Demonstrating and Implementing the Work-Based Access Control Policy for Cooperative Healthcare Environment Using XACML

Mohamed Abomhara and Henrik Nergaard

Department of Information and Communication Technology
University of Agder, Grimsatd, Norway
Email: {mohamed.abomhara, henrik.nergaard}@uia.no

Abstract—**This study focuses on collaborative activities that are best accomplished by organized groups of healthcare practitioners within or among healthcare organizations with the objective of accomplishing a specific task (a case of patient treatment). In our previous work, we proposed an access control model (work-based access control (WBAC)) that is suitable for collaborative healthcare systems in terms of addressing the issues of information sharing and information security. The current study extends on that work by demonstrating and implementing the WBAC access policy for a collaborative healthcare environment to support diverse domains of data authorization management with various constraints. The implementation is based upon using eXtensible Access Control Markup Language (XACML) with SunXACML. We explain the WBAC model for cooperative healthcare systems, introduces a software structure for WBAC implementation, implement the WBAC profile using XACML 2.0, specify permissions and define all authorization policies. Also, we validate the model and compare it with the existing solution to ensure that the model can fulfill and satisfy the main intended objectives. The experimental results demonstrate the efficiency and scalability of WBAC approach. It shows how the WBAC model simplifies decentralized administrative tasks (e.g., changing of team members and shifting responsibilities), thus enhancing the practicability of access control in dynamic collaboration environments.**

Keywords–XACML; Access control; Access control policy; Collaboration environments; Healthcare.

I. INTRODUCTION

Information flow concerns how the information should proceed to authorized entities [1], to whom the information should be propagated and what steps and methods should be used to ensure information flow [2]. Secure information flow comprises two related aspects: information confidentiality and information integrity [3]. Information confidentiality involves a set of rules that limits access or places restrictions on certain types of information. Information integrity seeks to prevent an accidental or malicious destruction of information. Different systems have various confidentiality and integrity requirements. For instance, a remote patient monitoring system will have high confidentiality requirements where data must be hidden from unauthorized entities as well as a high integrity checking against random errors due to information sensitivities [4]. Information confidentiality and integrity are increasingly dependent on how the information should flow, to whom the information should be propagated and what steps and methods should be used to ensure information flow.

Access control policies play an important role in ensuring that the information flow is controlled between authorized entities while preserving resource security in the face of inappropriate access [5, 6]. Access control policies specify which authorized entities (e.g., user or organization) can perform what operations on specific resources (e.g., files on electronic health records (EHRs) [7, 8]). In collaborative environments such as healthcare, it is not easy for traditional authorization mechanisms like role-based access control (RBAC) [9–11] and attribute-based access control (ABAC) [12, 13] alone to specify authorization constraints due to the complexity of a continuously growing as well as changing number of users and medical records. In addition to a lack of granularity, manageability and flexibility for the specification and maintenance of policies [14, 15].

Moreover, inconsistencies between the access control policies of various individuals or organizations are a common challenge [16]. Due to the dynamic nature of collaboration and team work, it is important to understand to what extent and under what conditions other parties are allowed access rights [17, 18]. It is also necessary to employ access control policies to control the way in which information or services are shared between different parties [19, 20]. In distributed environments, different participants (individuals or organizations) can play several different roles at a given time (e.g., resource owner, agent or consumer) [1, 21, 22]. Moreover, each participant manages their own resources and defines their own access control policies. Thus, participants collaborate with each other in various ways, which requires appropriate access control mechanisms in place to ensure that information is accessible only to those authorized to have access [23].

In our previous work [1, 21, 22, 24–26], work-based access control (WBAC) model was proposed. WBAC is extended with the team role concept. A team role classification based on Belbin team role theory [27, 28] was proposed [24]. The nine different team roles that Belbin identified were rephrased and classified into *thought*, *action* and *management* [24]. Role is used in conjunction with team role to handle access control in dynamic collaborative environments. Team member must be assigned to one team role (determined by their professional and/or technical knowledge) based on the goal, task and contributes towards achieving the team's objectives. The team role determine the finer role and the extend of access of each team member.

This study extends the previous work [1] to demonstrate and implement WBAC access policy for a collaborative healthcare environment to support diverse domains of data authoriza-

tion management with various constraints. The implementation is built based on eXtensible Access Control Markup Language (XACML) [29]. The aim is to simplify decentralized administrative tasks and thus enhance the practicability of access control in dynamic collaboration environments.

The remaining parts of this study are structured as follows: Section II presents usage scenarios of collaboration and healthcare data sharing followed by a detailed description of personal role, team role and resource classification. Section III provides an overview of XACML, demonstrates the modeling structures, authorization constraints, request model, policy model, experiments and result. Section IV presents WBAC authorization framework. Validation of the proposed WBAC model and comparison summary with existing solutions are presented in Section V. Discussion, conclusion and future work recommendations are provided in Section VI.

II. Background and Motivation

This section starts with with a short usage scenarios to better understand the collaborations in healthcare domain. This is followed by a description of personnel categories (personal roles, team roles) and the resources classification in WBAC.

A. Usage scenario: multiple healthcare practitioners cooperation among multiple healthcare organizations

As shown in Figure. 1, a patient named *Alice* is recently diagnosed with gastric cancer. Surgical removal of the stomach (gastrectomy) is the only curative treatment. For many patients, chemotherapy and radiation therapy are given after surgery to improve the chances of curing. *Alice* entered a cancer-treatment center at her chosen hospital (e.g., hospital A). *Alice* has a primary care doctor (*Dean*) who she regularly visits. Upon entering the hospital, *Alice* also sees an attending doctor (*Bob*) from the hospital. *Alice*'s health condition has caused some complications, so her attending doctor would like to seek expert opinions and consultation regarding *Alice*'s treatment from different hospitals (e.g., hospital B), including *Alice*'s specific primary care doctor who is fully informed about *Alice*'s medical history. Note that the invited practitioners are specialized in different areas, where some are specialists and others are general practitioners. Also, the final medical report of *Alice*'s treatment should be signed by appropriate practitioners using digital signatures [30, 31]. *Alice* should be able to verify the authenticity of the consultation results through the practitioner's digital signature [22, 32].

In such group consultation, also so-called multidisciplinary team consultation [33–35], it is noticeable that:

- Several healthcare professionals are involved in various roles to provide patient care. That includes primary care doctors, general physicians and specialists.
- The care team are formed dynamically and can be readily changed. For example, when *Alice*'s health condition causes some complications, her attending doctor wishes to seek expert opinions and consult with specialists. As a result of a request for a gastroenterology consultation, we assume a gastroenterologist (*Cara*) will join the care team.
- Every participant needs to obtain the medical records they request based on the health insurance portability and accountability act (HIPAA) [36, 37] minimal disclosure principle [38, 39].

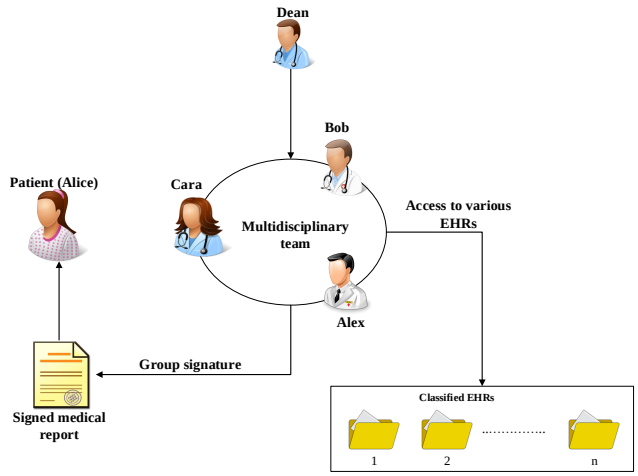


Figure 1. An example scenario of collaboration and sharing of healthcare data [26, p. 4]

- Sharing and accessing healthcare records with efficient coordination between healthcare practitioners is a critical function in access control models [40]. The main concern regards losing control of sensitive healthcare records while sharing them with multiple parties.

The act of managing the collaborative work in a given scenario must be defined clearly. By default, only the main practitioner (*Dean*) should be aware of the patient's personal information. The three other medical practitioners with supporting roles receive information based on their contributing roles based on "minimum necessary" standard to uses and disclosures for treatment [41]. The minimum necessary standard requires covered entities to evaluate their practices and enhance protection of health information as needed to limit unnecessary or inappropriate access to and disclosure of protected health information [41, 42].

B. Personnel categories: personal role

A role can be thought of as a set of permissions that a user or set of users can perform within the context of an organization [11, 43]. Permissions are allocated to roles by a system administrator. Such permissions include, for instance, the ability for a doctor to enter a diagnosis, prescribe medication, and add a entry to a record of treatments performed on a patient. Role can be organizational role in which participant has a common set of permissions for performing the job function associated by the name of the role. Example of hospital roles are medical practitioners, nurses and administrators (Figure 2). Moreover, role can be personal roles which represent an individual. They used to create a private workspaces for individuals [18]. Examples of personal roles include pediatric specialists, surgeons or pharmacists. As shown in Figure 3, the role of a pharmacist includes the permissions to dispense but not prescribe prescription drugs.

The role can be statically or dynamically assigned to subject. Static roles are predefined by the organization and manually assigned to users by system administrator, based on a specific organization policy, thereby authorizing users to use the roles' permissions. Membership in a static role is also revoked by a system administrator. The main issues with static roles are how to assigned and revoke them to users and how

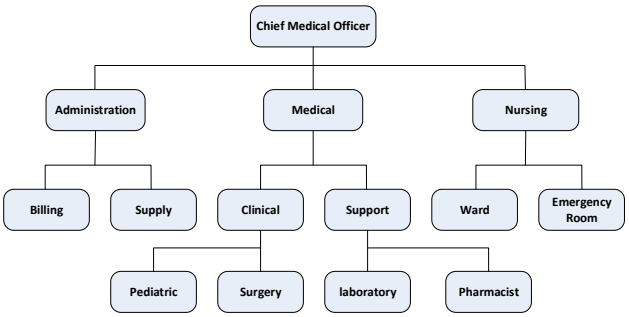


Figure 2. Example of organizational hospital chart

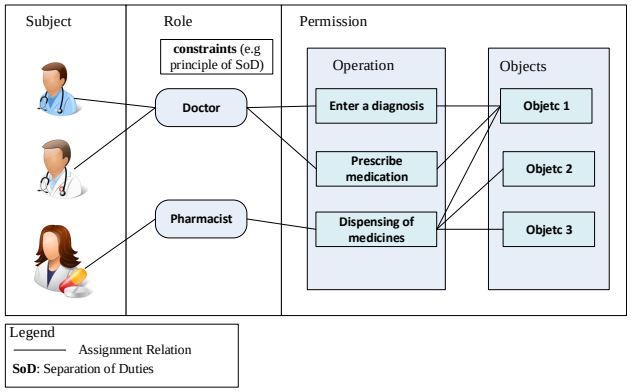


Figure 3. Subject, role and permission relationships

to guarantee that subject are assigned to appropriate roles. An appealing solution is to automatically assign/revoke users to roles. The dynamic role assignment approach has been studied by many researchers. *Al-Kahtani* and *Sandhu* [44] proposed a model to dynamically assign users to roles based on a finite set of rules defined by the organization. Moreover, *Alshehri et al.* [45] proposed a model which uses a concept of pseudorole, which is informally defined as a set of values of static attributes of subject. Although these model tried to solve the problem of assigning users into appropriate roles, it still inherit the major limitation of RBAC, including the lack of granularity and flexibility as well as dynamic adaptability specially in collaborative environments.

