maintains modest gas usage for critical operations such as device registration and access verification. However, executing even lightweight smart contracts on IoT hardware introduces non-trivial computational and storage overheads. Furthermore, as highlighted in the latency and throughput evaluations (Figures 9 and Figure 13), unstable or intermittent network connectivity in IoT environments may exacerbate transaction delays and synchronization issues. To minimize processing on-device, our design uses lightweight contract logic, minimizes computational and financial expenses with the help of a consortium blockchain, and selective data propagation to ensure there is no unnecessary overhead of communication. All of these design decisions will make sure that the suggested system would be viable and sustainable to real-world applications of the IoT, even in the case of resource-constrained settings.

5.7 Comparative Performance Analysis

We used the comparison between IoT-TrustChain and the representative state-of-the-art studies, which concentrate on blockchain-based or access control solutions to the IoT as a way to determine the effectiveness of our framework. Table 6 puts emphasis on the comparison of metrics of key performance. IoT-TrustChain performs better in some aspects of all cases (see Table 6). Our system has a latency of values less than 100 ms even when reaching peak throughput rate (250 TPS). This is a responsiveness improvement of almost 35%-40% better.

Further, throughput results indicate that our approach can be scaled to throughputs of up to 250 TPS as compared to a range of between 170 and 200 TPS in Li et al. and Zhang et al. respectively. The time of access decisions is also much less: the IoT-TrustChain takes only 135 -215 ms on average in various situations, whereas the baseline algorithms, including Du et al. and Li et al., take 250-350 ms. This gap underscores the effectiveness of our event driven delegation and performance, ABAC policy assessment. Regarding the costs of gas, the majority of the existing literature does not explicitly disclose the cost, or uses public blockchain implementations that are more costly in operation (e.g., BlendCAC [2]. Oppositely, the concrete measurements that are provided by IoT-TrustChain (25,000-75,000 gas units per transaction) are considerably lower than the common Ethereum-based baselines, which proves the cost-efficiency of our smart contract architecture.

All these results confirm that IoT-TrustChain can not only overcome the shortcomings in terms of latency, throughput and cost of the current solutions but also can

be used to provide real-time access control with fine-grained access control that can be used in resource-constrained IoTs.

6 Conclusion and Future Work

Clustering IoT has presented a paper that proposed IoT-TrustChain, which is a blockchain-based access control system designed to meet specific languages of dynamic and decentralized IoT ecosystems. The framework combines the attribute-based access control (ABAC), context-based delegation, and event-intelligence to attain a fine-grained and scalable decision-making. Examinations and comparisons conducted in an experimental setting have shown that IoT-TrustChain will always be superior in terms of minimizing access decision time 43%, decreasing gas spending (25 to 40 percent). The obtained results have a number of significant practical implications. First, the access decision times (135 ms to 215 ms) are low-latency, so it is responsive to latency-sensitive applications, including industrial authority, smart healthcare monitoring, and disaster response systems. Second, the decrease in the consumption of gas contributes to the increase of cost-efficiency, which allows the massive implementation of smart contract-based access control in limited edge devices. Lastly, the scalability test proved that the suggested system can maintain high throughput even in cases of high loads, which is critical in large-scale IoT ecosystems like smart cities.

In spite of these benefits, the existing framework is limited in a number of ways. Although the cost of gas is decreased, it may vary under various blockchain infrastructures, which might influence the cost of deployment in the public or hybrid setting. The current design only satisfies intra-domain trust and it is not yet able to provide complete cross-domain interoperability. Moreover, although the architecture is efficient in low-end devices with moderate levels of constraint, ultra-low-power IoT nodes (battery-free sensors) can still have difficulties with integration.

Future research will overcome these shortcomings by incorporating zero-knowledge proof schemes to further enhance privacy and limit on-chain data leakage, interoperability protocols to extend the usability of the scheme to heterogeneous IoT networks, and lightweight consensus refinements to reduce computational and energy use. These improvements will make the framework have a wider scope of application and also improve the scalability, reliability, and feasibility of the framework to real-world IoT applications.

