*electronics*

MDPI

*Article*

# Fine Grained Access Control Based on Smart Contract for Edge Computing

**Yong Zhu [1,2,3,*], Xiao Wu [4] and Zhihui Hu [2]**

[1] School of Computer Engineering, Jinling Institute of Technology, Nanjing 211169, China
[2] School of Computer Science & Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; hzhnjupt@163.com
[3] College of Computer and Information Technology Engineering, HOHAI University, Nanjing 210098, China
[4] White Matrix Inc., Nanjing 211899, China; ling@matrixdapp.com
* Correspondence: zhudz@jit.edu.cn; Tel.: +86-181-6809-2326

**Abstract:** Traditional centralized access control faces data security and privacy problems. The core server is the main target to attack. Single point of failure risk and load bottleneck are difficult to solve effectively. And the third-party data center cannot protect data owners. Traditional distributed access control faces the problem of how to effectively solve the scalability and diversified requirements of IoT (Internet of Things) applications. SCAC (Smart Contract-based Access Control) is based on ABAC (Attributes Based Access Control) and RBAC (Role Based Access Control). It can be applied to various types of nodes in different application scenarios that attributes are used as basic decision elements and authorized by role. The research objective is to combine the efficiency of service orchestration in edge computing with the security of consensus mechanism in blockchain, making full use of smart contract programmability to explore fine grained access control mode on the basis of traditional access control paradigm. By designing SSH-based interface for edge computing and blockchain access, SCAC parameters can be found and set to adjust ACLs (Access Control List) and their policies. The blockchain-edge computing combination is powerful in causing significant transformations across several industries, paving the way for new business models and novel decentralized applications. The rationality on typical process behavior of management services and data access control be verified through CPN (Color Petri Net) tools 4.0, and then data statistics on fine grained access control, decentralized scalability, and lightweight deployment can be obtained by instance running in this study. The results show that authorization takes into account both security and efficiency with the "blockchain-edge computing" combination.

**Keywords:** access control; SCAC; CPN; blockchain; edge computing



**Citation:** Zhu, Y.; Wu, X.; Hu, Z. Fine Grained Access Control Based on Smart Contract for Edge Computing. *Electronics* **2022**, *11*, 167. https://doi.org/10.3390/electronics11010167

Academic Editor: Pietro Manzoni

Received: 15 November 2021
Accepted: 30 December 2021
Published: 5 January 2022



## 1. Introduction

At present, hundreds of millions of IoT (Internet of Things) devices have been deployed in different fields of application to achieve the process of "data → information → value" by the analysis and research of big data [1]. However, there are risks of CIA (Data Confidentiality, Data Integrity, and Data Availability) for cloud-based applications due to data from a wide range of heterogeneous sensor and centralized data management. It cannot avoid the TPA (Third Party Auditors) for both data owners and data consumers [2]. A de-TPA (independent of TPA) distributed data management mode with secure and efficient is needed.

With a predicted 18 billion devices by 2022, the edge computing has become a technology with large influence across many IoT scenarios. However, the constrained capabilities of many IoT devices, as well as the current access control based on centralized and hierarchical architecture, bring new challenges in the domain [3].

- IAM (Identity and Access Management) for data security and privacy: Data managers can back up data without the knowledge of data producers and store it in various data

centers within the organization. IAM generally relies entirely on third parties, such as certification authorities.

- Performance of centralization data storage and processing: Cloud-based centralization model of data storage and processing cannot meet the expansion speed of IoT applications and diversified requirements of scenarios. A single centralized business server might become a bottleneck when access control queries and updates are frequent [4,5].

The massive, dynamic, and lightweight of network access devices in IoT are inherent and have coexisting characteristics [6]. Once these privacy information generated by the devices is leaked, it will bring huge losses to users. With the sustainable development of network applications, the security and privacy of transaction data has given rise to people's attention. It must meet the principle of "limited-open", meaning that only authorized users can access relevant information. As the cornerstone for data protection, access control can guarantee that data can only be accessed by users with relevant permissions [7]. The main access control models include RBAC, ABAC, CapBAC (capability-based access control), etc. [8]. Most of these are based on the concept of central trusted entity, which has many limitations. There are two emerging paradigms in post-cloud computing. Edge computing achieves efficient access control through cloud edge service orchestration and real-time task migration; with a blockchain in place, applications that could previously run only through a trusted intermediary, can now operate in a decentralized fashion, without the need for a central authority [9].

Edge computing is a three-tier architecture, including cloud, edge, and endpoints [10]. Its core component is the edge layer, which supports the access of various field devices downward (south) and can connect with the cloud upward (north). It integrates computing, storage, and network resources at the edge of the network to provide ICT power with the characteristics of distributed, low latency, and high bandwidth, enabling developers to quickly develop and deploy edge applications.

Based on the chained data structure, blockchain integrates a series of technologies such as cryptography, P2P (Peer-to-Peer) network, consensus mechanism, and on-chain script, which has the attributes of encryption, distribution, P2P, decentralization, and a shared database. It technically solves the security problems caused by the trust-based centralized model: Cryptography algorithm ensures the safe transfer of value, hash chain, and timestamp mechanism ensures the traceability and non-tamperability of data, consensus algorithm ensures the consistency of block data between nodes, and programmable code and smart contract in Turing complete realize flexible service. As a result, blockchain 2.0 and 3.0 have expanded from the financial field to industries, data applications, and other fields.

In the iterative IT process, there is always a balance between "high performance–well adaptation–security and reliability". With intraditional access control mode, resources are managed safely and orderly, which are viewed as static passive objects. And the access control is managed and executed by the core server. With the increasing complexity of application scenarios and requirements in IoT and edge computing, it is difficult to implement, and even lead to significant risks such as elastic scalability, single point of failure, load bottleneck, third-party data center risk, etc. At the same time, related IT is also developing rapidly: (1) Computing technology based on richfull stack platforms, such as docker, k8s, SDN (Software Defined Network), etc., EIS (Edge Intelligence Server), and its service orchestration close to end are added, connecting data center, edge computing, and end device layers. Thus tasks can be run in real time and resources be managed effectively. This provides a solid supporting environment for enhanced access control, including SCAC; (2) Blockchain developing: Blockchains developed from bitcoin applications in the initial stage to a secure data platform supporting smart contract and DApp (DecentralizationApplication). Flexible data transaction can be realized through smart contract programming. Of course, it can also be used as a SCAC carrier to complete a fine-grained access control strategy. In addition, the DApp properties of blockchain can be used to scale the system; (3) Pan-network technology: Most applications can communicate over the network. Edge

computing and blockchain domains can be connected through interfaces, such as SSH (Secure Shell) and MQTT (Message Queuing Telemetry Transport). In this way, after obtaining the access control policy stored on the smart contract in the blockchain, the resources in the edge computing are accessed with finegrained. Motivated by blockchain and edge computing technology features, the study aims to propose a more reasonable solution to the above problems. That is, a fine grained access control model using blockchain technology for edge computing, which operates in a smart contract, providing the access control information to IoT devices in security and real time. The goal of this work is to highlight the ways that blockchains and edge computing can be used together well. The key contributions of this article are as follows:

- The fine-grained SCAC mode has been explored in the combination of edge computing efficiency and blockchain security with access control strategies.
- It is proposed that the novel token can be used as the "key" to connect the elements of blockchain (value attribute of transaction), edge computing (entity authorized by access control policy), and Petri net model (condition of event fire) to build the fine-grained SCAC architecture.
- The rationality on typical process behavior of management services and data access control be verified through CPN tools 4.0, and then data statistics on fine grained access control, decentralized scalability, and lightweight deployment obtained by instance running.

