Department of Computer Science
University of Auckland
New Zealand

A new method of handling conflicts of interest in RBAC

Eunjung (Helena) Joo

June 2014

Supervisor: Professor Clark Thomborson

A dissertation submitted in fulfilment of the requirements of Bachelor of Science(Honours) in Computer Science

Abstract

Role-based access control (RBAC) has gained popularity due to its simple and cost-effective administration, that is, by assigning permission to user through the role. However, RBAC has shown a limitation, in that it cannot recognize a user's conflicts of interest due to its lack of flexibility in addressing various attributes of a user. For example, conflicts of interest may arise in a bank; that is, a loan manager, who holds a mortgage account at the bank where she works, wants to change her own 'client-grade' from silver to premium in order to lower her mortgage account interest rate. Since RBAC considers only the manager's role, this change will be accepted. In my analysis, conflicts of interest arises because the loan manager has two personas: they are both an employee, and a client, of the same bank. This dissertation discusses the use of 'persona' as an attribute in an attribute-based access control (ABAC) variant of RBAC, so that it can recognise and mitigate conflicts of interest.

Acknowledgements

I would like to express the deepest appreciation to my supervisor Clark Thomborson, who has continuously led me to broaden research area. Without his guidance and encouragement, this dissertation would not have been finished.

I would like to thank my husband for his love and support, and to my family.

I thank PhD. Yu-cheung for encouraging me throughput my research. Also, I thank persona research group members, Jason and Eva, for sharing ideas with me.

Finally, thank God for listening to my prayer all of the time.

Contents

1

Problem: insufficient handling of conflicts of interest in RBAC

1.1 Introduction

Conflicts of interest are considered a serious risk in organisations, since if it is not handled enough, this can lead to insider threat. I will begin with the example of a loan manger, Lena's case, who has conflicts of interest as an employee and a client in a bank she works at. To mitigate this risk, this dissertation is seeking a solution to limit access to sensitive data from those who have conflicts of interest.

The most relevant security mechanism might be access control. As role-based access control (RBAC) is currently the dominant access control model among the organisations, this dissertation will begin with discussing RBAC. Since RBAC has shown limitations to address various attributes of a user, it can not handle conflicts of interest. Therefore, I shift and broaden my research to Attribute-based access control (ABAC) in chapter 2. In addition, I will propose to add persona to ABAC to recognise and mitigate conflicts of interest. In chapter 3, a proposed solution will be evaluated if it can handle conflicts of interest and where it can be applied to. In chapter 4, challenges and the main contribution of this dissertation will be discussed.

1.2 Motivating examples

To understand the problem I will begin with two examples. Conflicts of interest may arise in a bank if an employee has various motives. Or it may occur in a law firm if a counsellor attempts to be involved in both conflicting cases and takes advantage of information gained from one another. Example 1 will be mainly discussed throughout this dissertation.

Example 1 Lena is a loan manager in a bank who wants to change her own 'client_grade' from silver to premium in order to lower her mortgage account interest rate. Here are some assumptions.

- Suppose a role hierarchy in a bank as shown in figure 1.1.
- A higher role inherits a set of permission of a lower role.
- Suppose that the name of the client table is 'client_info'
- Suppose that 'client_info' table has 'client_grade' field (i,e., bronze, silver, gold and premium).
- According to the 'client_grade', mortgage interest rate changes.
- A teller role has permission to assess the entitlement to update 'client_grade' field and ask for manager approval when it needs to be updated.
- A manager role has permission to approve updating 'client_grade'.

This shows that Lena is acting as both an employee and a client in the same bank.

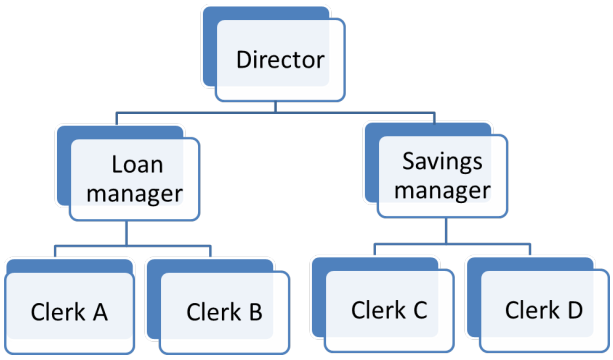

Figure 1.1: A role hierarchy diagram

Example 2 In a law firm, a counsellor, Carl wants to take another law case from company A which is related to the case from company B that he is currently working on. Since he can access sensitive information from company B, he can make a new case for company A successfully. Here, he has a dual role relationship and is taking advantage of information gained from the case he is currently working on.

1.3 Basic terminology

The most relevant security mechanism to mitigate the risk shown in motivating examples might be access control. Below are the key information security terms defined by NIST [20]. In this dissertation, a subject refers to an individual user in an organisation who can access and change the object. Object will be mostly used as database tables, records or fields.

Subject An active entity (generally an individual, process, or device) that causes information to flow among objects or changes the system state.

Object An entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are records, fields (in a database record), blocks, pages, segments, files, directories, directory trees, process, and programs, as well as processors, video displays, keyboards, clocks, printers, and network nodes. Devices such as electrical switches, disc drives, relays, and mechanical components connected to a computer system may also be included in the category of objects.

Access control is well defined by Bertino [8] as below:

> Access control determines which subjects can access which data under which circumstances and thus allows one to make sure that users can only access data according to their job responsibility.

Figure 1.2 shows a generic access control mechanism which typically includes a reference monitor. A reference monitor checks the requested access by subject to protected objects according to the access control policies [8].

Access control model can be divided as Discretionary Access Control (DAC) and Non-DAC which does not allow users to change access policies. Only administrators have authority to establish polices [17]. Mandatory Access Control (MAC) and Role-based Acess Control (RBAC) are good examples of Non-DAC. MAC is well known for its use

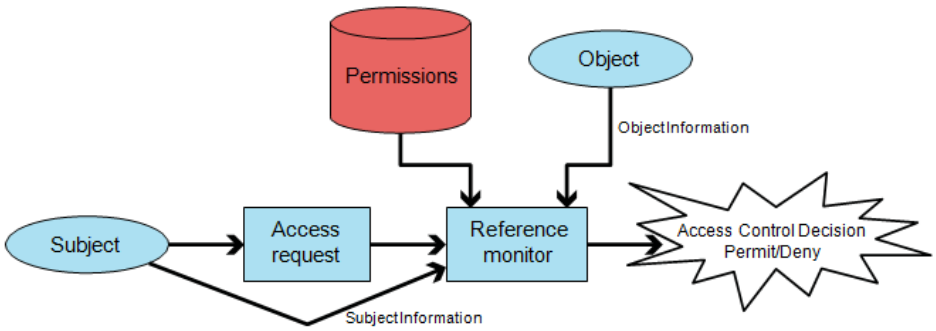

Figure 1.2: A generic access control mechanism architecture [8]

in military security, where an individual user cannot decide the classification of an object from Top Secret to Secret.

Figure 1.3 shows that while MAC and DAC were dominant models in the past, RBAC has been the most dominant model. Attribute-based Access Control (ABAC) is drawing attention and getting popular to replace RBAC. Gartner [15] predicts that ABAC will replace RBAC up to 70% to protect critical assets by 2020. Currently, it is reported that ABAC makes up less than 5% of the industry [15]. The scope of this dissertation is focused on those two modern access control models.

