Hindawi
Journal of Sensors
Volume 2022, Article ID 3314881, 16 pages
https://doi.org/10.1155/2022/3314881

Research Article

Tenant-Centric Attribute Semantic Access Control Policy Model for the Cloud Service Platform

Yang Yu,[1,2] **Linfu Sun**,[1,2] **and Shuhai Wang**[1,2]

[1]*School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China*
[2]*Manufacturing Industry Chain Collaboration and Information Support Technology Key Laboratory of Sichuan Province, Southwest Jiaotong University, Chengdu 610031, China*

Correspondence should be addressed to Linfu Sun; slxbr615765@163.com

Received 16 March 2022; Accepted 21 April 2022; Published 4 July 2022

Academic Editor: Yuan Li

In the open Internet environment, there is a cross-platform access control problem that when a tenant needs to access the business resources of other collaborative platforms through the cloud service platform, the cloud service platform only supports the tenant to access the business resources within the platform. When tenants need to access business resources through the cloud service platform, the authorization method of the cloud service platform is static and the authorization granularity is coarse-grained, so dynamic fine-grained authorization is not supported. To solve the above problems, based on ABAC, this paper proposes a tenant-centric attribute semantic access control policy model for cloud service platforms. The model and its application framework can automatically evaluate whether it has cloud service platform or cross-platform access control rights according to the change of the tenant's attributes so as to determine whether it can obtain the corresponding business resources. Through a practical case analysis, we prove that the application of ASACPM and its application framework to the cloud service platform have good flexibility, scalability, and practicability. In addition, we design some experimental scenarios to verify that the performance of ASACPM and its application framework meet our expectations and have good reliability, validity, and rationality.

1. Introduction

The cloud computing is an emerging business computing model that uses the Internet to access shared basic resources anytime, anywhere, on-demand, and conveniently, which is regarded as a major innovation in the information industry and has received widespread attention from the academia, industry, and business community [1, 2]. At the same time, it is also a challenging field that covers all aspects of the information industry and is driving the transformation of information technology to socialization, intensification, and specialization. The governments, hospitals, research institutions, and industry giants use cloud computing to solve their growing computing and storage problems [3, 4].

The cloud service platform (CSP) is a third-party public service platform that draws on the idea of cloud computing and uses software as a service (SaaS) as its core application mode, which adopts information technology and Internet of Things technology to share the business resources (i.e., business data, business processes, and business services) of different enterprises to different tenants, thus breaking the "information island" [5, 6]. The CSP hosts data, services, and applications for tenants in the pay-as-you-go feature, breaking the traditional service mode and automatically allocating business resources according to tenant needs, thereby saving enterprise costs, lowering information thresholds, and enjoying professional information services [7, 8]. The CSP builds an application environment based on a flexible and scalable multitenant architecture feature to meet the diverse application needs and personalized form requirements of tenants [9, 10].

In the open Internet environment, the application mode and characteristics of CSP bring the following problems: first, when tenants access business resources of other collaborative platforms, CSP only supports tenants to access business resources within the platform, so there is a problem of

cross-platform access control. For example, CSP and other collaborative platforms all adopt the traditional access control model, and it is impossible for CSP tenants to access the business resources of other collaborative platforms and vice versa. Second, when tenants access business resources, the CSP authorization method is static and the authorization granularity is coarse-grained, so dynamic fine-grained authorization is not supported. For example, the type of enterprise, role category, and collaboration relationship all change dynamically with the increase or decrease of enterprises in the enterprise alliance, and the permissions of enterprises to access CSP (that is, grant or cancel permissions) should have changed dynamically, but in fact, the existing CSP is to preset the permissions of tenants. The platform cannot automatically adjust the corresponding access permissions according to the changes of the properties of the tenants, and the granularity of authorization is coarse-grained.

Third, CSP has many characteristics, such as many types of enterprises, large number of tenants, wide range of roles, and complex and dynamic collaboration relationship, which pose new challenges for CSP to build an access control model.

In order to solve the above problems, based on attribute-based access control (ABAC) [5, 10], this paper proposes a tenant-centric attribute semantic access control policy model (ASACPM) for the CSP. In ASACPM, access requests, access decisions, and evaluation results are formally described through attribute semantics, where access request is used to describe the requester, requested business resources, current environment, credentials and access methods, etc., and access decision is used to evaluate the adaptation result of the access request and the policy set, and the evaluation result is used to determine whether the tenant can access the service resource. In the application framework of ASACPM, the access decision mechanism is used to evaluate whether the tenant has CSP or cross-platform access control rights, the attribute synchronization mechanism is used to ensure the consistency of the AA of CSP and the AA of collaborative platform (CP), and the dynamic fine-grained authorization mechanism ensures that the CSP has better flexibility and scalability.

2. Related Work

This section focuses on reviewing and analyzing the recent academic results of traditional access control models and attribute-based access control models. The traditional access control model is divided into the following three categories according to the authorization method: the discretionary access control (DAC) model, the mandatory access control (MAC) model, and the role-based access control (RBAC) model [11, 12] The traditional access control model is based on the identities or fixed identifiers and is suitable for the centralized and closed network environments. According to the application mode and characteristics of the CSP, the traditional access control model is difficult to adapt to the open and shared Internet environment.

In recent years, domestic and foreign scholars have carried out a lot of research on the traditional access control model, and based on this, combined with the characteristics of cloud computing or cloud services, some improved access control models are put forward. Reference [13] proposed an emergency-RBAC (E-RBAC) model, which controlled the system in emergency situations according to the model constraints. Taking medical and dispensing scenarios as examples, the effectiveness of the model was verified. Reference [14] proposed a cloud computing access control model based on RBAC, which introduced the dynamic variable mechanism and security level into the access control strategy, improving the security and flexibility of access control to a certain extent. Reference [15] introduced the concept of trust degree into the RBAC model and proposed a dynamic RBAC model based on trust in cloud computing environment, which first gave the calculation method of trust degree and then assigned permissions to users according to their role information and trust degree, improving the security of cloud resource or service access process. However, the above references [13–15] did not give a cross-domain or cross-platform access control method. Reference [16] proposed an adaptive model based on cloud computing environment on the basis of RBAC model, which solved the problem of dynamic change of roles, but the access request of the model was relatively high in the process of dynamic role transformation. Reference [17] proposed a fine-grained access control mechanism in the cloud computing environment, which was used to deal with the dynamic changes of user permissions, but it was difficult to manage and control permissions in the face of dynamic changes of multiple types of roles and reduce the flexibility and scalability of the cloud computing platform. Reference [18] proposed a role-based access control model using smart contracts, which utilized the smart contract technology of Ethereum to realize the cross-organizational utilization of roles and ensured the security and adaptability of the platform. However, the corresponding permissions cannot be automatically adapted according to the needs of tenants. Although the above references improve the traditional access control model, it is still not fully suitable for building a dynamic and fine-grained access control model.

In the aspect of the attribute-based access control model, reference [19] proposed a fine-grained access control scheme based on ciphertext policy attribute encryption (CPABE) and trusted execution environment (TEE), which could be used to mitigate sensitive information attacks and improve confidentiality, but this scheme did not consider cross-domain or cross-platform access control methods. Reference [20] proposed an attribute-based access control policy model and gave definition, decision-making process, and access control policy of the model, but how to implement the model was not described in detail. Reference [21] proposed a new decentralized access control scheme for secure data storage in clouds that supported anonymous authentication, which could effectively protect the privacy of the data, but it did not involve cross-domain or cross-platform data storage. Reference [22] proposed a new distributed access control scheme, which supported the storage security of anonymous

users to verify data, prevented replay attacks, and supported the creation, modification, and reading of data stored in the cloud; however, the scheme did not involve cross-domain or cross-platform data storage. Reference [23] proposed an access control architecture model for restricted medical resources in IoT, which provided authorized users with fine-grained access to services based on policies while protecting valuable resources from unauthorized access. Although the above references improve the attribute-based access control model, it is also not fully applicable to constructing a dynamic and fine-grained access control model.

In summary, in the Internet environment, it is a series of challenging problems to research and meet the access control requirements of different tenants, explore the access decision mechanism and dynamic fine-grained authorization mechanism, and realize the cross-platform access control of tenants. Therefore, this paper solves the above problems by building ASACM.

3. The Modeling Process of the ASACPM

3.1. Formal Definition of the ASACPM. In this section, the concepts of the ASACPM are defined and used throughout this paper. The ABAC does not directly define the authorization between the subjects and the objects but uses a multirelationship among the subject attributes, the object attributes, and the environment attributes to define the authorization. So far, there is no unified formal definition of the ABAC. This paper draws on the formal definition of the existing ABAC [5, 10] and then optimizes it based on this and finally gives the formal definition of the ASACPM.

Definition 1. The basic elements of ASACPM are composed of subject, object, environment, certificate, and action, among which subject refers to the tenant (i.e., the requester), object refers to the business resources of CSP or CP, environment refers to the context in which transactions are processed, certificate refers to the SSL certificate issued by certificate authority (CA), and action refers to the access mode of the tenant which is usually related to the object. Subject, object, and environment are represented by attributes (i.e., entities).