This paper is organized as follows: An introduction to research significance, goals, and contributions are given in Section 1. Section 2 provides the related work about networks architecture of edge computing, smart contracts on blockchain, and fine grained access control. Section 3 presents the data service on TDP (Trusted Data Platform) between DOs (Data Owner) and DCs (Data Consumer). The architecture of access control and model of SCAC are put forward in Sections 4 and 5, including policies of ABAC and RBAC, entities. and behavior with CPN. Section 6 discusses the model simulation by CPN-Tools and results of instance running on blockchain. Section 7 concludes the paper. Finally, future research problems are raised.

## 2. Related Work

### *2.1. Networks Architecture of Edge Computing*

Edge computing is based on cloud computing architecture and its communication protocols, such as openstack [11]. Openstack consists of control node, computing node, network node, and storage node. Among them, the control node is responsible for controlling other nodes, including virtual machine establishment, migration, network allocation, storage allocation, etc: The computing node is responsible for running the virtual machine; the network node is responsible for the communication between the external network and internal network; the storage node is responsible for additional storage management of the virtual machine. It provides an operating platform/toolkit, including openstack computing (code named Nova), openstack object storage (code named Swift), and openstack image service (code named grace), which are used to orchestrate cloud services. EdgeX Foundry [12] is a standardized interoperability framework for IoT edge computing. The South side at the bottom includes all IoT objects and the edge of the network; the North side at the top includes the cloud and part of the network. It abstracts device services layer, core services layer, supporting services layer, and export services layer. An EdgeX UI serves as a web-based interface for the operation and maintenance.

SDN (Software Defined Network) provides connectivity and management to deal with complexity and heterogeneity. SDIoT [13] architecture combining IoT and SDN facilitate virtualization and interoperability at SBI (SouthBound Interfaces) and NBI (NorthBound Interfaces). The compute-intensive and resource-limited applications can be efficiently managed at the edge. OpenFlow in SDN provides communication interfaces to the control plane and data plane. A RESTful SDIoT southbound adapter provides M2I connection through Modbus, OPC-UA, Lora, and other industrial communication protocols to collect

sensing data from IoT terminal nodes and its northbound adapter provides a connection to connect edge devices using HTTP protocol and send IoT data to the cloud platform [14]. MQTT [15] is an open, high throughput, low latency message transfer protocol that supports Pub/Sub mechanism with real-time performance and flexible programming advantages. It can use open protocols to connect microservices, streaming data, and analysis, and provide event data to multiple applications for real-time response. The new network architecture of SDN, which is programmable and separated from control and forwarding to realize the flexible management of the network. The point-to-point communication between ECN (Edge Computing Node) and SDN controller can achieve distributed trust authentication for blockchain. A smart contract can automate the protocol and authorization process among network users.

### 2.2. Smart Contracts on Blockchain

Blockchain is essentially a distributed shared ledger and database with the characteristics of decentralization, non-tampering, whole process trace, traceability, collective maintenance, openness, and transparency [16]. Blockchain 1.0 is marked by currency, blockchain 2.0 by smart contract, and blockchain 3.0 by decentralized application. Consensus enables all ledger nodes to reach an agreement to determine the validity of a record, which is also a means to prevent tampering. It enables non-trusting nodes to verify the credibility of a blockchain without the participation of a third party in a decentralized network [17]. Only after the distributed node (miner) successfully solves the computational puzzle of finding the qualified nonce value for the block header can its packaged blocks be added to the blockchain. Based on these trusted and tamper proof data, smart contracts can automatically execute some predefined rules and terms. Blockchain has brought two novel functions of "value representation" and "value transfer" to the digital world, bringing the internet from "information internet" to "value internet". Token represents the blockchain value in the digital world, which is reflected in four quadrants divided by two dimensions: The digital world/physical world/information internet/value internet and its on-chain through the smart contract and off-chain through the Oracle interface.

Smart contract is a computerized transaction protocol reside on the chain and have a unique address, which can not only carry out simple value transfer, but also set complex rules. It executes automatically and autonomously, whose software nature is equivalent to a special server daemon. Blockchain stores "states", and smart contract is the way for state transition. Programming languages, tools (smart contracts IDE), standards (ERC20), and operating environment (EVM) constitute the trusted intelligent platform for blockchain. For example, Ethereum's smart contracts run on EVM (Ethereum Virtual Machine) by Turing-complete programming language (Solidity). The logical connection among complex smart contracts is transformed into the form of program logic flow, whose life cycle is as follows: (1) Set Up (own an account book wallet, i.e., address); (2) Freeze (persistence by authentication); (3) Execution (condition trigger, update status, submit to blockchain, consensus verification) and (4) Finish (transaction and new status information are stored in blockchain). A blockchain that supports smart contracts takes this further and allows for multi-step processes. Smart contracts operate as autonomous actors, whose behavior is completely predictable. As such they can be trusted to drive forward any on-chain logic that can be expressed as a function of on-chain data inputs, provided that the data they need to manage is within their own reach. This whole process can be done via atomic peer-to-peer exchanges of tokens if the chain follows the transactional model.

### 2.3. Fine Grained Access Control

For the existing access control models, such as RBAC (cross organization access control and authentication) [18], ABAC (user identity attributes and access control policies cannot be modified by malicious users) [19], CapBAC (trusted database storage, blockchain record permission granting, use, circulation, etc.), the main purpose of research is to solve the multi-agency security trust problem [20] and the single point failure problem [21] caused

by the centralized authorization decision-making entities. Taking full advantage of the decentralized, tamper proof, traceable. and smart contract of blockchain, the access control model is built by blockchain as a trusted entity, such as static access control model [22], management method of scientific data source [23], access control model based on IoT event and query basic permission delegation [24], fine-grained access [25], and cross domain access model [26]. Ref. [27] presented LEDGE, an agile and secured software-defined edge computing system for resilient access management of mobile IoT, considering traditional mobile access system faces several challenges. In WSNs (Wireless Sensor Networks), the malicious sensor nodes are detected and identified successfully with blockchain [28], an edge node scheme is proposed to address the issue of jamming attack [29], and a light-weight anonymous authentication techniques (Medium Access Control) is presented to resolve the black-hole attack in real time [30].

It is a wise strategy of access control to take blockchain as a trusted platform and separate data business from services management. The business data and access permissions are stored in a transaction database with TPA, and the operation and transaction information of the data is recorded through smart contracts to prevent malicious users from destroying source data and permission from being tampered. Considering the lightweight nodes in edge computing, the characteristics of on-chain and off-chain are further utilized. Hashes and access control policies pointing to data are stored on the chain, and sensitive data is stored off the chain, which is managed by access control policies on the blockchain. A hierarchical management architecture is recommended for massive terminal nodes. The cluster is composed of multiple associated devices, and the cluster manager is set up as the gateway that installs the blockchain client to connect to the blockchain. Where blockchain clients can implement basic blockchain behaviors, such as running smart contracts and consensus functions, and manage multiple resources with blockchain wallet as an authorization manager point.