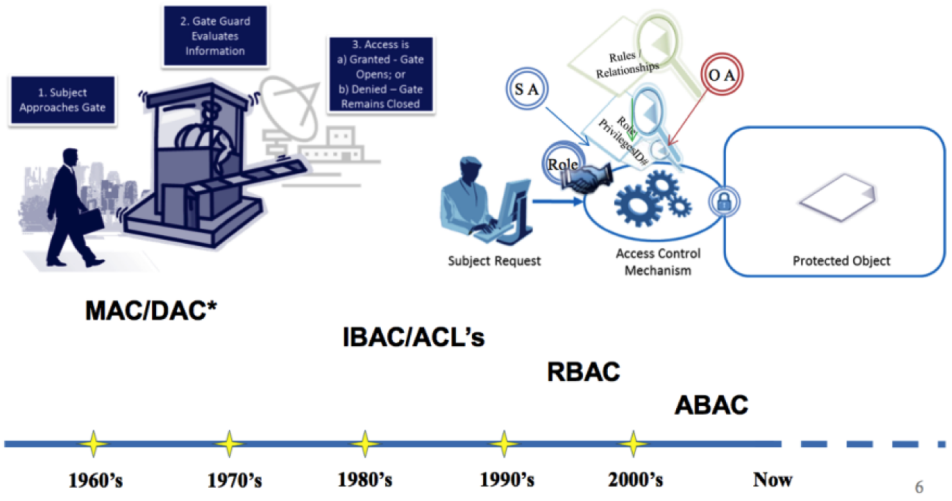

Figure 1.3: Access control models [1]

1.4 Role-based access control (RBAC)

RBAC is currently the dominant access control model due to its cost-effective and simple administration [8]. As the name implies, access decisions are based on the roles which users have in organisation such as bank manger, bank teller, doctor or counsellor. According to the roles, access permissions are assigned to restrict the use of objects. For example, within a bank, a teller role can include permission to receive the enquiry from clients to change their 'client_grade'. A bank manager, however, has access right to authorise this enquiry from a teller.

The main benefit of RBAC is illustrated in figure 1.4. Instead of assigning a massive set of permissions directly to the users, RBAC assigns permission to the roles. While users change frequently, roles are stable. Also, role hierarchy makes a higher role to inherit the number of permissions [14]. Therefore, granting permission tasks is simple and cost-effective [8].

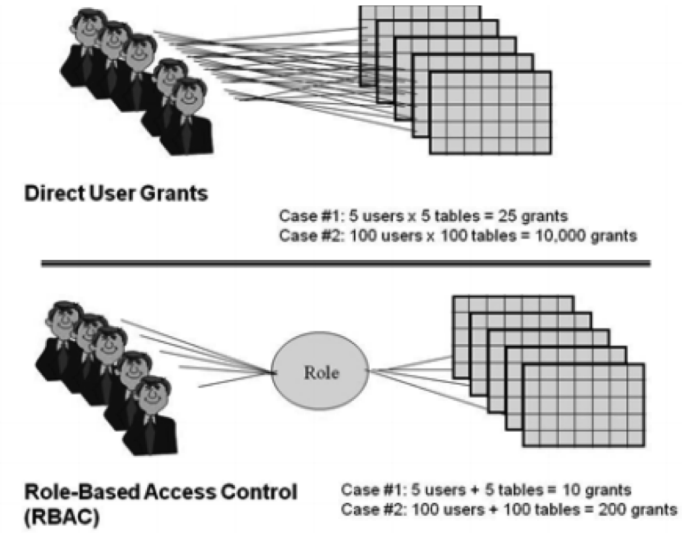

Figure 1.4: RBAC example [12]

A number of RBAC models have been studied [13, 14, 25]. This dissertation will discuss RBAC based on a NIST Standard RBAC model [26] which is organised with four levels as flat RBAC, hierarchical RBAC, constrained RBAC and symmetric RBAC. Since symmetric RBAC is not relevant to this dissertation, it will not be discussed in this section.

- Flat RBAC: This basic model describes the essential concept of RBAC (see figure 1.2). Users are assigned to roles and permissions are assigned to roles.
- Hierarchical RBAC: This model contributes to RBAC's efficiency by reducing administrative costs. The higher role can inherit the permissions of lower roles, without

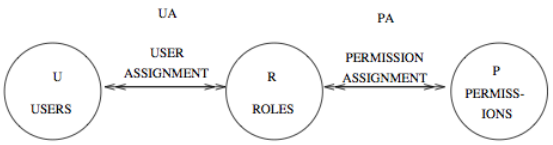

Figure 1.5: Flat RBAC model [26]

having to be assigned those permissions [26].

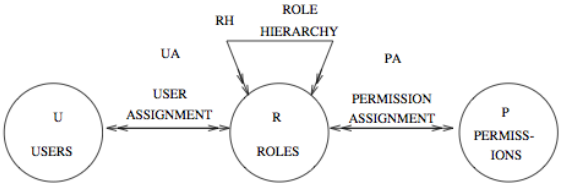

Figure 1.6: Hierachical RBAC model [26]

- Constrained RBAC: This model adds a requirement for enforcing separation of duties .

  According to [26], separation of duties can prevent fraud and major errors by giving a user a "reasonable level of authority" for their job functions. For example, two tasks of ordering products and approving product orders are commonly separated in organization with a separation of duties principle. Figure 1.7 illustrates a static separation of duties which do not assign conflicting roles to a single user. However, a dynamic separation of duties (see figure 1.8) allows a single user to assign conflicting roles, but those roles can not be activated in the same session simultaneously.

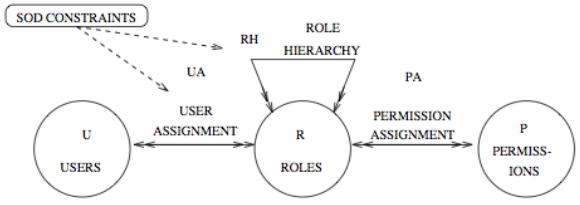

Figure 1.7: Constrained RBAC model (static SOD) [26]

Separation of duties will be reviewed later in section 1.6 as a way to handle conflicts of interest in RBAC.

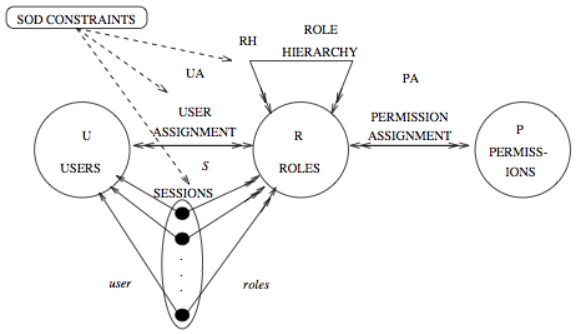

Figure 1.8: Constrained RBAC model (dynamic SOD)[26]

1.5 Conflicts of interest

Conflicts of interest have been studied in many fields such as law, the government and financial sector. The definition of conflicts of interest is well defined by Davis [11] as below:

> "A conflict of interest occurs when a personal or institutional interest interferes with the ability of an individual or institution to act in the interest of another party, when the individual or institution has an ethical or legal obligation to act in that other party's interest." [11]

Davis[11] dealt with conflicts of interest in several professions; laws, government, the market, business and heath care sector. For example, a lawyer who has two clients with opposed interests faces a conflict of interest. In an actual case, the prosecutor Marcia Clark negotiated a $4.2 million book deal for writing about the O.J. Simpson case.