Definition 2. In the ASACPM, all entities are described by attributes. The attribute is a variable consisting of the specified data type and value field, which can be represented by $\mathrm{Att} = (\mathrm{attn}, \mathrm{type}, \mathrm{Val}, \mathrm{R(Val)})$. Among them, attn indicates the attribute name, type indicates the data type, Val indicates the value range, and R(Val) indicates the relationship between different values as shown in Table 1. In Table 1, $\succ$ means precedence, $\succcurlyeq$ means inheritance, the meaning of $\prec$ and $\preccurlyeq$ is the opposite of $\succ$ and $\succcurlyeq$, Val means the value of the attribute, and $\mathrm{val}_1, \mathrm{val}_2 \in \mathrm{Val}$.

Definition 3. The attribute value pair (avp) represents the specific value of the attribute, which can be represented by avp = (attn, = ,val), and its mathematical expression is attn = val. In this paper, Savp, Oavp, and Eavp are used to represent the attribute value pairs of the subject, the object, and the environment, respectively.

Table 1: The relationship between different values.

Relationship	Mathematical expression
Comparative relationship	$\mathrm{R(Val)} = \{(\mathrm{val}_1 \bowtie_c \mathrm{val}_2) \mid \bowtie_c \in \{\leq, <, \geq, >, \neq, =\}\}$
Partial ordering relation	$\mathrm{R(Val)} = \{(\mathrm{val}_1 \bowtie_P \mathrm{val}_2) \mid \bowtie_P \in \{\preccurlyeq, \prec, \succcurlyeq, \succ, \neq, =\}\}$

Definition 4. The attribute predicate value pair (apvp) can be represented by apvp = (attn, $\bowtie_r$, val). Among them, $\bowtie_r \in \{\leq, <, \geq, >, \neq, =, \preccurlyeq, \prec, \succcurlyeq, \succ\}$ is a relational expression operator to limit the range of values of the attribute. Its mathematical expression is $\mathrm{attn} \bowtie_r \mathrm{val}$. In this paper, the Sapvp, Oapvp, and Eapvp are used to represent the attribute predicate value pairs of the subject, the object, and the environment, respectively.

Definition 5. The evaluation results of evaluating the apvp based on the avp are divided into three cases as shown in Table 2.

Definition 6. The tenant's standard access request (SAR) is defined as $\mathrm{SAR} = (\mathrm{s}, \mathrm{o}, \mathrm{e}, \mathrm{c}, \mathrm{a})$, where $\mathrm{s} = \{\mathrm{Savp}_1, \mathrm{Savp}_2, \cdots, \mathrm{Savp}_l\}$ represents the set of the Savp, $\mathrm{o} = \{\mathrm{Oavp}_1, \mathrm{Oavp}_2, \cdots, \mathrm{Oavp}_m\}$ represents the set of the Oavp, $\mathrm{e} = \{\mathrm{Eavp}_1, \mathrm{Eavp}_2, \cdots, \mathrm{Eavp}_n\}$ represents the set of the Eavp, *c* represents the tenant certificate, and *a* indicates the tenant's access method to the requested business resource, which includes browsing, adding, editing, deleting, and approving.

Definition 7. The ASACPM's policy is defined as $\mathrm{Pol} = (\mathrm{S}, \mathrm{O}, \mathrm{E}, \mathrm{C}, \mathrm{A}) \longrightarrow \mathrm{P}$, and the set of the policy is defined as $\mathrm{POL} = \{\mathrm{Pol}_1, \mathrm{Pol}_2, \cdots, \mathrm{Pol}_n\}$, where $\mathrm{S} = \{\mathrm{Sapvp}_1, \mathrm{Sapvp}_2, \cdots, \mathrm{Sapvp}_n\}$ represents the set of the Sapvp, $\mathrm{O} = \{\mathrm{Oapvp}_1, \mathrm{Oapvp}_2, \cdots, \mathrm{Oapvp}_n\}$ represents the set of the Oapvp, $\mathrm{E} = \{\mathrm{Eapvp}_1, \mathrm{Eapvp}_2, \cdots, \mathrm{Eapvp}_n\}$ represents the set of the Eapvp, $\mathrm{C} = \{\mathrm{C}_1, \mathrm{C}_2, \cdots, \mathrm{C}_n\}$ represents the set of the tenant certificate, $\mathrm{A} = \{\mathrm{A}_1, \mathrm{A}_2, \cdots, \mathrm{A}_n\}$ indicates the set of the access method (i.e., browsing, adding, editing, deleting, and approving), and $\mathrm{P} \in \{\mathrm{permit}, \mathrm{deny}\}$ represents the permission flag.

Definition 8. The evaluation results of evaluating the policy based on the SAR are divided into two cases as shown in Table 3.

When the Pol is applicable to the SAR (i.e., $[\mathrm{Pol}]_{\mathrm{SAR}} = \mathrm{True}$), since Pol's permission flag is $\mathrm{P} \in \{\mathrm{permit}, \mathrm{deny}\}$ (i.e., $[\mathrm{Pol}]_{\mathrm{SAR}} = \mathrm{True}$ has two states, the permissible state and the denial state), this paper defines the permissible state as $[\mathrm{Pol}]_{\mathrm{SAR}} = \mathrm{True}^{+} = \mathrm{permit}$ and the denial state as $[\mathrm{Pol}]_{\mathrm{SAR}} = \mathrm{True}^{-} = \mathrm{deny}$. In Table 3, if the $[\mathrm{Pol}]_{\mathrm{SAR}}$ is true (i.e., $[\mathrm{Pol}]_{\mathrm{SAR}} = \mathrm{True}$), then the final evaluation results (FERs) of the $[\mathrm{Pol}]_{\mathrm{SAR}}$ refer to the cases 1 and 2 in Table 4, and if the $[\mathrm{Pol}]_{\mathrm{SAR}}$ is false (i.e., $[\mathrm{Pol}]_{\mathrm{SAR}} = \mathrm{False}$),

Table 2: The evaluation results of evaluating the apvp based on the avp.

Cases	Evaluation results
Case 1	Description: given avp and apvp, if their attribute names are the same and the value of the avp is within the range defined by the apvp, then the evaluation result of the avp on the apvp is true; otherwise, the evaluation result is false, and the mathematical expression is as follows: $[\text{apvp}]_{\text{avp}} = \begin{cases} \text{True}, (\text{avp.attn} = \text{apvp.attn}) \wedge (\text{avp.val} \in \text{apvp.val}) \\ \text{False}, \text{otherwise} \end{cases}$
Case 2	Description: given the set of the attribute value pairs $\text{AVP} = \{\text{avp}_1, \text{avp}_2, \cdots, \text{avp}_n\}$ and apvp, if $\exists \text{avp}_i \in \text{AVP}$ makes $[\text{apvp}]_{\text{avp}_i}$ true, then the evaluation result (i.e., $[\text{apvp}]_{\text{AVP}}$) of the AVP on the apvp is true; otherwise, the evaluation result is false, among which $i \in N^*$, and the mathematical expression is as follows: $[\text{apvp}]_{\text{avp}} = \begin{cases} \text{True}, \bigvee_{i=1}^{n} [\text{apvp}]_{\text{avp}_i \in \text{AVP}} = \text{True} \\ \text{False}, \text{otherwise} \end{cases}$
Case 3	Description: given AVP and the set of the attribute predicate value pairs $\text{APVP} = \{\text{apvp}_1, \text{apvp}_2, \cdots, \text{apvp}_n\}$, if $\forall \text{apvp}_i \in \text{APVP}$ makes $[\text{apvp}_i]_{\text{AVP}}$ true, then the evaluation result (i.e., $[\text{APVP}]_{\text{AVP}}$) of the AVP on the APVP is true; otherwise, the evaluation result is false, among which $i \in N^*$, and the mathematical expression is as follows: $[\text{APVP}]_{\text{AVP}} = \begin{cases} \text{True}, \bigwedge_{i=1}^{n} [\text{apvp}_i \in \text{APVP}]_{\text{AVP}} = \text{True} \\ \text{False}, \text{otherwise} \end{cases}$

Table 3: The evaluation results of evaluating the policy based on the SAR.

Cases	Evaluation results
Case 1: single policy	Description: given $\text{SAR} = (s, o, e, c, a)$ and $\text{Pol} = (S, O, E, C, A) \longrightarrow P$, if $[S]_s \wedge [O]_o \wedge [E]_e \wedge (c \in C) \wedge (a \in A)$ is true, then the evaluation result (i.e., $[\text{Pol}]_{\text{SAR}}$) of the SAR on Pol is true, namely, Pol applies to SAR. Otherwise, the evaluation result is false, namely, Pol does not applies to SAR; the mathematical expression is as follows: $[\text{Pol}]_{\text{SAR}} = \begin{cases} \text{True}, [S]_s \wedge [O]_o \wedge [E]_e \wedge (c \in C) \wedge (a \in A) = \text{True} \\ \text{False}, \text{otherwise} \end{cases}$
Case 2: policy set	Description: given $\text{SAR} = (s, o, e, c, a)$ and $\text{POL} = \{\text{Pol}_1, \text{Pol}_2, \cdots, \text{Pol}_n\}$, if $\exists \text{Pol}_i \in \text{POL}$ makes $[\text{Pol}_i]_{\text{SAR}}$ is true, then the evaluation result (i.e., $[\text{POL}]_{\text{SAR}}$) of SAR on POL is true, namely Pol_i in POL applies to SAR. Otherwise, the evaluation result is false, namely, $\forall \text{Pol}_i$ in POL does not applies to SAR; the mathematical expression is as follows: $[\text{POL}]_{\text{SAR}} = \begin{cases} \text{True}, \bigvee_{i=1}^{n} [\text{Pol}_i \in \text{POL}]_{\text{SAR}} = \text{True} \\ \text{False}, \text{otherwise} \end{cases}$

Table 4: The final evaluation results.

