SecurOntology: A Semantic Web Access Control Framework

Ángel García Crespo

angel.garcia@uc3m.es
Escuela Politécnica Superior
Universidad Carlos III de Madrid, Spain
Av. Universidad 30, Leganés, 28911, Madrid, SPAIN
Phone: +34 91 624 9417
Fax: +34 91 624 9129

Juan Miguel Gómez-Berbís

Juanmiguel.gomez@uc3m.es
Escuela Politécnica Superior
Universidad Carlos III de Madrid, Spain
Av. Universidad 30, Leganés, 28911, Madrid, SPAIN
Phone: +34 91 624 5958
Fax: +34 91 624 9129

Ricardo Colomo-Palacios ***(Corresponding Author)***

ricardo.colomo@uc3m.es
Escuela Politécnica Superior
Universidad Carlos III de Madrid, Spain
Av. Universidad 30, Leganés, 28911, Madrid, SPAIN
Phone: +34 91 624 5958
Fax: +34 91 624 9129

Giner Alor-Hernández

galor@itorizaba.edu.mx
Division of Research and Postgraduate Studies
Instituto Tecnológico de Orizaba
Av. Oriente 9 852. Col Emiliano Zapata C.P. 94320, Orizaba, Mexico
Phone: (272)7244096
Fax: (272)7251728

SecurOntology: A Semantic Web Access Control Framework

Abstract

Security and privacy are key concerns for the current dimension the Internet has reached in our daily life. Policies representing resource access based on knowledge-oriented descriptions have gained momentum with the emergence of Semantic Technologies based approaches. Traditional access control frameworks were syntactic and error prone, lacking the necessary expressivity and efficiency of a solution where soundness and completeness of the underlying logics in access control descriptions could be critical to harness their potential. In this paper, we present the SecurOntology approach, which encompasses a three-fold strategy: a well-structured ontology for access control to resources, a logical declarative framework and a software architecture as a proof-of-concept of the advantages of this solution.

Key words:

Security, Semantic Web, Web Access Control, Description Logic, Rules

Acknowledgements

This work is supported by the Spanish Ministry of Industry, Tourism, and Commerce under the EUREKA project SITIO (TSI-020400-2009-148), SONAR2 (TSI-020100-2008-665), INNOVA 3.0 (TSI-020100-2009-612) and GO2 (TSI-020400-2009-127).

1 Introduction

The Internet-driven networked economy is evolving to the point where businesses are fully aware of the enormous business opportunities of online transactions. Nevertheless, full exploitation of these seemingly limitless opportunities will depend largely on non-functional concerns such as security or privacy and Web resources access control.

Traditionally, Role-based Access Control (RBAC, in short) [1] have been used to restrict system resources to authorized access within an organization. In a nutshell, roles are created for a number of profiles, which are granted specific permissions. These permissions to perform certain operations are assigned to specific roles and resources, so that, unlike Context-based Access Control (CBAC), RBAC does not consider any additional information context [2]. Hence, users are not assigned permissions directly, but only acquire them through their role (or roles) and the management of individual user rights becomes a matter of simply assigning the appropriate roles to the user.

In principle, understanding and characterizing these factors is crucial to improve the current sustainability of the e-commerce model, hence major concerns are rising. The fundamental basis of a successful approach hinges on the ability of trading partners to follow access control

policies from a meaningful and formal perspective. Using a declarative logical framework advocates Role-based Access Control resources to commit with specific unambiguous, decidable and machine-understandable policies. The goal of this work is to design, specify and implement a set of logic-based RBAC, ranging from the simplest security policy to complex multi-element composite policies, harnessing the potential of formal semantics.

The remainder of the paper is organized as follows. In section 2, we discuss the State of Art in access control to web resources in applications. In section 3, we describe a logical framework for access control decision based on Semantic Technologies which include the SecurOntology ontology, the underlying logical framework and a set of rules together of a use case to stress the potential of the SecurOntology approach. In section 4, a Semantic Access Control software architecture (SECO) is described through a set of loosely-coupled software components as a proof-of-concept implementation of the overall SecurOntology approach. Finally, section 5 concludes the paper and outlines our future work.

2 State of the Art

Role-based access control (RBAC) is being increasingly recognized as an efficient access control mechanism that facilitates security administration [1]. It can be seen as a newer alternative approach to mandatory access control (MAC) [3] and discretionary access control (DAC) [4], so in other words, RBAC enforces DAC and MAC [5]. RBAC has been proposed as an alternative approach to this traditional access control mechanisms both to simplify the task of access control administration and to directly support function-based access control [6]. Furthermore, it has been recently approved as a standard by the American National Standards Institute (ANSI) and a number of organisations are today applying this standard in specialised domains [7]. A key advantage of the RBAC model is that it simplifies authorization administration by assigning permissions to users through roles. Thus, it adds a layer of abstraction between users and their permissions [8].

RBAC groups individual users into roles that relate to their position within an organization and assigns permission to various roles according to their stature in the organization [9]. Separation of duty (SoD) and dependence constraints are examples of dynamic constraints and required in most commercial applications, including digital government, E-commerce, healthcare systems, and workflow management systems that can be addressed by using RBAC [10]. As a result of this, today, the RBAC model is one of the most established access models [11]. Because of its relevance, RBAC has been widely investigated and several extensions to it as well as possible applications have been proposed, including TRBAC [12], W-RBAC [13] and GeoRBAC [14] to cite just a few.