> "Relationships with family, friends, or other clients, simple greed, even the laudable ideal of reforming the law to better serve society, all may tempt the lawyer to dissever a client." [11]

This implies that when even distributing the law cases to lawyers, checking suitability regarding potential conflicts of interest is crucial. Relationships, goals, motivations and situations the user faces can affect job performance and if not handled carefully, in an extreme case, this can lead to fraud. This idea is deeply associated with the necessity to build contextual information about the user which we will call "Persona" in section three.

Dual role relationship is another example of conflicts of interest. Pope [24] first pointed the potential conflict of interest and loss of objectivity between professionals and students or supervisors. American Association for Counselling and Development Code of

Ethics(1986) reflects the concerns that dual relationship with clients who are close friends or relatives, or sexually intimate may impair the counsellor's objectivity and professional judgement. This code strongly emphasizes that the dual role relationship should be avoided. In the next section, I will demonstrate motivating examples of conflicts of interest.

1.6 How RBAC handles conflicts of interest

This section will apply the example of Lena's case to RBAC model, and evaluate how it handles conflicts of interest. Under RBAC, Lena's case can be drawn as figure 1.9. To change her own 'client_grade', a process is needed as following. Firstly, a client, Lena, requests a bank teller to change her 'client_grade'. Secondly, a bank teller will ask for an approval to a loan manger, who is also Lena. Finally, a loan manager Lena, who is also a client herself, will approve this request.

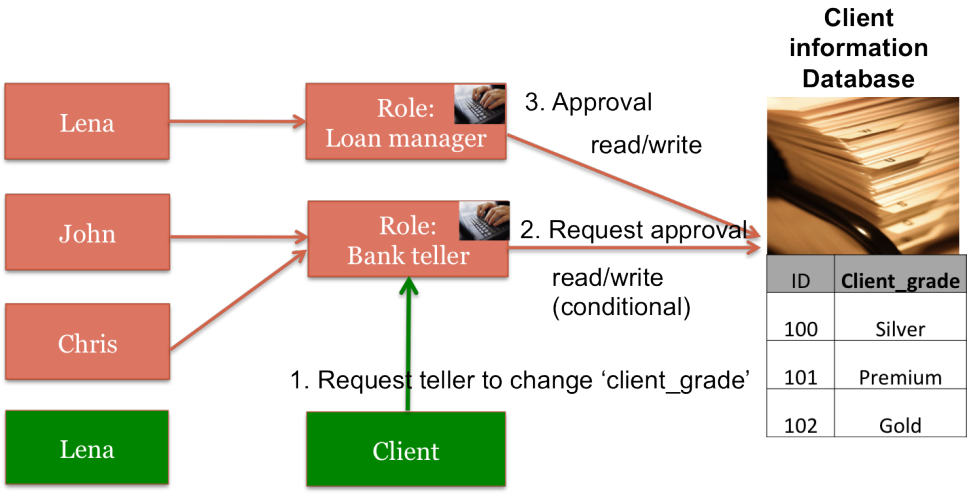

Figure 1.9: A process to change 'client_grade'

To handle conflicts of interest, RBAC provides a separation of duties solution [26]. Information Systems Audit and Control Association defines separation of duties below [7] :

> "A basic internal control that prevents or detects errors and irregularities by assigning to separate individuals the responsibility for initiating and recording transactions and for the custody of assets."

In RBAC, separation of duties is supported by the principle of least privilege. This principle avoids the problem of an individual having the ability to perform unnecessary and potentially harmful actions. There are three kinds of separation of duties.

- Static separation of duties: "If a user is authorized as a member of one role, the user is prohibited from being a member of a second role." [26].
- Dynamic separation of duties: "A user can be authorized for both roles but the roles cannot be activated simultaneously." [26].
- History-based separation of duties: "A same role cannot access the same object a certain number of times." [17].

Based on a static separation of duties solution, Lena cannot have both an employee and a client role. Also, RBAC cannot create a fine-grained rule which states that a user who has conflicting roles cannot modify 'client_grade' field. In an alternative way, fine-grained roles might be created to finely tune the access rules. However, this approach will create 'role explosion' [12, 16, 22] especially if there are many applications. As a result, RBAC will lose its main benefit of simplicity in administration.

There are some analysis [17, 19] which point out that a separation of duties might be avoided, especially in two cases.

First, even if a user is not assigned to conflicting roles, there is still possibility to gain sensitive permission. Role inheritance makes it possible to inherit lower role permission.

> "Although separation of duties is easy and efficient to accomplish with RBAC, if a single individual has access to all privileges needed to accomplish some critical function, then the system can be compromised regardless of the role structure." [17]

This implies that separation of duties can be compromised by creating a loophole in role structure [17]. For example, suppose two roles for ordering and approving are established with separation of duties within them. But if a third role is assigned to the permission to approve the orders and a user also has ordering role, then it is a violation of separation of duties.

Secondly, Jueneman [19] criticised that separation of duties may fail if there is not enough understanding about the user.

> "It should be pointed out that the requirement for separation of duties imposes a subtle requirement on the system with respect to the *global identification of users*...it does nothing to rule out the possibility that the same user may belong to a *different group*, or have an alternative network address ... a single user could appear to be two different individuals and thereby avoid the separation-of-duty exclusion. " [19]

In the example of Lena's case, it shows necessity to understand what kind of possible conflicts of interest exists within a single user. Unlike separation of duties, which is concerned between the conflicting roles, the idea of analysing possible conflicts of interest within a single user is a different approach to handling conflicts of interest.

In my analysis, Lena creates two personas: an employee and a client. And they show different motives and goals. Based on this idea, I tried to add personas to RBAC as seen in Figure 1.10. Firstly, I created a persona relation between roles and permission relations. Then, instead of assigning permission to roles, I assigned permission to personas.

However, this approach creates a complex data structure and loses the benefit of RBAC, just like 'role explosion'. See appendix 4.

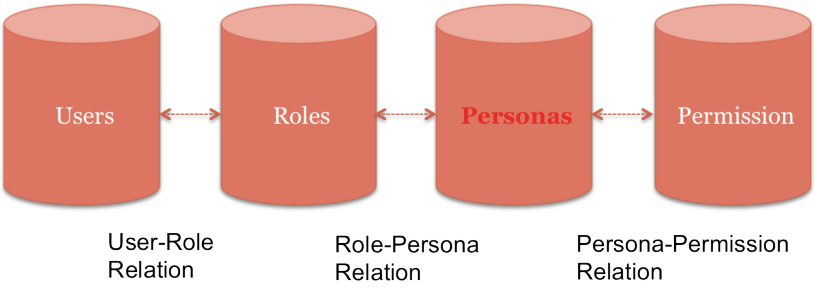

Figure 1.10: Adding persona to RBAC

1.7 Review: limitation of RBAC to handle conflicts of interest

The main limitation of RBAC is failing to take various attributes of a user into account when making access decisions. Since RBAC is considering only organisational 'roles', other attributes of a user, such as non-organisational roles or location of a user are not addressed. If RBAC considers those attributes, it will increase the number of roles and lead to 'role explosion'. In addition, RBAC is unable to create fine-grained access rule to limit access. Adding personas is not suitable due to complex data structure. I conclude that RBAC cannot handle conflicts of interest without a role explosion.

Therefore, a new approach is needed to address various attributes of a user and be able to create fine-grained access rules to limit access for those who have conflicting personas.