The problem of assigning users to role is out of the scope of this study. We assumed that the users within an organization has a role regardless of whether the role has been assigned statically or dynamically. We also believe that, WBAC model can adapt both approaches; static and dynamic subject-role assignments. In our modeling (Section III), we used static role assignment, where we assumed all subject have their roles assigned.

C. Personnel categories: proposed team role

Team is defined as a collection of subjects in specific roles with the objective of accomplishing a specific *work* [46]. Each team has a responsible team manager. Any of the subjects joining a team shares a common goal and may share a default set of permissions for their cooperative work. The notion of a *team role* is used in this study to restrict access permissions to those individuals who not only have the right organizational roles but also are associated to the cooperative *work* via team membership [24].

Regarding the process of collaboration and team work, access control model must be able to provide an efficient and secure platform for people to work together in a hospital without being deterred by restrictive enforcement of access control policies [17]. This can be a rather delicate situation to handle, given the fact that the fluidity of teamwork within the medical domain is often incongruent with technological security. To demonstrate this notion, we consider a scenario (Section II-A) involving four medical practitioners who are working together on a patient's case. For the sake of securing the patient's private (sensitive) data (e.g., mental illness records [47], etc.) [48], the collaboration must be clearly defined. By default, only the main practitioner should be aware of the patient's private information. The three other medical practitioners with supporting roles are given information based on their contributing roles. In order to achieve this, it is imperative to determine the finer roles of each team member. The team role of each member will subsequently determine the extent of access given. The concept of team roles is something that we see as integral to getting the team building process right [33].

Hospital personnel roles are often simplistically split into medical practitioners, nurses and administrators. However, their roles in a team can be further categorized using the team role theory (so-called also Belbin's team roles) [27, 28]. A good collaboration depends on more than team of people working together being enthusiastic and communicating well. Between them, they need the right mix of skills, resources and behaviors to serve the team, too [49, 50]. The Belbin' team role theory is a very useful for higher level team building processes as it helps an experienced facilitator identify the patterns that exist within any team and thus underpin their strengths and weaknesses. Team role theory contains a total of nine roles per group, which are classified into *thought*, *action* and *management* [24] as illustrated in Figure 4.

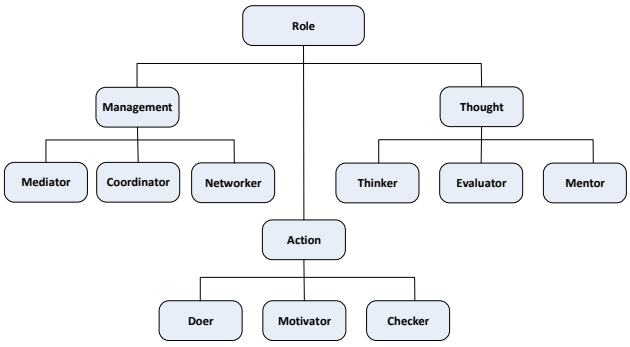


Figure 4. Taxonomy of team role [24, p. 217]

- ***Thought*** denotes a role that is dominated mostly by thinking, analyze problems and/or provide technical expertise. To be a successful thought collaborator, the person may need to understand the medical predicament in detail without necessarily knowing the patient. A worker in this role could be involved in devising strategies to confront particular medical enigmas. Thus, a cardiology specialist may offer his/her expertise regarding the best practices of performing a heart transplant on a child without being involved in the actual operation.

- ***Action***, as the labeling suggests, signifies being involved in task-related collaboration, such as meeting the patient for a medical check-up. Having an action role usually implies close interaction with the patient. Nevertheless, discretion is still feasible with care. For instance, an anesthesiologist needs to only know the patient's physical characteristics to prepare anesthetic. Who the patient is, or where the patient lives is not relevant to completing this task (this assumption is based on our review to [51] (preoperative evaluation and preparation for anesthesia and surgery).
- The ***management*** category comprises personnel who are mostly involved in managing others (e.g., guide, listen, delegate, and solve conflicts). These types of collaborators are adept at coordinating teamwork that is susceptible to social or psychological challenges. For example, in conflict management, they may have to resolve series of opposing diagnoses made by medical practitioners and that may otherwise escalate into serious altercations. In this regard, such personnel's need for information is inwardly oriented. They have a greater need to know personal information about others working at the hospital rather than of patients.

D. Resource classification

Medical records contain a wide range of information, not all of which may be shareable [52]. It could include personal names, phone numbers, addresses, appointment schedules, to do lists, as well as medical history and medical reports regarding patients, to name a few. Some elements of this information may be confidential and sensitive; others may be open for access. In an environment that supports resource sharing, unwanted parties could retrieve the confidential information causing information leakage and leading to the violation of patient privacy. One method to assure that resource sharing will prevent such confidential information leakage is to provide a mechanism to classify all information resources by their degree of share-ability [52, 53].

Medical records classification is infeasible and requires a great deal of effort and skills to accomplish. This is due to issues that, medical records include a variety of documentation of patient's history, diagnostic test results, and daily notes of a patient's progress and medications [54], to name a few. Moreover, healthcare providers can not decide on what appropriate information is really needed for treatment of a patient case. The HIPAA Privacy Rule [37, 55] is a set of standards to protect the privacy of patients' medical records as well as ensure how the health information is used, disclosed and maintained by healthcare organization and health care providers [56, 57]. Healthcare providers should inform and get a patient's permission (e.g., consent or authorization [1]) about how the patient's records are used or disclose? In general, information sharing needed for treatment, therefore, healthcare providers may use and disclose patient records for patient's treatment without a patient's authorization. This could occur during consultation between healthcare providers regarding a patient and referral of a patient by one provider to another. But in most cases when the healthcare providers are dealing with a sensitive information regarding the patient, patient authorization is required for disclosure. For example, The HIPAA Privacy Rule defines psychotherapy notes as "notes recorded by a health care provider who is a mental health professional documenting or analyzing the contents of a conversation during a private counseling session or a group, joint, or family counseling session and that are separate from the rest of the patient's medical record" [47]. Psychotherapy notes are treated differently from other mental health information because they contain particularly sensitive information and because they are the personal notes of the therapist that typically are not required or useful for treatment or health care operations purposes, other than by the mental health professional who created the notes. Therefore, with a few exceptions, healthcare providers must obtain the patient's authorization for any use or disclosure of such an information [47].

Resource within the WBAC is divided into two types, mainly $protected$ and $private$ resources. $Protected$ resources can be shared within a collaborative work. For example, consider scenario (Section II-A), we could say that $protected$ object contains resources related to *Alice*'s current case such as past surgical history, date related to abdominal CT scan (computed tomography scan) and gastroscopy data, to name a few. Contrary to the former type, the $private$ resources are highly classified pieces of information (e.g., name, data of birth, and address) within the medical records that would be shared during the collaborative work (only if needed). As such, the spreading of access control on the basis of collaboration will not affect the $private$ resources. It is meant to safeguard certain confidential information from being leaked out accidentally through collaborative means. In this study we assumed that, personal information (e.g., name, phone number, address, and /or IDs) and any medical records such psychotherapy notes [47] and sexually transmitted diseases (STD) records which are not related to the current medical case are $private$ resources.

III. XACML Profile for WBAC

In this section, we demonstrate and implement an WBAC model for a collaborative healthcare environment to support diverse domains of data authorization management with various constraints.

A. An overview of XACML

XACML is a standardized policy language by OASIS [29]. It defines the architecture, policies and messages of an access control system. XACML is a powerful and flexible policy language for heterogeneous distributed systems and is a general-purpose access control policy language [13, 58, 59]. According to the reference XACML architecture shown in Figure 5, the XACML model contains the following main entities [60, 61]:

- *The Policy Enforcement Point (PEP)* is an entity that intercepts a user's request to access a resource. The PEP forwards the request to the PDP to obtain the access decision (i.e., access to the resource is permitted or denied). PEP then acts on the received decision.
- *The Policy Decision Point (PDP)* is used to evaluate access requests against authorization policies and makes decisions according to the information contained in the request before issuing access decisions.

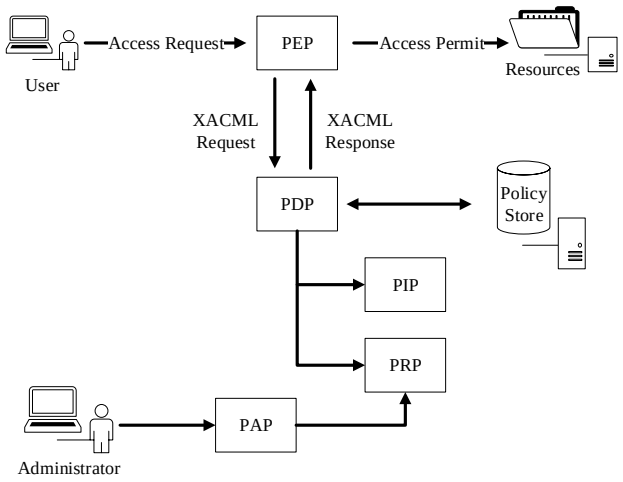


Figure 5. XACML framework

- *The Policy Information Point (PIP)* acts as the source of attribute values, or the data required for policy evaluation (i.e., a resource, subject, environment).
- *The Policy Retrieval Point (PRP)* is an entity that stores the XACML access authorization policies, typically in a database or filesystem.
- *The Policy Administration Point (PAP)* manages the access authorization policies.

The XACML core policy structure (Figure 6) consists of three components: the rule, policy and policy set [61]. The rule is a fundamental component of an XACML policy. The rule, policy and policy set have a target that PDP uses to quickly find the sub-policy parts applicable to making a decision regarding an access request.

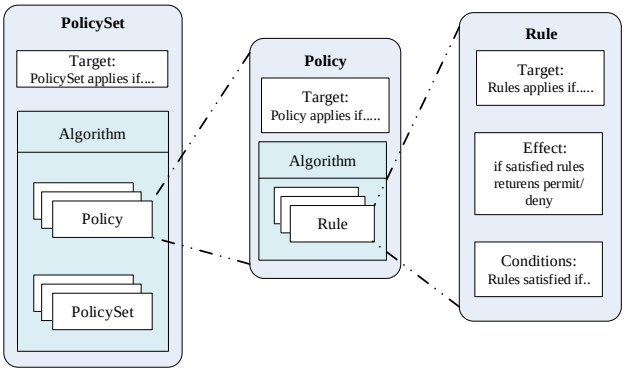


Figure 6. XACML policy structure

The target contains a set of attributes and their values for matching the subject, resource, action and environment, to check if the given rule, policy and policy set are applicable to a specific request. Several rules are grouped and encapsulated into policies and policies are grouped into policy sets. A rule consists of a condition and an effect that can be either a permission or denial associated with the successful evaluation of the rule. A condition represents an expression that refines the applicability of the rule beyond the predicates implied by its target. The correct evaluation of a condition returns the effect of the rule, while incorrect evaluation results in an error (*Indeterminate*) or the discovery that the condition does not apply to the request (*Not Applicable*).

