applied sciences

MDPI

Article

Hyperledger Fabric Access Control for Industrial Internet of Things

Dong-Her Shih [1], Ting-Wei Wu [1], Ming-Hung Shih [2,*], Guan-Wei Chen [1] and David C. Yen [3]

[1] Department of Information Management, National Yunlin University of Science and Technology, Douliu 64002, Taiwan; shihdh@yuntech.edu.tw (D.-H.S.); portraits1129@gmail.com (T.-W.W.); M10823021@yuntech.edu.tw (G.-W.C.)
[2] Department of Electrical and Computer Engineering, Iowa State University, 2520 Osborn Drive, Ames, IA 50011, USA
[3] Jesse H. Jones School of Business, Texas Southern University, 3100 Cleburne Street, Houston, TX 77004, USA; David.Yen@tsu.edu
* Correspondence: mshih@iastate.edu

Abstract: The Industrial Internet of Things (IIoT) plays an important role in Industry 4.0, but the existing IIoT systems could be vulnerable to a single point of failure and malicious attacks, failing to provide reliable services. IIoT devices have some particularities, such as mobility, limited performance, and distributed deployment, which are challenging to traditional centralized access control methods in the large-scale IIoT environment. To resolve the challenges, we propose an access control system for the Industrial Internet of Things. The system contains three smart contracts: device contract (DC), policy contract (PC), and access contract (AC). The device contract provides a method of storing the URL of the resource data generated by the equipment and a query method. The policy contract provides the function of managing the attribute-based access control (ABAC) of the administrator user. The access contract is the core program that implements the access control method for ordinary users. Combining ABAC and blockchain technology provides decentralized, fine-grained, and dynamic access control management for IIoT.

Keywords: industrial internet of things; blockchain; hyperledger; smart contract; access control

check for updates

Citation: Shih, D.-H.; Wu, T.-W.; Shih, M.-H.; Chen, G.-W.; Yen, D.C. Hyperledger Fabric Access Control for Industrial Internet of Things. *Appl. Sci.* **2022**, *12*, 3125. https://doi.org/10.3390/app12063125

Academic Editor: Zheng Chang

Received: 30 January 2022
Accepted: 15 March 2022
Published: 18 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Industry 4.0, or the fourth industrial revolution, is the trend of future development to achieve smarter manufacturing processes. The Industrial Internet of Things (IIoT) is a branch of IoT that connects industrial assets and provides the best solution to the difficulties encountered in the manufacturing industry, focusing on the industrial sector, and generating data to help decision makings. It has the potential to improve and enhance industrial assets, such as machinery and equipment, control systems, business processes, etc., which leads to the smart industries. IIoT interconnects industrial systems and physical objects using sensors and actuators. With the integration of wireless sensor networks, communication protocols, and Internet infrastructure, IIoT can provide smart and efficient industrial operations. Nevertheless, it is a technology of processing and exchanging large amounts of confidential data between various industrial machines, which could be exposed to multiple cyber threats.

As of today, the security level of many IIoT devices is low. The lack of a complete key management system, high-efficiency identity authentication, low fault tolerance, and many other problems have made the IIoT devices easy targets for the attackers to access illegally, causing serious consequences [1,2]. Access control is a common technology protecting resources from unauthorized access and has been widely used in various systems and environments. The traditional access control methods include discretionary access control (DAC), role-based access control (RBAC), and mandatory access control (MAC). However,

these methods were designed for centralized systems with the following shortcomings: single point of failure, low scalability, low reliability, and low throughput. These are the major challenges for traditional access control to meet the requirements of IIoT devices.

To solve the emerging security challenges of IIoT, there have been many proposals to use blockchain technology due to its decentralized nature. The integration of blockchain and IIoT has many advantages in improving and solving certain security issues. Blockchain is a distributed ledger technology that uses cryptographic sealing and tamper-proof technology to connect multiple nodes in a peer-to-peer (P2P) network without any involvement of the third party using the consensus algorithms such as Proof-of-Work, Proof-of-Stake, Proof-of-Assets, Proof-of-Elapsed-Time, Byzantium Practical Byzantine Fault Tolerance, etc. [3,4].

Zhang et al. [5] proposed an access control scheme based on Ethereum smart contracts which include three smart contracts: access control contract (ACC), judge contract (JC), and registered contract (RC). ACC implements policy-based authorization by checking the behavior of the object. JC is used to judge the wrong behavior and return the corresponding punishment. RC is used to register the above two smart contracts and provide updates, deletions, and other operations. The architecture is implemented through a personal computer, a laptop, and two Raspberry Pis. Liang et al. [6] proposed a security-based Fabric-based data transmission technology and implemented a transaction center in the IIoT, which solved the problems of low security, high management costs, and difficult supervision. Puri et al. [7] proposed a decentralized access control scheme for Internet of Things (IoT) devices using a single smart contract to reduce the communication cost between nodes. Nodes, also known as management centers, are designed to interact with devices, avoiding direct interaction between blockchain and IoT devices. It leads to six advantages: immediacy, accessibility, parallelism, lightweight, scalability, and transparency. Wang et al. [8] proposed an attribute-based access control method for the IoT, which saves attribute data through the blockchain. This method avoids data tampering and simplifies the access control protocol to meet the computing power of IoT devices.

The rapid development of the Internet of Things requires more complete distributed access control standards. Blockchain technology has four advantages of evaluation control, namely decentralization, encryption, scalability, and tamper resistance. As of today, blockchain technology has been developed to version 3.0. As its core technology, smart contracts build a safe and reliable operating environment for applications and endow the blockchain with more powerful functions. Therefore, based on the existing access control methods, many scholars have proposed various IoT access control methods by combining them with blockchain and smart contracts. IIoT looks very similar to IoT, yet there are many differences between them in terms of practical applications and security requirements. The main difference between the two lies in how they are used and the types of services they provide. Consumer-grade devices mainly provide services through the Internet of Things. The most important area for using IIoT is mission-critical systems. More differences can be found in respective architectures, communication, connectivity, data volume, latency, speed, and scalability.

Yeasmin and Baig [9] claimed that the IIoT is a machine-to-machine communication technology consisting of various networked industrial machines which generate, process, and exchange large amounts of data related to missions and critical safety infrastructure. IIoT also uses communication protocols to share confidential information, which is exposed to many cyber threats. IIoT can help make better business decisions and provide better scalability, connectivity, and efficiency. However, the centralized nature of IIoT devices makes them vulnerable to many cyber threats. Due to the heterogeneous networks used by different devices, the security problem of IIoT has become more critical. Since IIoT devices have been used in mission and critical security systems, security vulnerabilities need to be resolved and improved. First, the heterogeneity of IoT objects makes them vulnerable to different network attacks when connected to existing network systems. Secondly, existing security tools, such as encryption, firewalls, and intrusion detection systems, are generally not suitable for IIoT environments due to the resource-constrained nature of IoT devices

and the incompatibility between IoT standards and network protocols. Finally, inconsistent maintenance and updates of systems can make it difficult to test vulnerabilities in IIoT networks. With the rapid development of IIoT, more problems such as energy efficiency, performance, coexistence, interoperability, security, privacy, etc., still need to be resolved. One of the main requirements that IIoT needs to address is security and privacy, which have always been the focus of IIoT and IoT devices.

Since IIoT is an open, distributed, and heterogeneous system, achieving security has become a huge challenge. With IIoT, industrial infrastructure is vulnerable to cyber threats and attacks. The reason behind this is sharing information with the cloud to achieve efficiency and performance improvements, making industrial infrastructure more interconnected and smarter. IIoT devices make decisions transparently and share them among stakeholders to help with asset tracking. Such transparency of sensitive data attracts malicious attackers. In addition, IIoT handles tasks and safety-critical systems that generate large amounts of data to help make smart and important business-related decisions. Therefore, protecting the generated data and communication protocols becomes extremely important. IIoT devices are vulnerable to man-in-the-middle (MITM) attacks, device hijacking, distributed denial-of-service (DDoS), and permanent denial of service (PDoS). These security issues will affect valuable and sensitive data, leading to a huge impact on the industrial infrastructure that implements IIoT. Therefore, there is an urgent need to improve the security of IIoT to protect mission-critical systems and industrial infrastructure [10,11].

This research applies blockchain technology to the industrial Internet of Things access control, and proposes an architecture based on smart contracts and attribute-based access control (ABAC), which consists of policy contract (PC), device contract (DC), and access contract (AC) to provide dynamic access control management and solve the access control problems in the Industrial Internet of Things, and realize the trusted access mechanism of the Industrial Internet of Things system. This research also uses a one-time URL to ensure the security of data sharing once and provide a data platform for secure transactions between manufacturers.

2. Preliminary

This section will describe the Internet of Things, Industrial Internet of Things, the concept of blockchain, the Hyperledger platform, IIoT security issues, and access control.