In the next chater, I will broaden the scope of my research by discussing Attribute-based access control (ABAC) which addresses various attributes of a user. And I will suggest adding personas to recognise conflicts of interest. As a result, I will propose fine-grained access control rule which limits access to those who have conflicting personas.

2

Solution: adding persona in ABAC rule

As we have already studied, it is necessary to address attributes of a user to handle conflicts of interest. RBAC considers only one attribute of the user, their organisational role. ABAC is a standard way to address multiple attributes. In this chapter, I will discuss how to add persona to the list of attributes in ABAC.

2.1 Attribute-based access control: as a way of addressing different attributes

To satisfy the need of taking various attributes into account when making authorization decisions, non-ABAC systems may have authorization rules which are typically written in code and static. According to Gartner's research note [15], ABAC will replace RBAC by 70% by 2020. However, ABAC has currently only 5% in the industry. While RBAC works on a 'role' to make an access decision, ABAC has a complex rule set that can evaluate many different attributes [18]. A 'role' is considered as one of the user's attributes. Here is a definition of ABAC defined by Hu [18].

> An access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject,

assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions.

In the definition above, attributes means characteristics of the subject, object, or environment conditions. Attributes contain information given by a name-value pair [18]. Environment conditions means an operational or situational context in which access requests occur. Environment conditions are detectable environmental characteristics. Environment characteristics are independent of subjects or objects, and may include the current time, day of the week, location of a user, or the current threat level [18]. Figure 2.1 [18] shows how ABAC access control mechanisms work. When a subject requests access to an object, the access control mechanism evaluates rules, attributes of the subject, object, and environment conditions to compute a decision. The subject is given access to the object if authorized. For example, a rule that states "MPEG movies for adults can be downloaded only by users with age equal or greater than 18 years" (see Appendix A), allows access to all users whose age attribute is satisfied by the condition of being equal or higher than 18 [8]. Another example that "a doctor can only access to his patient's records within a hospital during working hours", verifies attributes of content, location of a subject, and time. In RBAC, this may create many doctor roles restricting permission according to content, location and time which will result in 'role explosion' [18]. Enterprise-scale ABAC scenario is shown in appendix B.

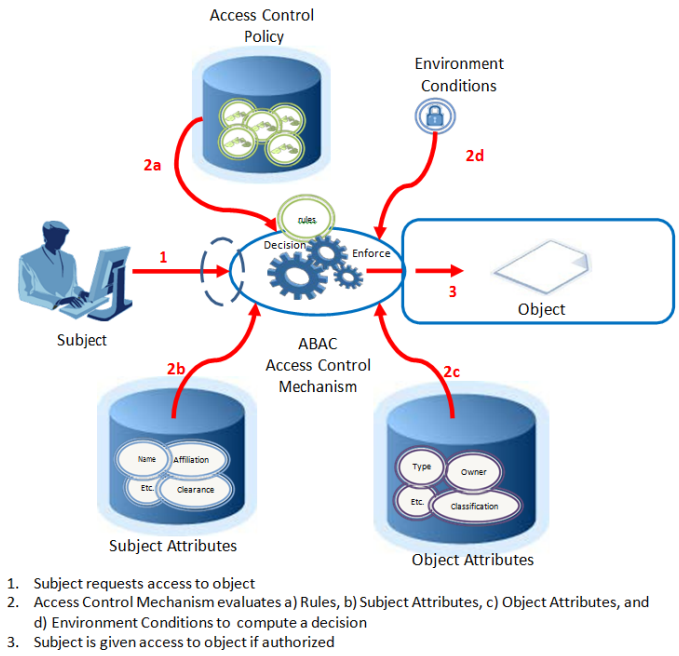

Figure 2.1: basic ABAC Scenario [18]

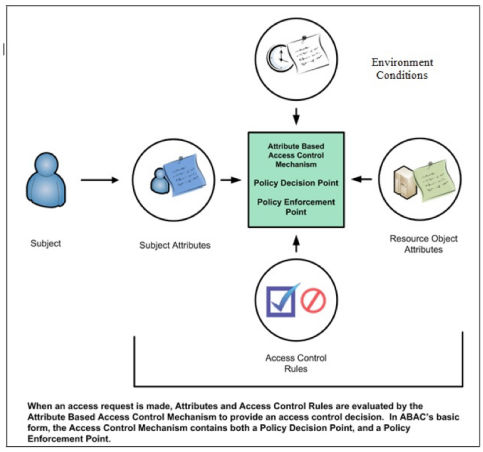

Figure 2.2: Core ABAC Mechanism [18]

Figure 2.2 enlarges ABAC mechanism. ABAC basically relies on the evaluation of attributes of the subject, object, environment conditions. This mechanism also includes policy enforcement point (PEP) and policy decison point (PDP). PEP intercepts user's access requests to a resource and makes a decision request to the PDP to obtain the access decision. PDP evaluates access requests according to the policy (see appendix C).

Universal Data Element Framework (UDEF) While the ABAC guideline document [18] does not suggest how to construct a wide range of attributes, research on Universal Data Element Framework(UDEF) [4] may contribute to organise those attributes. UDEF has been studied by open group of UDEF project [4] based on ISO/IEC 11179 (Metadata Registry (MDR) standard) [6] to reduce data management cost in a large enterprise which typically has many different data stored in various applications. Therefore, the main purpose of this project is to establish the UDEF as the universally-used naming, categorization and indexing system for enterprise data. Definition of UDEF is as following [4]:

> UDEF is a framework for categorizing, naming, and indexing data. It assigns to every item of data a structured alphanumeric tag plus a controlled vocabulary name that describes the meaning of the data.

2.2 Persona

Several months ago, I found the NIST document "Guide to Attribute Based Access Control (ABAC) Definition and Considerations (February, 2014)" [18] which briefly mentioned persona as a user's attribute as below.

> In the course of a person's life, he or she may work for different organizations or act in different roles, they may inherit different privileges tied to those roles. The person may establish different personas for each organization or role and amass different attributes related to each persona.

The idea above possesses a risk of data leak which can lead to insider threats, especially if this person is working for competing organisations.

I will define persona as a combination of user's attributes that show different motives and goals to the others. For example, an employee who works for two different companies creates two employee personas with different attributes. Figure 2.3 demonstrates an object diagram of the example. John Smith is working for IBM and also for Oracle. He creates two employee personas with different attributes related to them. A risk of a data leak might arise if he attempts to access to the information on competing software products. Therefore, a fine-grained access control is needed for conflicting personas.

In the next section, I will demonstrate how adding persona helps to recognise a concern of the loan manager case discussed in section 1.2.