In view of the above results, the following research scheme is adopted in the study: First, taking the node data objects in the blockchain and edge computing as entities, the corresponding operations and activities of access control are abstracted as behaviors to build the Petri Net model. Then, after simulation verification, the performance metrics is evaluated by instance running. Finally, SCAC guidelines are obtained, such as building a trusted platform TPA based on blockchain, efficient and flexible access control strategy combined with edge computing service orchestration and blockchain smart contract, low-overhead edge server interaction, active access control authorization mechanism, etc. That is a fine grained access control model using blockchain technology for edge computing, which operates in a smart contract, providing the access control information to the nodes in security and real time. The goal of this work is to highlight the ways that the blockchains and edge computing can be used together well.

## 3. Data Service on Blockchain

Data service provides service interface, service function, and data description information with a certain service protocol. Meanwhile, according to the business logic requirements, the services are divided into modules to encapsulate various operations of data entities. Data owners obtain data through edge nodes or IoT devices and store the data to a third-party data service platform for hosting. Data consumers retrieve data and obtain applications on the data service platform. However, the third-party data platform is at risk of tampering data and data voyeur, and the key nodes are vulnerable to DDoS (Distributed Denial of Service) attacks.

The decentralized property of blockchain as the underlying core technology has become an emerging large-scale network data sharing technology in a trustless environment. Data services based on distributed data storage and tamper-proof characteristics can ensure the integrity of service process records. The data service with smart contract formulates service specifications and sets service processes. It is becoming a common infrastructure for building DApp. The smart contract running on the blockchain describes

complex logic as code, which becomes the implementation of contract-based automation protocol. Business functions are decoupled from system management through layered services. That is, at the business level, specific business logic is realized and automatic business process specifications are customized through smart contracts, and data security is ensured by the attributes of blockchain data layer; at the management level, the interactive management with visual system is implemented by using RESTful programming mode combined with the edge computing architecture. NBI in WEB provides access entrance for user application, SBI connects data source in IoT protocol to set up access control strategy. The intelligent management of data service using smart contract achieves data management and service customization. Compared with centralized services, the functions that record the transaction of service process in data center are upgraded through blockchain distributed data storage and transaction services, so that users who use data services on TDP have the power to independently manage data. There is a TDP model consisting of DOs, DCs, and blockchain-based data service as shown in Figure 1.

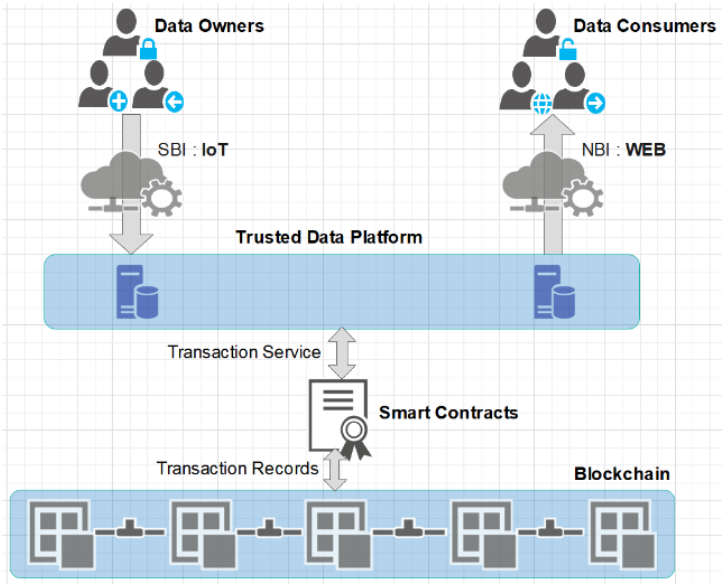



**Figure 1.** Blockchain-based TDP.

The information of both DOs and DCs is recorded and the account of each party is used to interact in blockchain through smart contract. All the on-chain transactions with smart contract can be transparently audited. The basic transaction records include: (1) Data storage: Store user data in the blockchain as a secure and tamper proof record. (2) Data verification: The miner node packs the transaction data and broadcasts it to the whole network. The consistency of data state is guaranteed by the consensus mechanism in the trustless environment, which make malicious nodes bear the high cost to attack TDP. (3) Data retrieval: The hybrid storage method of on-chain and off-chain is adopted, that is, key information such as identification, index, and summary are stored on the chain, and detailed description information is stored off the chain, which improve the retrieval efficiency. It can decouple application services from data management, contribute to the universality and expansibility of service model, and combine with edge computing mode. According to the idea of interface oriented programming, smart contract encapsulates the core business logic of data service that provide service business interfaces and service access interfaces, and separates the logical conditions from the data state. The main transaction service include: (1) Core services: Provide data retrieval, data release, data request, and other function services to realize automatic process control. (2) Permission service: Data access authorization is implemented to provide security for users and edge nodes according to access policies (ABAC, RBAC, etc.). (3) Registration service: All operations are identified by the blockchain account address as the unique identity in data services. Users write

account address and attribute information into the blockchain through smart contract. (4) Other services: Such as data service log query, service performance statistics and quality evaluation, service orchestration, and other value-added functions. Smart contracts have the ability to execute complex transaction protocols, which trigger on-chain functions to enable smart contracts to automatically execute relevant business logic.

DIaaS, including storage server technologies, protocols, standards, and architecture is available as part of clouded services, and further implemented by smart contract on the blockchain, which is fully decentralized. The tracking and audit of data interaction records can be implemented without the third-party supervision. And the automatic service interaction function can be achieved through smart contract without data service center. After DIaaS is started, the data would become valid and accessible to other nodes by the blockchain consensus. The publish/subscribe mode is selected to manage service calls, using MQTT protocol with the Pub/Sub mode for edge computing for reference. Compared with the traditional B/S and C/S modes, it has the advantages of low coupling, strong scalability, and asynchronous real-time. The data management architecture based on blockchain and edge computing supports distributed data management, storage, and access control. Access control uses smart contracts to register, broadcast, and revoke access authorization, and to create specific transactions to define access control policies. Smart contract can code the interaction rules between entities, and then is automatically executed when triggered. The active access control mechanism uses smart contract technology to realize the interaction logic between edge server and blockchain, so that users authorized by data owner can access node data in TDP. The edge server only maintains part of the state information, which greatly reduces the storage overhead and enhances the scalability of the system.

## 4. Architecture of Access Control

The main access control models are RBAC and ABAC, which are summarized as follows:

The access permission is composed of triplet (Who, What, How) in RBAC. "Who" performs "How" operations on "What or Which". That is, the operations of "subject" on "object". Where "Who" is the owner or subject of the permission, such as user and role. "What" is an resource or object, and "How" is an operation or activity. RBAC accord with the principle of minimum permission to decompose all permissions into a fine-grained subset of permissions and defines them into corresponding roles, which are assigned to corresponding subjects. Its basic elements are User, Role, Session, and Permission, which can be constructed several forms of models, including RBAC0 (Core RBAC), RBAC1 (Hierarchal RBAC), RBAC2 (Constraint RBAC), and RBAC3 (Combines RBAC).

The RBAC structure and behavior of edge computing systems in different states and life cycle phases can be described as follows: Roles involve a variety of user types in the whole application life cycle, such as designers, engineers, operators, operation, and maintenance personnel and owners. Its access permissions to the edge computing system should be limited and set from the perspective of security and reliability, such as design, configuring, deployment, retrieval, charging, etc. The behavior of the whole system includes process, business, function, and other elements and scenario-related elements include time, space, trigger conditions, results, and constraints.