Cases	Evaluation results	FERs
Case 1	$[\text{Pol}]_{\text{SAR}} = \text{True}^+$ $[\text{POL}]_{\text{SAR}} = [\text{Pol}_1]_{\text{SAR}} \vee [\text{Pol}_2]_{\text{SAR}} \vee \cdots = \text{True}^+ \vee \text{True}^+ \vee \cdots = \text{True}^+$	{permit}
Case 2	$[\text{Pol}]_{\text{SAR}} = \text{True}^-$ $[\text{POL}]_{\text{SAR}} = [\text{Pol}_1]_{\text{SAR}} \vee [\text{Pol}_2]_{\text{SAR}} \vee \cdots = \text{True}^- \vee \text{True}^- \vee \cdots = \text{True}^-$	{deny}
Case 3	$[\text{POL}]_{\text{SAR}} = [\text{Pol}_1]_{\text{SAR}} \vee [\text{Pol}_2]_{\text{SAR}} \vee \cdots = \text{True}^+ \vee \text{True}^- \vee \cdots = \text{True}^+ \vee \text{True}^-$	{permit, deny}
Case 4	$[\text{Pol}]_{\text{SAR}} = \text{False}$ $[\text{POL}]_{\text{SAR}} = [\text{Pol}_1]_{\text{SAR}} \vee [\text{Pol}_2]_{\text{SAR}} \vee \cdots = \text{False} \vee \text{False} \vee \cdots = \text{False}$	{False}

then the FER of the $[\text{Pol}]_{\text{SAR}}$ refers to the case 4 in Table 4.

In Table 3, if the $[\text{POL}]_{\text{SAR}}$ is true (i.e., $[\text{POL}]_{\text{SAR}} = \text{True}$), then obviously, the $\bigvee_{i=1}^{n} [\text{Pol}_i]_{\text{SAR}} = [\text{Pol}_1]_{\text{SAR}} \vee [\text{Pol}_2]_{\text{SAR}} \vee \cdots \vee [\text{Pol}_n]_{\text{SAR}}$ is true; namely, there is one or more $[\text{Pol}]_{\text{SAR}}$ such that the $[\text{POL}]_{\text{SAR}}$ is true. If one $[\text{Pol}]_{\text{SAR}}$ is true, then the FERs are the same as the result of Table 3, and if at least two $[\text{Pol}]_{\text{SAR}}$ are true, then after performing the disjunction operation, the FERs of the $[\text{POL}]_{\text{SAR}}$ refer to the cases 1-3 in Table 4. If the $[\text{POL}]_{\text{SAR}}$ is false (i.e., $[\text{POL}]_{\text{SAR}} = \text{False}$; specifically, all $[\text{Pol}]_{\text{SAR}}$ is false), then after performing the disjunction operation, the FER of $[\text{POL}]_{\text{SAR}}$ refers to the case

4 in Table 4. When the FER is {permit, deny}, this type of result can be handled by the rejection priority principle, the license priority principle, the first application principle, and the unique application principle [24].

In summary, given the set of logical variable as $U = \{[\mathrm{Pol}_1]_{\mathrm{SAR}}, [\mathrm{Pol}_2]_{\mathrm{SAR}}, \cdots, [\mathrm{Pol}_n]_{\mathrm{SAR}}\}$ and the function of U as $[\mathrm{POL}]_{\mathrm{SAR}} = \bigvee_{i=1}^{n} [\mathrm{Pol_i}]_{\mathrm{SAR}}$, ask if there is a logical variable $[\mathrm{Pol}]_{\mathrm{SAR}}$ in U that satisfies $[\mathrm{POL}]_{\mathrm{SAR}}$. Obviously, this is a typical SATISFIABILITY problem (SAT for short), according to the Cook's theorem; the SAT problem is the NP-complete problem. It can be deduced that the access control problem in this paper is also the NP-complete problem.

Based on the above definition, this paper gives the mathematical expressions of the conditions, constraints, and objective functions necessary for the ASACPM (access control problem). Among them, the mathematical expressions of the conditions and constraints of the ASACPM are as follows.

$$\begin{cases} \mathrm{SAR} = (s, o, e, c, a), \forall \mathrm{SAR} \in \mathrm{CSP}, \\ \mathrm{POL} = \{\mathrm{Pol}_1, \mathrm{Pol}_2, \cdots, \mathrm{Pol}_n\}, \mathrm{Pol} = (S, O, E, C, A) \longrightarrow P. \end{cases} \tag{1}$$

In this paper, the conditions and their constraints (i.e., Equation (1)) are divided into multivariate relationship and affiliation relationship. Among them, the multivariate relationship refers to the relationship between the entity attributes of the SAR (i.e., s, o, e) and attribute conditions of the Pol (i.e., S, O, E). The affiliation relationship refers to whether the certificate and action (i.e., c, a) of the *SAR* belong to Pol's certificate set and action set (i.e., C, A), where the entity attribute refers to the subject attribute, the object attribute, and the environment attribute (i.e., Definition 1).

The mathematical expression of the ASACPM objective function is as follows.

$$[\mathrm{Pol}]_{\mathrm{SAR}} = \begin{cases} \mathrm{True}, \left([S]_s \wedge [O]_o \wedge [E]_e \wedge (c \in C) \wedge (a \in A) = \mathrm{True}\right), \\ \mathrm{False}, \text{otherwise}, \end{cases} \tag{2}$$

$$[POL]_{SAR} = \begin{cases} \mathrm{True}, \bigvee_{i=1}^{n} [Pol_i \in POL]_{SAR} = \mathrm{True}, \\ \mathrm{False}, \text{otherwise}. \end{cases} \tag{3}$$

Given the conditions and constraints (Equation (1)), the ASACPM first obtains the evaluation result by calculating the objective function (Equations (2) and (3)) and then obtains the FER based on the evaluation result and Table 4 and finally according to the FER which determines whether the tenant has access to the platform's business resources. In the policy evaluation, $[\mathrm{Pol}]_{\mathrm{SAR}}$ in Equations (2) and (3) is true if and only if all the multivariate relationships and affiliation relationships are true; that is, Pol applies to SAR.

3.2. Implementation of the ASACPM. Policy evaluation algorithm (PEA) is the core algorithm for implementing ASACPM. It is mainly used to evaluate whether a tenant's access request meets the requirements of the policy set. The detailed process and pseudocode are as follows:

Step 1. Given the necessary conditions as $\mathrm{SAR} = (s, o, e, c, a)$ and $\mathrm{POL} = \{\mathrm{Pol}_1, \mathrm{Pol}_2, \cdots, \mathrm{Pol}_n\}$, the constraints are $\forall SAR \in CSP$ and $Pol = (S, O, E, C, A) \longrightarrow P$, that is, Equation (1).

Step 2. The conditional SAR and POL (i.e., Equation (1)) are taken as parameters to the objective function (i.e., Equations (2) and (3)) for the calculation to obtain the evaluation result.

Step 3. According to the evaluation result and Table 4, the FER is obtained.

4. Application Framework of the ASACPM

4.1. Framework of the ASACPM. The overall framework of the tenant-centric attribute semantic access control policy model for cloud service platform is shown in Figure 1. It consists of four modules, namely, tenant and platform module, CA module, web module, and decision module. These modules can not only complete the corresponding tasks independently. They can also communicate with each other and work together. The tenant first initiates an access request and then establishes an SSL secure channel to evaluate the access request and next return the result to form a complete closed loop. The detailed description of the meaning and function of each module is as follows.

4.1.1. Tenant and Platform Module. Tenants (i.e., requesters) refer to enterprise users in the enterprise alliance consisting of suppliers, manufacturers, distributors, service providers, and agents. The CSP is a third-party public service platform that provides business resources (such as business data, business services, and business processes) to different tenants through the Internet, while the CP can also provide different business resources for the tenants (complementing the resources of the CSP). This paper builds the ASACPM so that the tenant can not only access the business resources of the CSP but also access the business resources of the CP through the CSP, thereby breaking the "information island."