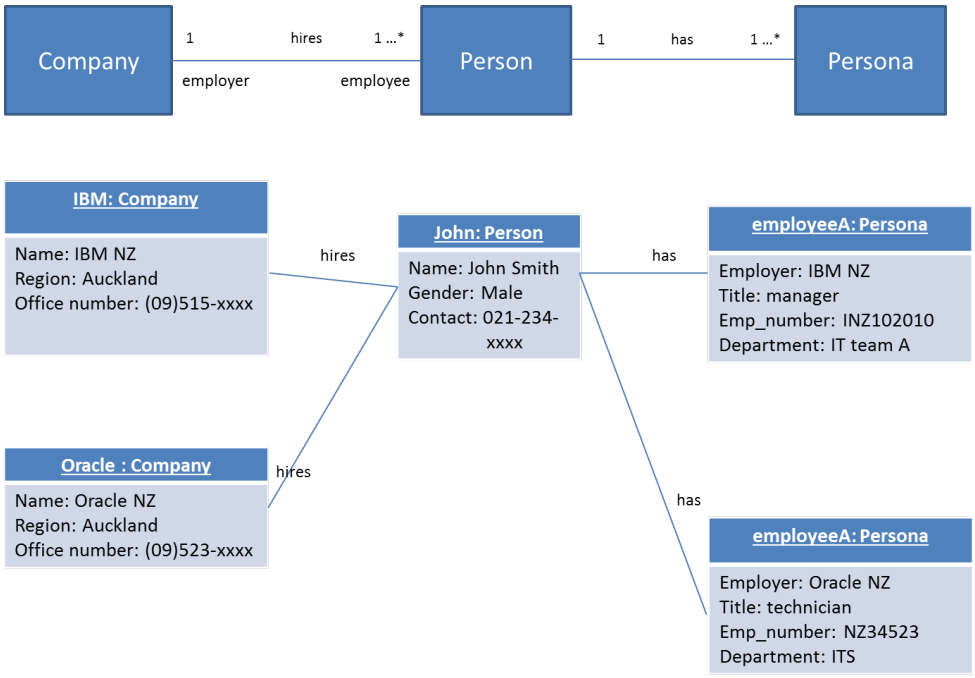

Figure 2.3: two company example

2.3 Design

Handling conflicts of interest can be achieved by creating a strong ABAC rule by limiting an access request which has conflicting personas. Personas will be represented with an object diagram and UDEF person object tree model.

2.3.1 Representing personas

Persona object diagram As seen in the example in section 1.3, Lena has conflicting personas of an employee persona and a client persona. This can be illustrated with an object diagram. See figure 2.4.

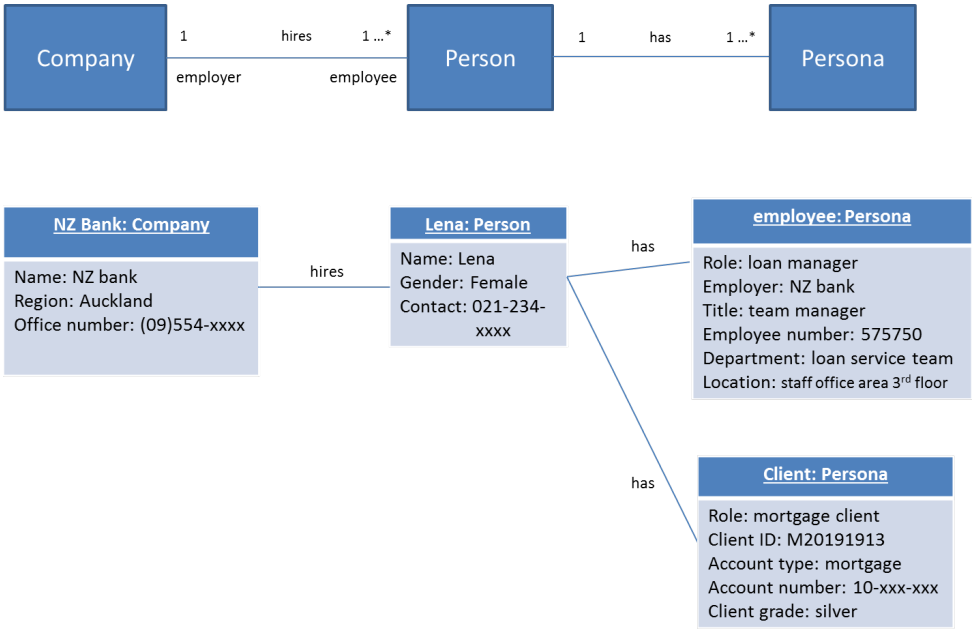

Figure 2.4: personas of Lena

To limit access to sensitive data that might cause conflicts of interest, it is important to recognise conflicting personas. However, it is challenging to identify which personas belong to which person. It is even more challenging if persona attributes are distributed in various applications. Persona might not have any attributes to identify whom it belongs to. Therefore, it is necessary to organise and index persona attributes. UDEF person object trees can be used as a categorisation scheme for personas.

UDEF person object tree According to UDEF, person is described as "any data or information about human being that is relevant to the enterprise" [4]. A current work of a person object tree [4] (see figure 2.5) categorises a person into many types, such as an employee, customer, patient, investor and so on. Those types of person can be applied

to represent persona. For example, if a person is an employee and also a customer or if a person is a doctor and also a patient at the same time may have different motives and goals to the organisation.

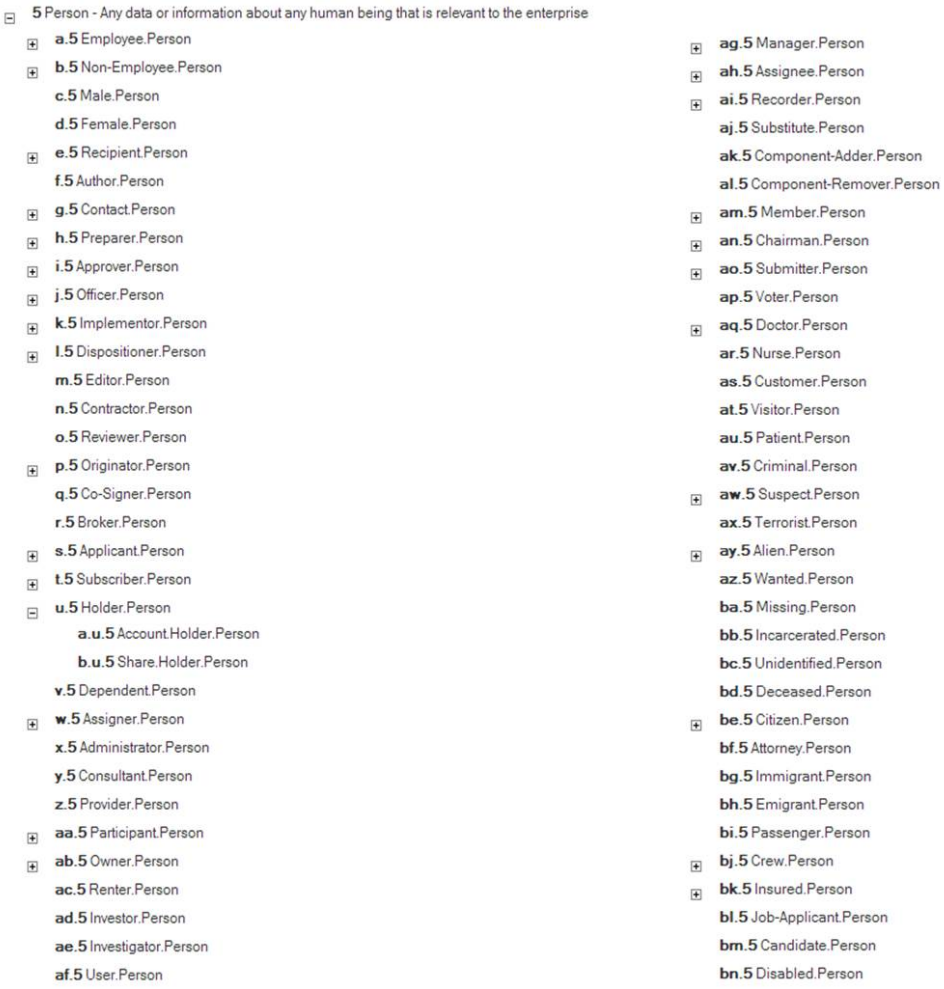

Figure 2.5: UDEF person object tree [4]

Back to the motivating example, since Lena is an employee and a client, elements in an UDEF person object tree [4] 'a.5 Employee.Person' , 'as. 5 Customer.Person' or 'a.u.5 Account.Holder.Person' can be applied to represent Lena's persona. This categorization scheme is useful to create an ABAC policy to authenticate persona and as a result, limit access to a user who has a conflicting persona.