The attributes of ABAC can be described by a quad (S, O, P, E). S represents the subject attribute, that is, the attribute of all entities that actively initiate the access request, such as age, name, occupation, etc.; O represents the object attribute, that is, the attribute of accessible resources in the system, such as documents, pictures, audio, and video data resources; P represents the permission attribute, that is, various operations on object resources, such as reading, writing, creating, and deleting files or databases. E refers to the environment attribute, that is, the environment information when the access control process occurs, such as the time when the user initiates the access, and the geographical or

network location of the system, whether there is concurrent access to the same information, etc. These attributes are independent of the subject and accessed resource [7].

Based on the attributes of user, resource, operation, and operation context, the proposed ABAC takes the attributes of subject and object as the basic decision-making elements to decide whether to grant its access permission, which separates policy management from permission determination. The policy description XACML (eXtensible Access Control Markup Language) inherits the platform independence of XML. XACML-based access control policy can use the same description policy in multiple different systems. It is universal and suitable for the description of access control policy in distributed and dynamic environment. Security administrators can define corresponding policies and rules to refine access control permissions, such as distributed fine-grained authorization methods, hierarchical control framework of cloud computing, and lightweight XACML. The XEngine system converts XACML policy rules and requests into digital representations and ABE access control policy expression is described by the access structure.

The architecture based on RBAC and ABAC is shown in Figure 2 [31].

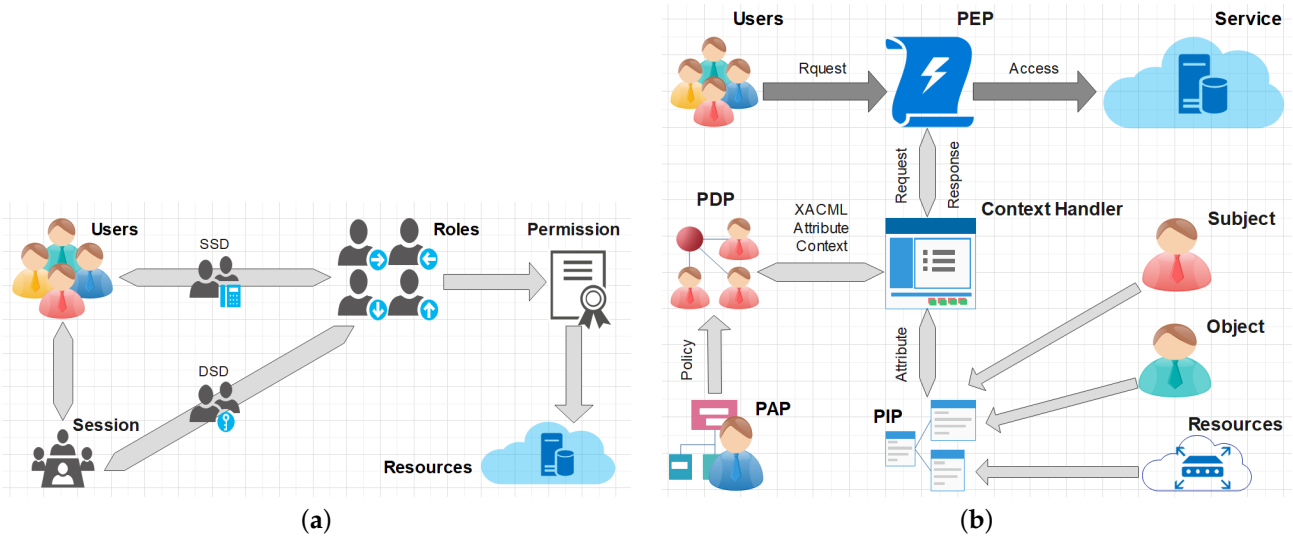



**Figure 2.** The architecture based on RBAC and ABAC. (**a**) RBAC based Architecture. (**b**) ABAC based Architecture.

Terminology notes: SSD means Static Separation of Duty, and DSD means Dynamic Separation of Duty in RBAC; Policy Enforcement Point is represented as PEP, Police Administration Point as PAP, Policy Decision Point as PDP, and Policy Information Point as PIP in ABAC.

The blockchain-based identity and access control policies records access events, tamper proof logs, and stores specified policies to enhance the privacy protection ability of edge computing, which can implement ABAC and RBAC well. RBAC-SC uses a general infrastructure to represent the trust relationship in RBAC, and Ethereum's smart contract technology to achieve the cross organizational access of roles. ABACV1, a secure access control model combining RBACV1 and ABAC, can perform fine-grained access control by applying Ethereum's blockchain smart contracts. In the decentralized storage framework, data owners can build identity and certificate management mechanisms and specify access policies to achieve fine-grained access control of data. Through the identity-based token management in the blockchain network, BlendCAC realizes the functions of registration, propagation, and revocation of access authorization by using smart contracts, and the distributed, scalable, lightweight, and fine-grained access control of the IoT. IoT-oriented architecture defines a new node, called management hub that requests access control information from the blockchain on behalf of the IoT devices/ECN. In addition to that, the solution involves a single smart contract that defines all the operations allowed in the access

control system. BBIAC (Blockchain-Based IoT Access Control) framework introduces the attribute in the process of IoT authorization. The blockchain's own distributed structure and identity authentication method provide massive and dynamic support for the model. Its security and multi-agency trust can deploy large-scale computing and storage in the blockchain to support lightweight IoT devices/ECNs.

The framework based on smart contract provides access control methods for subject and implements access authentication for static and dynamic based on predefined policies by checking subject behavior. Meanwhile, a lightweight protocol is constructed for ECN, which uses edge computing to reduce latency and run multi-chain cross-domain access in the cloud. Finally, a fine-grained, lightweight, and cross domain trusted access control mechanism-based blockchain for edge computing is realized.

## 5. Model of SCAC

The basic entities in model of SCAC are defined as follows:

- The RBAC-based model $M_{RBAC}^{ACSC}$ is represented as a quad (U, Ro, P, S), where elements represent User, Role, Permission, and Session respectively;
- And ABAC-based model $M_{ABAC}^{ACSC}$ is represented as a quad (S, O, P, E), where elements represent Subject, Object, Permission, and Environment, respectively.

The SCAC permission policy can be put to good use in ABAC, RBAC, and their mixed modes. To be compatible with patterns such as ABAC and RBAC, and $_{RBAC}^{ACSC}$ and $M_{ABAC}^{ACSC}$ can be set as base models. On the one hand, entities and their attributes, behaviors, events, and other elements can be extended to adapt to a wider range of scenarios. On the other hand, the basic elements of both are defined in the access control policy, and the corresponding authorization is obtained in the smart contract by executing different options to realize fine grained access control. The entities, relationship and behavior of $_{RBAC}^{ACSC}$ and $M_{ABAC}^{ACSC}$ are described as follows:

- Attribute A is a triple (attr, Val, R < Val >), representing the attribute name, value range, and values, respectively. After creating a resource object, the attribute policy for the resource object needs to be set [32];
- Attribute predicate AP is a triple (attr, $\propto$, Val). Where $\propto \in \{\wedge, \vee, \neg, \cup, \cap, \subseteq, \supseteq, \in, \forall, \exists\}$ is an operator to limit the value range of the attribute, which constitutes the attribute policy expression of the resource object;
- ACL consists of a role list and an attribute policy list, which record the roles allowed to access the resource and the attribute policy expression of the resource respectively;
- Rule is as (ID, Tgt, Eff, C, OE, AE), where ID is identification and Tgt is target $\in$ {Resource, Subject, Action, Environment}. Eff namely effect, which is the decision result matching the rule according to the access request. Condition is a boolean expression. OE and AE are obligation expressions and advice expressions, respectively;
- The core element policy P is (ID, Tgt, RS, RC). P $\leftarrow$ (AP, A), that is, the AP implements the attribute A set operation. The values permit and deny represent positive and negative authorization, respectively. Others, for example, ID namely identification, RS = $\{r_1, r_2, \ldots, r_n\}$ is rule set, and RC, namely, rule combining algorithm;
- There is also a top-level element PolicySet = (ID, Tgt, PS, PC) to contain several policies or other policy sets, where PS = $\{P_1, P_2, \ldots, P_n\}$ is the policy set, and PC namely the policy combining algorithm. They form a hierarchical relationship among rule, policy, and Policyset.