4.1.2. CA Module. The CA is responsible for the issuance and management of SSL certificate, including tenant certificates and server certificates. The main role of the SSL certificate is to implement data encryption transmission and communication entity identity authentication [7], so it can solve many security problems of business resources in the open Internet environment. In ASACPM, SSL certificate can be used not only to build authorization relationships between tenant and business resource but also to establish trust relationships between tenant and server or between cloud platform and collaborative platform. Therefore, when the tenant first uses the platform, the unique tenant certificate is issued by the CA, and this certificate is also saved to the policy administration point of the decision-making body to form the tenant certificate set of the policy which is one of

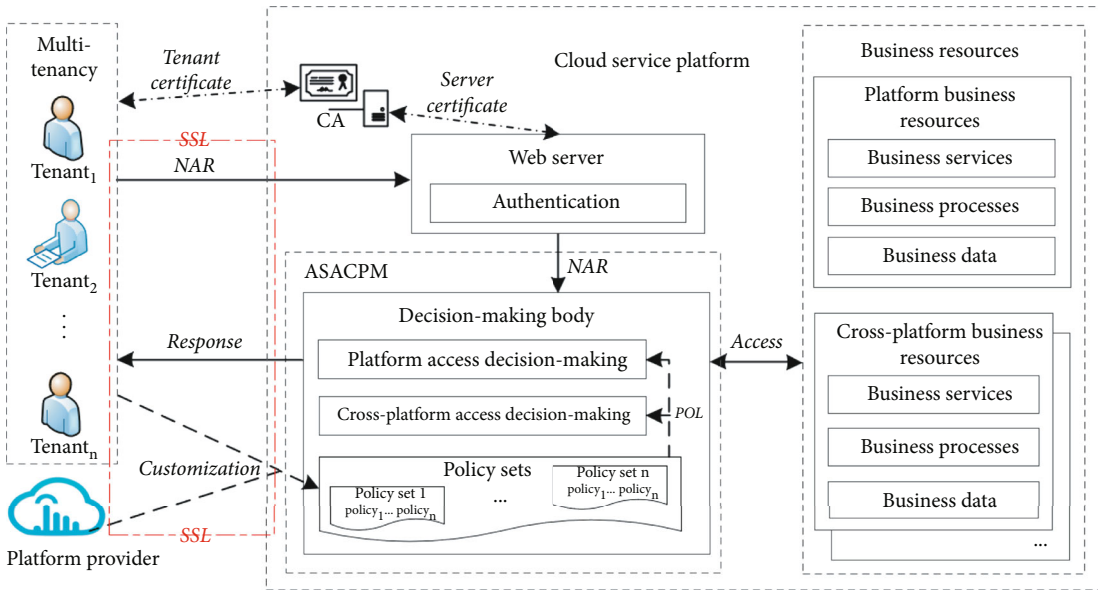

Figure 1: The framework of the tenant-centric attribute semantic access control policy model for cloud service platform.

the basic elements of the calculation of the objective function (Equations (2) and (3)). The version adopted by SSL is V3.0, and the specification adopted by CA is X.509 V3.

4.1.3. Web Module (Web Server). When the tenant accesses the business resources of the CSP, a secure SSL channel (that is, data encryption transmission) is established between the tenant and the web server. When an SSL session is generated, firstly, the web server sends the server certificate to the tenant, and the tenant automatically analyzes the server certificate and verifies the identity of the web server. Secondly, the tenant sends the tenant certificate to the web server, and the web server verifies the identity of the tenant. The verification content includes the certificate authority, the validity period of the certificate, and whether the certificate has been tampered with or revoked. Finally, after the two parties authenticate successfully and establish an SSL secure channel, the web server sends the received tenant's native access request (NAR) to the decision-making body. In addition, this paper mainly focuses on the access control and dynamic fine-grained authorization of business resources. Therefore, the specific processes of constructing SSL secure channel and identifying communication entity authentication can be referred to reference [7], which will not be researched in detail in this paper.

4.1.4. Decision Module (Decision-Making Body). According to the conditions and constraints (Equation (1)) and the objective function (Equations (2) and (3)), the decision-making body is divided into conditional part and decision part, which are described in detail below.

(i) Conditional part: the policy/policy set is formulated by the tenant and the platform provider according to the sharing requirements of the business resources and the management requirements of the CSP, respectively, which is a necessary condition for the access decision. The policy consists of explicit element and implicit element; among them, the explicit element refers to the subject, the object, the environment, and the action; the entity attribute set of the first three elements and action set need to be customized by the tenant (platform provider) and stored in the policy administration point of the decision-making body; and the implicit element refers to the certificate, which is composed of the tenant certificate set and stored in the policy administration point of the decision-making body, so it does not require tenant (platform provider) customization. The access request has five elements: the subject, the object, the environment, the certificate, and the action, respectively, for describing the entity attribute set of the requester, the entity attribute set of the requested business resource, the entity attribute set of the current environment, the tenant certificate, and the access method. It is another necessary condition for the access decision. Since the business resources accessed by tenants may belong to different platforms, this article divides access request into platform access request and cross-platform access request

(ii) Decision part: the ASACPM's access decision mechanism is based on conditions and constraints (Equation (1)) through the calculation of the objective function (Equations (2) and (3)) to obtain the evaluation results, and then based on the evaluation results and Table 4 to obtain the FERs, and finally based on the FERs whether the tenant has access to the platform's business resources. Since access requests are divided into the platform access request and cross-platform access request, the access decision mechanism is divided into the platform access decision and cross-platform access decision, and the

detailed description refers to Section 4.2. The ASACPM's attribute synchronization ensures attribute authority consistency across platforms and provides support for access decision mechanisms, and the detailed description refers to Section 4.3. The ASACPM's dynamic fine-grained authorization mechanism restricts dynamic tenant access requests by increasing the policy's attribute conditions, and the detailed description refers to Section 4.4

4.2. Access Decision Mechanism. The eXtensible Access Control Markup Language (XACML) [25] is an open standard language based on the eXtensible Markup Language (XML) that can be used to determine the general access control policy language for request/response and the framework for executing authorization policies. This language is well interoperable, versatile, and extensible while also supporting dynamic fine-grained access control. This paper adopts XACML to provide a unified writing specification for tenant's access request, platform's policy, and decision-making, which makes the access control system of different platforms universal and then gives the detailed process of access decision implementation.

According to the process of executing the authorization policy, the decision-making body is divided into the policy enforcement point (PEP), the policy decision point (PDP), the attribute authority (AA), and the policy administration point (PAP), in which PEP and PAP belong to the conditional part, PDP belongs to the decision part, and AA can belong to both the conditional part and the decision part. The platform access decision and cross-platform access decision include the PEP, PDP, AA, and PAP, and their functions are described in detail as follows:

(i) The function of the PEP is to translate the NAR into the SAR based on the attribute table provided by the AA and then obtain the decision result from the PDP. If the FER is permit, the tenant is permitted access to the business resource; otherwise, the tenant is denied access to the business resource

(ii) The function of the PDP is to calculate the objective function (Equations (2) and (3)) according to the conditions and constraints (Equation (1)) to obtain the evaluation result and obtain the FER based on the evaluation result and Table 4

(iii) The function of the AA is to store and manage the attribute table (which includes the subject, the object, and the environment) and the relationship between attributes and their values and provide attribute support for the PEP and PDP

(iv) The function of the PAP is to store and manage the policy sets formulated by the tenant and provide the necessary condition (i.e., policy sets) for PDP.

4.2.1. Platform Access Decision Mechanism

(1) Process of Platform Access Decision. The process of platform access decision is shown in Figure 2. When a tenant accesses the business resources of the CSP, first the CSP establishes an SSL channel between the tenants and the web server according to the tenant certificate and server certificate, and then, the web server sends the received NAR to the decision-making body, and finally, the decision-making body executes the platform access decision mechanism to obtain the final evaluation result.

Step 1. When the tenant initiates the NAR, the identity of tenant and web server is authenticated according to the tenant certificate and the server certificate, then the SSL channel between them is established, and next, the NAR is sent to the web server.

Step 2. The web server receives the NAR and sends the NAR to the decision-making body of the CSP.

Step 3. After the PEP of the decision-making body receives the NAR, the process of the platform access decision handling NAR is as follows.

(a) The AA provides attribute support for PEP and PDP, including the attribute tables and relationship between attributes and their values

(b) The PEP transforms the NAR into SAR = (s, o, e, c, a) based on the attribute table provided by the AA. Since all the object (i.e., business resource) stored in the object attribute table of the AA has a corresponding subordinate label (i.e., the platform to which the resource belongs), the PEP can identify the SAR as the platform access request and send the SAR to the PDP

(c) The PDP receives the SAR, which obtains the policy set POL = $\{\text{Pol}_1, \text{Pol}_2, \cdots, \text{Pol}_n\}$ from the PAP. In the case where the given conditions are SAR and POL, the evaluation result is calculated by Algorithm 1, and the calculation process needs to map the entity attribute of the policy to a specific value or a range of values according to the relationship between the attribute and its value provided by the AA

(d) The PDP obtains the FER (i.e., mapping the evaluation result to the FER) based on the evaluation result and Table 4 and then sends the FER to the PEP

Step 4. The PEP performs the FER, which permits or denies the tenant access to the CSP's business resources.

(2) Platform Access Decision Algorithm. Platform access decision algorithm (PADA) is one of the core algorithms for implementing access decision mechanism, which is mainly used to process platform access requests initiated by tenants.

4.2.2. Cross-Platform Access Decision Mechanism

(1) Process of Cross-Platform Access Decision. The process of cross-platform access decision is shown in Figure 3. When a

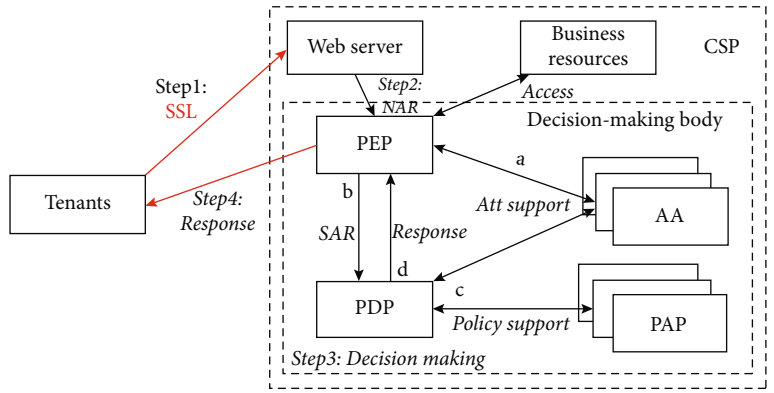

Figure 2: The process of platform access decision.