2.3.2 ABAC rule

To decide an access request, for example, 'Is Lena allowed to modify her own record?', it is necessary to make an access policy. To represent the requirements that specify how information is managed and who, under what circumstances, may access what information, Natural Language Policy (NLP) can be considered. Hu [18] also pointed out that Natural Language Policy (NLP) in ABAC can be created based on a conflict of interest factor. Again in Lena's example, to prevent Lena from changing her own record in the 'client_info' table, access control rules have to finely tune several requirements. First, access control mechanisms have to check if Lena has a 'client' persona. Once authenticated as a user holding a 'client' persona, 'read-only' action to her record in 'client_info' table is allowed with time and location constraints, which are only during working hours (8AM to 5PM) and within staff office area (3rd floor). Therefore, she cannot change her own record. The following pseudo ABAC rule shows flexibility by allowing dynamic access control according to attributes and environment conditions.

```
Access control rule:
# Working hours = 8AM to 5PM
1: If a user has a client persona c
2: Allow access to resource 'client_info' table  with  c.client_id
          then action is read-only to the record with c.client_id
          condition on time = working hours
          condition on location = staff office area on 3rd floor
```

ABAC access rule can be represented by XACML(eXtensible Access Control Markup Language). However, it is beyond the scope of this dissertation and adding persona to XACML can be a future research direction. See appendix A.

3
Evaluation

ABAC provides various attributes to address contextual information. However, a set of attributes are passively enumerated only to verify the access rules. Persona is designed to organise relevant attributes to show a motive and goal to recognise potential conflicts of interest. This section will evaluate if the suggested solution is more secure than ABAC to handle conflict of interest. Then, I will discuss where this solution can be applied.

3.1 Security: is it more secure than ABAC to handle conflicts of interest?

In an organisation, a security analyst might be in charge of assessing the risk of system and as a first step, he or she may identify sensitive data to be protected. In addition, a security analyst can check if there is any conflicts of interest between personas when accessing sensitive data.

For example, in Lena's case, the 'client_info' table needs to be classified as sensitive data to be protected. And access rule needs to restrict an access request if a user has both an employee and a client persona. By restricting the modifying operation to her own record in the'client_info' table, conflicts of interest can be handled. Therefore, security risks will be mitigated.

However, ABAC rules without persona will be very long to address all relevant attributes which will lead to 'attribute explosion' [21] (see figure 3.1). Persona organises a user's attributes acording to similar motive and goal.

On the other hand, if it fails to recognise persona or fails to create the ABAC rule, it may not handle conflict of interest. It might be challenging to develop a certain quality of attributes of a user. A user's attributes can be provisioned by multiple attribute authorities such as human resources, security and organization leadership [18]. While security authority provisions clearance attribute, other authorities involve name attribute, user's current tasking, physical location, and the device from which a request is sent. These attributes should not be altered by individuals and needs to be managed strictly.

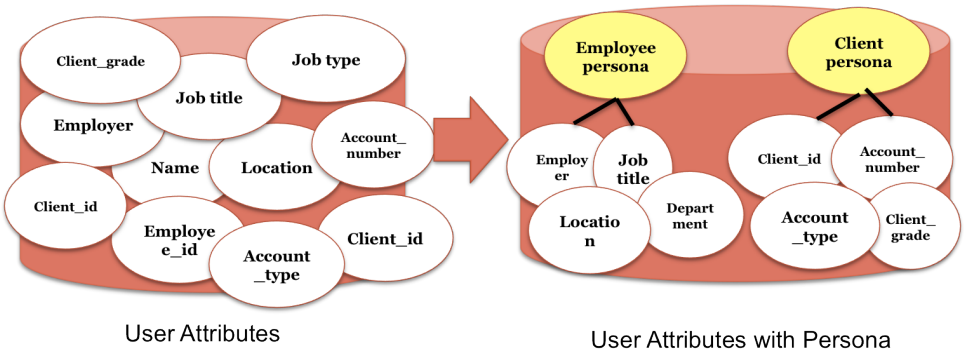

Figure 3.1: Attributes with personas

3.2 Feasibility

This new method can be applied to a financial sector to finely tune access control. As we have seen in Lena's case, sensitive tasks such as changing employee's own records or family records might cause conflicts of interest. By providing fine-grained access control, this risk can be avoided. In addition, this method can be used to prevent or detect insider trading. According to Securities Markets Act 1988 [3], insiders are categorised as primary or secondary insiders.

> insiders are defined as the public issuer, a director of the public issuer, officers of the public issuer, an employee of the public issuer and a substantial security holder (these people are known as primary insiders). People who obtain information in confidence from primary insiders are also insiders (known as secondary insiders).

Securities Markets Act 1988 [3] clarifies that an information insider of a public issuer must not trade securities of the public issuer in section 8. The proposed method can detect the

breach by checking if an insider has security holder persona. Furthermore, the concern that "a person can be an insider if they have a relationship with the public issuer (i.e. are primary or secondary insiders). " might be detected by building persona of an insider that he connects with. But as it is stated in [3], this problem is complicated and can avoid the prohibition by disguising any connection between the secondary insider and the source.

Also, in the law sector, this method can be applied to prevent a dual role relationship by recognising conflicting persona such as current client or family a counsellor has. In the example of Carl in section 1.3, he attempts to take advantage of information gained from the current case he is working on. Under the new method, he is not eligible to be involved in the new case because he has conflicting persona.

However, adopting ABAC is considered to be challenging since it requires changes in other systems as well. For example, identity access management system (IAM) needs to consider potential access-giving attributes instead of focusing on roles and entitlements which need to be attested and certified [15].

3.3 Limitation

There are several requirements to support the proposed solution. First, a sensitive object to be protected should be identified. Second, based on comprehensive understanding of a user, a certain depth of personas should be built. Finally, based on requirements above, fine-grained access rules should be created.

However, if any of those requirements are not valid, this solution is not able to recognise conflicts of interest.

In addition, it was pointed out that accountability could be lost [18] in generic ABAC. Since ABAC relies on attributes when deciding access request. Accountability refers to the ability to track who access the resource with which account (user ID). Therefore, to assure accountability, there will need mechanism to track the access by a user's account.

4
Discussion

In this chapter, I will briefly discuss the current access control trend towards ABAC. Then I will discuss challenges, contribution, limitation of this dissertation and finally, I will discuss future research directions.

I started this research to address conflicts of interest, especially in the RBAC model.

I tried to add personas to generic RBAC data relations, however, it was unable to handle complicated data structures. To achieve fine-grained access control, it requires sophisticated roles by creating 'role explosion'. Moving to ABAC gave flexibility to handle various attributes and complicated access rule.

For this reason, ABAC has drawn attention to finely tune access control. According to the Gartner's research note [15], ABAC will be the dominant access control in near future.