The key of SCAC is the permission carried by the smart contract. The policies and operations of access control are defined in the smart contract and triggered by blockchain transactions to achieve fine-grained access control. The description of class smart contract attributes and methods are shown in Table 1:

**Table 1.** Class smart contract attributes and methods.

| | Name | Note | | Name | Note |
|---|---|---|---|---|---|
| **Attrbutes** | +ID | Identity | **Methods** | #aPre | Attribute Prediicate |
| | #A | Attribute | | #regMan | Register Manager |
| | #Ru | Ruler | | #regDev | Register Device |
| | #T | Token | | +query | Information query |
| | −ACL | Access Control List | | #assign | Mode assignment |
| | −P | Policy | | −adminAC | Administrate Access Control |

Smart contracts for different access control models can inherit Table 1 to adapt to different scenarios.

In addition to entities defined above, the basic sets also includes user set U, role set Ro, operation set Op, object set Ob, session set Ss, etc. The relationship between entities and events, such as the relationship between subject-object and permission, user role assignment, role permission binding, and token events, can be established through methods such as AP operation in smart contract. For example:

```
AP ⊆ A × P                                  // attribute-policy assignment
OPA ⊆ AP × O                                // object-attribute-policy assignment
assignUP ( u, ap) =  ap ∈ AP | (ap,u) ∈ UPA  // assign method mapped to the user attribute policy set
assignOP(o, aps) =  ap ∈ AP | ( ap,u) ∈ OPA  // assign method mapped to the object attribute policy set
```

Each resource object maintains its own ACL. The following algorithm fragment gives a state transitions description for the creation of the ACL core data structure:

```
#1 S1: S ← Φ;                                          // clear ACL initial state
#2 S2: ACL_P ←(P_{i}, P_Ro, ACL_R ← Ro_{id};           // create ACL attribute policy list ACLP and role access list ACLR
#3 S3: Ro ← Ro ∪ Ro_id, P_{i} ← P_{i} ∪ P_id;          // add the role access list and attribute policy list of role id to ACL
#4 S4: Ro ← Ro_id IF id IN P_Ro ∀ id;                  // record the role id in ACL role list, if attribute policies match.
```

Typical SCAC processes are formally described as follows:

- Registration. Every user of data owner and subject need to register an account in the blockchain where access control is deployed. The registration for user of data consumer and object must be authorized by the former to set attributes and access control policies.
  regUser( user: ID, ap, ss ): U ∪ {user | user ∉ U}, A ∪ {a | ( a,p) ∈ AP}, P ∪ {p | ( a,p) ∈ AP}, Ss ∪ {ss ← ∅ }
- Access. Complete access includes access requests and access resources. That is, access to object resources is obtained by requesting access permissions. With the help of token, attribute, role assignment, and permission transfer are carried out in the blockchain-edge computing network to achieve fine-grained access control.
  adminAC( user,o,acl,ap ): assignU→U∪{user,o} | (user,o) ∈ U ∪ assignAP→{ACL∪ acl,AP ∪ ap | acl ∈ ACL,ap ∈ AP }
- Validation and query. CheckP matches all policy attributes to be checked, such as ACL, rule, token, etc. A boolean value is returned with the AND operation ∩ to determine whether the check passes. Information query is a non-destructive operation to be provided to all users.
  checkP ( acl,a,p ): ∀ p ( ∩ $acl_i$ ⊆ ACL ) ∩ ( ∩ $a_i$ ⊆ A )

SCAC provides two access control policy modes:

1. The data owner publishes the access control policy of the resource in the blockchain at first. Then the data consumer requests permission from the SCAC to access control. The runtime smart contract determines whether to grant access permission.
2. The data consumer first request an access to the authorization service carried on the edge server. Then the access permission is granted to it if the policy permits and the blockchain records the access transaction. In the above process, the smart contract provides the function of the automatic permission grant and transfer. The authorization process is permanently recorded in the blockchain that is regarded as TDP to prevent tampering.

The complete SCAC for edge computing includes two domains: Blockchain and edge computing, associated with manager, hub, device, etc., and shares TDP. It is proposed that the token can be used as the "key" to connect the elements of blockchain (value attribute of transaction), edge computing (entity authorized by access control policy), and Petri net model (condition of event fire) to build the fine-grained SCAC based on ABAC and RBAC. In the Petri net model, token is the key condition of event fire in state transition, such as registration, ACL and attribute initialization, role binding, access request and resource access, policy check and assign, and permission transfer, which are represented by E = {e1, e2, . . . , en}.

The Petri net model is used to describe the system architecture and the behavior of each object. The top-level module of SCAC in a hierarchical CPN model built based on the above entities and their attributes are shown in Figure 3. Place XTab contains the data structure of access control elements based on smart contracts, which constitutes the policy set of smart contracts. The two main substitution transitions handlerMan and handlerData demonstrate the decoupling of management services and data business.

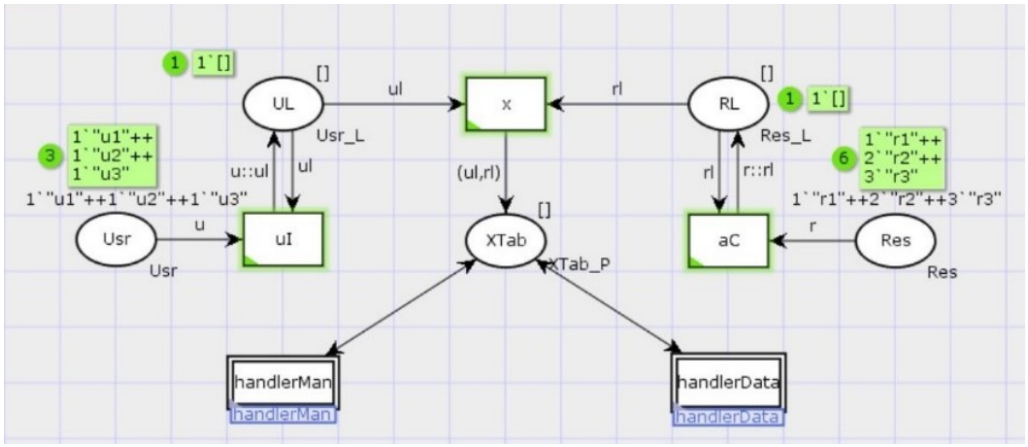



**Figure 3.** Top-level module of SCAC in a hierarchical CPN model.

The basic color sets of CPN are defined as follows:

- User ID. The user obtains the unique identity ID when registering, which generates the corresponding hash value as the key.
  colset Usr = STRING; colset Usr_L = list Usr.
- Basic attributes of ACL, role, ruler, etc. They are product data structures that contain attributes, such as users and resources.
  colset XTab_P = product Usr_L × Res_L.
- Policy P. Integrating multiple factors such as rules, RC, and PC, it works under the combined action of data owners, data consumers, managers, subjects, and objects. Its data sets vary according to access control policy patterns.
- Smart contract address. SCAC uses smart contract that provides services through address for various authorizations. It can be mapped into the "IP: Port" form for distributed edge computing server deployment to facilitate fine-grained access control.
  colset cSCAddr = product IP × Port.