PDP can use different rules, policies and policy sets to make a decision for a specific request. Therefore, conflict might occur between multiple policies when policies offer different authorization decisions. Thus, XACML provides a set of combining algorithms for combining rules and policies to solve a decision conflict between multiple policies [61]. The most commonly utilized combining algorithms are as follows:

1) Deny-overrides algorithm: combines decisions in such a way that if any rule or a policy evaluates denial, then the decision is "deny".
2) Permit-overrides algorithm: combines decisions such that if any rule or a policy evaluates permission, then the decision is "permit".
3) First-applicable algorithm: combines decisions in such a way that the final decision is made based on the first rule or policy in the policy file.
4) Only-one-applicable algorithm: This combining algorithm exists only to combine policy sets and policies. It cannot be used to combine rules. It returns the effect of the unique policy in the policy set that applies to the request; whether Deny or Permit [61].

Based on the combining algorithm used, PDP computes the authorization decision corresponding to the given access request. PDP evaluation is based on the rule, policy and policy set, for which the PDP returns the authorization decision, *Permit*, *Deny*, *NotApplicable* or *Indeterminate*. PDP returns to PEP a sequence of actions called *"obligation"* that should be performed in conjunction with enforcing the authorization decision applied to the access request given.

B. Collaborative work and XACML policy

The *work* model for WBAC (Figure 7) postulates that the entire nature of collaboration can be centralized by the work concept. Here, each *work* is connected to three main components; *personnel* (Section II-B and II-C), *patient* and *resource* (Section II-D). Managing the access control of collaborative work is an interplay between these components.

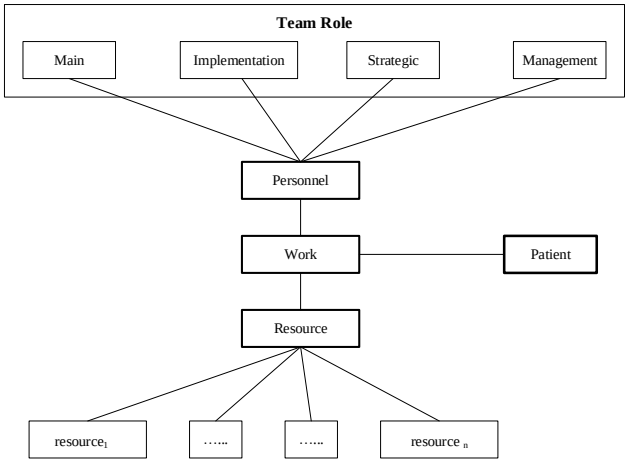


Figure 7. Work model for WBAC

Every resource in WBAC is considered a collaborative entity when it is assigned a *workID*. The *workID* connects

the resource to its corresponding work or project that is cooperatively done. By default, a resource does not have a *workID*. This implies that it is not a collaborative resource and thus, cannot be shared. To clarify the idea of managing security through a centralized work, consider the scenario below (Figure 8). Three resources (*resource1*, *resource2* and *resource3*) are all tied to a certain work. As such, all of them contain a *workID* to establish this connection. However, *resource4* is not connected to any work entity. Thus, it does not contain a *workID* and can only be accessed through the main policy.

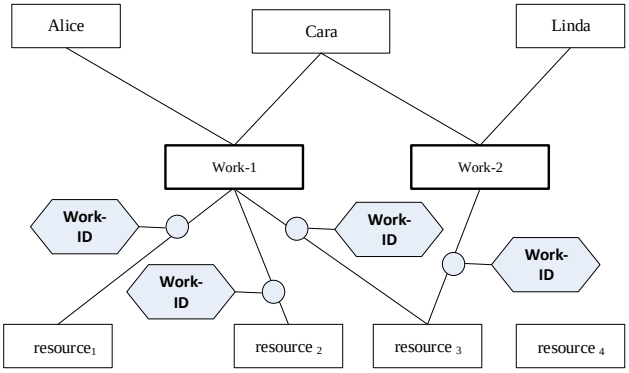


Figure 8. Work and shared resources

Any action that a subject (e.g., healthcare provider) would do on a resource (e.g., patient EHR) is defined entirely within the policy. A dynamic policy with dual inclination is proposed in WBAC [21, 24], whereby the normal policy of enforcing access control is contained within the main policy. On the other hand, any policy that mediates between resource sharing and collaborative work is covered by the collaboration policy. This way, better access control management is achievable. The main policy depends on the roles of the personnel in the organization (e.g., Dean is a general practitioner). PDP only considers the main policy if the personnel possess roles. The collaborative policy is dependent on team roles. In this respect, even if personnel do not have the required roles, they can still gain access upon invitation to collaborate. The team role provides a demarcation between the roles of personnel within a collaboration work and it restricts the role that each team member can have. A person can have various team roles, whereby each is tied to a different collaborative work.

C. Initiation of collaborative work

To begin, the initial situation for access control of which a patient visits the hospital and registers herself. Here, access is given to the physician that she comes in contact with, as well as the nurses at the health institution. As shown in the scenario patient name is *Alice* and her primary care doctor named *Dean*.

In this case of collaborative work shown in case scenario (Section II-A), the workflow of every healthcare practitioner is as follows:

- The primary care doctor (*Dean*) could not solve *Alice*'s case . He invites multidisciplinary team including *Bob*, *Cara* and *Alex* to help. In this team consideration (Figure 9), *Dean* is the core physician of the

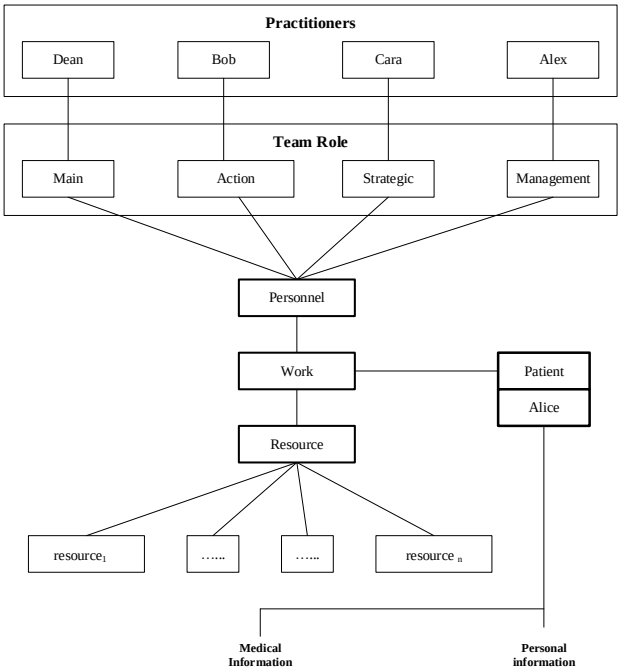


Figure 9. Scenario for team consideration

collaborative work. He serves as the team leader. He is responsible for initiating the work (treatment of *Alice*'s case) and choosing the practitioners (group of doctors) who may be required to attend *Alice*'s consultation and treatment. This implies that he possesses the *main* team role. In other words, he owns the collaborative work initiated. Therefore, full access is given to *Dean* with regard to the information related to the patient. He can access the personal information of the patient as well as the medical records (*private* and *protected* resources). Moreover, the primary care doctor must revoke the group upon completion of the patient's diagnosis consultation.

- *Bob* helps *Dean* with the operational part of the case. Operation refers to a series of responsibilities that entail interaction with the patient. *Bob* needs to see *Alice* on a face-to-face basis to perform various tasks that are related to her recovery. In this respect, there is a need for *Bob* to know personal and medical information about *Alice* to perform his duty effectively. *Bob* is involved in the action part of the collaboration. Therefore, his team role falls under the category of *action*.
- *Cara* has more of a strategy role. She is responsible for helping *Dean* solve the medical case. There is no need for *Cara* to meet *Alice* personally on a day-to-day basis. In fact, *Cara* is only required to analyze the medical situation and suggest a possible solution. *Cara*'s *thought* role within the team implies a rather clear indication of the access that she needs. Since *Cara* is predominantly preoccupied with diagnosing the disease, there is no urgent need for her to know the patient's personal information. As such, she is only given access to the patient's medical information as per her *thought* team role.

- With the increasing number of physicians working on *Alice*'s case, their interaction can become more complex. For instance, if there exists a competition between conflicting diagnoses given by *Bob* and *Cara*, which would gain priority? This is where *Alex* comes in. He contributes to the team by coordinating the interaction of the other members by taking on the team management role. To work effectively, *Alex* does not really need to know the patient's personal information. However, he must be aware of the patient's medical information to enable coordination. Furthermore, *Alex* must also be informed of the work information related to the physicians. In effect, access to certain staff and medical information of the client are given to *Alex*.

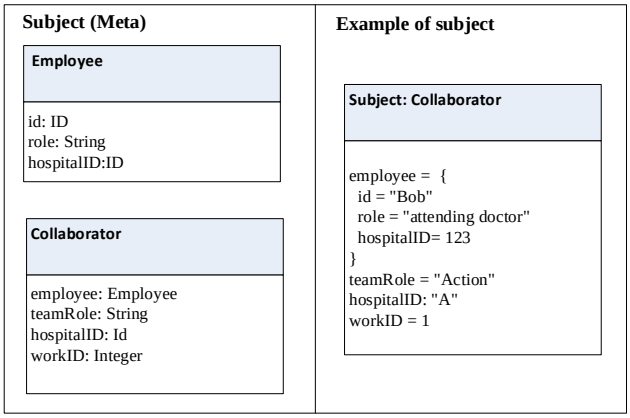
In addition, *Alice* may have some historical health information (e.g., mental illness or sexually transmitted diseases (STD), etc.), to which the group (or some of the group) of specialists and practitioners do not have to have access. As we assume in Section II-D that each resource in the system are divided into type, mainly *private* and *protected* during the collaborative work. Each shared resource is tied to the set of collaborative roles or team roles that can access it. In effect, the selected roles will determine the extent of collaborative access.

D. Modeling structures

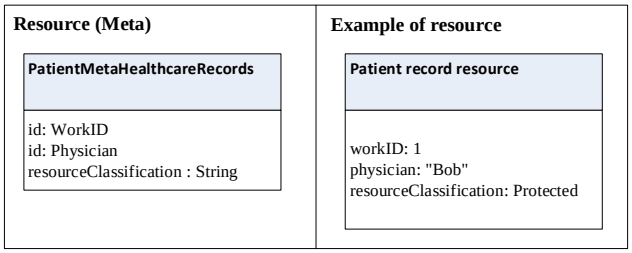
With the WBAC model, the policy is defined as a tree structure that narrows the combination of attributes presented in an access request. Access to a specific resource is granted when the whole policy tree has found possible matches to the request; the result from rule evaluation is then combined upwards to the outer-most policy using the combining algorithm defined at that level. The result is then sent back to the PEP.

The XACML structure of our model is as follows:

1) Subjects, resources and actions are elements defined by identifier/value pairs (Figure 10). Subjects (e.g., healthcare providers) are entities that send an access request to perform an action (e.g., read or write) on a resource (patient EHRs). The subject is modeled based on the minimal number of attributes required to make different decisions the policy is built to handle. Examples of identifiers are *role*, *employeeID*, *hospitalID* and/or *patientID* (a patient for whom the physician is responsible), to name a few. For the collaborative part, the information about the subject also includes the team identifier for the current collaboration work. As shown in Figure 10(a), physician *Bob* has been assigned the role of attending doctor in the hospital to perform some tasks. He is invited to a collaborative work (*work No 1*) and is assigned the team role *action* to perform some tasks in *Alice*'s treatment.
2) *Collaboration members* comprise a group of healthcare providers (specialists or general practitioners) who are invited to a collaborative work (in our case *Alice*'s treatment). Based on the given scenario, *Dean* is responsible for initiating the work and choosing the practitioners (team of doctors) who may be required to attend *Alice*'s consultation and treatment.
Bob, *Cara* and *Alex* joined the team and are assigned team roles based on the required job functions. Table I presents the policy data used as an input for XACML. An action represents the operation that a subject can perform on a resource, e.g., *read* and *write* operations. In our model, we also consider several resource attribute as show in Figure 10(b). We also assume the resource are classified into two categories *private* and *protected*.