2.1. Internet of Things

In 1995, Bill Gates proposed the concept of the Internet of Things in "The Road to the Future". The term Internet of Things was first proposed in 1999 by Kevin Ashton of the Automatic Identification Center of the Massachusetts Institute of Technology. The International Telecommunication Union (International Telecommunication Union, ITU) formally proposed the concept of the Internet of Things (IoT) in 2005. Later, in 2009, IBM proposed the concept of "Smart Earth". The Internet of Things (IoT) once again attracted widespread attention from all walks of life. The Internet of Things consists of actual objects, such as home appliances, machines, vehicles, etc., along with embedded sensors and APIs, and other devices, to form information connection and network exchange [8].

Green [12] published the seven-layer architecture at the Internet of Things World Forum. The functions of each layer are described below:

- Physical Devices and Controllers: Multiple physical devices and controllers are considered as "Things" in the Internet of Things. There exist all types of sensors, devices, machines, or intelligent edge nodes, to receive and transmit data between terminal equipment.
- Connectivity: The most important factor of IoT is the connectivity with reliable and instant data transmission. Different technology has been explored to propagate data such as radio frequency identification (RFID), cellular networks (4G/5G), Wi-Fi, Bluetooth, ZigBee, routers, and switches.

- Edge Computing: Edge computing is a decentralized computing architecture, where a huge number of services are processed by a central node to divide into smaller tasks and distribute to edge nodes for processing. The edge node is close to the terminal device, which can make the transmission speed and data processing faster and reduce the delay. It is suitable for processing big data. Edge software has many service applications, and some can be completely independent of vertical applications. The most typical application is Microsoft's Cognitive Service, an intelligent data analysis model based on Azure's machine learning. Setting up edge nodes in the place closest to the data source can provide the computing power of the cognitive service data analysis model. Without the need to send data to the cloud, costs on transmission time and analysis results can be reduced.
- Data Accumulation: Data accumulation is to convert real-time data to static format, filter, and reduce the amount of data before being sent to database storage. The stored data can then be used by the other applications when it is not timely.
- Data Abstraction: Data abstraction unifies data formats from different databases and integrates them into the same place. Identity verification and authorization could be used to protect data through regularization, and indexing could provide quick access.
- Applications: Applications on personal computers, smart devices, and other equipment could verify data analysis results, receive early warning notifications, and provide the correct information to support users on decision makings and actual actions.
- Collaborations and Processes: Collaborations and processes are critical in IoT since most actions require the involvement of multiple devices and users.

2.2. Blockchain Concept

Blockchain is the core technology of Bitcoin. Bitcoin was developed by Satoshi Nakamoto in 2008 as a new idea of electronic money by using a peer-to-peer network to create an electronic transaction system without relying on trust. Blockchain has two major characteristics: decentralization and non-modification. Decentralization is a special core feature of the blockchain where data is stored on different nodes, and each node needs to be self-verified and managed. The non-modification property will protect the data from being modified once the data have been written into the blockchain. Data written into blocks are protected by the hash method, making the data easy for verification but difficult to be cracked and returned to the original data. Blockchain platforms such as Bitcoin and Ethereum use proof of work (POW), which requires the high computing power of each miner node to solve complex computing problems, verify new blocks, and add them to the blockchain classification. The higher the computing resources a miner has, the higher the probability of winning the puzzle. The winning miner broadcasts the correct answer to the puzzle to other miners. Based on the current state of the blockchain, if all transactions are valid, they will be treated as one block [13,14].

2.3. Hyperledger Fabric

The popularity of virtual currency has aroused the world's attention to blockchain, but this type of blockchain belongs to the public chain and has the following disadvantages: (1) low transaction throughput (about seven transactions per second), (2) long transaction time (an hour per transaction), (3) high cost of computing resources due to the PoW consensus algorithm, (4) branching problem, where only the longest chain is valid when multiple chains of transactions are generated, (5) privacy issues with the ledger being public [11]. To solve the above problems, the Hyperledger Fabric, an open-source project of the Linux Foundation, introduced the modular blockchain architecture standard for enterprise blockchain platforms [15]. Specifically designed to be the foundation for the development of enterprise-level applications and industrial solutions, the open modular architecture uses plug-and-play components to address a wide variety of use cases. Hyerledger Fabric is an open and proven enterprise and decentralized general ledger platform with advanced privacy controls that allow only data sharing among network participants. It contains code

and contracts that exist in a decentralized blockchain network. Transactions are traceable but irreversible, which builds trust between organizations and allows businesses to make informed decisions, saving time and reducing costs.

Smart contracts enable blockchain systems to evolve from simple cryptocurrency platforms to general-purpose transaction systems. However, the consensus model of the Nakamoto blockchain also has its performance bottleneck. Nakamoto's consensus model is called the "Order-Execute" model., as shown in Figure 1 [16,17]. First, a consensus is formed on the order of transactions, and then all nodes execute them in sequence. After receiving the block, all nodes execute each transaction to verify and update the ledger after their correctness has been confirmed. This is the main reason for the lack of efficiency of the blockchain since every single transaction must be repeatedly executed by each node.

Figure 1. Order-execute architecture.

In addition to the efficiency issue, the "order-execute" model comes with some limitations. (1) Serialize execution: each node must rely on a "sort-execute" model to update, limiting the scalability. (2) Enforce confidentiality: transactions must be executed by all nodes, and the state saved in the blockchain must be public so that each node can read these states, which is not conducive to the confidentiality of trans-actions, and nodes can learn the operation and state of any smart contract.

Lack of efficiency, privacy, and high development costs make it more difficult for businesses to embrace blockchain technology. Unlike public networks, corporate networks cannot be used by any ordinary users without permission. The inter-enterprise network environment is the opposite of the public network. The inter-enterprise collaboration network is limited with fixed participants, real-name authentication, and access control. In a public blockchain, data and transaction logics are transparent to the public, which leads to private da-ta leakage. A new architecture called Execution-Order-Validate is proposed in Hyperledger Fabric, Fabric Blockchain requires its members to register through a trusted membership service to interact with the blockchain. As shown in Figure 2, all transactions will be executed by some of the first designated nodes. If the results are consistent, a consensus will be formed. Transactions without a consensus will be deleted. Transactions with a consensus will be ordered and packaged into blocks. The block broadcasts to all nodes to verify the transaction, and the status can be updated if the verification result is correct.

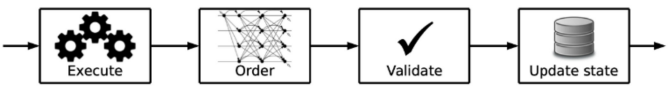

Figure 2. Execute-order-validate architecture.

The difference between the Order-Execute architecture and the Execute-Order-Validate architecture is that the transactions on the smart contracts of the former are usually tied together after reaching a consensus. Order-Execute architecture limits scalability and requires sequential execution of transactions until all peer's approval is completed. The Execute-Order-Validate architecture lets transactions execute before the blockchain reaches consensus in the chain, without systematic reliance on native cryptocurrencies. This is a major advancement over existing blockchain platforms.

Hyperledger is mainly composed of clients, peer nodes, ordering services, and a channel to jointly maintain a ledger. Only members of the channel can read the ledger data. Different channels can be constructed at the same time with each channel being an independent blockchain network.

2.4. Industrial Internet of Things (IIOT)

The Internet of Things can be divided into consumer Internet of things (CIoT) and industrial Internet of things. The CIoT includes a product platform that adds smart devices to a single customer. The industrial Internet of things can improve the performance of output value, product quality, and traceability. However, IIoT systems need to meet the security requirements of a specific operational technology (OT) environment, and the characteristics are different from the IoT.

2.4.1. IIoT Concept

IIoT indicates a network with billions of industrial devices, factories, and machines filled with sensors and connected to the internet for collecting and sharing data. The advent of tiny low-cost sensors and high-bandwidth wireless networks now means that even the smallest devices with a certain level of intelligence, can monitor, collect, and share data with other devices. All the collected data can be analyzed to improve the efficiency of business processes. IIoT can help companies better understand their business processes, and by analyzing data from sensors, it can improve the efficiency of their processes and even open new revenue streams. IIoT brings greater insight to broad supply chains, allowing businesses to coordinate and increase efficiency.

2.4.2. IIoT Security Issue

The IIoT has blurred the traditional IT and OT infrastructure boundaries and added a level of confusion to the integration of the two systems. With the development of the IIoT, the traditional role of OT systems is changing. The proliferation of IoT and devices has greatly increased the scope of cyberattacks against OT systems [18].

