Advances in Engineering Software 69 (2014) 26–36

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Cloud service access control system based on ontologies

Chi-Lun Liu *

Department of Multimedia and Mobile Commerce, Kainan University, No. 1, Kainan Road, Luzhu Shiang, Taoyuan County, Taiwan

ARTICLE INFO

Article history:
Received 10 June 2013
Received in revised form 18 November 2013
Accepted 15 December 2013
Available online 17 January 2014

Keywords:
Cloud service
Access control
Ontology
Conflict detection
Payment status
Service level

ABSTRACT

Cloud service is a new and distinctive business model for service providers. Access control is an emerging and challenging issue in supporting cloud service business. This work proposes a new access control mechanism called cloud service access control (CSAC). The CSAC mechanism considers payment status and service level as the two essential characteristics of cloud service. Ontology is a theoretical foundation for the CSAC mechanism. Inconsistent access control policies are detected by a set of proposed policy conflict analysis rules. Inappropriate user accesses are inhibited by access control policies according the proposed access denying rules. System architecture is designed to support the CSAC mechanism. A case study is provided to demonstrate how CSAC works. Finally, an evaluation is conducted to measure the concept explosion issue in CSAC.

1. Introduction

Cloud computing is a model that servers can dynamically provisioned and configured to deliver services in a scalable and low-management-effort manner [12,1]. Many users are increasing realizing that they can use cloud services to gain applications and resources rapidly, flexibly, and cheaply [16]. In cloud, service providers usually do not know their users in advance. It is difficult to assign users to roles in access control policies [17]. Therefore proposing a new access control mechanism for the new context of cloud computing is an important theoretical and practical issue.

Cloud can be classified in fourfold: private, public, community, and hybrid clouds. In *private cloud*, an organization provides its own cloud services for its internal users [10,16]. *Public cloud* means that a cloud services provider offers access for external and public users who are typically billed on a pay-as-you-use basis [10]. *Hybrid cloud* is a mixture of public and private clouds. Finally, *community cloud* is provisioned for exclusive use by a specific community constituted by a group of organizations that have shared concerns [11].

Cloud service business usually needs to consider two elements: payment status and Service Level Agreement. *Payment status* can be defined as a stage of payment process in cloud providers' accounting records. In practice, cloud computing access control models usually considers service accounting records of cloud computing providers [17]. Cloud computing often utilizes the pay-per-use strategy to charge service fees. Pay-per-use means cloud resource charges are based on the quantity used [3]. The above discussion indicates that payment status for service fees is important for cloud access control. *Service Level Agreement* (SLA for short) is a part of service contract that defines the level of service and controls the use of computing resources [17]. In general, cloud computing service providers define SLA as guarantee about expected service level that the enterprise has committed to provide [3]. And SLA should be considered in cloud computing access control models [17].

RBAC (Role-Based Access Control) is a widespread used mechanism in business. The RBAC model uses application-specific and organizational role structure to control objects' accesses in organizational-wide applications [4]. Considering the attributes, other than role and object, that affect users' permission causes *concept explosion* and limits RBAC's strengths. The concept explosion is twofold: *role explosion* and *object explosion*. *Role explosion* means that the number of roles can easily explode [4]. Role explosion occurs when users' and context-specific attributes, such as payment status, affects users' permissions. The idea of *object explosion* defined in this work means the number of resources can easily explode. Object explosion occurs when resources' attributes, such as service level, are considered in access control mechanisms.

Conflicts are inevitable in access control policies. Here is a conflict example about multiple roles for the same person. In this example, policy A is: role X in company Y **can not use** resource Z in Cloud C. And policy B is: role I in community J **can use** resource Z in Cloud C. Therefore policy A contradicts policy B when person W plays both role X and role I.

To handle the cloud service business requirements and reduce concept explosion in cloud service, this work is based on and extend RBAC to proposes the cloud service access control (CSAC) system to automatically support for access control policy analysis

* Tel.: + 886 3 341 2500x6070; fax: +886 3 341 2373.
E-mail address: tonyliu@mail.knu.edu.tw

0965-9978/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.advengsoft.2013.12.006

and execution in cloud services. The CSAC system is designed to support the CASC method. The CSAC method is comprised of a process, policy and requirement metadata, and rules. The process describes the steps of CSAC method. The policy and requirement metadata reveals what information should be considered in access control policy and user requirement specification. The rules are used to analyze policy conflict and deny inappropriate accesses.

Ontology is the theoretical foundation of the CSAC system. Ontology is a shared and explicit conceptualization to represent a domain knowledge structure [5]. The ontology is an appropriate methodology to accomplish a common consensus of communication and to facilitate knowledge management [2]. In the CSAC system, ontology can provide necessary semantic information to execute policy conflict analysis and access denying rules.