(a) Example of subject attributes



(b) Example of resource attributes

Figure 10. Subjects, resources and actions are elements defined by identifier/value pairs

E. Authorization constraints

We describe the authorization constraints based on our team role classification and our usage-scenario (Section II-A) as follows:

- The subject (healthcare provider) who is assigned the primary doctor role can access both *private* and *protected* resources of the patient for whom he/she is responsible. Figure 11 shows a part of XACML policy ensuring that the primary doctor has a clearance to access medical records.
- A collaborative work must be active, such that team members can work on it. Assuming the value set assigned to a work is its identifier, and if there is no work, the field will not be present in a request.
- Only a subject (healthcare providers) who is a member of the care team and is assigned the *action* team role can access *private* and *protected* resources, but only if needed (inevitably). In this model, we assume the healthcare provider who is assigned the *action* team role needs to access private resources because he/she needs to see a patient on a face-to-face basis to perform various tasks related to the patient's recovery. In this respect, there is a need for the healthcare provider

TABLE I. Tabular structure of policy data

Subject	Job Function	Team Role	Object Type	Action	Permission
Dean	Primary Doctor	Main role	Private and protected	Read/write	Permit
Bob	General practitioner	Action	Private and protected	Read	Permit
Cara	Gastroenterologist	thought	Protected	Read	Permit
Alex	Medical coordinator	Management	Protected	Read	Permit

```
<!--
  Policy ensuring that the primary physician has clearance to access medical records
-->
<Policy PolicyId="team:manager:doctor:record:access:policy" RuleCombiningAlgId="
rule-combining-algorithm:permit- overrides">
 <Target>
  <Subjects>
  <Subject>
    <SubjectMatch MatchId=" string-equal">
      <AttributeValue DataType="string">doctor</AttributeValue>
      <SubjectAttributeDesignator DataType="string" AttributeId="subject:role"/>
    </SubjectMatch>
  </Subject>
  </Subjects>
 </Target>
 <Rule RuleId="isPrimaryDoctor" Effect="Permit">
  <Target/>
  <Condition>
   <Apply FunctionId=" string-equal">
    <Apply FunctionId=" string-one-and-only">
     <AttributeSelector
     RequestContextPath="//Resource/ResourceContent/record/patient/physician"
     DataType="string"/>
    </Apply>
    <Apply FunctionId=" string-one-and-only">
     <SubjectAttributeDesignator AttributeId="subject:id" DataType="string"/>
    </Apply>
  </Apply>
  </Condition>
 </Rule>
</Policy>
```

Figure 11. Policy structure for main team role

to know personal and medical information about the patient to perform his/her duty effectively. Figure 12 presents part of XACML policy for *action* team role. It can be seen that a team member who is assigned to the *action* team role (e.g., *Bob*) is allowed access (read only) on both the personal and medical information of the patient (*private* and *protected* resources). Note that in other scenarios, a healthcare provider who is assigned the *action* team role might not need to know private information about the patient.

- Only a subject (healthcare providers) who is a member of the care team and who is assigned to the *thought* team role can access *protected* resources, which are approved for collaboration works. This healthcare provider is predominantly preoccupied with diagnosing the disease, and there is no urgent need for him/her to know the patient's personal information. In fact, he/she is only required to analyze the medical situation and suggest a possible solution. Figure 13 displays a part of XACML policy structure for *thought* team role. In our model (Figure 13), personnel assigned the *thought* team role are permitted access only to *protected* resources (e.g., any resources related to the current case of the patient).
- Healthcare providers who are assigned the *management* team role are responsible for coordinating the other team members' interaction by managing meetings and resolving problems with conflicting diagnoses made by other team members. Figure 14 presents a part of XACML policy structure for *management* team role. The healthcare provider does not really need to know the patient's personal information. However, he/she must be aware of the patient's medical information to enable coordination (Figure 14). Similar to the *thought* team role, personnel assigned the *management* team role are permitted access only to *protected* resources. The difference between the *thought* and *management* team roles is the need for personnel assigned to the *management* team role to have access to team member (healthcare provider) records to be informed of specialist information related to the team members (physicians) in order to coordinate the collaborative work effectively.

```
<Policy PolicyId="actioneer:policy" RuleCombiningAlgId="
rule-combining-algorithm:permit-overrides">
     <VariableDefinition VariableId="WorkID">...</VariableDefinition>
 <Target>...</Target>
    <Rule RuleId="permitRead" Effect="Permit">
     <Target>
    <Resources>
       <!--
        Action collaborator. shall have access to protected journals of type:
    { personalInformation . medicalHistory . patientNote . treatmentSummary }
       -->
        <Resource>
         <ResourceMatch MatchId=" string-equal">
          <AttributeValue DataType="string">personalInformation</AttributeValue>
          <AttributeSelector DataType="string"
          AttributeId="//Resource/ResourceContent/record/type"/>
         </ResourceMatch>
        </Resource>
        <Resource>
         <ResourceMatch MatchId=" string-equal">
          <AttributeValue DataType="string">medicalHistory</AttributeValue>
          <AttributeSelector DataType="string"
          AttributeId="//Resource/ResourceContent/record/type"/>
         </ResourceMatch>
        </Resource>
        <Resource>
         <ResourceMatch MatchId=" string-equal">
          <AttributeValue DataType="string">patientNote</AttributeValue>
          <AttributeSelector DataType="string"
          AttributeId="//Resource/ResourceContent/record/type"/>
         </ResourceMatch>
        </Resource>
        <Resource>
         <ResourceMatch MatchId=" string-equal">
          <AttributeValue DataType="string">treatmentSummary</AttributeValue>
          <AttributeSelector DataType="string"
          AttributeId="//Resource/ResourceContent/record/type"/>
         </ResourceMatch>
        </Resource>
    </Resources>
     <Actions>
      <Action>
        <ActionMatch MatchId=" string-equal">
          <AttributeValue DataType="string">read</AttributeValue>
          <ActionAttributeDesignator DataType="string" AttributeId="action-id"/>
        </ActionMatch>
      </Action></A
          ctions>
     </Target>
       <Condition> <VariableReference VariableId="WorkID"/>
   </Condition>
</Rule> </Policy>
```

Figure 12. Policy structure for action team role

F. Request model

The XACML request contains the attributes related to subject, resource and action with their corresponding values. For example, in our case and as depicted in Figure 15, we

```
<Policy PolicyId="thought:policy" RuleCombiningAlgId=" rule-
combining-algorithm:permit-overrides">
  <VariableDefinition  VariableId="WorkID">...</VariableDefinition>
  <Target>...</Target>
  <RuleRuleId="protected:resource:rule"Effect="Permit">
    <Target>
   <Resources>
    Thought collaborator  shall have access to protected journals of
                    type: { medicalHistory . treatmentSummary }
      <Resource>
        <ResourceMatch  MatchId=" string-equal">
          <AttributeValue DataType="string">medicalHistory</AttributeValue>
          <AttributeSelector  DataType="string"
          AttributeId="//Resource/ResourceContent/record/type"/>
        </ResourceMatch>
      </Resource>
      <Resource>
        <ResourceMatch  MatchId="string-equal">
          <AttributeValue DataType="string">treatmentSummary</AttributeValue>
          <AttributeSelector  DataType="string"
          AttributeId="//Resource/ResourceContent/record/type"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>
        <ActionMatch MatchId="string-equal">
          <AttributeValue DataType="string">read</AttributeValue>
          <ActionAttributeDesignator DataType="string"AttributeId="action-id"/>
        </ActionMatch>
      </Action>
    </Actions>
    <Environments>
    <Environment>
    <EnvironmentMatch MatchId="string-equal">
    <AttributeValue DataType="string"> Hospital.A.Domain </AttributeValue>
    <AttributeSelector DataType="string"
    </EnvironmentMatch>
  </Environment>
 </Environments>
    </Target>
    <Condition>
      <VariableReference  VariableId="WorkID"/>
   </Condition>
  </Rule>
</Policy>
```

Figure 13. Policy structure for thought team role

```
    <Policy  PolicyId="Management:policy"  RuleCombiningAlgId=" rule-combining-
    algorithm:permit-overrides">
      <VariableDefinition   VariableId="WorkID">...</VariableDefinition>
      <Target>
        <Subjects>...</Subjects>
      </Target>
      <Rule RuleId="managment"Effect="Permit">
        <Target>
        <Resources>
            <!--
    Management collaborator shall have access to protected journals of type:
    { medicalHistory .  treatmentSummary, Doctors information }
            -->
            <Resource>
              <ResourceMatch   MatchId=" string-equal">
                <AttributeValue DataType="string">medicalHistory</AttributeValue>
                <AttributeSelector DataType="string"
                AttributeId="//Resource/ResourceContent/record/type"/>
              </ResourceMatch>
            </Resource>
            <Resource>
              <ResourceMatch   MatchId=" string-equal">
                <AttributeValue DataType="string">treatmentSummary</AttributeValue>
                <AttributeSelector DataType="string"
                AttributeId="//Resource/ResourceContent/record/type"/>
              </ResourceMatch>
            </Resource>
          </Resources>
          <Actions>
            <Action>
              <ActionMatch   MatchId=" string-equal">
                <AttributeValue   DataType="string">read</AttributeValue>
                <ActionAttributeDesignator  DataType="string"AttributeId=" action:action-id"/>
              </ActionMatch>
            </Action>
          </Actions>
        </Target>
        <Condition>
          <VariableReferenceVariableId="WorkID"/>
        </Condition>
      </Rule>
    </Policy>
  </PolicySet>
</PolicySet>
```

Figure 14. Policy structure for management team role

have attribute *Subject:Role* and its value *General practitioner*, and attribute *ResourceClassification* and its value *protected* as well as an action value *write*. This information is necessary for authorization decision-making. When PDP evaluates the request against the policy, the attribute names and attribute values are compared according to criteria defined in the policy.

```
<Request>
  <Subject>
    <AttributeAttributeId="subject:id"DataType="string">
      <AttributeValue>Bob</AttributeValue>
    </Attribute>
    <AttributeAttributeId="subject:role"DataType="string">
      <AttributeValue>General  practitioner</AttributeValue>
    </Attribute>
    <Attribute  AttributeId="subject:collaboration:work"  DataType="string">
      <AttributeValue>1</AttributeValue>
    </Attribute>
    <Attribute  AttributeId="subject:collaboration:role"  DataType="string">
      <AttributeValue>action</AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <ResourceContent>
      <record>
        <patient>
         <physician>Dean</physician>
          <work>1</work>
        </patient>
        <classification>protected</classification>
      </record>
    </ResourceContent>
    <AttributeAttributeId="resource-id"DataType="string">
      <AttributeValue>patientRecord</AttributeValue>
    </Attribute>
  </Resource>
  <Action>
    <AttributeAttributeId="action-id"DataType="string">
      <AttributeValue>write</AttributeValue>
    </Attribute>
  </Action>
</Request>
```