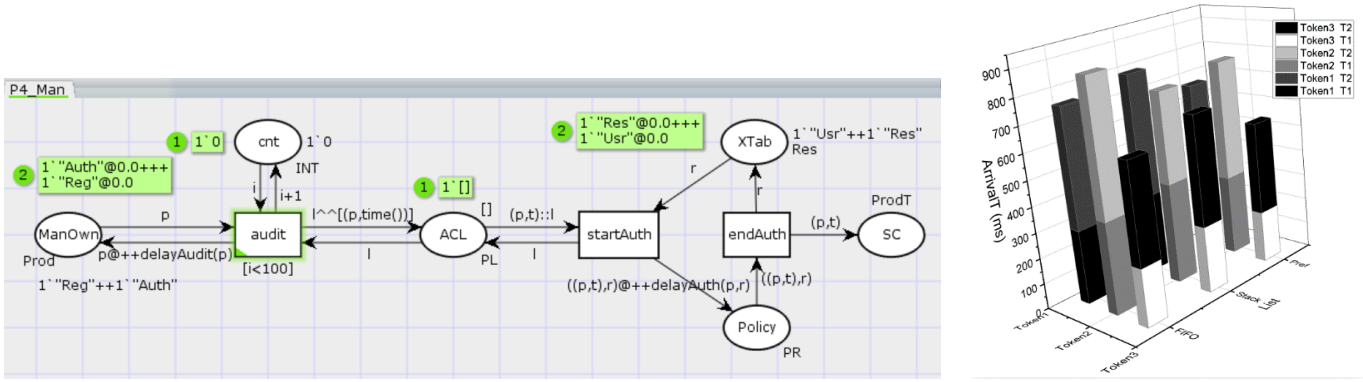
## 6. CPN Simulation and Instance Running Results

The main service processes of SCAC are as follows:

1. Network Set-up. It consists of edge computing/IoT and blockchain.
2. Node Registration. After registration based on the attributes and roles, the fine-grained access control can be authorized dynamically according to the corresponding policy. There can be multi-managers managed by consensus mechanism in a decentralized blockchain.
3. Policy Definition. As the core of the system, it defines the rules for resource and service access, and takes the blockchain smart contract as the carrier to carry out access control behaviors such as authorization, propagation, and transfer.
4. Policy Modification & Upgrade. It helps SCAC for diverse application scenarios, such as IIoT, AIoT, etc. to optimize data services and automated process management.
5. Service Discovery. It is an effective way to provide services, such as DIaaS, which export a unique address to network by a smart contract carrying the access control policy.

First, network set-up is carried out by the manager, whose tasks include policy definition, exposing services to edge computing and blockchain, and auditing user registration. Then, the network follows the policy for fine-grained access control. Limited to space, only simulations for typical process behavior in management services and data access control are presented below.

The management service is performed by the users with manager and data owner role, mainly including user registration service, resource attribute assignment, role permission binding, data publishing and request audit, data management service, etc. Key elements include access control policies, smart contract entity, attribute and role related algorithms, and data management sessions. Its behavior and metrics statistics are shown in Figure 4:



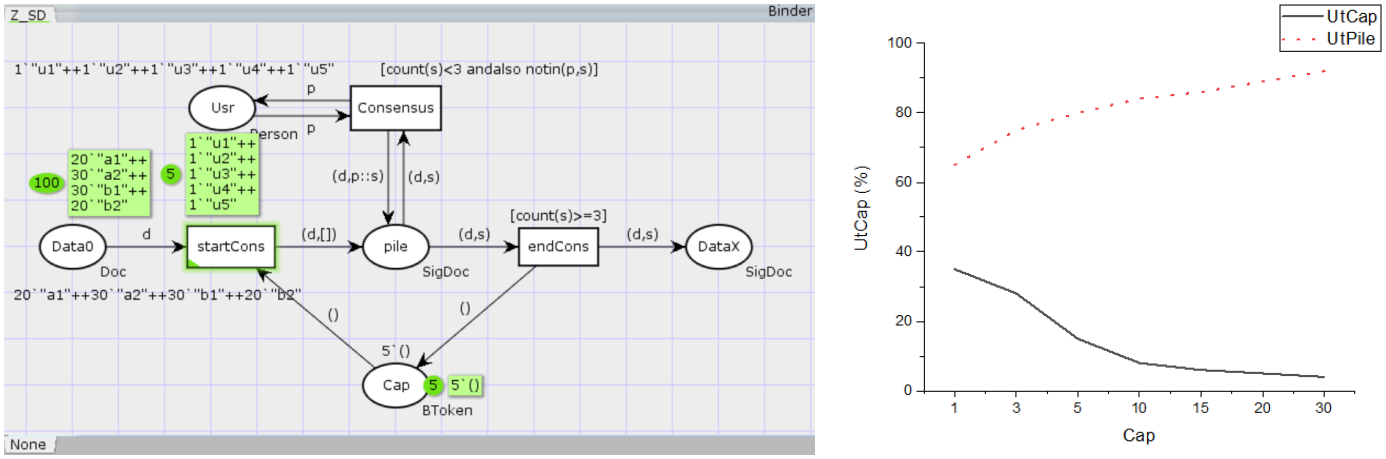

(**a**) (**b**)

**Figure 4.** Behavior and metrics statistics for management service. (**a**) CPN Model. (**b**) Data Statistics.

Complex management services can be represented and implemented through chained processes. At first, various services can be applied to users after register; ACL and other basic elements are formed after audit, and then policy is generated according to the access object. Finally, it is encapsulated as a smart contract to provide resource access token to achieve distributed fine-grained access control. The rationality of CPN model behavior can be easily verified. Different list processes, such as FIFO, stack, and preference settings, perform differently in monitor of CPN tools 4.0 by defining timed tokens. Among them, the preference setting mode is the most flexible to adjust smart contract by writing the corresponding function to customize management services. Simulation runs the management service process in each mode for 100 times, by taking list (FIFO, stack, and preference settings) and token (initial tokens in the place ManOwn and XTab) as dimensions to detect the average arrival time of audit and auth transition, so as to evaluate its performance.

The blockchain-based data access mode needs to verify the identifier and the permission information of users, and confirm the data transaction through the consensus. In other

words, it should be able to describe the operation behavior of multi-user access control. Its behavior and metrics statistics are shown in Figure 5.




(**a**) (**b**)

**Figure 5.** Behavior and metrics statistics for data access. (**a**) CPN Model. (**b**) Data Statistics.

The initial data Data0 is stacked in the buffer Pile after entering the consensus process. Multi-user Usr authenticates Data0 through the consensus, which is finally loaded as dataX to access after the consensus process is completed. On the basis of verifying the rationality of data access behavior, the operating efficiency for different buffer capacity utilization is surveyed by monitor in CPN-Tools 4.0. Specifically, the token sizes of place Cap and Pile were monitored to guide how buffer sizes could be designed to achieve an optimal match between data processing and users consensus. It can be seen from the simulation statistical results that with the increase of the initial token size of place Cap, the utilization rate of Pile increases monotonically and slightly. It shows that increasing the buffer capacity can not effectively improve the data channel efficiency. As timed token and multi-entity competition are not taken into account in the model, the above statistical data is slightly simplified.