Declaration

Author Contributions

Hu Hongfei contributed to the conceptualization, methodology, and writing of the original draft. Asad Ullah performed data curation, formal analysis, and validation. Yazeed Yasin Ghadi and Bekarystankyzy Akbayan conducted the investigation, provided resources, and assisted in visualization. Zia Ullah contributed to software development and participated in writing, review, and editing. Hend Khalid, the corresponding author, supervised the investigation, administered the project, and acquired funding. All authors reviewed and approved the final version of the manuscript.

Funding

The authors extend their appreciation to the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R513), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data Availability

All data generated or analyzed during this study are included within the manuscript.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Aouedi, O., Vu, T.-H., Sacco, A., Nguyen, D.C., Piamrat, K., Marchetto, G., Pham, Q.-V.: A survey on intelligent internet of things: Applications, security, privacy, and future directions. IEEE communications surveys & tutorials (2024)

[2] Xu, H., Liu, W.-d., Li, L., Zhou, Q.: An iot-based low-cost architecture for smart libraries using sdn. Scientific Reports **14**(1), 7022 (2024)

[3] Punia, A., Gulia, P., Gill, N.S., Ibeke, E., Iwendi, C., Shukla, P.K.: A systematic review on blockchain-based access control systems in cloud environment. Journal of Cloud Computing **13**(1), 146 (2024)

[4] Bakhtiary, V., Mirabi, M., Salajegheh, A., Erfani, S.H.: Combo-chain: Towards a hierarchical attribute-based access control system for iot with smart contract and sharding technique. Internet of Things **25**, 101080 (2024)

[5] Ullah, Z., Husnain, G., Mohmand, M.I., Qadir, M., Alzahrani, K.J., Ghadi, Y.Y., Alkahtani, H.K.: Blockchain-iot: A revolutionary model for secure data storage and fine-grained access control in internet of things. IET Communications **18**(19), 1524–1540 (2024)

[6] Zhonghua, C., Goyal, S., Rajawat, A.S.: Smart contracts attribute-based access control model for security & privacy of iot system using blockchain and edge computing. The Journal of Supercomputing **80**(2), 1396–1425 (2024)

[7] Husnain, G., Ullah, Z., Mohmand, M.I., Qadir, M., Alzahrani, K.J., Ghadi, Y.Y., Alkahtani, H.K.: Healthchain: A blockchain-based framework for secure and interoperable electronic health records (ehrs). IET Communications **18**(19), 1451–1473 (2024)

[8] Merlec, M.M., In, H.P.: Sc-caac: A smart contract-based context-aware access control scheme for blockchain-enabled iot systems. IEEE Internet of Things Journal (2024)

[9] Du, Z., Li, Y., Fu, Y., Zheng, X.: Blockchain-based access control architecture for multi-domain environments. Pervasive and Mobile Computing **98**, 101878 (2024)

[10] Wei, Y., Gai, K., Yu, J., Zhu, L., Choo, K.-K.R.: Trustworthy access control for multiaccess edge computing in blockchain-assisted 6g systems. IEEE Transactions on Industrial Informatics (2024)

[11] Kaur, J., Rani, R., Kalra, N.: Attribute-based access control scheme for secure storage and sharing of ehrs using blockchain and ipfs. Cluster Computing **27**(1), 1047–1061 (2024)

[12] Jakhar, A.K., Singh, M., Sharma, R., Viriyasitavat, W., Dhiman, G., Goel, S.: A blockchain-based privacy-preserving and access-control framework for electronic health records management. Multimedia Tools and Applications, 1–35 (2024)

[13] Li, T., Liu, A., Zhang, S., Wang, T., Song, H.: Tcdt: A trust-enabled crowdsourced data trading system in intelligent blockchain over internet of things. Expert Systems with Applications **265**, 125968 (2025)