Figure 15. Example of an XACML access request

G. *Policies and policy sets model*

The XACML collaboration model begins with a top-level policy set containing one policy for handling a case where the subject is the patient's primary physician and a policy set for the different collaboration cases as shown in Figure 16.

```
PolicySetId="patient-collaboration"PolicyCombiningAlgId="
policy-combining-algorithm:first-applicable">
  <Target>...</Target>
  <!--
     Policy ensuring that the primary physician has clearance to access medical records
  -->
  <Policy PolicyId="team:manager:doctor:record:access:policy" RuleCombiningAlgId="
  rule-combining-algorithm:permit-overrides">...
  </Policy>
  <!--   CollaborationPolicies   -->
  <PolicySet PolicySetId="collaboration:policy:set" PolicyCombiningAlgId="
  policy-combining-algorithm:deny-override">
    <PolicyDefaults>...</PolicyDefaults>
    <Target>...</Target>
    <Policy PolicyId="thought:policy" RuleCombiningAlgId=" rule-
    combining-algorithm:permit-overrides">...</Policy>
    <Policy PolicyId="actioneer:policy" RuleCombiningAlgId=" rule-
    combining-algorithm:permit-overrides">...</Policy>
    <Policy PolicyId="Management:policy" RuleCombiningAlgId=" rule-
    combining-algorithm:permit-overrides">...</Policy>
  </PolicySet>
</PolicySet>
```

Figure 16. Screenshot of top-level policy set

The top-level policy combines the results based on first applicability, meaning that if the requesting subject is the patient's primary doctor, he/she will get access to records regardless of collaboration. PDP will receive all policies as inputs, where each policy has an element known as "target" (described in Section III-A). As depicted in Figure 17, the target element's attribute values (subject, resource, action and environment) are matched with the incoming request (Figure 15) attribute values to decide whether a particular policy is

applicable to a given request. If the request attributes match the target's attributes, the policy will be evaluated further. Else, PDP decides the given request is not applicable to the policy.

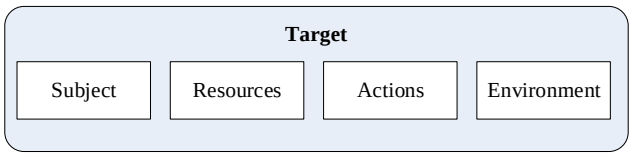


Figure 17. Target element

Within the subject element, XACML uses a sub-element called "subjectMatch" (Figure 18) to define matching criteria for policy. A Subject match element contains two parameters; attribute name and attribute value which are used to compare attribute value with the relevant data type in the policy. XACML engine also uses a sub-element called "SubjectAttributeDesignator" (Figure 18) to look for values from the XACML request related to attribute values from incoming subject (in request). Similarly to "SubjectAttributeDesignator", "ResourceAttributeDesignators" will be used to look for resource in XACML request and "ActionAttributeDesignators" will be used to look for action in the XACML request. The same pattern is applied to "EnvironmentAttributeDesignators".

```
<!--
   Policy ensuring that the primary physician has clearance to access medical records
-->
<Policy PolicyId="team:manager:doctor:record:access:policy"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <PolicyDefaults>
    <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
  </PolicyDefaults>
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch   MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
          DataType="http://www.w3.org/2001/XMLSchema#string">doctor</AttributeValue>
          <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
          AttributeId="subject:role"/>
        </SubjectMatch>
      </Subject>
    </Subjects>
  </Target>
  <Rule RuleId="isPrimaryDoctor" Effect="Permit">
    <Target/>
    <Condition>
      <Apply   FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <Apply   FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          <AttributeSelector
          RequestContextPath="//Resource/ResourceContent/record/patient/physician"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply>
        <Apply   FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          <SubjectAttributeDesignator AttributeId="subject:id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>
```

Figure 18. Sample policy with subject match element

Assuming the subject element in the XACML request (Figure 15), the XACML engine will evaluate the target element by first building an XACML evaluate function contain an *"AttributeId"* and *"AttributeValue"* and de-reference value in "SubjectAttributeDesignator" after matching metadata in policy and XACML request and retrieve appropriate values from request.

The attributes element contains attributes of the entity making the access request. There can be multiple subjects in the form of additional attributes elements with different categories, and each subject can have multiple attributes. In our case (Figure 15), there is only one subject, and the subject has number of attributes. An example of the subject's attribute is subject's identity, expressed as a name ("*Bob*"). Resource element represents the actual resource which subject is trying to access. In the Figure 15 there is an attributes element contains attributes of the resource to which the subject *Bob* has requested access. The resource identified by its classification, which is *protected*. Action element represents subject's activity on a resource (e.g., read and write). An attributes element contains attributes of the action that the subject *Bob* wishes to perform on the resource which is "write". The PDP processing this request context locates the policy in its policy repository. It compares the attributes in the request context with the policy target. The PDP now compares the attributes in the request context with the target of the one rule in this policy.

Figure 11 displays an example of a policy ensuring that the primary physician has clearance to access medical records. While the target element evaluates the applicability of a policy, the rule element implements the actual authorization logic. The primary physician policy has one rule as demonstrated also in Figure 11, which permits access. If the rule's condition is evaluated as true, the output of the rule will be "permit" where the primary physician field in the resource content patient metadata the same identifier for the subject. Condition is a *Boolean* expression (true or false) that refines the applicability of the rule beyond the predicates implied by its target. The effect of rule indicates the outcome of the rule based on the condition evaluation. Two values are allowed: "permit" and "deny".

Collaboration policies are divided into three sub-policy sets from the main policy set, as shown in Figure 16. Each policy set is for one specific team role and the rule that applies to this team role. To evaluate collaborative work, the subject *workID* is matched with that of the resource and must be equal for access to be granted and combined with other constraints, such as *read* or *write* effect. An instance of one collaboration policy is shown in Figures 12, 13 and 14. In figure 13 for example, the subject assigned the *thought* team role is granted access (read access only) to the *protected* resource type if the *workID* matches the active *workID*.

H. Experiments and result

The WBAC model has been implemented using XACML 2.0. Verifying that this implementation of WBAC can be used as part of an XACML policy was done using the Java SunXACML implementation [62] to run a PDP, testing the policy against different requests. Sun's XACML Implementation was originally created in Sun Microsystems Research Laboratories by members of the Internet Security Research Group. It provides complete support for all the mandatory features of XACML and a number of optional features. It also provides support for parsing both policy and request/response documents, determining applicability of policies, and evaluating requests against policies. There are APIs for adding new functionality as needed and writing new retrieval mechanisms for finding things like policies and attributes. All of the standard attribute's types, functions, and combining algorithms are supported [62].

In our experiment, we assume that the PDP is configured to be deny-based which means that any response which is *indeterminate* or *not applicable* is seen as a *deny* response. The WBAC policy was tested by using the attributes based on the data models shown in Figure 10(a) and Figure 10(b) to build access control requests as shown in Figure 15. Both valid and invalid values were set for the different attributes to verify that access was permitted and denied correctly.

The experiments showed that the WBAC model granted access correctly to subjects matching the same work as the resource for the expected cases. Invalid request such as a subject *work* with the value 2, while the resource *work* value set to 1. Since the policy is only implemented with rules needed for permitting access when requests is matched the PDP responded with a *indeterminate* answer, which is interpreted as a deny response when the PDP is deny-based.

IV. WBAC AUTHORIZATION FRAMEWORK

In WBAC, users obtain privileges through roles and team roles. The decision function (PDP) makes a decision for a request permission based on the authorized role and authorized team role. If a role is assigned to a user and is activated, the user will get all permissions associated with the active role. As for team role, the permission a user will get is based on which team he/she is a member of and his/her authorized team role in that team as well as whether the collaborative work is active or not. As shown in the request model (Figure 15), the request should contain all information (attributes) about the user, operation and object including the user's authorized role and authorized team role.

WBAC enables determining if the user, once identified, is permitted to access the resource. According to Figure 19, WBAC is a combination of authentication and authorization processes aimed at managing and securing access to system resources while also protecting resource confidentiality and integrity, among others.

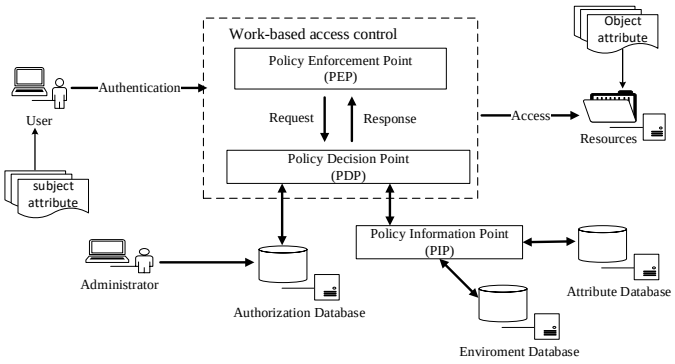


Figure 19. Access control mechanism for WBAC [32, p. 185]

Authentication entails validating the identity establishment between two communicating parties, showing what or who the user is. Authorization checks if the user can access the resources he/she has requested. When a user requests access to a system resource, the user must first authenticate him/herself to the system. In our work [22, 32], we proposed an attribute-based authentication (ABA) scheme, which is a way to authenticate users by attributes or their properties. Second, the WBAC authorization process decides to permit or deny the access request based on the authorization policies. PEP intercepts a user's request to access an object and then forwards the request to PDP to obtain the access decision (permit or deny). PDP receives the request from PEP and combines the user with the object information (attribute value described in Section III), then checks if they satisfy the authorization policies (Figure 19). If so, the subject's access request is granted and will be enforced by PEP.

A. Evaluation process and decision-making

Figure 20 presents a sequence diagram of the authorization evaluation process for the WBAC model. When a user sends an access request (Figure 15) to perform an operation on an object, PEP intercepts the call request and forwards it to PDP (access decision function) to check whether the user has permission to perform the requested operation on the object. The authorization system decides if the user has permission to carry out the requested operation by checking three layers: the first RBAC layer, the secondary RBAC layer and the ABAC layer.

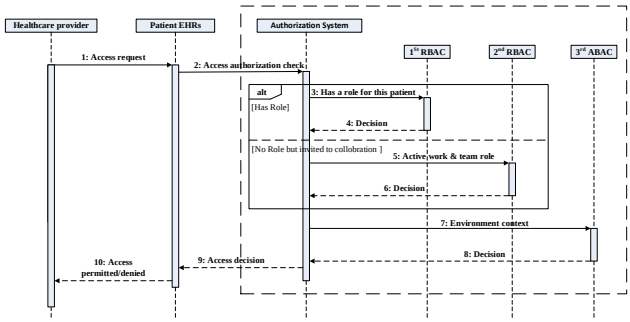
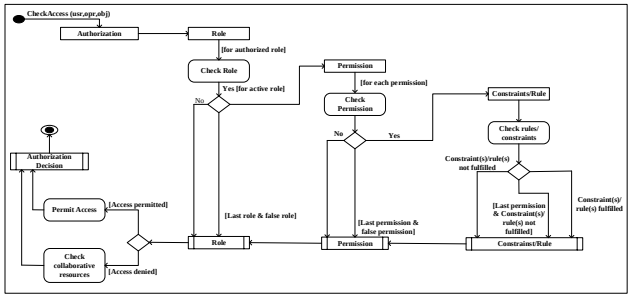
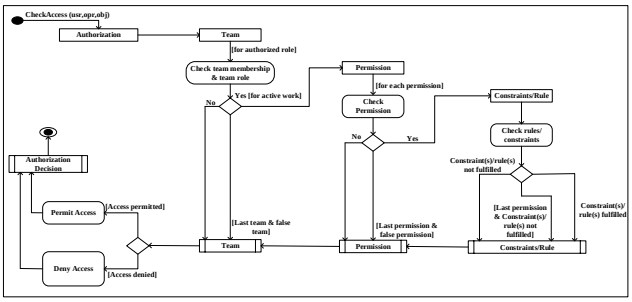


Figure 20. Sequence diagram of authorization process

The entire authorization process is shown in Figure 21. The authorization system is responsible for making an authorization decision on an access request by checking if the access request should be permitted or denied. The access checking operation starts with gathering all attribute values in the access request (e.g., user role, object, and operation attributes) followed by checking the user's state – whether the user is in the user set. If the user is active, the checking process continues with a role check, team role check, and permission check, otherwise the checking process stops and returns the value "no".