1. Insecure IoT Gateway: IoT gateways are responsible for connecting IIoT devices with the cloud. IoT gateways access and analyze sensor data to make important decisions. Since the gateway is the medium for communication between IIoT devices, a secure design is required. IIoT devices become vulnerable to cyber threats such as MITM and DDoS. If IoT gateways are inadequately secured and compromised by these cyber threats, the entire IIoT network is at serious risk [7].
2. Inefficient and Insecure Protocol, Server, and Access: We discuss this security issue on each of the following layers. Data collection layer, where most IIoT devices are located at. Devices might be vulnerable to viruses which lead to the construction of huge IoT botnet DDoS attacks. The data transport layer is responsible for combining sensor networks, mobile networks, and the internet. The communication protocol commonly used by IIoT is MQ Telemetry Transport (MQTT), which does not provide any encryption and authentication. As a result, intercepting data while communicating with other devices becomes easier and more vulnerable to threats [19]. Data processing layer, where the data are processed with requirements of communication with servers, historical data servers, and remote monitoring of terminals. Firewalls cannot identify forged packets that conform to protocols and access control rules. Therefore, they are very easy to evade firewalls. Therefore, to transmit data securely, a secure communication protocol is required. Due to the use of MQTT transmission, the way of sharing data becomes vulnerable to network attacks, and more importantly, there should be an access control protocol so that only authorized parties can access the data. Thus, confidentiality, integrity, and availability of data will be achieved.
3. Unsafe Cyber-Physical System: Cyber-Physical System (CPS) combines the Internet (cyber world) with the real world (physical). Many advanced systems adapted CPS such as self-driving cars, airplanes, smart homes, and smart factories. CPS is the foundation of IIoT, which handles a large amount of information and can better control and monitor the program. CPS is composed of IT equipment and connected, so there are many security loopholes, which become a concern for IIoT security [20]. CPS is vulnerable to cyber-attacks in the following ways [7]:

Equipment, is vulnerable to physical attacks such as side-channel attacks and reverse engineering. Software is vulnerable to malicious attacks including Trojans and viruses. Communication Protocol is vulnerable to protocol attacks such as MITM and DDoS attacks. CPS Manufacturing System is vulnerable to social attacks, such as phishing and social engineering.

4. Integration of information systems and operating systems: Industrial Control System (ICS) is a general term covering different information systems and technologies, such as Supervisory Control and Data Acquisition (SCADA), Distributed Control System (DCS), Programmable Logic Controller (Programmable Logic Controller, PLC), and so on. With the use of IIoT, the repositioning of the Industrial Control System (ICS) has greatly impacted the industry. The connection to the Internet and the sudden integration of IT in ICS makes it vulnerable to cybersecurity risks. Because OT systems are not designed with security in mind, it increases the risk of cyber-attacks on the IIoT. According to Industrial Cybersecurity 2018, most companies implementing IIoT (65%) believe there is a high risk of security breaches and attacks due to the sudden integration of OT in the IIoT environment. Unprotected network connectivity of OT environments and uncertain security issues due to the combination of IT and OT are both risky for IIoT technology. Therefore, secure IT and OT and secure ICS are required to protect the IIoT from damage.
5. Unsafe SCADA: SCADA is a network of smart devices used to control and monitor specific machines or systems or processes. Machines or systems are connected to smart devices with the help of sensors. It helps to study the output produced by the device and make important decisions. SCADA provides control of the software using a programmable logic controller (PLC) that is part of the IIoT. This makes SCADA part of the IIoT and why it is important to implement the IIoT. According to SCADA systems having security vulnerabilities, these vulnerabilities have also become a threat to the IIoT. These security issues include Trojans, worms, DoS attacks, etc. Therefore, a secure SCADA is required to increase the efficiency and enhance the functionality of the IIoT.

2.5. Access Control

Access control is a security measure that regulates user access to resources by verifying user permissions. Three important parts of the access control model are composed of identification, authentication, and authorization. Access control allows the subject (Subject) to use credentials to identify whether it is a legitimate user and allow the subject to access resources. The main access control models are described below.

2.5.1. Discretionary Access Control (DAC)

DAC develops a policy and decides who can access the objects. DAC allows legitimate users or groups to access regulated objects, and some users can also grant their access to other users. Many operating systems (OS) on servers such as Linux, UNIX, and Windows NT/SERVER have discretionary access control. To achieve this function, the system should identify the user's identity, and then allow or restrict the user's use of object resources according to the permissions in the access control list. Typically, the control rights of an object can be modified by privileged or administrative users.

2.5.2. Mandatory Access Control (MAC)

MAC is a "stricter" system that assigns labels to data and users according to the security levels, and the access control mechanism determines whether to grant or deny users access to resources based on the security labels [21]. Users can only access the corresponding objects according to the permissions granted by the security label as shown in Figure 3. Each security label has different security levels. These range from Unclassified (anyone can access) to Confidential to Secret and finally (we believe) to Top Secret; other countries use similar classifications. The security levels of principals and objects are

compared when performing access control. It is a method to force the subject to obey the access control policy. The MAC access control method implements the one-way flow of messages through gradient security labels, which effectively prevents Trojan horse attacks. However, MAC has the drawbacks of large workload, inconvenient and inflexible management, over-emphasizing confidentiality, and lack of the consideration of continuous workability of the system and the manageability of authorization.

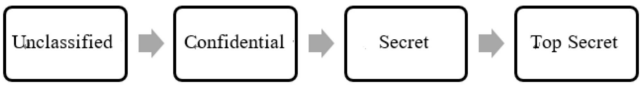

Figure 3. Four-layer security label.

2.5.3. Role-Based Access Control (RBAC)

RBAC divides the permissions of users in the system into roles, providing a simple, easy-to-manage, and fine-grained access control method [21]. In RBAC, each role is associated with one or multiple sets of permissions, and each permission may be assigned to multiple role groups too. Roles are assigned to users which eventually associate users with permissions. Compared to previous access control models, RBAC has greater flexibility in providing access rights rather than providing access to each user individually. All-access rights associated with a user can be easily audited by checking the permissions associated with the associated role. It also makes it easy to identify risk exposures associated with users.

2.5.4. Attribute-Based Access Control (ABAC)

In ABAC, a subject's access to an object can be determined by different attributes of the subject. The basic idea of ABAC is not to assign permissions directly between subjects and objects, but to allow all authorization based on the subject's attributes [21]. Attributes play a critical role in ABAC to grant permissions to authorized users such as name, location, IP address, location, etc. We chose ABAC for the access control method in this research.

3. System Architecture

In this section, we propose a distributed access control framework based on smart contracts, as shown in Figure 3. We first introduce the smart contract design and its functions in this framework.

3.1. Smart Contract Design

The proposed smart contract design consists of three contracts: policy contract, device contract, and access contract.

3.1.1. Policy Contract (PC)

PC manages the policies using the following functions.

- Auth(): The administrator and the data consumer define the ABAC Policy and transmit the request to the blockchain system. The administrator encrypts the data with the PC node's public key and then signs the request with the private key. The PC calls Auth() to authenticate the admin using its public key and decrypt the data using its private key.
- CheckPolicy(): The function to validate the ABAC policy, as shown in Algorithm 1.

Algorithm 1: Check ABAC policy

```
Input: ABAC Policy
Output: True or False
<AS,AO,AE,AP>←ABAC Policy
IsOK = True
for item in AS do:
   if item ∉ <userID, role>then
      IsOK = False
   endif
end for
for item in AO do:
   if item ∉ <DeviceID, MAC>then
      IsOK = False
   endif
end for
if AE != 1 or 0
   IsOK = False
end if
for item in AP do:
   if item ∉ <createTime,EndTime,allowedIP>then
      IsOK = False
    end if
end for
 return IsOK
```

- AddPolicy(): The function to add the ABAC policy to the state database (SDB) in the Hyperledger, as shown in Algorithm 2.

Algorithm 2: Add policy to blockchain

```
Input: ABAC Policy
Output: Error or null
@implement SmartContract Interface
API stub ChaincodeStub←Invoke()
if CheckPolicy (ABAC Policy) == False
     return Error ('Wrong Policy')
end if
ID ← SHA256(ABAC Policy.AS+ ABAC Policy.AO)
err ← APIstub.PutState(ID, ABAC Policy)
if err != null then
      return Error (err. Text)
end if
 return null
```

- DeletePolicy(): Deletion of the policy occurs in two cases: when the administrator actively deletes the policy by calling this function, or when the CheckAccess() method is executed and the "end-Time" shows expired, then it will call this function in the PC to delete the relevant policy, as shown in Algorithm 3.

3.1.2. Device Contract (DC)

DC is mainly responsible for storing the resource URL of the device in the SDB and generating a one-time URL. DC has 2 input parameters: {DeviceID, URL} with the following functions.

- AddURL(): The function to store the URL in SDB using DeviceId the primary key.
- GetOne-TimeURL(): The function to generate a one-time URL by querying SDB with the DeviceId.

Algorithm 3: Delete ABAC Policy form blockchain

```
Input: AS, AO
Output: Error or null
@implement SmartContract Interface
API stub ChaincodeStub←Invoke()
ID ←SHA256(AS + AO)
err ←APIstub.GetState(ID)
if err ! = null then
    return Error (err. Text)
end if
APIstub.DeleteState(ID)
if err != null then
    return Error (err. Text)
end if
 return null
```

3.1.3. Access Contract (AC)

AC manages the access control between the subject (user) and the object (resource/data). CheckAccess() is the function we designed to manage AC. It calls CheckPolicy() in the PC with the subject (AuS) and object (AuO) to query the corresponding ABAC policy (ABACP). If the returned result is null, it means that no policy complies with the request, and an error will be returned directly indicating no permission. If the returned result is not null, one or more ABACPs will be obtained. Then it verifies whether the environment attributes (AE) of the request match the AE of ABACP and returns 1 if permission is granted. Finally, GetOne-timeURL() function in DC is called to get the URL of the resource and return it to the user, as shown in Algorithm 4.