Input: Given $SAR = (s, o, e, c, a)$ and $POL = \{Pol_1, Pol_2, \cdots, Pol_n\}$, among them, $Pol = (S, O, E, C, A)$
Output: FER
1: Policy_Evaluation ($SAR = (s, o, e, c, a)$, $POL = \{Pol_1, Pol_2, \cdots, Pol_n\}$) /* (Equation (1)) */
2: FOR each k in n do
3: IF $[S(k)]_s \wedge [O(k)]_o \wedge [E(k)]_e \wedge (c \in C(k)) \wedge (a \in A(k)) = True$ /* Performing the conjunction operation (namely Equations (2) and (3)) */
4: $[Pol_k]_{SAR} = True$
5: $[POL]_{SAR} += \vee [Pol_k]_{SAR}$
6: ELSE
7: $[Pol_k]_{SAR} = False$
8: $[POL]_{SAR} += \vee [Pol_k]_{SAR}$
9: END IF
10: END FOR
11: $[POL]_{SAR} = [Pol_1]_{SAR} \vee [Pol_2]_{SAR} \vee \cdots \vee [Pol_k]_{SAR}$ /* Performing the disjunction operation (namely Eq. (2)) */
12: $[POL]_{SAR} \longrightarrow FER$ /*$[POL]_{SAR}$ maps to FER*/
13: Return FER
14: End Policy_Evaluation

Algorithm 1: Policy evaluation algorithm (PEA).

Input: Tenant's NAR
Output: *Permit* or *deny* to access business resources for the CSP
1: $\forall NAR$
2: PEP_Fuction (NAR)
3: Getting attribute tables from the AA
4: Translating NAR to SAR
5: Executing PDP_Fuction (SAR)
6: Executing FER
7: *Permit* or *deny* to access business resources
8: END PEP_Fuction
9: PDP_Fuction (SAR)
10: Getting SAR from the PEP
11: Getting $POL = \{Pol_1, Pol_2, \cdots, Pol_n\}$ from the PAP
12: Getting the relationship between attributes and their values from the AA
13: Policy_Evaluation (SAR, POL) /* Executing **Policy Evaluation Algorithm (PEA)***/
14: Return FER
15: END PDP_Fuction

Algorithm 2: Platform access decision algorithm (PADA).

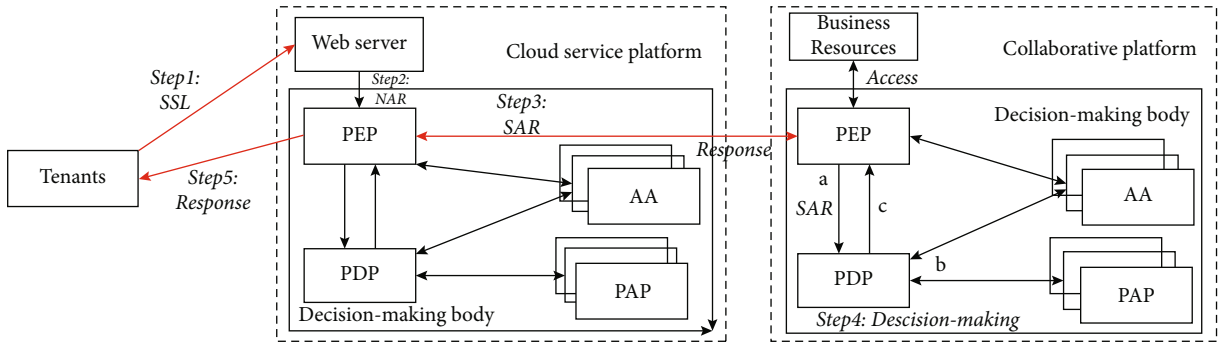

Figure 3: The process of cross-platform access decision.

Input: Tenant's *NAR*
Output: *Permit* or *deny* to access business resources of the CP
1: ∀*NAR*
2: CSP.PEP_Fuction (*NAR*)
3: Getting attribute tables from the AA
4: Translating *NAR* to *SAR*
5: IF *SAR* is the cross-platform access request
6: Sending *SAR* to CP.PEP
7: ENDIF
8: END CSP.PEP_Fuction
9: CP.PEP receives *SAR* and verifies the integrity of *SAR*
10: CP.PEP_Fuction (*SAR*)
11: Executing CP.PDP_Fuction (*SAR*)
12: Executing *FER*
13: *Permit* or *deny* to access business resources of the CP
14: END CP.PEP_Fuction
15: CP.PDP_Fuction (*SAR*)
16: Getting *SAR* from the CP.PEP
17: Getting $POL = \{Pol_1, Pol_2, \cdots, Pol_n\}$ from the CP.PAP
18: Getting the relationship between attributes and their values from the *CP.AA*
19: Policy_Evaluation (*SAR*, *POL*)/∗ Execute **Policy Evaluation Algorithm (PEA)**∗/
20: Return *FER*
21: END CP.PDP_Fuction

Algorithm 3: Cross-platform access decision algorithm (CADA).

tenant accesses the business resources of the CP through the CSP, first the CSP establishes an SSL channel between the tenants and the web server according to the tenant certificate and server certificate, and then, the web server sends the received NAR to the decision-making body, and finally, the decision-making body executes the cross-platform access decision mechanism to obtain the final evaluation result.

Step 1. When the tenant initiates the NAR, the identity of tenant and web server are authenticated according to the tenant certificate and the server certificate, then the SSL channel between them is established, and next, the NAR is sent to the web server.

Step 2. The web server receives the NAR and sends the NAR to the decision-making body of the CSP.

Step 3. After the CSP.PEP (i.e., the PEP of the decision-making body of the CSP) receives the NAR, it is transformed into SAR = (s, o, e, c, a) based on the attribute table provided by the AA. Since all the object (i.e., business resource) stored in the object attribute table of the AA has a corresponding subordinate label (i.e., the platform to which the resource belongs), the CSP.PEP can identify the SAR as the cross-platform access request and send the SAR to the decision-making body of the CP.

Step 4. After the CP. PEP (it has a similar meaning to the CSP.PEP) receives SAR, the process of the cross-platform access decision handling SAR is as follows.

(a) The CP.PEP verifies the integrity of the SAR and sends it to the CP.PDP

(b) The CP.PDP receives the SAR, which obtains the policy set $POL = \{Pol_1, Pol_2, \cdots, Pol_n\}$ from the CP.PAP. In the case where the given conditions are SAR and POL, the evaluation result is calculated by Algorithm 1, and the calculation process needs to map the entity attribute of the policy to a specific value or a range of values according to the relationship between the attribute and its value provided by the CP.AA

(c) The CP.PDP obtains the FER (i.e., mapping the evaluation result to the FER) based on the evaluation result and Table 4 and then sends the FER to the CP.PEP

Step 5. The CP.PEP performs the FER, which permits or denies the tenant access to the CP's business resources based on the CSP.

(2) Cross-Platform Access Decision Algorithm. Cross-platform access decision algorithm (CADA) is one of the core algorithms for implementing access decision mechanism, which is mainly used to process cross-platform access requests initiated by tenants.

4.3. Attribute Synchronization Mechanism

4.3.1. Methods of Attribute Synchronization. The dynamic characteristics of CSP lie in the fact that new enterprises join and old enterprises exit from the enterprise alliance at any time [7, 10]. Therefore, the CSP grants or cancels the access permissions of the enterprise users with the dynamic characteristics, which makes the information of the subject attribute (SA) often change. If the SA changes, the CSP needs to update the subject attribute table in the AA and then informs the CP to update the subject attribute table in the AA; that is, the CSP needs to ensure the consistency of the AA of different platforms. If the attribute table of the CSP or CP is not updated, it will affect the policy evaluation (i.e., the calculation of Equations (2) and (3) will be affected).

When the CSP or CP updates the SA in the subject attribute table, if the SA is being used by other operations, a conflict will occur. Therefore, this paper introduces the PV operation [26] to solve the mutual exclusion problem of attribute update and invoke of the CSP or CP (i.e., attribute synchronization problem), the PV operation is related to the processing of semaphore, and the P operation and V operation must appear in pairs. The P and V are from the initials of Dutch words, P is usually explained as proberen (i.e., "to test" or " to try"), and V is usually explained as verhogen (i.e., "increase") [27]. We set a semaphore $\mathcal{S}$ for the SA. If the SA is not invoked by other operations, the initial value $\mathcal{S} = 1$ is given. If the SA is invoked by another operations, the initial value $\mathcal{S} = 0$ is given.

The detailed definition of the PV operation: executing a P operation means that the platform performs the attribute update operation, so the value of $\mathcal{S}$ is decremented by 1. When $\mathcal{S} \geq 0$, which means that the platform can update the SA in the subject attribute table, and when $\mathcal{S} < 0$, which means that the SA is invoked by other operations and the platform cannot perform the update operation, the update operation is in the wait state. Executing a V operation means releasing the update operation of the platform or the SA invoked by other operations, so the value of $\mathcal{S}$ is incremented by one. When $\mathcal{S} \leq 0$, it means that the update operation in the wait state is awakened to make it run (i.e., executing the update operation).