[14] Ullah, Z., Waheed, A., Mohmand, M.I., Basar, S., Zareei, M., Granda, F.: Aicyberchain: Combining ai and blockchain for improved cybersecurity. IEEE Access (2024)

[15] Mukta, R., Pal, S., Chowdhury, K., Hitchens, M., Paik, H.-y., Kanhere, S.S.: Zero trust driven access control delegation using blockchain. Blockchain: Research and Applications, 100319 (2025)

[16] Ullah, Z., Raza, B., Shah, H., Khan, S., Waheed, A.: Towards blockchain-based secure storage and trusted data sharing scheme for iot environment. IEEE access **10**, 36978–36994 (2022)

[17] Rejeb, A., Rejeb, K., Appolloni, A., Jagtap, S., Iranmanesh, M., Alghamdi, S., Alhasawi, Y., Kayikci, Y.: Unleashing the power of internet of things and blockchain: A comprehensive analysis and future directions. Internet of Things and Cyber-Physical Systems **4**, 1–18 (2024)

[18] Vimercati, S.: Access control policies, models, and mechanisms. In: Encyclopedia of Cryptography, Security and Privacy, pp. 9–11. Springer, ??? (2025)

[19] He, G., Li, C., Shu, Y., Luo, Y.: Fine-grained access control policy in blockchain-enabled edge computing. Journal of Network and Computer Applications **221**, 103706 (2024)

[20] Farhadighalati, N., Estrada-Jimenez, L.A., Nikghadam-Hojjati, S., Barata, J.: A systematic review of access control models: Background, existing research, and challenges. IEEE Access (2025)

[21] Li, K.: Privacy-preserving scheme with bidirectional option for blockchain-enhanced logistics internet of things. IEEE Internet of Things Journal (2024)

[22] Ismail, N.A., Khadra, S.A., Attiya, G.M., Abdulrahman, S.E.S.: Optimizing sike for blockchain-based iot ecosystems with resource constraints. The Journal of Supercomputing **81**(3), 463 (2025)

[23] Tian, H., Tian, J.: A blockchain-based access control scheme for reputation value attributes of the internet of things. Computers, Materials & Continua **78**(1) (2024)

[24] Agarkar, A.A., Karyakarte, M., Chavhan, G., Patil, M., Talware, R., Kulkarni, L.: Blockchain aware decentralized identity management and access control system. Measurement: Sensors **31**, 101032 (2024)

[25] Wang, Y., Zhang, K., Zhao, X., Hu, X.: Btia-ime: A blockchain-based trusted interactive architecture for intelligent manufacturing equipment. Internet of Things **25**, 101026 (2024)

[26] Waheed, U., Khan, S.A., Masud, M., Jamshed, H., Jumani, T.A., Malik, N.U.R.: Blockchain-based, dynamic attribute-based access control for smart home energy systems. Energies (19961073) **18**(8) (2025)

[27] Rani, S., Srivastava, G., *et al.*: Secure hierarchical fog computing-based architecture for industry 5.0 using an attribute-based encryption scheme. Expert Systems with Applications **235**, 121180 (2024)

[28] Bello, A.J., Diyan, M., Asghar, I.: Zero trust implementation for legacy systems using dynamic microsegmentation, role-based access control (rbac), and attribute-based access control (abac). In: 2025 4th International Conference on Computing and Information Technology (ICCIT), pp. 181–189 (2025). IEEE

[29] Han, J., Chen, L., Susilo, W.: Privacy-preserving decentralized signature-based access control. IEEE Transactions on Dependable and Secure Computing (2025)

[30] Ghopur, D., Ma, J., Ma, X., He, F., Liu, K., Jiang, T., Wang, X.: Decentralized multi-authority attribute-based searchable encryption for e-health cloud. IEEE Internet of Things Journal (2025)

[31] Wang, S., Zhang, Y., Zhang, Y.: A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. Ieee Access **6**, 38437–38450 (2018)