The smart contract can encode the interaction rules and logic between edge computing and blockchain entities, and automatically execute when the corresponding conditions are triggered. Then through the blockchain running environment and its interface, that is, BaaS (Blockchain as a Service) mode, the data access instance in SCAC is actually run to obtain the operation status of fine-grained access control, decentralized scalability and lightweight deployment on the device.

1. Fine grained access control: Transfer tokens through the network, and modify the parameters in the blockchain smart contract to dynamically transfer permissions with different policies.
   The basic smart contract is first written and published as a framework in blockchain to facilitate implementation. And then the tokens as parameters be sent based on WEB protocol from edge computing to the blockchain to update the smart contract. The smart contract status information can be queried similarly. The manager or data owner has the authorization to change the corresponding smart contract to implement different service agreements. However the data consumer node must obtain a permission token from the smart contract to access data resources, so as to achieve secure fine-grained access control for edge computing.
   Through continuous access to edge computing network resources, different policy tokens are transmitted between networks to test the access cumulative time and its stability. The scripts of different tasks for test are issued, and rich performance metrics can be obtained in the dashboard of edge computing and blockchain network as shown in Figure 6.

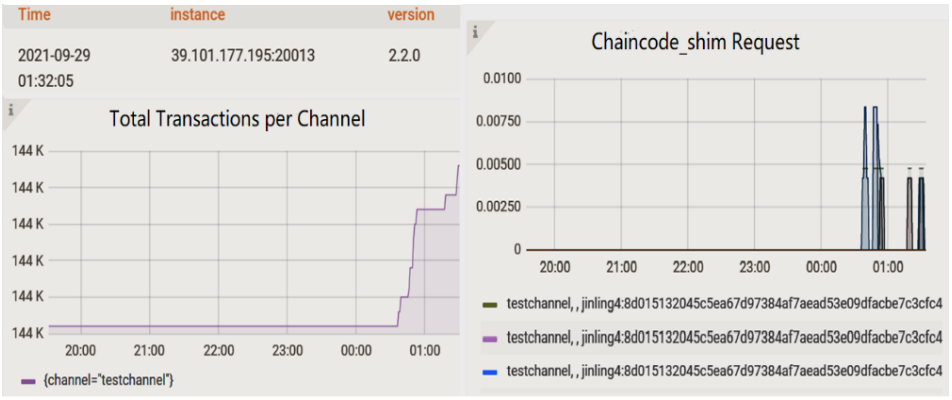



**Figure 6.** Instance of smart contract transaction.

The relevant data statistics come from the blockchain network. For the performance statistics methods of edge computing, see another research literature of the author [33].

2. Decentralized scalability: Test the response time of one and cumulative data access to observe the performance loss by increasing the number of access control policies and edge computing nodes.
   In the test, some nodes adopted ECN in WISE-PaaS, and others were simulated by virtual machine environment and Docker. With the increase of nodes, routing algorithms and traffic control and other factors will inevitably consume more network delay, resulting in system performance degradation. In view of the current cutting-edge network technologies, such as k8s, SDN, EC, etc., have been able to make up for the above performance losses and provide many methods for managing large networks. Moreover, blockchain has a natural decentralized attribute, and edge computing also has strong distributed scalability. Therefore, SCAC scalability is controllable.
3. Lightweight deployment on device: Make full use of node with limited ICT resources in edge computing environment. Terminal devices deploy only embedded systems and applications and use lightweight protocols. The EIS is used as the access decision-making entity to manage lightweight device nodes and interact with smart contracts to meet lightweight challenges for IoT devices.

Through CPN tools model verification and instance running results, compared with the existing access control mode, the following tradeoffs are shown in Table 2.

**Table 2.** Other studies vs. our research.

| SN | Other Studies | Our Research | Comparison & Trade-Off |
|---|---|---|---|
| 1 | As an emerging decentralized application, blockchain technically solves the security problems caused by the trust-based centralized model [8]. | As a new DApp, blockchain has changed from a trusted database storage access to an automatic access control using blockchain smart contract. | Other studies guarantees security based on blockchain decentralization. Our research further leverages smart contract programmability to achieve fine-grained access control. |

**Table 2.** *Cont.*

| SN | Other Studies | Our Research | Comparison & Trade-Off |
|---|---|---|---|
| 2 | For traditional RBAC, ABAC, and other methods, the management is simplified in a hierarchical manner [8]. | Combining RBAC and ABAC, management and services are decoupled based on XACML policies. | Other studies are layered by architecture. Our research is divided by process and suitable for different scenarios. |
| 3 | The policies and the rights exchanges are publicly visible and deployed on the Bitcoin blockchain [20]. | Service exposure and discovery are achieved through EIS, and users interact by the MQTT and SSH protocol. | Users can know at any time in other studies. Our research is more intelligent and fully integrated into edge computing. |
| 4 | A hierarchical framework comprising four tangible layers for the 'Industry 4.0' era [26]. | There are 2-layer architecture, and 2 main sub-modules among them: handlerMan and handlerData. | Other studies is suitable for Industry 4.0. Our research builds a uniform ACL (XTab) to integrate various access control. |
| 5 | Blockchain-based system for secure mutual authentication [26]. | Token and its parameters are passed between edge computing and blockchain, and it can realize native modeling in Petri net. | Other studies integrates attribute signature, multi-receivers encryption, and message authentication code. Our research use the token to connect the elements of blockchain, edge computing, and thePetri net model to build the fine-grained SCAC architecture. |

## 7. Conclusions

As a new DApp, blockchain has changed from a trusted database storage access to an automatic access control using blockchain smart contract, namely SCAC. The SCAC model is based on ABAC and RBAC with attributes as basic decision elements and authorization by role, which has been explored in the combination of edge computing efficiency and blockchain security with the access control strategies. It is proposed that the novel token can be used as the "key" to connect the elements of a blockchain (value attribute of transaction), edge computing (entity authorized by access control policy), and the Petri net model (condition of event fire) to build the fine-grained SCAC architecture. The rationality on typical process behavior of management services and data access control are verified through CPN tools 4.0, and then data statistics on fine grained access control, decentralized scalability, and lightweight deployment obtained by instance running in this study. The results showed that authorization takes into account both security and efficiency with the "blockchain-edge computing" combination.

There are still two issues that are worth further research:

1. Enhance the data attribute description and access behavior ability of the model. With the help of color, timed and hierarchical PN characteristics, enhance token elements, and design 4M (multi-organization member information sharing, multi-mode strategy, multi-channels transaction, and multi-node transmission competition) fidelity model with complex tasks for cloud computing, IoT, and other application scenarios.
2. Design the interface between blockchain and edge computing to realize automatic fine-grained access control mechanism. Although data can be shared by database, its flexibility and efficiency are not ideal. The deployment, operation, and maintenance of the system can be completed, learning from the current cloud computing and edge computing technologies and taking account blockchain as a BaaS node.

Our conclusion is that the blockchain-edge computing combination is powerful to cause significant transformations across several industries, paving the way for new business models and novel distributed applications. Blockchains give us resilient, truly decentralized P2P systems and the ability to interact with peers in a trustless, auditable manner. In addition, with the help of edge computing "cloud-edge-end" resource collaboration and service orchestration, SCAC allow us effectively to automate complex data service processes.