Algorithm 4: Check user's access

```
Input: ABAC-Request
Output: Error or one-time URL
< AuS;AuO;AuE >←GetAttributes(ABAC-Request)
P=<P1,P2,P3 . . . ..Pn>←PC.QueryPolicy(AuS, AuO)
if P = = Null then
    return Error
end if
for P in< P1,P2,P3 . . . ..Pn > do
  <AP,AE>←P
  if Value(ApP) = = 'deney' then
    continue
  if AuE∩ApE then
    continue
    URL←DC.GETONE_TIMEURL(AuO)
end for
  if One-time != Null then
    return One-time URL
  else
    return
end if
```

3.2. Hyperledger-IIOT System Structure

The architecture we proposed is for a blockchain-based access control system for the IIoT, which consists of four parts: the user, the blockchain, the smart gateway, and the IIoT device, as shown in Figure 4.

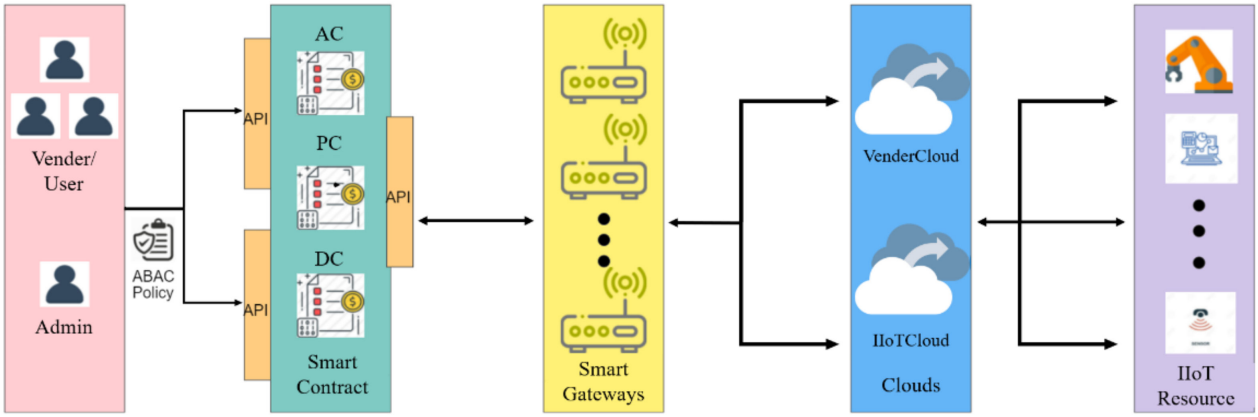

Figure 4. System Architecture.

3.2.1. User

The system divides users into two types: administrators, and vendor/users (data consumers). The administrator is responsible for managing the blockchain system and maintaining the related programs of the smart gateway. The administrator needs to have specific credentials to access the blockchain system. Data consumers obtain resources by sending requests to the blockchain through attribute-based authorization.

3.2.2. Blockchain

As the core of the system, all nodes need to be certified before joining the blockchain system. The system can implement access control through smart contracts.

3.2.3. Smart Gateway

The bridge between the device and the blockchain system can receive information from the device, thereby avoiding the pressure of the device's direct access to the blockchain system.

3.2.4. Industrial IoT Devices

Industrial IoT devices are quite large, and it is impossible to directly deploy the device as a peer node of the blockchain, which will cause a burden on the blockchain. IoT devices have unique MAC addresses or device IDs that can be distinguished from other devices. Whenever a device generates a new resource, a message containing the resource URL is sent to the smart gateway. In this system, the MQTT protocol is used as a message transmission protocol.

3.3. Hyperledger-IIOT System Flow

Lu [22] proposed the types of policy alliances suitable for domestic enterprises to adopt, and classified them as "vertical policy alliances", "horizontal policy alliances", "comprehensive policy alliances" and "project-based policy alliances". Taking the above four scenarios as the research background, as shown in Figures 4 and 5, the entire system workflow mainly includes the following parts. This section details the intermediate steps in each section. The connection between equipment resources and manufacturers is shown in Figures 4 and 5. Sequence numbers 1–8 in Figure 4 show a brief workflow, and Figure 6 shows a sequential diagram of the context diagram.

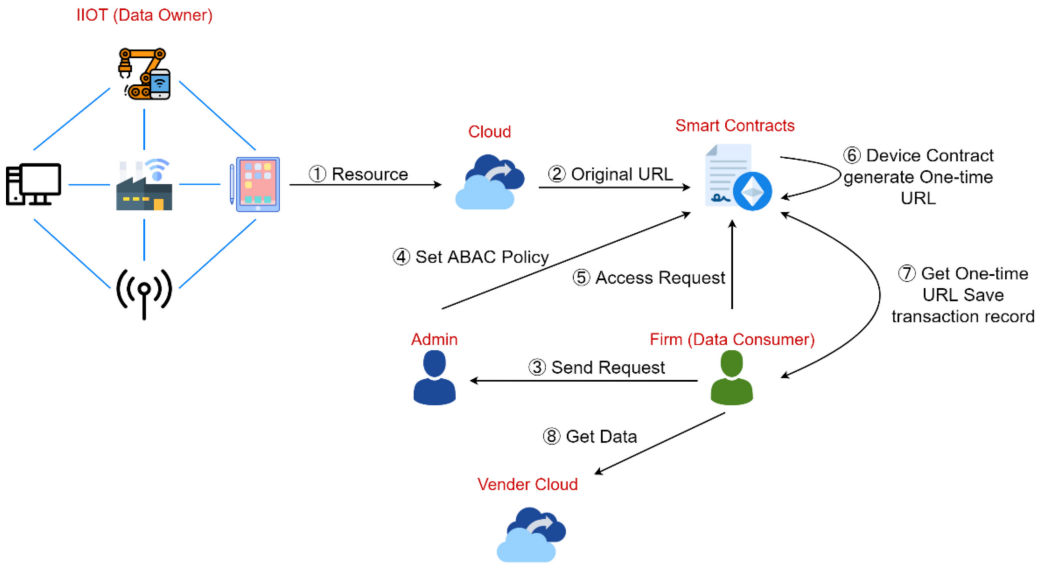

Figure 5. IIoT (Industrial Internet of Things) Operational Scenario.

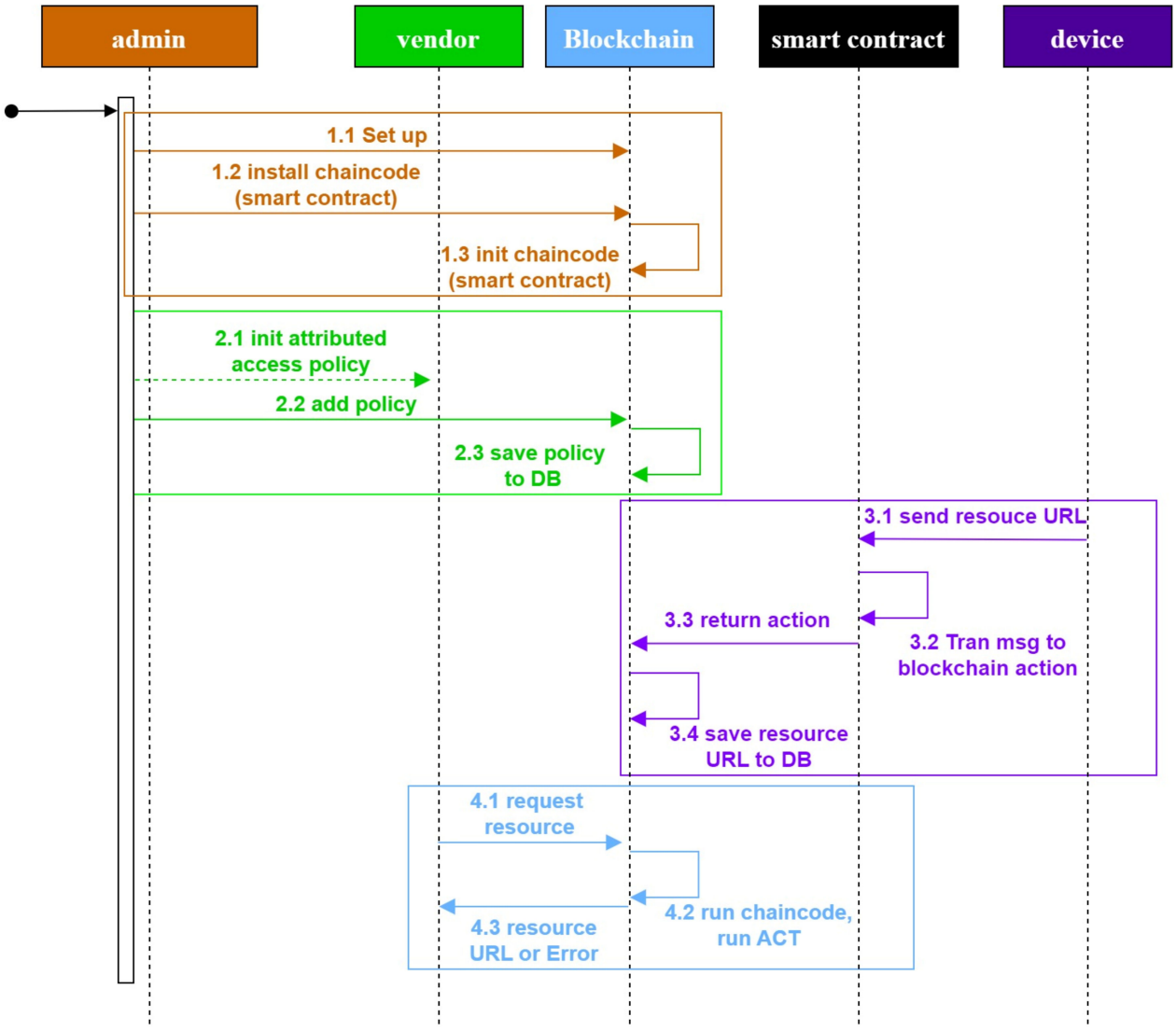

Figure 6. System sequence diagram.