> By 2020, 70% of enterprises will use ABAC as the dominant mechanism to protect critical assets, up from less than 5% today.

However, ABAC is currently taking only 5% in the industry. This is because ABAC requires the support from other systems such as identity management system or attribute provisioning system.

Instead of moving straight to ABAC , there are some hybrid approaches to slowly adopt the ABAC from the legacy of the RBAC model [10, 22]. This implies that people

do not want to lose the benefit of simple administration from RBAC, but need more flexible and fine-grained access control from ABAC.

In contrast, there is an opposite security approach called "People-Centric Security" (PCS) [27]. While ABAC focuses on tightly controlling employees access, PCS is trust based and relying on personal responsibility on individual employees for the protection of information.

Persona has been studied mostly in user centric design or requirement engineering [9, 23]. SAP screen persona [2] is a good example for implementing persona to maximise usability. Users can modify GUI screens according to personas and this leads to productivity.

While persona is mentioned as an attribute in a recent ABAC literature [18], it was not further discussed nor implemented. The primary contribution of this dissertation is to find persona as an important attribute in ABAC in an early stage, and embed persona as a key attribute to ABAC for the purpose of handling conflict of interest.

In addition, persona contributes to simplify rule sets. According to [21],"In ABAC you must keep a very structured eye on your rule sets otherwise you will swap the role explosions with explosions of entries in the rule set.". By connecting relevant attributes of a user, persona contributes to avoid complicity of attributes in rule sets.

However, due to time limits, I was unable to implement or build a prototype for the method. It will be a future research direction to implement ABAC by adding persona and place conflicts of interest scenarios. Evaluating security and feasibility will become possible after a prototype is complete.

Creating access rules and representing personas with existing XACML (eXtensible Access Control Markup Language) can also be a research direction.

Even though this method has limitations, I would conclude that persona is an important attribute for allowing it to recognize and mitigate conflicts of interest in an organisation. In addition, persona is not used as a common term for user-centric design, but used as a key attribute for the use in access control mechanism. Lastly, this dissertation broadens the use of ABAC to recognise and mitigate conflicts of interest.

A

Example of XACML (eXtensible Access Control Markup Language)

Attached is an example of XACML (eXtensible Access Control Markup Language) which states that MPEG movies for adults can be downloaded only by users with age equal or greater than 18 years. [8]

```
<Policy ID = P1>
<Target>
      <Subjects> <Subject> GroupName = IBMOpenCollaboration </Subject>
</Subjects>
      </Target>
<Rule ID = R11 Effect = Permit>
      <Target>
           <Subjects> <Subject> Designation = Professor </Subject>
</Subjects>
           <Resources> <Resource> FileType = Source </Resource>
</Resources>
           <Actions> <Action> Type = Read </Action> </Actions>
           <Environments> <Environment> Time = (8AM, 6PM)
</Environment> </Environments>
      </Target>
      <Condition> (FileSize < 100MB) </Condition>
</Rule>
<Rule ID = R12 ..> ... . </Rule>
```

Figure A.1: Example of XACML [8]

B

Example of Enterprise ABAC Scenario

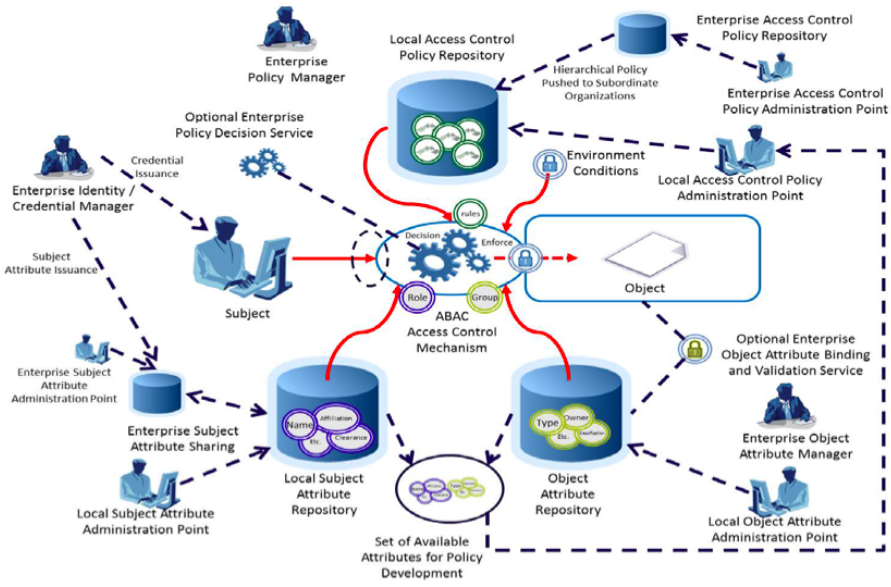

Figure B.1: Example of Enterprise ABAC Scenario [16]

C

Example of Access Control Mechanisms Functional Points

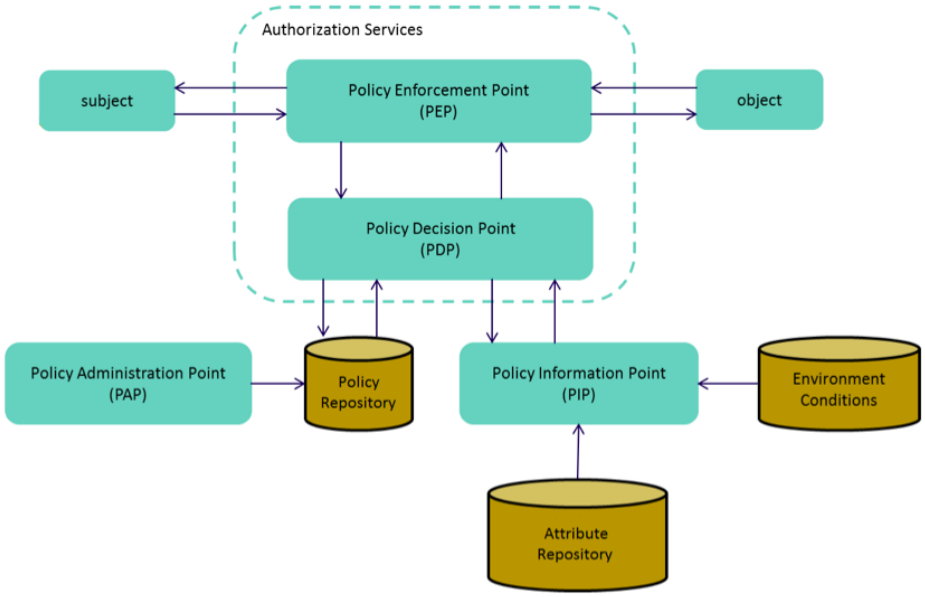

Figure C.1: Example of Access Control Mechanisms Functional Points [16]