(a) Activity diagram of the role check authorization process



(b) Activity diagram of the team role check authorization process

Figure 21. Activity diagrams of the WBAC authorization process

The role check process (Figure 21(a)) performs a role lookup to check if the role is assigned to the respective user. Only when the user is assigned the role the check process continues with the permission check, otherwise it stops and returns the value "no" and the check access operation investigates the collaborative resources (Figure 21(b)). The permission lookup process checks whether the requested operation on the respective object is assigned to the corresponding role and if the input request object is equal to the permission object. If the requested operation is permitted by the role, the check access operation return "yes" and continues with the constraint and rule check on the ABAC layer. To provide a fine-grained access control, the third layer (ABAC) enforces extra constraints such as environment and context constraints. It is not sufficient to grant access only when the user holds the appropriate role.

In case the permission in the request is not assigned any role or the constraint check returns "no", the check access operation further investigates the collocation policy (Figure 21(b)). The check access operation checks the user memberships in a team and if permission is granted by the team role. If the request is permitted by the respective team role and the input "request object" is equal to a permission object, the check access operation returns "yes" and continues with the constraint and rule check on the ABAC layer; otherwise the check process stops and the access request is denied.

Consider *Alice*'s case presented in Section II-A with four healthcare providers: *Dean*, *Bob*, *Cara*, *Alex*. If *Dean* sends a request to read *Alice*'s file in *Alice*'s *private* objects, the check access operation checks if the permission (e.g., read *Alice*'s private object) is assigned to *Dean*'s role (primary doctor). Based on the our defined policy (Table I), *Dean* is assigned the primary doctor role and the permission (read *Alice*'s private object) is assigned to the primary doctor role. Therefore, based on the role and permission checks, *Dean* is permitted to perform the operation "read" on *Alice*'s *private* objects. However, granted access based on an appropriate role is not sufficient. Thus, WBAC facilitates more fine-grained access by checking the third layer (ABAC) for additional constraints, for example if *Dean* is permitted to read a file form a certain location at a particular time. In *Dean*'s case, the authorization system checks only the main policy set, where the requesting subject is the patient's primary doctor.

If *Bob* sends a request to access *Alice*'s EHRs, the access policy (Table I) shows that *Bob* is assigned a general practitioner role, but based on the permission check, permission (read *Alice*'s private object) for example is not assigned to the general practitioner role; hence, the permission check returns "no" and the check access operation continues checking the collaboration resources (Figure 21(b)). In our model, it is assumed that Bob joined *Alice*'s treatment team and is assigned an *action* team role. Therefore, *Bob* is a member of the team and holds an *action* team role. The team check returns "yes" and the check access operation continues with permission checking. Permission (read *Alice*'s private object) is assigned to the *action* team role, thus *Bob* is permitted to read *Alice*'s *private* objects.

V. Validation of the Proposed WBAC Model

In this section, we present a validation of WBAC to ensure that WBAC strikes a balance between collaboration and safeguarding sensitive patient information.

A. Informal Validation of WBAC

This informal validation examines the core functions of access control models [10]. The core function as following:

- **Initiation of collaborative work**: the process of initiating the collaborative work (discussed in Section III-C).
- **Policy structure**: Policy is a statement of what is, and what is not allowed and policy structure is a procedure for enforcing the policy in the system (discussed in Section III-G).
- **Alteration of policy for collaborative work**: the process of altering access control policies by WBAC model to meet the requirements of the organization and collaborative work. Consider again the case of *Alice* of which *Cara* plays a *thought* team role in deciding the best treatment for *Alice*'s case. Since *Cara* does not need to see the patient on a face to face basis, she often contemplates upon the decision making from her local hospital (we assumed that *Care* is invited from hospital B). This implies that the shared resources for the *thought* team role are not accessed at the hospital. The alteration that enables the aforementioned scenario can be seen below (Figure 22). Observe that the first rule (Figure 13) allows anyone from the *thought* team role to read the shared information locally (e.g., in hospital A). On the other hand, the second rule alters the former policy. The physician with *thought* team role can access the shared resources from other location (e.g., in hospital B). In both cases however, only the read access is given. Also, the modification will be done in the collaboration policy set. There is no need to modify any policy in the main policy set.

```
<Policy PolicyId="thought:policy" RuleCombiningAlgId="rule-
combining-algorithm:permit-overrides">
 <PolicyDefaults>...</PolicyDefaults>
 <VariableDefinition VariableId="WorkID">...</VariableDefinition>
 <Target>...</Target>
 <Rule RuleId="protected:resource:rule" Effect="Permit">
  <Target>
   <Resources>...</Resources>
   <Actions>...</Actions>
   <Environments>
   <Environment>
   <EnvironmentMatch MatchId="string-equal">
   <AttributeValue DataType="string"> Hospital.A.Domain </AttributeValue>
   <AttributeSelector DataType="string"
   </EnvironmentMatch>
   </Environment>
   <Environment>
   <EnvironmentMatch MatchId="string-equal">
   <AttributeValue DataType="string"> Hospital.B.Domain </AttributeValue>
   <AttributeSelector DataType="string"
   </EnvironmentMatch>
   </Environment>
   </Environments>
  </Target>
  <Condition>...</Condition>
 </Rule>
</Policy>
```

Figure 22. Policy structure that involves alteration

- **Alteration of permission for collaborative work**: the process of altering assigned permissions to subjects to access an resource. The permission of accessing the resources that are related to the collaborative work is reliant on the given team roles. This could change dynamically. For instance, supposed that *Dean* is answering a compelling medical emergency call that forces him to leave the country. To ensure the

fluidity of the collaborative work, he promotes *Bob* as the *main* team role. With the new team role, *Bob* is given much greater control over the collaboration. Therefore, there is a need to change the permission to reflect the new role more accurately. This is simply done by altering the *action* team role that was initially defined for *Bob* to the *main* team role as shown below (Figure 23). The change only affects this particular collaborative work and nothing else.

Personnel	Team Role
Bob	Action

⇩

Personnel	Team Role
Bob	Main

Figure 23. Alteration of permission

- **Termination of collaborative work**: the process of deleting all assigned permissions to collaborative Work. Via successful collaboration, the right diagnosis for *Alice* is obtained. After receiving the required treatment, *Alice* is now fully recovered and left the hospital. The collaboration between *Dean*, *Bob*, *Cara* and *Alex* is no longer needed. Subsequently, *Dean* completes the final report for *Alice* and withdraws the collaborative work. Now, supposed that in the future, if *Bob* is inclined to review the diagnosis, then he must request for access again. When the owner of the collaborative work deletes or withdraws the project at hand (Figure 24), all the access to the shared resources, including those that contain the medical or personal information of the patient are revoked. The *workID* that is tied to their access is therefore deleted. Deletion may entail an exhaustive search by the system to guarantee complete removal of access to shared parties. In effect, the other collaborators will cease to have access over the information related to the work. A timestamped log entry of when a work participant entered the work flow should be made, and a corresponding timestamp of when the *work* was completed (which is when the work rights were revoked).

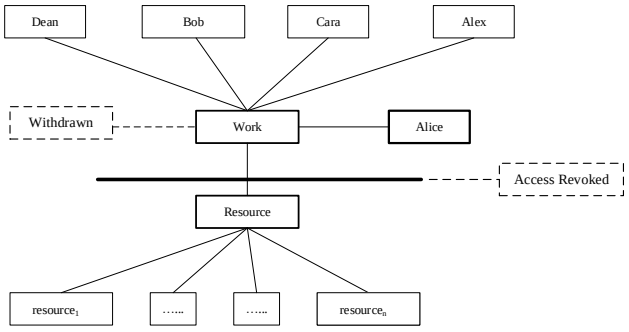


Figure 24. Work withdrawn to terminate collaboration

B. Comparing WBAC with the existing solutions

Researchers have made the best effort to propose an access control model that balance between security and collaboration requirements [23, 63, 64]. A numerous of research trends on access control approaches have been presented such as RBAC, ABAC, team-based access control (TMAC) [46], task-based access control (TBAC) [65], context-based TMAC (C-TMAC) [66], team task based RBAC (TT-RBAC) [67] and group-based RBAC (GB-RBAC) [68]. In this section, we compare them to understand better the differences between these approaches. Comparison is imperative and aims at well defining the appropriate access control model for our model. The main evaluation criteria for access control in collaborative system were presented in number of studies [45, 64]. The assessment criteria with respect to healthcare collaborative environments as follows:

1) **Personalized permission**: Patients must be informed of the collaboration and should be given the right to choose who can have access to their records.
2) **Selective confidentiality**: Certain patient information is highly sensitive. Thus, patients should be able to withhold information that remains confidential.
3) **Flexibility and adaptability**: Flexibility is the access control model's ability to support frequent changes in policy, whereas adaptability is used to evaluate the access control's ability to adapt to different healthcare scenarios and environments.
4) **Fine-grained control**: The access control model should support fine-grained subjects, objects and access rights. This is a granular level at which rules can be applied not only to roles but also to individuals regarding one or many controlled objects [64].
5) **Groups of users: assignment and revocation**: in collaborative work, common tasks are undertaken by a group of people (a team). Therefore, an access control model supports the team's notion and facilitates specifying access rights for teams. Also, the model should have the capability to revoke subjects' access rights to objects.
6) **Policy specifications and maintenance**: The access control model should allow for scalability and easy extension and modification of subjects' access rights to objects. Also. it should provide means of ensuring correct enforcement of the policy or constraint specification.
7) **Design for collaborative healthcare systems**: This criterion indicates whether the access control solution was designed specifically for collaborative healthcare systems.

Table II summarizes our comparative analysis of the RBAC, ABAC, TMAC, TBAC, C-TMAC, TT-RBAC, GB-RBAC and WBAC models. It can been seen that WBAC meets the requirements of collaborative healthcare environments better than the other models. The WBAC model solves the problems of personalized permission and selective confidentiality, whereby, as described above, access to objects is controlled based on the classification of teams into three classes according to the team members' tasks they will carry out in the collaborative work. RBAC, TT-RBAC, GB-RBAC, TMAC and other models do not consider team classification. They deal with all teams in the same way, which can confuse

TABLE II. Comparative analysis of the RBAC, ABAC, TMAC, TBAC, C-TMAC, TT-RBAC, GB-RBAC and WBAC models

Access Control models	Assessment Criteria						
	1	2	3	4	5	6	7
MAC	No	High	Low	High	No	Complex	No
DAC	Yes	Low	Low	Low	No	Complex	No
RBAC	No	Low	Medium	Low	Yes	Simple	Yes
ABCA	No	Yes	High	High	Complex	Complex	No
TMAC	No	Low	High	Yes	Yes	Simple	No
TBAC	No	Medium	Low	High	No	Complex	No
C-TMAC	No	Low	High	Yes	Yes	Complex	No
TT-RBAC	No	Medium	High	Medium	Yes	Complex	Yes
GB-RBAC	No	High	Medium	Low	Yes	Complex	No
WBAC	Yes	High	High	High	Yes	Simple	Yes

security administrators and object owners. In WBAC, the patient will be informed about the team formation and to what information each team member will get access based on the assigned team role. WBAC supports selective confidentiality well because it is possible to assign a specific object to each member in a given team based on the object and team role classifications.