3.3.1. Industrial IoT Devices

This research proposes a blockchain system based on attribute access control, which stores industrial IoT data outside the blockchain, and access control is performed on the blockchain. The smart contract is responsible for access control and evaluating access requests. The data owner can store the data in the cloud. Through the smart contract, the data owner can set the price to be paid to access the data. The data consumer sends an access request and then decides whether to grant permission according to the policy contract. The smart contract will also record the transaction information between the data owner and the consumer. Provide a data marketplace where data owners sell data and consumers buy data.

The vertical policy alliance relationship has seven steps, as shown in Figure 7. In Step 1, the cooperative manufacturer will send a request to the manager and discuss the permission set with the manager. Step 2, the administrator sets the cooperating manufacturer to the same group and sets the ABAC authority to the PC. Step 3, the cooperative manufacturer sends an access request to the AC. Step 4 is for AC to check whether the user has permission to access the PC. Step 5, PC sends an allow or error message to the AC. Step 6, AC replies to the DC with allowing or error. Step 7, if the permission is approved, the cooperative manufacturer will obtain a one-time URL. The difference between the horizontal policy alliance relationship and Figure 7 is that in the first step, manufacturers will individually send requests to managers, and managers set different permissions for different manufacturers.

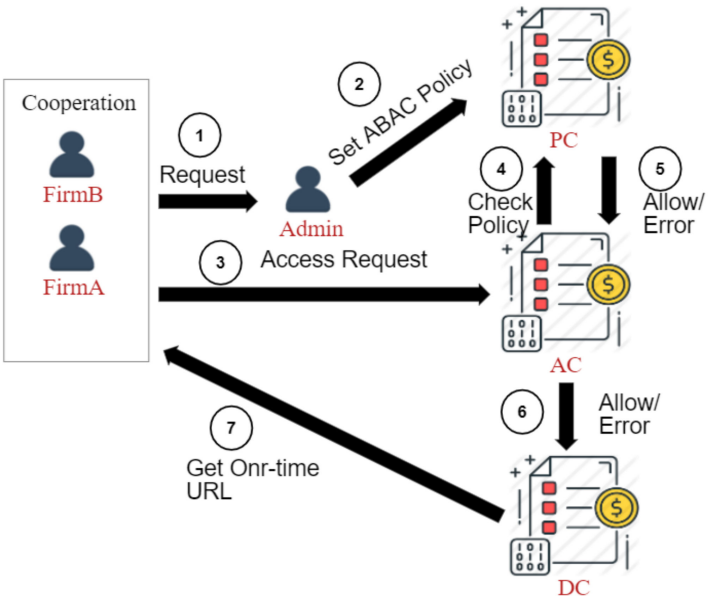

Figure 7. Vertical Strategic Alliances Relationship Scenario.

The situation in the project-based policy alliance relationship is shown in Figure 8. It is assumed that product A must be composed of two tasks, T1 and T2, and the tasks are assigned to FirmA, FirmB, FirmC, FirmD and FirmE for execution. A is a product concept that can be mapped to Substantial knowledge and knowledge required to execute T2, including product-related knowledge and knowledge of supporting product activities; FirmA can share data with FirmC, T1 tasks are jointly executed by FirmA and FirmB, and T2 tasks are jointly executed by FirmC, FirmD and FirmE, so they can share data.

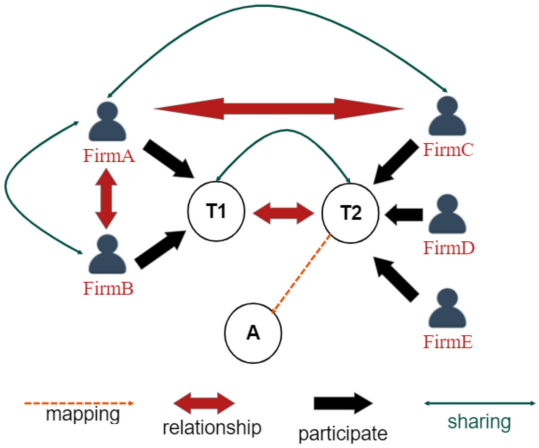

Figure 8. Example of project alliances data sharing.

3.3.2. One-Time URL

In this research, a blockchain-based controller manages identity and access control policies and acts as a tamper-proof log of access events, and uses one-time URLs to secure data access only once. A one-time URL has the following characteristics: (1) Once a URL is used, it cannot be used again. (2) The URL will expire after a certain period. (3) The administrator can revoke a valid URL, and if the user tries to use this URL again, an error message will be seen. As shown in Figure 9, the data owner first uploads the industrial IoT data to the cloud, and the cloud will store the data uploaded by the data owner individually, and then upload the original URL to the blockchain. After the data consumer sends a request to the blockchain, the identity will be verified. If the identity is eligible, the access record between the data owner and the data consumer (manufacturer) will be saved in the blockchain and transmitted once through the device contract Sexual URLs to data consumers.

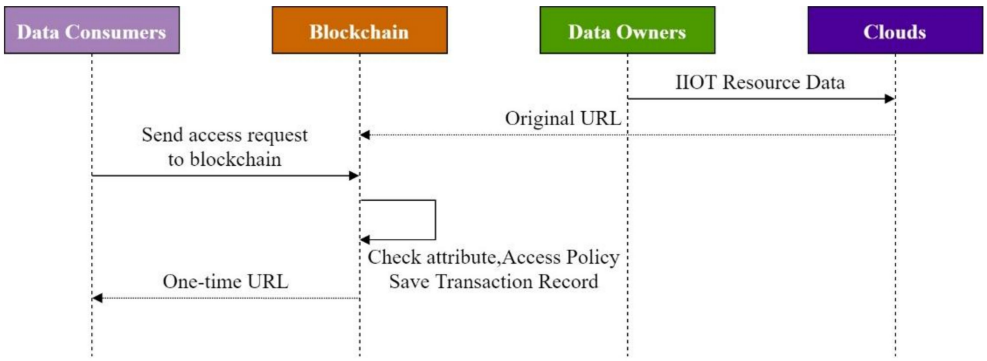

Figure 9. One-time URL context sequence diagram.

The sequence diagram of the access situation is shown in Figure 10. First, the data consumer will register with the administrator, and the administrator will then add permissions to the data consumer. The data owner will upload the data to the cloud, and the device contract in the smart contract obtains the original URL and stores it in the state database in the system. When a data consumer makes an access request to the system, the system will confirm from the policy contract and the access contract whether they have permission to access. If it is allowed, the system will return a one-time URL to the data consumer, otherwise, it will return an Error.

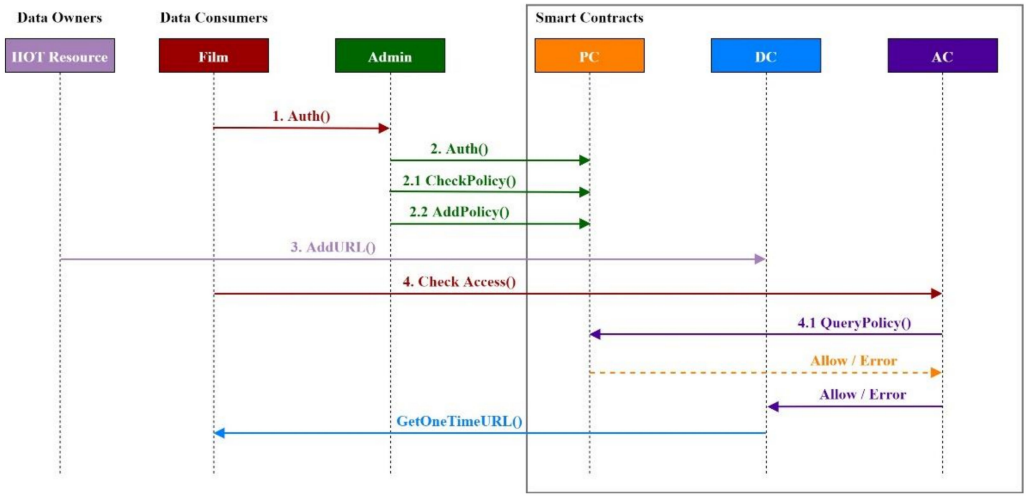

Figure 10. Access control sequence diagram.