D

RBAC Data Model

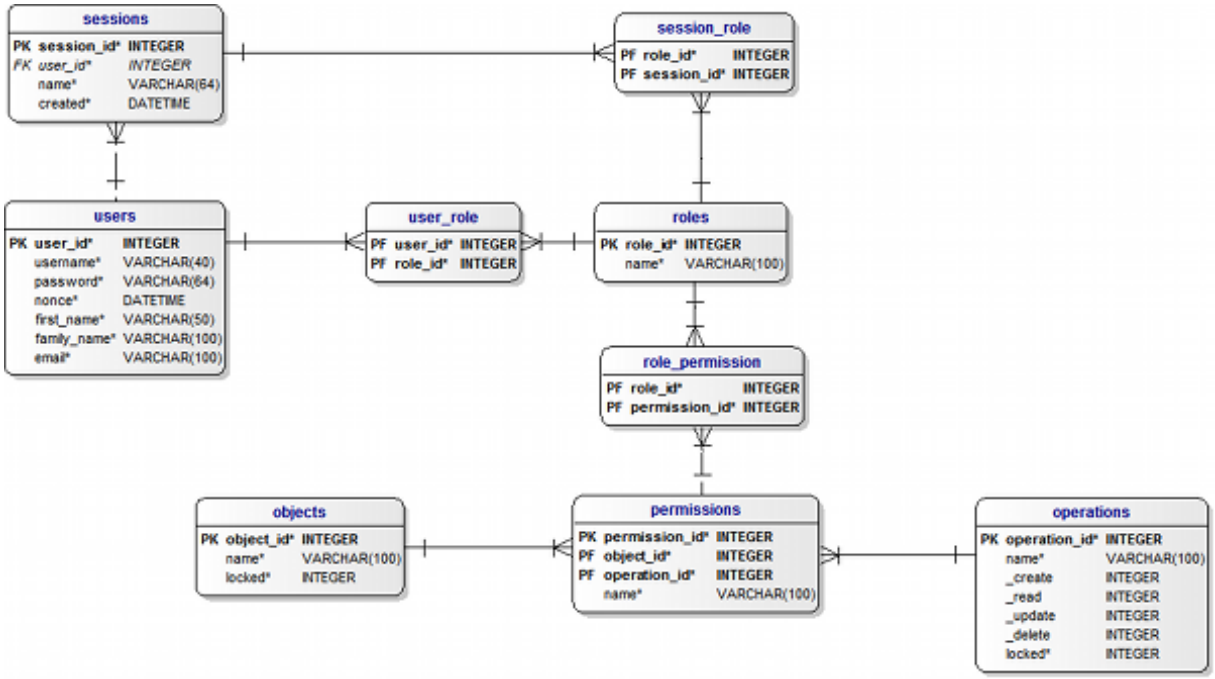

Figure D.1: Example of RBAC Data Model [5]

USER		
ID	NAME	ADDR
U1	JOHN	PUKE
U2	HELENA	ALBANY
U3	CLARK	CITY

PERSONA	
ID	NAME
P1	CUSTOMER
P2	EMPLOYEE
P3	SERVICE VENDOR

ROLE	
ID	NAME
R1	MANAGER
R2	CREDIT CARD HOLDER
R3	SERVER MAINTENENCE
R4	SYSTEM ADMIN

USER_PERSONA		
ID	UESR	PERSONA
UP1	U1	P1
UP2	U1	P2
UP3	U2	P1
UP4	U2	P3
UP5	U3	P1
UP6	U3	P2

PERSONA_ROLE		
ID	PERSONA	ROLE
PR1	P1	R2
PR2	P2	R1
PR3	P2	R4
PR4	P3	R3

Figure D.2: Example of complex relations by adding personas to RBAC

References

[1] NIST (National Institute of standards and Technology) (2013) ABAC workshop presentation. Retrieved from http://csrc.nist.gov/projects/abacjuly2013_workshop/presentations.html/. Accessed June 2014.

[2] SAP screen persona. Retrieved from http://wiki.scn.sap.com/wiki/display/Img/SAPScreenPersonas?original_fqdn=wiki.sdn.sap.com/. Accessed March 2014.

[3] New Zealand legislation. Retrieved from http://www.legislation.govt.nz/act/public/1988/0234/latest/DLM139727.html/. Accessed May 2014.

[4] Universal Data Element Framework. Retrieved from http://www.opengroup.org/udef/. Accessed March 2014.

[5] RBAC Data Model (2010). Retrieved from https://www.mind-it.info/nist-rbac-data-model/. Accessed June 2014.

[6] ISO/IEC 11179. Metadata standards. Retrieved from http://metadata-standards.org/. Accessed April 2014.

[7] Information Systems Audit and Control Association. Separation of duties. Retrieved from http://www.isaca.org/. Accessed January, 2014.

[8] Elisa Bertino. Data protection from insider threats. *Synthesis Lectures on Data Management*, 4(4):1–91, 2012.

[9] Alan Cooper et al. *The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity*, volume 261. Sams Indianapolis, 1999.

[10] Ed Coyne and Timothy R Weil. ABAC and RBAC: Scalable, flexible, and auditable access management. *IT Professional*, 15(3):0014–16, 2013.

[11] Michael Davis and Andrew Stark. Conflict of interest in the professions. 2001.

[12] Aaron Elliott and Scott Knight. Role explosion: Acknowledging the problem. In *Software Engineering Research and Practice*, pages 349–355, 2010.

[13] David Ferraiolo, Janet Cugini, and D Richard Kuhn. Role-based access control (RBAC): Features and motivations. In *Proceedings of 11th Annual Computer Security Application Conference*, pages 241–48, 1995.

[14] David F Ferraiolo and D Richard Kuhn. Role-based access controls. *15th National Computer Security Conference, 1992, pp. 554 - 563.*

[15] Gartner. Predicts 2014: Identity and access management (2013). Retrieved from `http://www.gartner.com/`. Accessed April 2014.

[16] Vincent C Hu and Karen Ann Kent. *Guidelines for access control system evaluation metrics*. US Department of Commerce, National Institute of Standards and Technology, 2012.

[17] Vincent C Hu, David Ferraiolo, and D Richard Kuhn. *Assessment of access control systems*. US Department of Commerce, National Institute of Standards and Technology, 2006.

[18] Vincent C Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. Guide to Attribute Based Access Control (ABAC) definition and considerations. *NIST(National Institute of standards and Technology) Special Publication*, 800:162, 2014.

[19] Robert R Jueneman. Integrity controls for military and commercial applications. In *Aerospace Computer Security Applications Conference, 1988., Fourth*, pages 298–322. IEEE.

[20] Richard Kissel. *Glossary of key information security terms*. DIANE Publishing, 2011.

[21] Niels Knuzen. Analysis of SAPs ability to use ABAC, SAP community network, 27 december 2013. Retrieved from `http://scn.sap.com/docs/DOC-50389/`. Accessed June 2014.

[22] D Richard Kuhn, Edward J Coyne, and Timothy R Weil. Adding attributes to role-based access control. *IEEE Computer*, 43(6):79–81, 2010.

[23] Granville Miller and Laurie Williams. Personas: Moving beyond role-based requirements engineering. *Microsoft and North Carolina State University*, 2006.

[24] Kenneth S Pope, Hanna Levenson, and Leslie R Schover. Sexual intimacy in psychology training: results and implications of a national survey. *American Psychologist*, 34(8):682, 1979.

[25] Ravi S Sandhu, Edward J Coynek, Hal L Feinsteink, and Charles E Youmank. Role-based access control models yz. *IEEE computer*, 29(2):38–47, 1996.

[26] Ravi S. Sandhu, David F. Ferraiolo, and D. Richard Kuhn. The NIST model for role-based access control: towards a unified standard. In *ACM Workshop on Role-Based Access Control*, pages 47–63, 2000.

[27] Tom Scholtzrtner. Consider a people-centric security strategy (2013). Retrieved from `http://www.gartner.com/`. Accessed May 2014.