The CSAC system is comprised of seven modules: access grant and denying, resource usage monitor, payment status monitor, policy specification, policy conflict detector, conflict resolver, and ontology management. *Access grant and denying* module executes access control policies and access denying rules to decide which access is allowed or not. *Resource usage monitor* module measures how many resources are used, such as used storage size. *Payment status monitor* indicates that each user's accounting record about paying bills. The cloud administrator uses *policy specification* module to specify access control policies. *Policy conflict detector* module uses policy conflict rules and ontologies to automatically analyze inconsistencies between policies. The cloud administrator uses *conflict resolver* module to prioritize inconsistent policies to resolve conflicts. Finally, *ontology management* module is used to add, modify, and delete concepts and relationships.

This paper is organized as follows. Section 2 discusses related works about access control mechanisms. Section 3 proposes the CSAC method which is constituted of the process, metadata, and rules. Section 4 proposes the CSAC system architecture and database design. Section 5 provides a case study to demonstrate how the CSAC method works. Section 6 evaluates the concept explosion situation in CSAC and RBAC. The final conclusion section includes research contributions and further works.

2. Literature review

Several related works in the literature are summarized in Table 1. Le et al. [7] propose an access control method in ubiquitous hospital information system. It authorizes access permission according to user activities. Masoumzadeh and Joshi [9] use ontologies and policy rules to control access in online social networks. Hu et al. [6] use semantic web technology to control cloud computing access. Zhu et al. [18] extend Role-Based Access Control (RBAC) to the cloud computing context. Ruj et al. [15] provides an attribute-based access control mechanism in data storage cloud service.

The most above access control methods focus on cloud computing. In these works, few methods discuss the relationships between the payment issue and access control mechanism. Few works consider service level issue as an important element in the access control methods. In the other hand, ontology is used in the two methods in Table 1. Only an existing method in Table 1 discusses how to use ontology to analyze policy conflicts. Therefore this work proposes an ontological access control method which concerns payment status and service level agreement.

3. CSAC method

The main contribution of this work is to design and evaluate the CSAC system. Before we introduce the CSAC system design, the rationale behind the CSAC system is the CSAC method. The CASC method is an extended version of the prior work [8]. The CASC method is introduced in this section.

3.1. CSAC process

Four actors and six steps in Fig. 1 depicts how to use the proposed access control method. The four actors are the end user, administrator, access control server, and cloud. The six-stages process in the CSAC method is introduced as follows. In the first step, the end user requests the access control server to get a specific cloud computing service. Then the access control server uses policies to check the end user's authority for this cloud service in step 2. The policies are managed by the administrator in step 0. If the end user has the authority, the access control server requests and acquire necessary resource in the cloud in step 3 and 4. If the end user has no authority, step 3 and step 4 are skipped. And the access control server records service logs in step 5. The final step is to deliver the service result to the end user.

3.2. CSAC metadata

The core elements in the proposed metadata are fivefold: payment status, role, access permission, service level, and cloud. The metadata (in Fig. 2) comprises two cloud access control policies and one user requirement. The six relationships ($R_{PS1-PS2}$, R_{U1-U2}, R_{R1-R2}, $R_{AP1-AP2}$, $R_{SL1-SL2}$, and R_{C1-C2}) are between access control policy ACP_1 and ACP_2. These relationships are used by the proposed policy conflict analysis rules (summarized in Table 2) to analyze conflicts between two policies. And the other five relationships ($R_{PS1-PS3}$, R_{R1-R3}, $R_{AP1-AA3}$, $R_{SL1-RSL3}$, and R_{C1-C3}) are between access control policy ACP_1 and user requirement UR_3. These five relationships are utilized by the proposed access denying rules (summarized in Table 3) to deny and allow access.

3.3. Policy conflict analysis and access denying rules

This work proposes four policy analysis rules summarized in Table 2 for detecting conflicts. These rules are developed based on the proposed metadata in Fig. 2. These rules are introduced as follows.

$Rule_{PCA1}$ for policy opposition conflict:

IF an equality or a kind relationship exists between payment status PS_1 and PS_2, an equality or a kind relationship exists between role R_1 and R_2, an antonym relationship is between access permission AP_1 and AP_2, an equality or a kind relationship exists between service level SL_1 and SL_2, and an equality, a kind, or a composition relationship is between cloud C_1 and C_2, THEN a policy opposition conflict occurs between access control policy ACP_1 and ACP_2.

$Rule_{PCA2a}$ for policy exclusion conflict:

IF there is no equality and no kind relationship exists between role R_1 and R_2, a exclusion relationship is between access permission AP_1 and AP_2, an equality or a kind relationship exists between service level SL_1 and SL_2, and an equality, a kind, or a composition relationship is between cloud C_1 and C_2, THEN a policy exclusion conflict occurs between access control policy ACP_1 and ACP_2.

$Rule_{PCA2b}$ for policy exclusion conflict:

IF there is no equality and no kind relationship exists between payment status PS_1 and PS_2, a exclusion relationship is between access permission AP_1 and AP_2, an equality or a kind relationship exists between service level SL_1 and SL_2, and an equality, a kind, or a composition relationship is between cloud C_1 and C_2, THEN