Considering TBAC and TT-RBAC, tasks in healthcare environments usually have their own (different) characteristics and it is difficult to establish in advance access based on tasks. For instance in *Alice*'s case, it is hard to identify what task *Bob* has. In the WBAC model, as *Bob* is assigned the *action* team role, he would have all tasks related to preparing *Alice* for operation. Examples of *Bob*'s tasks are laboratory work (e.g., taking all blood tests required for the operation) and physical examination (e.g., physical examination based on gathered information related to past and current medical history, surgical history, family history, social history (use of tobacco, alcohol and illegal drugs), history of allergies, and current and recent drug therapy [51] to name a few). *Cara* is assigned the *thought* team role. Therefore, her tasks might be for example preoperative risk assessment (e.g., function of the patient's preoperative medical condition) and treatment recommendations after surgery (e.g., pain management post-op [69]). In these cases, access privileges are assigned to healthcare providers according to their team roles and not their tasks. Holding a team role would allow healthcare providers to access multiple information (based on the selective confidentiality requirement), which would allow them to work on multiple tasks related to the patient's treatment. Thus, healthcare providers assigned to the team would be permitted to access the selected objects required for performing their duties.

In terms of fine-grained control, WBAC focuses on the user's role, user's team roles and target object; therefore, it can be said WBAC is classified as fine-grained access control. WBAC reduces over-privilege access arising from frequent specifications when using role in RBAC by classifying the team and objects. The level of fine-grained control access (granularity) to objects that can be authorized to healthcare providers is managed and controlled based on individual scenarios (active work, which is the patient's treatment). Although fine-grained control is very complicated in healthcare environments, WBAC's policy can be implemented using XACML, and the more information that is considered to define a rule, the finer-grained the resulting access control will be. XACML can specify rules in terms of attribute values (e.g., attributes about users, resources, actions, and the environment) that can be of various types, such as strings and integers (Section III-D), making WABC very fine-grained.

WBAC supports an easy means of adding, changing, manipulating, and specifying a team of users. Regarding groups of users, assignment and revocation are similar to TMAC, C-TMAC and TT-RBAC, except that in WBAC the team is classified based on team role. Moreover, in WBAC, a team can be assigned to a collaborative work at any granularity based on the team members' team roles. In general and as explained in [70], using the concept of role in RBAC and its extension greatly reduce the management complexity of user assignment and revocation. Thus, employing the team role concept in WBAC helps solve the problem of user assignment and revocation in the case of team work.

Policy specification and policy enforcement in WBAC are the same as in RBAC. WBAC supports means of specifying and managing policies as well as using appropriate policy languages such as XACML (Section III), which allows extensions or modifications in a simple and transparent manner. The proposed dual policy [24] is to ensure system scalability, especially in collaborative environments, where governance policies require different organizational entities to have different responsibilities for administering various aspects of policies and their dependent attributes.

WBAC has a number of advantages including flexibility in terms of permission administration management, since roles and team roles can be updated without updating permissions for every user. Moreover, it is fairly easy to assign and revoke users based on their roles and team roles. We believe that WBAC handles personalized permissions well and meets our expectation of allowing fine-grained access control, and it enhances the practicability and manageability of access control in dynamic collaboration environments.

VI. Discussion and Conclusions

In this section, discussions, conclusions and future works are presented.

A. Discussion

To prevent any violation of the access control policy of an organization, most classical access control models like RBAC and ABAC define users rights precisely, based on subject and object elements. When several subjects and objects are involved, the subject-object model cannot deliver satisfactory security management. In collaborative environments such as healthcare, it is challenging to predefine all access needs based on the subject-object model. One example of such a situation is explained in our case scenario (Section II-A), which may not

be predictable and it would be hard to express the condition of who should join the collaboration and when *Dean* necessitates collaborative support from other parties. Moreover, in deciding on the extent and limit of resource sharing, For instance, in the case of *Alice*'s treatment, which sensitive data should be disclosed to an assisting practitioner so collaboration can be effective, and which should be hidden to safeguard the patient's privacy? Another important matter is the correctness of the policy. Access policy adoption may be limited if the intended policies are not implemented efficiently and consequently thus perform poorly.

WBAC was proposed to address these concerns and support the security and collaboration requirements in access control [23, 63, 64]. The major contributions of the WBAC model include ensuring that access rights are dynamically adapted to the actual needs of healthcare providers and providing fine-grained control of access rights with the least privilege principle, whereby healthcare providers are granted minimal access rights to carry out their duties. In our case scenario, it was noted that general practitioner *Dean* could not solve Alice's case alone. He invited a multidisciplinary team including *Bob*, *Cara* and *Alex* to help. In this team, *Dean* is the core physician in the collaborative work and servers as the group manager. He is responsible for initiating the *work* (*Alice*'s treatment case) and choosing practitioners (group of doctors) who may be required to attend *Alice*'s consultation and treatment. This implies that *Dean* holds the main role. In other words, he owns the initiated collaborative work. Therefore, *Dean* is given a full access (based on his role as primary physician, Figure 11) with regard to patient-related information. *Bob*, *Cara* and *Alex* are assigned to team roles based on the job function they will perform in *Alice*'s treatment. In our previous work [21], we formally describe and showed how each user joins the team and how each should be assigned at least one team role; a team role can be assigned to none or multiple users in many teams.

In this study, we showed how XACML can be used to implement the WBAC model policy and how XACML combining algorithms can be used to manage the inconsistencies between different policy sets. We selected XACML because it has been proven to be adaptable to specifying several common access control methods, such as RBAC and ABAC. Moreover, XACML has become very popular in both academia and industry as a standard for combining, maintaining and exchanging access control policies. It is an architecture for evaluating authorization requests and for issuing authorization decisions. The experiments we conducted demonstrated the applicability of XACML to supporting collaborative and distributed domains in sharing access control of specific resources. However, It still come with some limitation in the expressive power of higher-order logic such as the expressions of separation of duty (SoD) constraints and domain constraints.

Our implementation only covers access request for medical records resources, but by using similar matching technique as for the *work* attribute, it is possible to extend this to other polices that are also active during collaboration. An example of this could be for persons with the $management$ team role, which should also have access to the personal files like those in the same collaborative work team.

XACML offers extensibility and pluggability which enables the policy presented in this work to be not only a standalone policy, but it could also be a small part of a larger collection of policies. Possible extensions of the base collaboration policy could, for example, by sub-roles (Figure 4) of each primary collaboration roles. This could give even more granularity for specific cases for example if a medical employee in the $management$ team role.

B. Conclusions and Future work

The WBAC model was proposed by introducing the team role concept and modifying the user role assignment model from RBAC and ABAC works. The team role of each team member will subsequently determine the extent of access given. Moreover, the level of fine-grained control of access (granularity) to objects that can be authorized to healthcare providers is managed and controlled based on the job required.

The WBAC model utilizes role, team role and WBAC policies to perform an access control evaluation process. First, it checks the access request to verify whether the requesting user possesses a valid role specified in the system. If the requesting user holds the right role, WBAC will check the permission associated with the role and then inspect the rule(s) within the main WBAC policies for additional constraints on access. In other models such as RBAC, failure in this stage results in the complete termination of the decision process. WBAC, however, treats this differently. If the requesting user does not hold a valid role (in most cases, the requesting user might be an outsider who is invited to collaborative work and does not hold a role in the organization), WBAC investigates further to determine whether the requesting user is part of the collaborative work. If so, the respective user's team role is extracted and examined for whether the requesting user possesses a valid team role over the resource. WBAC also checks the permission associated with the team role and checks the rule(s) within WBAC collaborative policies for additional constraints on access.

In the future, the plan is to develop and prototype the WBAC functionality to understand the possible difficulties in managing the model during actual implementation; model performance validity could also be evaluated in terms of resource consumption, e.g., time and computational capability. In additional, future research is required to incrementally develop additional types of constraints and policies, to further investigate how the WBAC and access delegation can be enriched to support the various needs on information access management in case of emergency (break-glass policy [71–73]), and to examine the generalizability of the enhanced WBAC model for other applications in healthcare environments such as clinical education and biomedical research.

ACKNOWLEDGMENT

The authors would like to thank Geir M. Køien for the support in investigating and typesetting this work.

REFERENCES

[1] M. Abomhara and H. Nergaard, "Modeling of work-based access control for cooperative healthcare systems with xacml," in Proceedings of the Fifth International Conference on Global Health Challenges (GLOBAL HEALTH 2016), 2016, pp. 14–21.

[2] A. C. Myers and B. Liskov, A decentralized model for information flow control. ACM, 1997, vol. 31, no. 5.

[3] D. Hedin and A. Sabelfeld, "A perspective on information-flow control," in Proceedings of the 2011 Marktoberdorf Summer School, 2011.

[4] M. Abomhara and G. M. Køien, "Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks," Journal of Cyber Security, vol. 4, pp. 65–88.

[5] P. Samarati and S. C. de Vimercati, "Access control: Policies, models, and mechanisms," in International School on Foundations of Security Analysis and Design, vol. 2171. Springer, 2001, pp. 137–196.

[6] P. Li, Y. Mao, and S. Zdancewic, "Information integrity policies," in In Proceedings of the Workshop on Formal Aspects in Security & Trust (FAST. Citeseer, 2003.

[7] P. K. Sinha, G. Sunder, P. Bendale, M. Mantri, and A. Dande, Electronic health record: standards, coding systems, frameworks, and infrastructures. John Wiley & Sons, 2012.

[8] J. H. Carter, Electronic health records: a guide for clinicians and administrators. American College of Physicians (ACP) Press, 2008.

[9] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," Computer, no. 2, 1996, pp. 38–47.

[10] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed nist standard for role-based access control," ACM Transactions on Information and System Security (TISSEC), vol. 4, no. 3, 2001, pp. 224–274.

[11] D. Ferraiolo, D. R. Kuhn, and R. Chandramouli, Role-based access control. Artech House, 2003.

[12] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (abac) definition and considerations," NIST Special Publication, vol. 800, 2014, p. 162.

[13] D. Ferraiolo, R. Chandramouli, V. Hu, and R. Kuhn, "A comparison of attribute based access control (abac) standards for data service applications:extensible access control markup language (xacml) and next generation access control (ngac)," NIST Special Publication (800-178), vol. 800, no. 178, 2016. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-178.pdf

[14] M. H. Kang, J. S. Park, and J. N. Froscher, "Access control mechanisms for inter-organizational workflow," in Proceedings of the sixth ACM symposium on Access control models and technologies. ACM, 2001, pp. 66–74.

[15] S. Oh and S. Park, "Task–role-based access control model," Information systems, vol. 28, no. 6, 2003, pp. 533–562.

[16] R. A. Shaikh, K. Adi, L. Logrippo, and S. Mankovski, "Inconsistency detection method for access control policies," in Information Assurance and Security (IAS), 2010 Sixth International Conference on. IEEE, 2010, pp. 204–209.