4. Results

4.1. Hyperledger-IIoT Architecture

The Hyperledger-IIoT architecture in this study is shown in Figure 11. The IIoT data are uploaded to the cloud through Wi-Fi, and then the URL is stored in the blockchain. The blockchain stores the URL in the hyperledger database. Administrators and data consumers perform operations and identity verification through the Hyperledger SDK. After verification, corresponding actions are given according to different permissions.

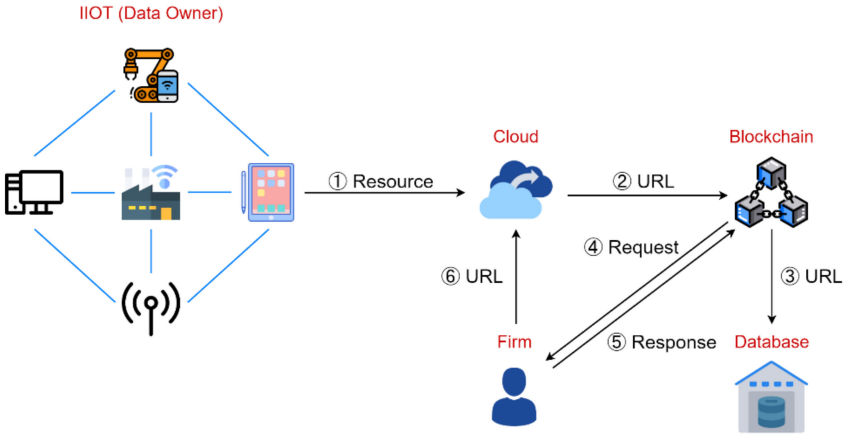

Figure 11. Hyperledger-IIoT Architecture.

4.2. Hyperledger-IIoT Implementation Process

As mentioned in Section 3, there are three kinds of smart contracts in this research (PC, DC, and AC). The smart contracts are written in the Go programming language. This section will introduce the three smart contracts in Hyperledger-IIoT environment. Table 1 shows the system environment and configuration.

Table 1. Development and Test Environment.

OS	Ubuntu 18.04.5
Language	Golang
Docker	v19.03.13
Docker-compose	v1.25.5
Hyperledger fabric	v2.1

4.2.1. Hyperledger-IIoT Management Rank Implement

The biggest difference between Hyperledger Fabric and other blockchain systems is being private. Hyperledger Fabric registers all members through the Membership Service Provider (MSP). Credentials need to be created for members before establishing a Hperledger Fabric network. Hyperledger Fabric provides the ability to create channels, allowing participants to create a separate ledger for transactions as shown in Figure 12. This feature becomes extremely important when some of the participants in the network are competitors because these participants do not want all the information to be open to all participants in the network. Only participants in the same channel will have the ledger in the channel, and other participants who are not in the channel will not see the ledger. After the Peer Node and the Orderer Node are successfully established, the channel is established, and the channel is added to the ledger, and create a genesis block, as shown in Figure 13.

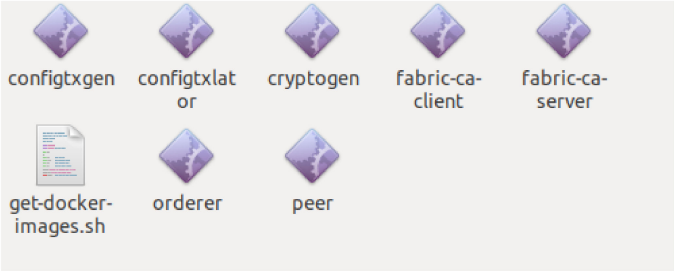

Figure 12. Member certificate.

Figure 13. Channel establishment.

Chaincode is a program written in Go language that can implement predefined interfaces. Chaincode runs in a protected Docker container and can be initialized and managed by the state of the transaction reconciliation submitted by the application. After the establishment of Hyperledger Fabric network, the smart contract is then written. The administrator can use the Fabric SDK to deploy the smart contract to peer nodes, install the chaincode, and initialize it as shown in Figure 14.

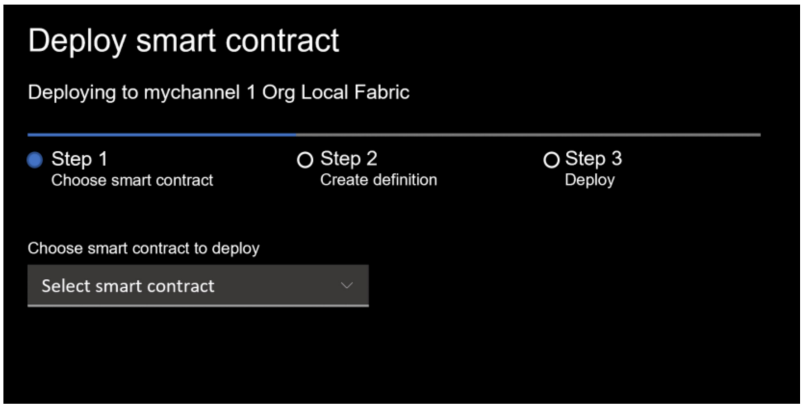

Figure 14. Deploy smart contracts.

The policy contract provides a method for operating ABACP. Figure 15 shows the administrator's identity to set permissions for different manufacturers. Figure 16 shows the user can log in through this interface. Figure 17 shows an example of an attribute-based access control policy request (ABACPR). An administrator can use the policy contract node to perform asymmetric encryption, and the policy contract calls Auth() to verify the identity

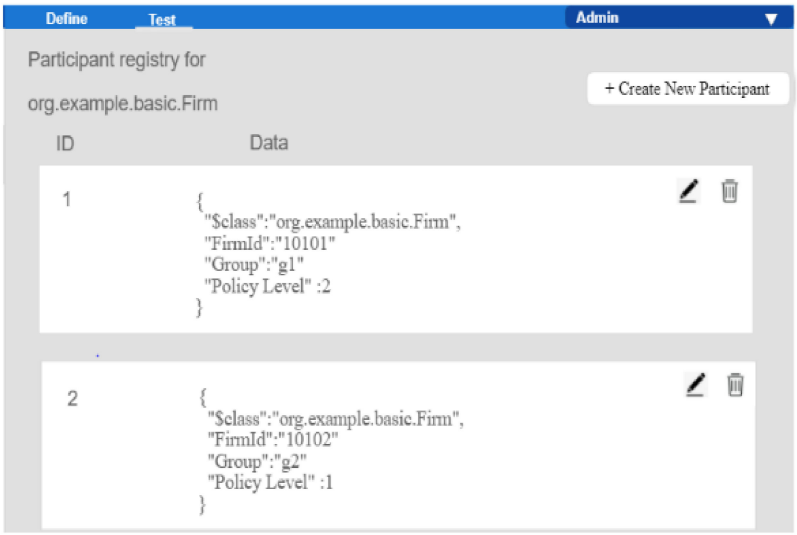

Figure 15. Administrator setting permissions.

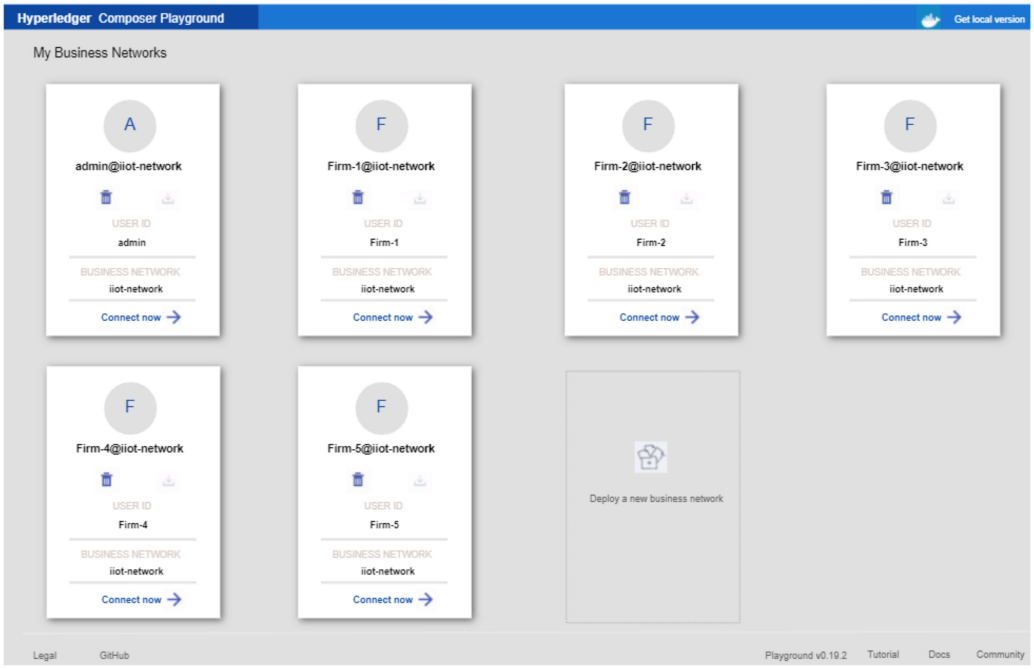

Figure 16. Login interface.