Input: $\mathcal{S} = 0$ or $\mathcal{S} = 1$
Output: Performing the update operation
IF the attribute or its value is changed
  IF the attribute or its value is invoked
    $\mathcal{S} = 0$
    Executing P_Operation ($\mathcal{S}$)
    Executing V_Operation ($\mathcal{S}$)
  ELSE
    $\mathcal{S} = 1$
    Executing P_Operation ($\mathcal{S}$)
    Executing V_Operation ($\mathcal{S}$)
  END IF
END IF
P_Operation ($\mathcal{S}$: Semaphore)
  $\mathcal{S} = \mathcal{S} - 1$
  IF $\mathcal{S} < 0$
    Wait ($\mathcal{S}$)
  ELSE
    Performing the update operation
  END IF
END P_Operation
V_Operation ($\mathcal{S}$: Semaphore)
  $\mathcal{S} = \mathcal{S} + 1$
  IF $\mathcal{S} = <0$
    Resume ($\mathcal{S}$)
    Performing the update operation
  ELSE
    Releasing the update operation
  END IF
END V_Operation

Algorithm 4: Attribute synchronization algorithm (ASA).

The PV operation can not only update the subject attribute table but also update other attribute tables and relationships between attributes and their values. Attribute synchronization (i.e., the PV operation) not only ensures the consistency the AA of the CSP and the AA of the CP but also solves the mutual exclusion problem of attribute update and invoke, and it is also the basis of access control mechanism and dynamic fine-grained authorization mechanism.

4.3.2. Attribute Synchronization Algorithm. Attribute synchronization algorithm (ASA) adopts PV operation to solve the mutual exclusion problem of attribute updating and invoking (i.e., attribute synchronization problem).

4.4. Dynamic Fine-Grained Authorization Mechanism. The ASACPM defines a dynamic fine-grained authorization mechanism that determines the tenant's permissions based on the SAR evaluation of the Pol and applies to an open,

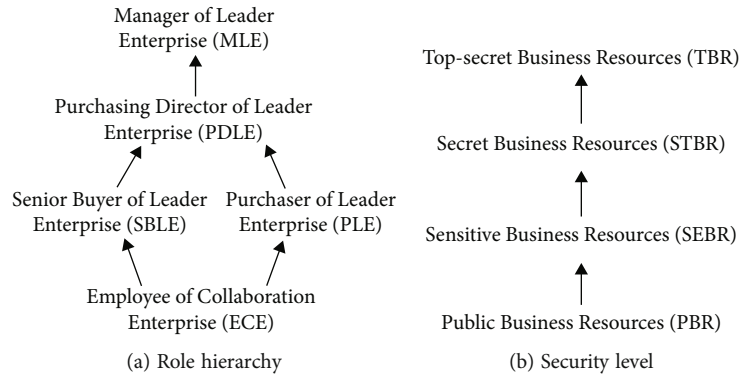

Figure 4: The relationship between different attribute values.

shared, and complex Internet environment. In the case that the SAR is unchanged, the owner of the business resource restricts the tenant's access to the business resources by increasing the attribute condition of the policy, thereby forming the finer-grained attribute condition constraint, and the formation process can refer to Section 5.1.

It can be seen from the above that the dynamic and fine-grained authorization mechanism of the model defines the multivariate relationship between the entity attributes of SAR and the attribute conditions of Pol. If and only if all multivariate relationships (i.e., $[S]_s$, $[O]_o$, and $[E]_e$) and affiliation relationships (i.e., $c \in C$ and $a \in A$) are true, $[\mathrm{Pol}]_{\mathrm{SAR}}$ or $[\mathrm{POL}]_{\mathrm{SAR}}$ (i.e., Equations (2) and (3)) can be true, thereby resulting in that Pol applies to SAR (i.e., the authorization is valid). When the entity attributes of SAR remain unchanged, we can conclude that the more attribute conditions of Pol, the finer the granularity of access control, which fully shows that the mechanism has good scalability.

In addition, when the AA of different platforms is the same and the policy set remains unchanged, no matter how the SA of the tenants in the enterprise alliance changes, the $[\mathrm{Pol}]_{\mathrm{SAR}}$ or $[\mathrm{POL}]_{\mathrm{SAR}}$ can only be true if all the multivariate relationships and affiliation relationships are true; that is, the tenant can access the business resources. Therefore, this mechanism can effectively solve the problem of dynamic characteristics of access permissions, which fully shows that it has good flexibility.

In summary, this paper makes the CSP not only have good scalability and flexibility through the dynamic fine-grained authorization mechanism but also can adapt to future development and application.

5. Analysis and Evaluation of the ASACPM

5.1. Case Analysis. The ASACPM and its application framework are applied to the ASP-/SaaS-based manufacturing industry value chain collaboration platform [24], and the specific process of the ASACPM and its application framework performing access decisions are described through corresponding business scenarios. We use the SA to represent the subject attribute, the OA to represent the object attribute, and the EA represent the environment attribute. According to the actual business requirements, the relationship between the attributes of some SAs and OAs is given, as shown in Figures 4(a) and 4(b), respectively, where the srole represents the role hierarchy of the SA and the obsl represents the security level of the OA. Each level of business resources includes business data, business services, and business processes.

Table 5 gives the SAR of the tenant, Table 6 shows the simple policy set (i.e., POL), and the definitions of the srole and obsl are shown in Figure 4.

Given the tenant's SAR (Table 5) and POL (Table 6) (i.e., given the input of Algorithm 1), the evaluation results (ERs) are calculated by Algorithm 1, and the ERs are mapped to the FERs, which are shown in Table 7. This paper assumes that "T^+" means True$^+$, "T^-" means True$^-$, and "F" means False; the default authorization is deny (i.e., "F" is deny).

It can be seen from Table 7 that for the SAR_2 and SAR_6, since there is no suitable policy in the POL, the FER is the default authorization (deny); that is, the tenant cannot add or browse the business resources of the CSP or CP. For the SAR_5, there are two suitable policies in the POL, the ER includes deny and permit. According to the license priority principle [5, 9], the FER is permit; that is, the tenant can delete the business resources of the CSP or CP, among which the license priority principle means that the ER of at least one policy is T^+, and the FER is {permit}.

Based on the above example, we describe in detail the implementation process of the dynamic fine-grained authorization mechanism of ASACPM and its application framework. It is known from Section 4.4 that this mechanism defines the multivariate relationship between the entity attributes of the SAR and the attribute conditions of the Pol, which limits the SAR by increasing the attribute condition of the Pol.

For example, we add an attribute condition to the S of the Pol_1 in Table 6 as shown in Table 8, give the tenant's SAR (Table 5), and then calculate the ERs based on the Pol_1 and SAR through Algorithm 1, and finally, the ERs are mapped to the FERs as shown in Table 9. It can be seen from Table 9 that the Pol_1 is not applicable to all SARs in Table 5, where st = CU represents the attribute condition added in the S, st represents the requester (i.e., tenant), and CU represents the automobile enterprise alliance, which

Table 5: The SAR of the tenant.

SAR	s	o	e	c	a
SAR_1	srole = ECE	obsl = PBR	etime = 11 : 30	C_1	Browsing
SAR_2	srole = SBLE	obsl = SEBR	etime = 13 : 30	C_2	Adding
SAR_3	srole = PLE	obsl = STBR	etime = 10 : 30	C_3	Editing
SAR_4	srole = PDLE	obsl = TBR	etime = 15 : 30	C_4	Approving
SAR_5	srole = MLE	obsl = STBR	etime = 10 : 30	C_5	Deleting
SAR_6	srole = ECE	obsl = PBR	etime = 11 : 30	None	Browsing

Table 6: The simple policy set.

POL	S	O	E	C	A	P
Pol_1	srole ⩾ ECE	obsl ⩾ PBR	8 : 30 < etime < 17 : 00	$\{C_1, \cdots, C_m\}$	Browsing	Permit
Pol_2	srole ⩾ SBLE	obsl = PBR	8 : 30 < etime < 17 : 00	$\{C_1, \cdots, C_m\}$	Browsing, adding	Permit
Pol_3	srole ⩾ PLE	obsl ⩾ STBR	9 : 30 < etime < 17 : 30	$\{C_1, \cdots, C_m\}$	Editing, deleting	Deny
Pol_4	srole ⩾ SBLE	obsl = SEBR	9 : 00 < etime < 18 : 00	$\{C_1, \cdots, C_m\}$	Editing	Permit
Pol_5	srole ⩾ PDLE	obsl = TBR	9 : 30 < etime < 17 : 30	$\{C_1, \cdots, C_m\}$	Approving	Deny
Pol_6	srole ⩾ PDLE	obsl⩽STBR	7 : 00 < etime < 19 : 30	$\{C_1, \cdots, C_m\}$	Deleting	Permit

Table 7: The final evaluation results.

SAR	Pol_1	Pol_2	Pol_3	Pol_4	Pol_5	Pol_6	ERs	FERs
SAR_1	T^+	F	F	F	F	F	$\{T^+\}$	{permit}
SAR_2	F	F	F	F	F	F	$\{F\}$	{deny}
SAR_3	F	F	T^-	F	F	F	$\{T^-\}$	{deny}
SAR_4	F	F	F	F	T^-	F	$\{T^-\}$	{deny}
SAR_5	F	F	T^-	F	F	T^+	$\{T^-, T^+\}$	{deny, permit}
SAR_6	F	F	F	F	F	F	$\{F\}$	{deny}

Table 8: The Pol_1.