[17] X. H. Le, T. Doll, M. Barbosu, A. Luque, and D. Wang, "An enhancement of the role-based access control model to facilitate information access management in context of team collaboration and workflow," Journal of biomedical informatics, vol. 45, no. 6, 2012, pp. 1084–1107.

[18] W. Wang, "Team-and-role-based organizational context and access control for cooperative hypermedia environments," in Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots: returning to our diverse roots. ACM, 1999, pp. 37–46.

[19] C. E. Rubio-Medrano, C. D'Souza, and G.-J. Ahn, "Supporting secure collaborations with attribute-based access control," in Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on. IEEE, 2013, pp. 525–530.

[20] Q. Li, X. Zhang, M. Xu, and J. Wu, "Towards secure dynamic collaborations with group-based rbac model," computers & security, vol. 28, no. 5, 2009, pp. 260–275.

[21] M. Abomhara, H. Yang, and G. M. Køien, "Access control model for cooperative healthcare environments: Modeling and verification," in 2016 IEEE International Conference on Healthcare Informatics (ICHI). IEEE, 2016, pp. 46–54.

[22] M. Abomhara and H. Yang, "Attribute-based authenticated access for secure sharing of healthcare records in collaborative environments," in Proceedings of the Eighth International Conference on eHealth, Telemedicine, and Social Medicine (eTELEMED 2016), 2016, pp. 138–144.

[23] H. Shen and P. Dewan, "Access control for collaborative environments," in Proceedings of the 1992 ACM conference on Computer-supported cooperative work. ACM, 1992, pp. 51–58.

[24] M. Abomhara and G. M. Køien, "Towards an access control model for collaborative healthcare systems," in Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2016), vol. 5, 2016, pp. 213–222.

[25] M. Abomhara and M. B. Lazrag, "Uml/ocl-based modeling of work-based access control policies for collaborative healthcare systems," in 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom). IEEE, 2016, pp. 1–6.

[26] M. Abomhara, H. Yang, G. M. Køien, and M. B. Lazreg, "Work-based access control model for cooperative healthcare environments: Formal specification and verification," Journal of Healthcare Informatics Research, 2017, pp. 1–33. [Online]. Available: http://dx.doi.org/10.1007/s41666-017-0004-7

[27] R. M. Belbin, Team roles at work. Routledge, 2012.

[28] R. Meredith Belbin, "Management teams: Why they succeed or fail," Human Resource Management International Digest, vol. 19, no. 3, 2010.

[29] T. Moses et al., "Extensible access control markup language (xacml) version 2.0," Oasis Standard, vol. 200502, 2005.

[30] D. Boneh, E. Shen, and B. Waters, "Strongly unforgeable signatures based on computational diffie-hellman," in International Workshop on Public Key Cryptography. Springer, 2006, pp. 229–240.

[31] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in cryptographers Track at the RSA Conference. Springer, 2011, pp. 376–392.

[32] M. Abomhara and H. Yang, "Collaborative and secure sharing of healthcare records using attribute-based authenticated access," International Journal on Advances in Security Volume 9, Number 3 & 4, 2016, pp. 148–195.

[33] C. Borrill, M. West, D. Shapiro, and A. Rees, "Team working and effectiveness in health care," British Journal of Healthcare Management, vol. 6, no. 8, 2000, pp. 364–371.

[34] C. Taylor, A. J. Munro, R. Glynne-Jones, C. Griffith, P. Trevatt, M. Richards, and A. J. Ramirez, "Multidisciplinary team working in cancer: what is the evidence?" BMJ, vol. 340, 2010, p. c951.

[35] P. Mitchell, M. Wynia, R. Golden, B. McNellis, S. Okun, C. E. Webb, V. Rohrbach, and I. Von Kohorn, "Core principles & values of effective team-based health care," Washington, DC: Institute of Medicine, 2012.

[36] S. J. Dwyer III, A. C. Weaver, and K. K. Hughes, "Health insurance portability and accountability act," Security Issues in the Digital Medical Enterprise, vol. 72, no. 2, 2004, pp. 9–18.

[37] R. Nosowsky and T. J. Giordano, "The health insurance portability and accountability act of 1996 (hipaa) privacy rule: implications for clinical research," Annu. Rev. Med., vol. 57, 2006, pp. 575–590.

[38] R. Zhang and L. Liu, "Security models and requirements for healthcare application clouds," in Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. IEEE, 2010, pp. 268–275.

[39] B. Fabian, T. Ermakova, and P. Junghanns, "Collaborative and secure sharing of healthcare data in multi-clouds," Information Systems, vol. 48, 2015, pp. 132–150.

[40] F. T. Alotaiby and J. X. Chen, "A model for team-based access control (tmac 2004)," in Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on, vol. 1. IEEE, 2004, pp. 450–454.

[41] G. J. Annas, "Hipaa regulations–a new era of medical-record privacy?" The New England journal of medicine, vol. 348, no. 15, Apr 10 2003, pp. 1486–90. [Online]. Available: https://search.proquest.com/docview/223930472?accountid=45259

[42] J. L. Agris, "Extending the minimum necessary standard to uses and disclosures for treatment: Currents in contemporary bioethics," The Journal of Law, Medicine & Ethics, vol. 42, no. 2, 2014, pp. 263–267.

[43] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (rbac): Features and motivations," in Proceedings of 11th annual computer security application conference, 1995, pp. 241–48.

[44] M. A. Al-Kahtani and R. Sandhu, "A model for attribute-based user-role assignment," in Computer Security Applications Conference, 2002. Proceedings. 18th Annual. IEEE, 2002, pp. 353–362.

[45] S. Alshehri and R. K. Raj, "Secure access control for health information sharing systems," in Healthcare Informatics (ICHI), 2013 IEEE International Conference on. IEEE, 2013, pp. 277–286.

[46] R. K. Thomas, "Team-based access control (tmac): a primitive for applying role-based access controls in collaborative environments," in Proceedings of the second ACM workshop on Role-based access control. ACM, 1997, pp. 13–19.

[47] U. D. of Health, H. Services et al., "Hipaa privacy rule and sharing information related to mental health." [Online]. Available: http://www.hhs.gov/hipaa/for-professionals/special-topics/mental-health/

[48] R. Gajanayake, R. Iannella, and T. Sahama, "Privacy oriented access control for electronic health records," electronic Journal of Health Informatics, vol. 8, no. 2, 2014, p. 15.

[49] S. A. Wheelan, Creating effective teams: A guide for members and leaders. Sage Publications, Inc, 1999.

[50] C. Borrill and M. West, "Developing team working in health care: A guide for managers," Aston: Aston Centre for Health Service Organisation Research, Aston University, 2001.

[51] A. Zambouri, "Preoperative evaluation and preparation for anesthesia and surgery," Hippokratia, vol. 11, no. 1, 2007, pp. 13–21.

[52] K. Asif, S. I. Ahamed, and N. Talukder, "Avoiding privacy violation for resource sharing in ad hoc networks of pervasive computing environment," in Proceedings of the 31st Annual International Computer Software and Applications Conference-Volume 02. IEEE Computer Society, 2007, pp. 269–274.

[53] O. R. Kurkovsky, O. Rivera, and J. Bhalodi, "Classification of privacy management techniques in pervasive computing," International Journal of u-and e-Service, Science and Technology, vol. 11, no. 1, 2007, pp. 55–71.

[54] J. Thomas et al., "Medical records and issues in negligence," Indian Journal of Urology, vol. 25, no. 3, 2009, p. 384.

[55] M. A. Rothstein, "Hipaa privacy rule 2.0," The Journal of Law, Medicine & Ethics, vol. 41, no. 2, 2013, pp. 525–528.

[56] C. for Disease Control, Prevention et al., "Hipaa privacy rule and public health. guidance from cdc and the us department of health and human services," MMWR: Morbidity and mortality weekly report, vol. 52, no. Suppl. 1, 2003, pp. 1–17.

[57] U. D. of Health, H. Services et al., "Summary of the hipaa privacy rule," Washington, DC: Department of Health and Human Services, 2003.

[58] J. F. Alqatawna, E. Rissanen, and B. Sadighi, "Overriding of access control in xacml," in Policies for Distributed Systems and Networks, 2007. POLICY'07. Eighth IEEE International Workshop on. IEEE, 2007, pp. 87–95.

[59] A. X. Liu, F. Chen, J. Hwang, and T. Xie, "Designing fast and scalable xacml policy evaluation engines," Computers, IEEE Transactions on, vol. 60, no. 12, 2011, pp. 1802–1817.

[60] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone et al., "Guide to attribute based access control (abac) definition and considerations (draft)," NIST Special Publication, vol. 800, 2013, p. 162.

[61] N. Li, Q. Wang, W. Qardaji, E. Bertino, P. Rao, J. Lobo, and D. Lin, "Access control policy combining: theory meets practice," in Proceedings of the 14th ACM symposium on Access control models and technologies. ACM, 2009, pp. 135–144.

[62] S. Microsystems, "Official project web site sun's xacml implementation." [Online]. Available: http://www.sunxacml.sourceforge.net

[63] B. Alhaqbani and C. Fidge, "Access control requirements for processing electronic health records," in Business Process Management Workshops. Springer, 2008, pp. 371–382.

[64] W. Tolone, G.-J. Ahn, T. Pai, and S.-P. Hong, "Access control in collaborative systems," ACM Computing Surveys (CSUR), vol. 37, no. 1, 2005, pp. 29–41.

[65] R. K. Thomas and R. S. Sandhu, "Task-based authorization controls (tbac): A family of models for active and enterprise-oriented authorization management," in Database Security XI. Springer, 1998, pp. 166–181.

[66] C. K. Georgiadis, I. Mavridis, G. Pangalos, and R. K. Thomas, "Flexible team-based access control using contexts," in Proceedings of the sixth ACM symposium on Access control models and technologies. ACM, 2001, pp. 21–27.

[67] W. Zhou and C. Meinel, "Team and task based rbac access control model," in Network Operations and Management Symposium, 2007. LANOMS 2007. Latin American. IEEE, 2007, pp. 84–94.

[68] Q. Li, M. Xu, and X. Zhang, "Towards a group-based rbac model and decentralized user-role administration," in Distributed Computing Systems Workshops, 2008. ICDCS'08. 28th International Conference on. IEEE, 2008, pp. 441–446.

[69] T. Gould, D. Crosby, M. Harmer, S. Lloyd, J. Lunn, G. Rees, D. Roberts, and J. Webster, "Policy for controlling pain after surgery: effect of sequential changes in management." Bmj, vol. 305, no. 6863, 1992, pp. 1187–1193.

[70] A. Mohammad, T. Khdour, G. Kanaan, R. Kanaan, and S. Ahmad, "Analysis of existing access control models from web services applications perspective," J. Comput, vol. 3, 2011, pp. 10–16.

[71] A. D. Brucker and H. Petritsch, "Extending access control models with break-glass," in Proceedings of the 14th ACM symposium on Access control models and technologies. ACM, 2009, pp. 197–206.

[72] M. Hafner, M. Memon, and M. Alam, "Modeling and enforcing advanced access control policies in healthcare systems with sectet," in Models in Software Engineering. Springer, 2007, pp. 132–144.

[73] S. Marinovic, R. Craven, J. Ma, and N. Dulay, "Rumpole: a flexible break-glass access control model," in Proceedings of the 16th ACM symposium on Access control models and technologies. ACM, 2011, pp. 73–82.