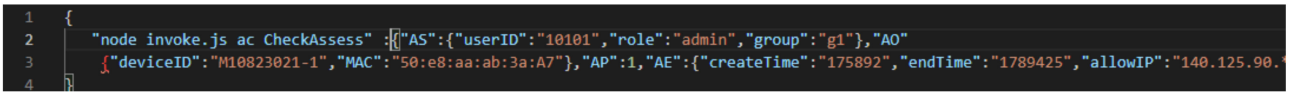

Figure 17. Check Access results.

4.2.2. Hyperledger-IIoT Data Consumer Rank Implement

After the identity verification through the blockchain, the equipment contract provides a method for the supplier to obtain a one-time URL. Figure 18 shows the information after

the data consumer obtains the URL. Figure 19 shows that the manufacturer obtains relevant information through a one-time URL, such as the data number and the owner of the data. Figure 20 shows that after the manufacturer uses the URL or does not use the one-time URL within the time specified by the system, the one-time URL becomes invalid.

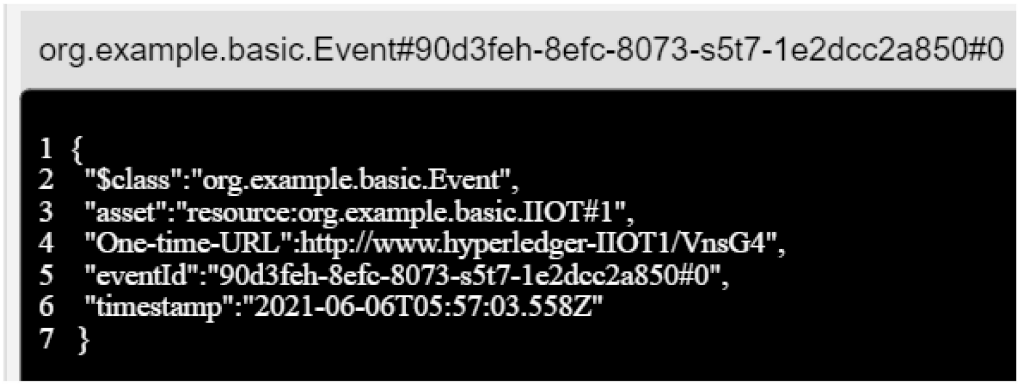

Figure 18. Manufacturers obtain URL-related information.

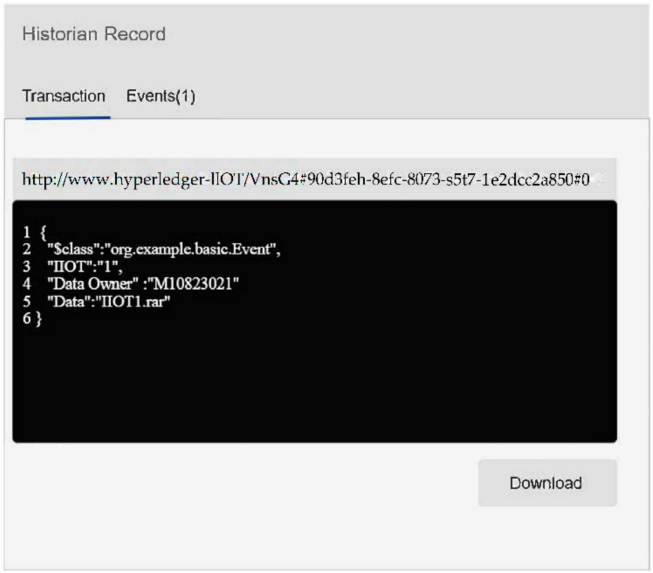

Figure 19. One-time URL profile.

Figure 20. One-time URL information invalid.

5. Evaluation and Discussion

We use Hyperledger Explorer in this research to test the performance of Hperledger-IIoT, simulating clients requesting permissions and accessing the platform. We simulate pol-icy contracts (PC), access contracts (AC), and device contracts (DC) under different numbers of clients and their time costs by setting the number of clients to be 10, 100, 200,

and 1000, respectively under the same system environment. The test results are shown in Figures 21–24. Figure 21 shows that the time spent by DeletePolicy in the policy contract is relatively high compared to the other functions. Figures 22 and 23 analyzes the cost of adding and obtaining URLs for different node numbers. Figure 24 analyzes the cost of access contracts with different numbers of nodes. The throughput of the system increases with the increase in the number of requests. When the throughput reaches a certain value, it tends to be stable. When the number of users increases, the throughput has no obvious downward trend.

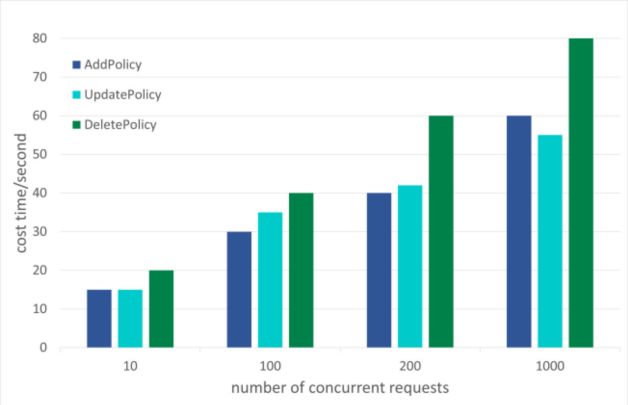

Figure 21. Time cost of Policy Contract (PC).

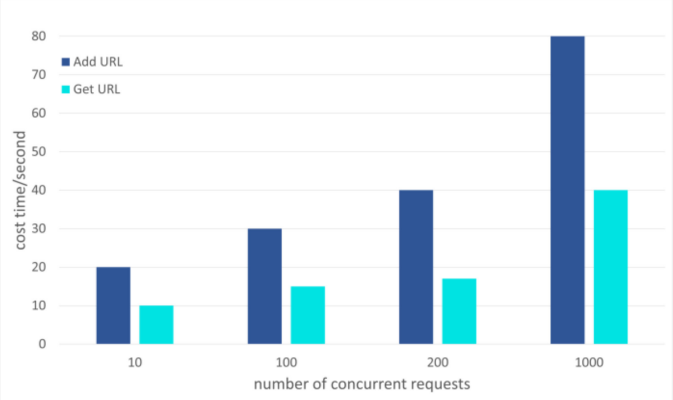

Figure 22. Time cost of Device Contract (DC).

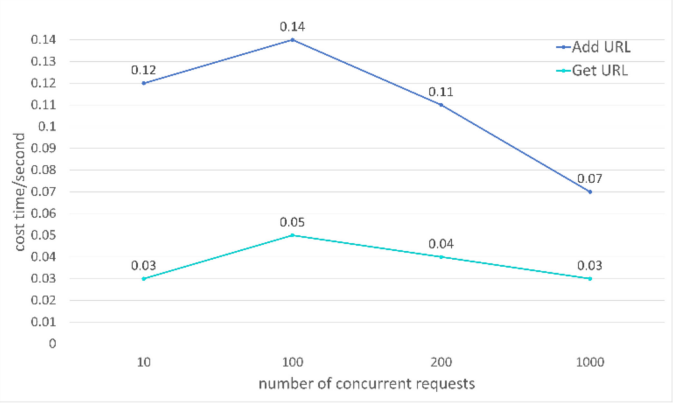

Figure 23. Time cost curve of Device Contract (DC).

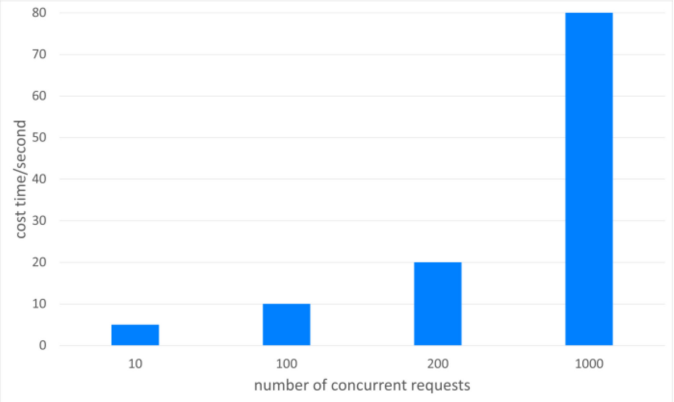

Figure 24. Time cost of Access Contract (AC).

By comparing the time cost of Hyperledger-IIoT and PoW consensus algorithms under different node trees from 10 to 100, as shown in Figure 25 and Table 2, the time required for Hperledger-IIoT to reach consensus is less than PoW. Because Hyperledger's consensus is different from PoW, the consensus is driven by individual nodes who are responsible for authorizing a transaction and push into blockchain those nodes known as endorsers node, and the distribution of blocks is given to special nodes called Orderers. Hyperledger-IIoT can have higher throughput in an environment with a large number of requests sent and can reach consensus more efficiently in a distributed system.