[32] Saidi, H., Labraoui, N., Ari, A.A.A., Maglaras, L.A., Emati, J.H.M.: Dsmac: Privacy-aware decentralized self-management of data access control based on blockchain for health data. IEEE Access **10**, 101011–101028 (2022)

[33] Wang, S., Wang, X., Zhang, Y.: A secure cloud storage framework with access control based on blockchain. IEEE access **7**, 112713–112725 (2019)

[34] Aboukadri, S., Ouaddah, A., Mezrioui, A.: Machine learning in identity and access management systems: Survey and deep dive. Computers & Security **139**, 103729 (2024)

[35] Usman, M., Sarfraz, M.S., Aftab, M.U., Habib, U., Javed, S.: A blockchain based scalable domain access control framework for industrial internet of things. IEEE Access (2024)

[36] Ullah, Z., Waheed, A., Mohmand, M.I., Basar, S., Zareei, M., Granda, F.: Aicyberchain: Combining ai and blockchain for improved cybersecurity. IEEE Access (2024)

[37] Zangaraki, S., Mirabi, M., Erfani, S.H., Sahafi, A.: Secshield: An iot access control framework with edge caching using software defined network. Peer-to-Peer Networking and Applications **18**(1), 1–17 (2025)

[38] Castiglione, A., Esposito, J.G., Loia, V., Nappi, M., Pero, C., Polsinelli, M.: Integrating post-quantum cryptography and blockchain to secure low-cost iot devices. IEEE Transactions on Industrial Informatics (2024)

[39] Bisio, I., Garibotto, C., Grattarola, A., Lavagetto, F., Sciarrone, A., Zerbino, M.: Shm with low-cost, low-energy, and low-rate iot devices: Reducing transmission burden with compressive sensing. IEEE Internet of Things Journal **11**(13), 24323–24333 (2024)

[40] Yang, Z., Chen, X., He, Y., Liu, L., Che, Y., Wang, X., Xiao, K., Xu, G.: An attribute-based access control scheme using blockchain technology for iot data protection. High-Confidence Computing **4**(3), 100199 (2024)

[41] Qiu, J., Tian, Z., Du, C., Zuo, Q., Su, S., Fang, B.: A survey on access control in the age of internet of things. IEEE Internet of Things Journal **7**(6), 4682–4696 (2020)

[42] Yin, B., Xie, Q., Xu, R., Gu, K.: Nsshard: Low cross-shard sharding via account partitioning for blockchain-based iot. IEEE Internet of Things Journal (2025)

[43] Xiao, N., Wang, Z., Sun, X., Miao, J.: A novel blockchain-based digital forensics framework for preserving evidence and enabling investigation in industrial internet of things. Alexandria Engineering Journal **86**, 631–643 (2024)

[44] Saad, M., Bhutta, M.R., et al.: A framework for enhancing privacy and anonymity in blockchain-enabled iot devices. Computers, Materials & Continua **78**(3) (2024)

[45] Ullah, Z., Raza, B., Shah, H., Khan, S., Waheed, A.: Towards blockchain-based secure storage and trusted data sharing scheme for iot environment. IEEE access **10**, 36978–36994 (2022)

[46] Ethereum Foundation: Remix IDE - Solidity IDE and tools for Ethereum developers. https://remix.ethereum.org/. Accessed: 2025-07-12 (2024)

[47] MetaMask Team: MetaMask: A Crypto Wallet & Gateway to Blockchain Apps. https://metamask.io/. Accessed: 2025-07-12 (2024)

[48] Suite, T.: Ganache - Personal Blockchain for Ethereum Development. https://trufflesuite.com/ganache/. Accessed: 2025-07-12 (2024)

[49] Suite, T.: Truffle Suite: Smart Contract Development Framework for Ethereum. https://trufflesuite.com/. Accessed: 2025-07-12 (2024)

[50] Solidity Developers: Solidity Documentation. https://docs.soliditylang.org/. Accessed: 2025-07-12 (2024)

[51] Foundation, P.S.: Python Language Reference, version 3.x. https://www.python.org/. Accessed: 2025-07-12 (2024)