**Author Contributions:** Conceptualization, Y.Z.; methodology, Y.Z.; software, X.W.; validation, Y.Z.; writing—original draft preparation, Z.H.; writing—review and editing, Z.H.; visualization, Z.H.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the fund of State Key Laboratory of Computer Architecture (ICT, CAS) grant number CARCHA202010, the JIT fund grant number 0620025, the fund of Science and Technology Development Center, Ministry of Education grant number 2018A03040 and Communication chip video detection software platform grant number jit-h-2019-143.

**Acknowledgments:** The authors acknowledge the fund of State Key Laboratory of Computer Architecture (ICT, CAS) (grant no. CARCHA202010), the JIT fund (grant no. 40620025), the fund of Science and Technology Development Center, Ministry of Education (grant number 2018A03040), and Communication chip video detection software platform, jit-h-2019-143.

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

## References

1. Zhang, H.C. Research on IoT Data Service Model and Key Mechanism Based on Blockchain. Master's Thesis, Beijing University of Technology, Beijing, China, 2019.
2. Liu, B.; Yu, X.L.; Chen, S.; Xu, X.; Zhu, L. Blockchain Based Data Integrity Service Framework for IoT Data. In Proceedings of the 2017 IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA, 25–30 June 2017; pp. 468–475.
3. Ericsson Mobility Report: On the Pulse of the Networked Society. Ericsson, Tech. Rep., November 2017. Available online: https://www.ericsson.com/mobility-report (accessed on 20 August 2021 ).
4. Cheng, G.J.; Huang, Z.J.; Deng, S.G. Data management based on blockchain and edge computing for Internet of things. *J. Internet Things* **2020**, *4*, 1–9.
5. Novo, O. Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT. *IEEE Internet Things J.* **2018**, *5*, 1184–1195. [CrossRef]
6. Shi, J.; Li, R.; Song, T. Blockchain-based access control framework for Internet of things. *J. Comput. Appl.* **2020**, *40*, 931–941.
7. Fang, L.; Ying, L.H.; Guo, Y.C.; Fang, B.X. A Survey of Key Technology in Attribute-Based Access Control Schema. *J. Comput.* **2017**, *40*, 1680–1698.
8. Shi, J.S.; Li, R. Survey of blockchain access control in Internet of things. *J. Softw.* **2019**, *30*, 1632–1648.
9. Christidis, K.; Devetsikiotis, M. Blockchains and smart contracts for the internet of things. *IEEE Access* **2016**, *4*, 2292–2303. [CrossRef]
10. Edge Computing Reference Architecture 3.0. 2018. Available online: http://www.ecconsortium.org/Lists/show/id/334.html (accessed on 11 September 2021).
11. OpenStack. October 2021. Available online: https://docs.openstack.org/zh_CN/ (accessed on 16 August 2021).
12. Liu, F.; Tang, G.; Li, Y.; Cai, Z.; Zhang, X.; Zhou, T. A survey on edge computing systems and tools. *Proc. IEEE* **2019**, *107*, 1537–1562. [CrossRef]
13. Rafique, W.; Qi, L.; Yaqoob, I.; Imran, M.; Rasool, R.U.; Dou, W. Complementing IoT services through software defined networking and edge computing: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1761–1804. [CrossRef]
14. ADVANTECH. WISE-PaaS IIoT Cloud Platform Architecture and Service. 2021. Available online: https://docs.wise-paas.advantech.com/en (accessed on 19 September 2021).
15. Message Queuing Telemetry Transport (MQTT). June 2016. Available online: https://www.iso.org/standard/69466.html (accessed on 4 October 2021).
16. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: www.bitcoin.org/bitcoin.pdf (accessed on 24 October 2021).
17. Wu, J.G.; Liu, T.L.; Li, J.Y.; Huang, J.Y. Research Progress on Blockchain Technology in Mobile Edge Computing. *Comput. Eng.* **2020**, *46*, 1–13.
18. Cruz, J.P.; Kaji, Y.; Yanai, N. RBAC-SC: Role-based access control using smart contract. *IEEE Access* **2018**, *6*, 12240–12251. [CrossRef]

19. Maesa, D.D.F.; Mori, P.; Ricci, L. Blockchain based access control. In Proceedings of the IFIP International Conference on Distributed Applications and Interoperable Systems, Neuchâtel, Switzerland, 19–22 June 2017 ; Springer: Cham, Switzerland, 2017; pp. 206–220.
20. Hashemi, S.H.; Faghri, F.; Campbell, R.H. Decentralized user-centric access control using pubsub over blockchain. *arXiv* **2017**, arXiv:1710.00110.
21. Outchakoucht, A.; Hamza, E.S.; Leroy, J.P. Dynamic access control policy based on blockchain and machine learning for the internet of things. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 417–424. [CrossRef]
22. Zhang, Y.; Kasahara, S.; Shen, Y.; Jiang, X.; Wan, J. Smart contract-based access control for the internet of things. *IEEE Internet Things J.* **2018**, *6*, 1594–1605. [CrossRef]
23. Ramachandran, A.; Kantarcioglu, D. Using blockchain and smart contracts for secure data provenance management. *arXiv* **2017**, arXiv:1709.10000.
24. Ali, G.; Ahmad, N.; Cao, Y.; Asif, M.; Cruickshank, H.; Ali, Q.E. Blockchain based permission delegation and access control in Internet of Things (BACI). *Comput. Secur.* **2019**, *86*, 318–334. [CrossRef]
25. Lin, C.; He, D.; Huang, X.; Choo, K.-K.R.; Vasilakos, A.V. BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. *J. Netw. Comput. Appl.* **2018**, *116*, 42–52. [CrossRef]
26. Ma, M.; Shi, G.; Li, F. Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario. *IEEE Access* **2019**, *7*, 34045–34059. [CrossRef]
27. Wu, D.; Huang, X.; Xie, X.; Nie, X.; Bao, L.; Qin, Z. LEDGE: Leveraging edge computing for resilient access management of mobile IoT. *IEEE Trans. Mob. Comput.* **2021**, *20*, 1110–1125. [CrossRef]
28. Almaiah, M.A. A New Scheme for Detecting Malicious Attacks in Wireless Sensor Networks Based on Blockchain Technology. In *Artificial Intelligence and Blockchain for Future Cybersecurity Applications*; Springer: Cham, Switzerland, 2021 ; p. 217.
29. Adil, M.; Almaiah, M.A.; Omar Alsayed, A.; Almomani, O. An anonymous channel categorization scheme of edge nodes to detect jamming attacks in wireless sensor networks. *Sensors* **2020**, *20*, 2311. [CrossRef] [PubMed]
30. Adil, M.; Khan, R.; Almaiah, M.A.; Al-Zahrani, M.; Zakarya, M.; Amjad, M.S.; Ahmed, R. MAC-AODV based mutual authentication scheme for constraint oriented networks. *IEEE Access* **2020**, *8*, 44459–44469. [CrossRef]
31. Chen, J. Research on XACML Strategy Optimization Method. Master's Thesis, Nanjing University of Science & Technology, Nanjing, China, 2015.
32. Li, W.G.; Zhao, F.Y. RABC Permission Access Control Model with Attribute Policy. *J. Chin. Comput. Syst.* **2013**, *34*, 328–331.
33. Zhu, Y.; Huang, C.; Hu, Z.; Al-Dhelaan, A.; Al-Dhelaan, M. Blockchain-Enabled Access Management System for Edge Computing. *Electronics* **2021**, *10*, 1000. [CrossRef]