POL	S	O	E	C	A	P
Pol_1	srole ⩾ ECE, st = CQ	obsl ⩾ PBR	8 : 30 < etime < 17 : 00	$\{C_1, \cdots, C_m\}$	Browsing	Permit

means that the requester is affiliated with the automobile enterprise alliance.

The above cases fully demonstrate that ASACPM and its application framework can solve the access control problem of CSP through the access decision mechanism and dynamic fine-grained authorization mechanism. The application of these two mechanisms enables CSP not only to have good flexibility, scalability, and practicability but also to adapt to future development and applications.

5.2. Correctness Analysis. The ASACPM and its application framework proposed in this paper are through CA and SSL certificates to restrict attackers from invading CSP or CP to steal or tamper with tenant's business resources. So the

Table 9: The final evaluation results of the Pol_1.

SAR	Pol_1	ERs	FERs
SAR_1	F	$\{F\}$	{deny}
SAR_2	F	$\{F\}$	{deny}
SAR_3	F	$\{F\}$	{deny}
SAR_4	F	$\{F\}$	{deny}
SAR_5	F	$\{F\}$	{deny}
SAR_6	F	$\{F\}$	{deny}

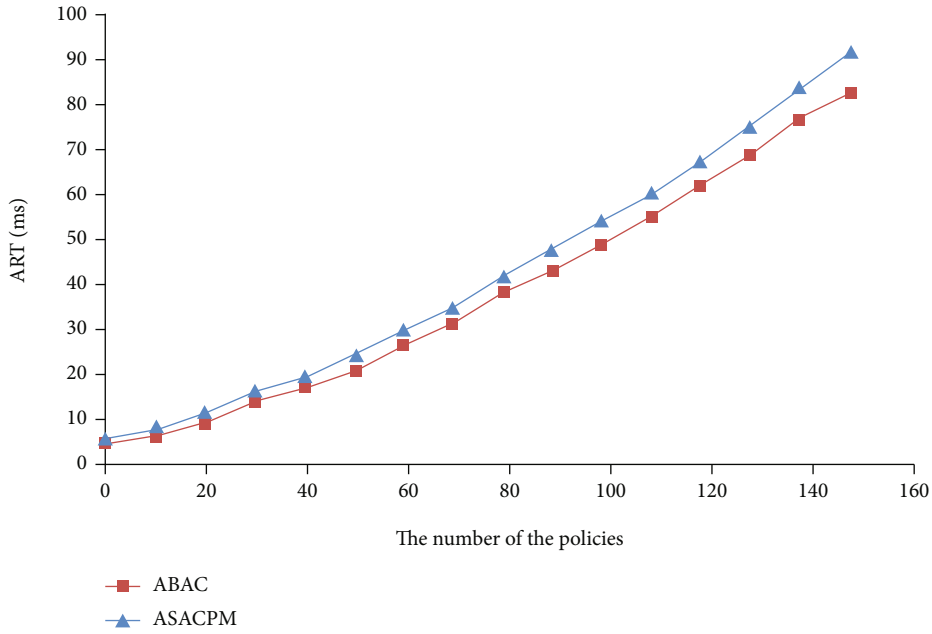

Figure 5: Comparison of the execution overhead of ABAC and ASACPM.

ASACPM and its application framework ensure that the confidentiality and integrity of business resources will not be destroyed by attackers. In order to prove this conclusion is correct, we use the reductio ad absurdum (RAA) to prove the safety performance of the model.

We assume that the attacker obtains the correct tenant certificate and obtains access permission of the CSP by repeatedly sending the virtual attribute set in a polynomial time. Attacking this target requires the following three parameters $<x, y, z_i>$, where x represents the number of attributes required to access the business resources of the CSP or CP, y represents the number of attribute conditions of all policy sets, $y \gg x$, and z_i represents the number of values for each attribute. The attempted time required for the attacker to find the correct combination is $C_y^x = A_y^x/x! = y!/(x!(y-x)!)$. Therefore, the computational complexity of finding out the correct access permissions is $T(x, y, z_i) = C_y^x \times \prod_{i=1}^{i} z_i = y!/(x!(y-x)!) \times \prod_{i=1}^{i} z_i$. This attempt cannot be accomplished within a polynomial time, which has conflict with the assumed condition, so the correctness of the above conclusion is proved.

5.3. Model Evaluation. The ASACPM and its application framework are applied to the ASP-/SaaS-based manufacturing industry value chain collaboration platform, and a thousand policies are written for evaluating the performance of ASACPM and its application framework. Development tools include XACML, JDK1.7, Eclipse4.5, and Mysql5.6. LAN is taken as the test environment, the computer configuration at client side is Intel Core i7-4790 3.6 G, 8 G memory, and Windows 10 enterprise operating system, and the computer configuration at server side is Intel Core i7-4790 3.6 G, 16 G memory, and Windows server 2012 operating system.

In the execution overhead experiment of the ABAC and ASACPM, the NAR of 16 different tenants is arbitrarily selected, the NAR is sent to the decision-making body first, and then, the different policy sets (the number of policies in each policy set is different and is within 0-160) are evaluated based on the NAR, and next, the result is returned to the tenant, and the time spent in the entire process is called the response time. The same NAR was repeatedly performed 20 times, and the corresponding response time was recorded and then averaged to form an average response time (ART). Under different access control schemes (ABAC and ASACPM), the relationship between the number of the policies in the policy set and the ART is shown in Figure 5.

It can be seen from Figure 5 that the difference in execution overhead between ABAC and ASACPM is relatively small, and as the number of policies increases, their ART is also growing. In the case of the same number of policies, the ABAC's ART is slightly slower than the ASACPM's ART, but their gap is within 10 ms. The ASACPM proposed in this paper needs to establish an SSL secure channel, so its implementation overhead is relatively long, but its security is better than the ABAC (i.e., the data plaintext transmission). In combination with actual needs, the number of policies and the amount of data transmission are usually small, and their ART is relatively short, so the execution overhead of the ASACPM has little effect on the overall performance of the CSP.

In the experiment on the access decision, we divide the experimental scenario according to the type of the access decision and then evaluate different policy set (the number of policies in each policy set is different and is within 0-1000) based on different NAR. Repeat the execution of the same NAR 20 times, record the corresponding decision-making time, and then take the average to form

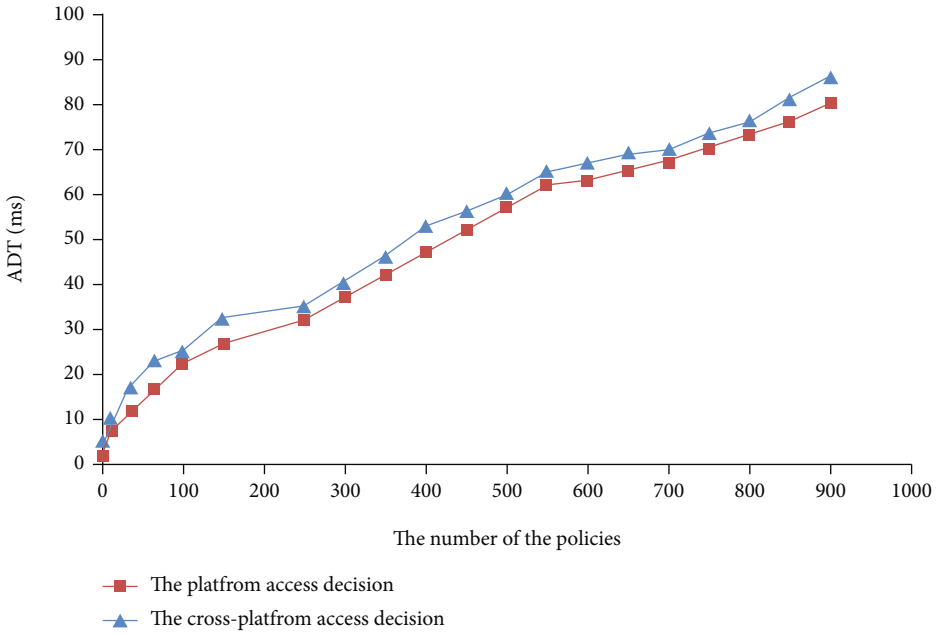

Figure 6: Comparisons of relationship between the number of the policies and ADT.