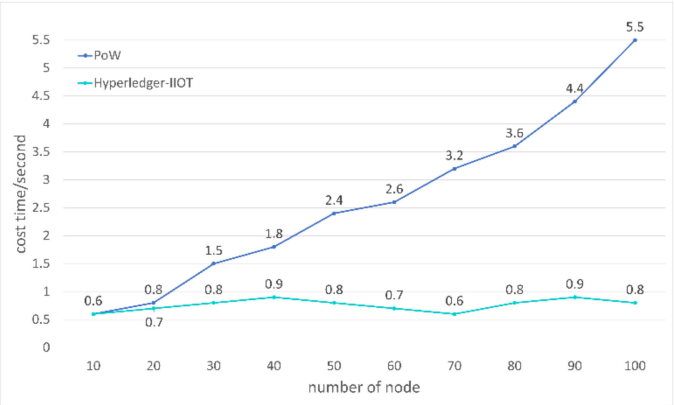

Figure 25. Consensus speed comparison between Hyperledger-IIoT and PoW(Proof of Work).

Table 2. Consensus Speed.

	Hyprtledger-IIoT	
Node	**Consensus Speed (Second)**	**PoW**
10	0.6	0.6
50	0.8	2.4
80	0.8	3.6
100	0.8	5.5

We also compare our work with other studies for access control, blockchain, access log, access timeliness, and usage of one-time URL, as shown in Table 3. Zhang et al. [5] proposed a framework based on smart contracts, it consists of multiple access control contracts (ACCs), a judgment contract (JC), and a registration contract (RC), to achieve distributed and trusted access control to IoT systems. Liu et al. [15] proposed an IoT access control system named fabric-iot, the system is based on the Hyperledger Fabric blockchain framework and attribute-based access control (ABAC). Sun et al., [23] integrate a

permissioned blockchain (HLF), an attribute-based access control (ABAC), and an identity-based signature (IBS) to build a secure, lightweight, and cross-domain based Blockchain-based IoT access control system.

Table 3. Comprehensive comparison.

Literature	Access Control	Blockchain	Access Log	Access Timeliness	One-Time URL
[5]	V	V	V	X	X
[15]	V	V	V	X	X
[23]	V	V	V	X	X
Our study	V	V	V	V	V

This research combines Hyperledger technology with the ABAC model. It adapts the blockchain to execute smart contracts, implement ABAC policies, and upload records of legal access to the blockchain. In addition, using a one-time URL ensures the security of the shared data sharing once.

6. Conclusions

Intelligentization has become the essential technological axis in the 21st century in the future IIoT era. The distribution of information security weaknesses has begun to rise, and potential threats are more likely to affect the industrial Internet of Things system through information security vulnerabilities. With the continuous transmission of a large number of rich data streams, the legality of sharing each other's data becomes more important. Once the data is leaked or the data has been maliciously tampered with, it will create some undesirable chain reactions. This research combines Hyperledger Fabric with an attribute-based access control model and utilizes the decentralization, tamper-proof, traceability, and other characteristics of blockchain to solve the shortcomings of traditional centralized access control in the environment of the Internet of Things. Nevertheless, by adding a one-time URL mechanism to data illegal reuse protection, our proposed system can also protect data sharing from misuse in an industrial IoT environment.

Author Contributions: Conceptualization, D.-H.S.; Data curation, T.-W.W. and G.-W.C.; Funding acquisition, D.-H.S.; Investigation, T.-W.W. and M.-H.S.; Methodology, D.-H.S., T.-W.W. and M.-H.S.; Project administration, D.-H.S. and D.C.Y.; Resources, G.-W.C.; Software, G.-W.C.; Supervision, D.C.Y.; Validation, M.-H.S. and G.-W.C.; Visualization, T.-W.W. and D.C.Y.; Writing—original draft, M.-H.S.; Writing—review & editing, D.C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: Taiwan Ministry of Science and Technology (MOST 109-2410-H-224-022 and MOST 110-2410-H-224-010).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was partially supported by the Taiwan Ministry of Science and Technology (grants MOST 109-2410-H-224-022 and MOST 110-2410-H-224-010). The funder has no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. AlAbdullatif, A.; AlAjaji, K.; Al-Serhani, N.S.; Zagrouba, R.; AlDossary, M. Improving an identity authentication management protocol in IIoT. In Proceedings of the 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 1–3 May 2019; pp. 1–6.
2. Djenna, A.; Harous, S.; Saidouni, D. Internet of Things Meet Internet of Threats: New Concern Cyber Security Issues of Critical Cyber Infrastructure. *Appl. Sci.* **2021**, *11*, 4580. [CrossRef]
3. Kaur, S.; Chaturvedi, S.; Sharma, A.; Kar, J. A Research Survey on Applications of Consensus Protocols in Blockchain. *Secur. Commun. Netw.* **2021**, *2021*, 6693731. [CrossRef]
4. Gourisetti, S.N.G.; Cali, Ü.; Choo, K.-K.R.; Escobar, E.; Gorog, C.; Lee, A.; Lima, C.; Mylrea, M.; Pasetti, M.; Rahimi, F.; et al. Standardization of the Distributed Ledger Technology cybersecurity stack for power and energy applications. *Sustain. Energy Grids Netw.* **2021**, *28*, 100553. [CrossRef]
5. Zhang, Y.; Kasahara, S.; Shen, Y.; Jiang, X.; Wan, J. Smart Contract-Based Access Control for the Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 1594–1605. [CrossRef]
6. Liang, W.; Tang, M.; Long, J.; Peng, X.; Xu, J.; Li, K.-C. A Secure FaBric Blockchain-Based Data Transmission Technique for Industrial Internet-of-Things. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3582–3592. [CrossRef]
7. Puri, V.; Priyadarshini, I.; Kumar, R.; Kim, L.C. Blockchain meets IIoT: An architecture for privacy preservation and security in IIoT. In Proceedings of the 2020 IEEE International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 13–14 March 2020; pp. 1–7.
8. Wang, S.; Hou, Y.; Gao, F.; Ji, X. A novel IoT access architecture for vehicle monitoring system. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 639–642.
9. Yeasmin, S.; Baig, A. Permissioned Blockchain-based Security for IIoT. In Proceedings of the 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, 9–12 September 2020; pp. 1–7.
10. Li, D.; Yao, Y.; Shao, Z.; Wang, L. From digital Earth to smart Earth. *Chin. Sci. Bull.* **2014**, *59*, 722–733. [CrossRef]
11. Zhong, R.Y.; Xu, X.; Klotz, E.; Newman, S.T. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering* **2017**, *3*, 616–630. [CrossRef]
12. Green, J. The Internet of Things Reference Model. In Proceedings of the Internet of Things World Forum (IoTWF) White Paper, Seoul, Korea, 6–8 March 2014.
13. Abuhashim, A.; Tan, C.C. Smart Contract Designs on Blockchain Applications. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–4.
14. Hylock, R.H.; Zeng, X. A Blockchain Framework for Patient-Centered Health Records and Exchange (HealthChain): Evaluation and Proof-of-Concept Study. *J. Med. Internet Res.* **2019**, *21*, e13592. [CrossRef] [PubMed]
15. Liu, H.; Han, D.; Li, D. Fabric-iot: A Blockchain-Based Access Control System in IoT. *IEEE Access* **2020**, *8*, 18207–18218. [CrossRef]
16. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Yellick, J. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portuga, 23–26 April 2018; pp. 1–15.
17. Nasir, Q.; Qasse, I.A.; Abu Talib, M.; Nassif, A.B. Performance Analysis of Hyperledger Fabric Platforms. *Secur. Commun. Netw.* **2018**, *2018*, 3976093. [CrossRef]
18. Yu, X.; Guo, H. A Survey on IIoT Security. In Proceedings of the 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS), Singapore, 28–30 August 2019; pp. 1–5.
19. Mugarza, I.; Flores, J.L.; Montero, J.L. Security Issues and Software Updates Management in the Industrial Internet of Things (IIoT) Era. *Sensors* **2020**, *20*, 7160. [CrossRef] [PubMed]
20. Wells, L.J.; Camelio, J.A.; Williams, C.B.; White, J. Cyber-physical security challenges in manufacturing systems. *Manuf. Lett.* **2014**, *2*, 74–77. [CrossRef]
21. El Sibai, R.; Gemayel, N.; Bou Abdo, J.; Demerjian, J. A survey on access control mechanisms for cloud computing. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3720. [CrossRef]
22. Lu, K. A Study on the Success Factors of Taiwanese Enterprises' Strategic Alliance. In Proceedings of the Industrial Management International Academic Research Association, Kaohsiung, Taiwan, 27 May 2001; pp. 14–31. (In Chinese)
23. Sun, S.; Du, R.; Chen, S.; Li, W. Blockchain-Based IoT Access Control System: Towards Security, Lightweight, and Cross-Domain. *IEEE Access* **2021**, *9*, 36868–36878. [CrossRef]