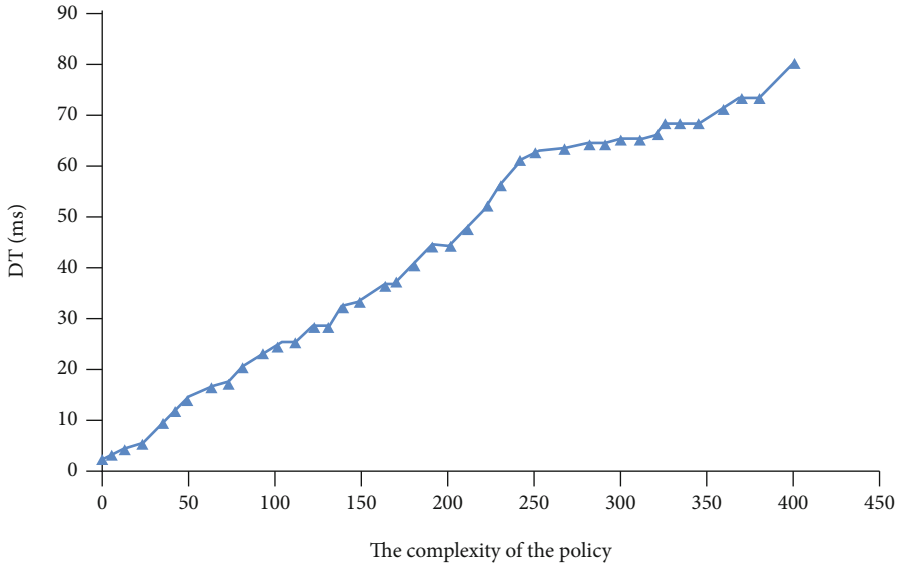

Figure 7: The relationship between the complexity of the policy and the DT.

the average decision-making time (ADT). In the scenario where the access decision is different types, the relationship between the number of the policies in the policy set and the ADT is as shown in Figure 6.

It can be seen from Figure 6 that the ADT of the platform access decision and cross-platform access decision is slowly increasing with the increase of the number of policies. The ADT of the cross-platform access decision is relatively long. In the case where the number of policies is the same, the ADT gap between it and the platform access decision is within 7 ms. In combination with actual needs, when evaluating policies based on the SAR, since the number of policies is usually less than 200 and the ADT is relatively short, the impact of the access decision on the overall performance of the CSP is relatively small.

In the experiment on the dynamic fine-grained authorization, the complexity of a policy is related to the number of the attribute conditions it owns. The more the attribute conditions of the policy, the higher its complexity and the finer the granularity of access control. In this experiment, we first give the SAR of the tenant, then arbitrarily select 40 policies, next evaluate the policies based on the SAR, and finally record their decision time (DT). The relationship between the complexity of the policy and the DT is shown in Figure 7.

It can be seen from Figure 7 that as the complexity of the policy increases (i.e., the granularity of the access control is also finer), the corresponding DT is also gradually increasing, and we can conclude that the relationship between them is proportional. In combination with actual requirements, the complexity of the policy is usually within 200 attribute conditions, and the corresponding DT has a small gap. Therefore, the dynamic fine-grained authorization has less impact on the overall performance of the CSP.

In summary, in the Internet environment, the above experimental results show that the performance of the ASACPM and its application framework proposed in this paper meets our expectations and is generally satisfactory.

6. Conclusions

In the Internet environment, traditional cloud service platforms have cross-platform access control problems and do not support dynamic fine-grained authorization methods. This paper proposes a tenant-centric attribute semantic access control policy model for cloud service platforms. First, this paper formally describes ASACPM through attribute semantics and evaluates whether the tenant's access request conforms to the requirements of the policy set through ASACPM. Secondly, this paper presents the application framework of ASACPM, in which the access decision mechanism is used to evaluate whether the tenant has CSP or CP access control permissions, the attribute synchronization mechanism is used to ensure the consistency of the AA of the CSP and the AA of the CP, and the dynamic fine-grained authorization mechanism ensures better flexibility and scalability of CSP. Finally, through a practical case analysis, this paper proves that the application of ASACPM and its application framework to CSP has good flexibility, scalability, and practicability. In addition, we design some experimental scenarios to verify that the performance of ASACPM and its application framework meets our expectations and has good reliability, validity, and rationality.

Since the application framework of ASACPM adopts CA and SSL certificate to ensure the security, integrity, and confidentiality of data, its execution overhead is relatively high in the application process. Therefore, the main work in the future is to further optimize the web server and shorten the construction time of the SSL secure channel. In addition, the visual interface and policy customization format are optimized to improve the experience of tenants.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Plan of China (Nos. 2018YFB1701500 and 2018YFB1701502).

References

[1] Z. C. Zhou and L. Zhao, "Cloud computing model for big data processing and performance optimization of multimedia communication," *Computer Communications*, vol. 160, pp. 326–332, 2020.

[2] S. N. Mthunzi, E. Benkhelifa, T. Bosakowski, C. G. Guegan, and M. Barhamgi, "Cloud computing security taxonomy: from an atomistic to a holistic view," *Future Generation Computer Systems*, vol. 107, pp. 620–644, 2020.

[3] L. Wang, D. Chen, Y. Hu, Y. Ma, and J. Wang, "Towards enabling cyberinfrastructure as a service in clouds," *Computers & Electrical Engineering*, vol. 39, no. 1, pp. 3–14, 2013.

[4] M. Z. Wang and Q. L. Zhang, "Optimized data storage algorithm of IoT based on cloud computing in distributed system," *Computer Communications*, vol. 157, pp. 124–131, 2020.

[5] Y. Yu, L. Sun, and Y. Ma, "Access control model for attribute-based cloud manufacturing collaboration platform," *Computer Integrated Manufacturing Systems*, vol. 23, no. 1, pp. 196–202, 2017.

[6] Y. Yu, L. F. Sun, C. H. Ren, and M. Han, "Bilateral matching model of business resources for multi-service value chain," *Computer Integrated Manufacturing Systems*, vol. 27, no. 5, pp. 1397–1409, 2021.

[7] Y. Yu, L. Sun, Y. Ma, and S. Wang, "Data security model for industrial chain collaborative SaaS platform," *Computer Integrated Manufacturing Systems*, vol. 22, no. 12, pp. 2911–2919, 2016.

[8] M. Hasan and B. Starly, "Decentralized cloud manufacturing-as-a-service (CMaaS) platform architecture with configurable digital assets," *Journal of Manufacturing Systems*, vol. 56, pp. 157–174, 2020.

[9] X. Li, G. Feng, and C. Chen, "The access control model based on attribute," *Journal of Communications*, vol. 29, no. 4, pp. 90–98, 2008.

[10] Y. Yu, L. Sun, and Y. Ma, "Multi-tenant form customization technology for collaborative cloud service platform in industrial chain," *Computer Integrated Manufacturing Systems*, vol. 22, no. 9, pp. 2235–2244, 2016.

[11] K. Ma, G. Yang, and Y. Xiang, "RCBAC: a risk-aware content-based access control model for large-scale text data," *Journal of Network and Computer Applications*, vol. 167, article 102733, 2020.

[12] Y. D. Wang, J. H. Yang, C. Xu, X. Ling, and Y. Yang, "Survey on access control technologies for cloud computing," *Journal of Software*, vol. 26, no. 5, pp. 1129–1150, 2015.

[13] F. Nazerian, H. Motameni, and H. Nematzadeh, "Emergency role-based access control (E-RBAC) and analysis of model specifications with alloy," *Journal of Information Security and Applications*, vol. 45, pp. 131–142, 2019.

[14] Z. Tan, Z. Tang, and R. Li, "Research on trust-based access control model in cloud computing," in *IEEE Joint International Information Technology and Artificial Intelligence Conference*, pp. 339–344, Washington, DC, USA, 2011.

[15] M. Zhao and Z. Yao, "Cloud computing access control model based on RBAC," *Journal of Computer Applications*, vol. 32, no. S2, pp. 267–270, 2012.

[16] Y. Jung and M. Chung, "Adaptive security management model in the cloud computing environment," in *International Conference on Advanced Communication Technology*, pp. 1664–1669, Washington, DC, USA, 2010.

[17] G. Lin, S. He, and H. Huang, "The cloud computing access control model based on behavior security model," *Journal of Communication*, vol. 33, no. 3, pp. 59–66, 2012.

[18] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: role-based access control using smart contract," *IEEE Access*, vol. 6, pp. 12240–12251, 2018.

[19] Y. K. Fan, S. L. Liu, G. Tan, and F. Qiao, "Fine-grained access control based on trusted execution environment," *Future Generation Computer Systems*, vol. 109, pp. 551–561, 2020.

[20] X. Cheng, X. Chen, B. Zhang, and Y. Yang, "Attribute-based access control policy model," *Computer Engineering*, vol. 36, no. 15, pp. 130–133, 2010.

[21] S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized access control with anonymous authentication of data stored in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 384–394, 2014.

[22] K. Fan, H. Y. Xu, L. X. Gao, H. Li, and Y. Yang, "Efficient and privacy preserving access control scheme for fog-enabled IoT," *Future Generation Computer Systems*, vol. 99, pp. 134–142, 2019.

[23] S. Pal, M. Hitchens, V. Varadharajan, and T. Rabehaja, "Policy-based access control for constrained healthcare resources in the context of the Internet of Things," *Journal of Network and Computer Applications*, vol. 139, pp. 57–74, 2019.

[24] Southwest Jiaotong University, "ASP/SaaS-based manufacturing industry value chain collaboration platform," 2022, http://www.autosaas.cn/.

[25] A. Liu, C. Fei, J. Hwang, and X. Tao, "Designing fast and scalable XACML policy evaluation engines," *IEEE Transactions on Computers*, vol. 60, no. 12, pp. 1802–1817, 2011.

[26] A. Ren, X. Wang, X. Luo, and L. Ruan, *Operating System Practical Tutorial*, Tsinghua University Press, 3th edition, 2012.

[27] A. Silberschatz, G. Gagne, and P. Galvin, *Operating System Concepts*, John Wiley & Sons, Inc., 8th edition, 2008.