New technologies such as Web services or Semantic Web increase the complexity and the dependencies of IT systems with respect to access control [15]. RBAC model is particularly suitable for Web applications [16] because it can define a diverse set of access control policies [8]. Thus, the adaptation of RBAC to new technologies has been a common starting point. As a result access control frameworks have been evolving from OASIS XACML (Extensible Access Control Markup Language) [17] or X-RBAC [2] which were based on XML to describe the access rights and lacked on machine interpretation; to O-RBAC [18] that adapt RBAC to semantic web technologies by exporting its domain to an ontology specification.

The arrival of the Semantic Web represents a revolution for the form of access and storage of information. The term "Semantic Web" was coined by Berners-Lee, Hendler & Lassila [19], to describe the evolution from a document-based web towards a new paradigm that includes data and information for computers to manipulate. The Semantic Web enables automated information access based on machine-processable semantics of data. The Semantic Web was defined by these authors as “an extension of the current web in which information is given well defined meaning”. Formal ontologies [20] play an essential role in the Semantic Web vision, because they provide structured vocabularies that describe a formal specification of a shared conceptualization. Ontologies were developed in the field of Artificial Intelligence to facilitate knowledge sharing and reuse [21]. The Semantic Web provides an alternative solution to represent the comprehensive meaning of integrated information and promises to lead to efficient data management by establishing a common understanding [22]. Until recently, the use of Semantic Web languages has been limited primarily to representing Web content and services. But there are just a few of its possibilities. And both enhancing the Semantic Web with security [23] and adding semantics to security models are prolific research lines. Thus, taking into account just the latter, apart from the effort by [18], some more initiatives has been adopted to use this new technology in the field of security. In [24] authors used Ontology Web Language (OWL) to Specify Constraints of RBAC, KAoS uses OWL as the basis for representing and reasoning about policies within Web Services, Grid Computing, and multi-agent system platforms [25], Rei [26] is a policy language that is grounded in a semantic representation of policies in RDF-S. In [27] a comparison of KAoS, Rei, and Ponder [28] is made. In other relevant works, Kwon and Moon [29] uses semantic web as well to model and specify constraints of RBAC and Chen [30] uses an ontology-based access control approach to enable Knowledge sharing in virtual enterprises. In a recent work, Finin et al. [31] studied different ways to support the NIST Standard RBAC model in OWL.

In the work of Bonatti and Olmedilla [32], authors merge the ideas of previously published papers [33, 34, 35] to study the requirements required to develop a successful Semantic Policy Framework. In referred study, authors analyze KAos and Rei together with two trust based initiatives, PeerTrust [36] and Protune [37] to bring a recommendation to develop the new semantic system. Artz & Gil [38] provide an overview of trust research in computer science relevant to the Semantic Web, and more recently, Blanco et al. [39] identify the main initiatives and compare them using ONTOMETRIC [40].

In the context of web-based communities that arises from the so called Social Web or Web 2.0, Chowdhury et al. [41] a framework for privacy in social communities by exploiting semantic web technologies, following the steps of previous efforts that combine somehow semantics and web 2.0 [42, 43]

Actually, the motivation of setting up a new framework instead of using an existing one is to achieve our main goal of fulfilling a certain amount of requirements. Those requirements were hence not addressed by the current approaches and are listed in what follows. First of all, a logic-based ontology-driven approach was necessary to harness the expressiveness and potential of knowledge-based systems. Secondly, a proper fully-fledged software architecture dealing with ontologies repositories, logic-based systems and the interrelation with the underlying formal languages was also expected. Finally, last but not least, the actual approach

is absolutely turned towards “semantics” versus “syntactic” approaches, since current state of the art in security RBAC models were based in the latter.

To sum up, in this work we are addressing all those requirements and provide a general solution with a broad scope to encompass a number of features that will be outlined in the remainder of the paper, but hinge on the implementation of a declarative logical framework to support the RBAC techniques.

The SecurOntology approach is different to the previous approaches but builds on their strengths and forthcomings. First of all, we will be using a declarative logical language in order to express a set of policies based on resource access and control, based on Semantic Web languages, such as the Web Ontology Language (OWL) in its OWL-DL variant. Secondly, we believe one of the advantages of SecurOntology face to traditional RBAC systems is precisely the expressive power of formal, sound and logically complex descriptions of a number of resources access. Since RBAC systems are syntactic, it would be difficult to express a number of highly precise and relevant queries. We will elaborate on this issue in section 4.

Finally, SecurOntology combines the expressivity of the underlying formal descriptions with an architecture and evaluation that can be superposed or applied to a number of typical Web applications, such as Context Management Systems (CMS), or to Web data silos (e.g. financial data silos observing the Sarbanes-Oxley Act regulations) or even personal critical data.

3 A Logical Framework to Make Semantic-based Access Control Decisions

We promote using semantic technologies as the main aspect of our work in order to support and improve access control, taking advantage of the expressiveness and reasoning features provided by these technologies.

Semantic representation can be done through ontologies, the cornerstone technology of the Semantic Web. Ontologies provide structured vocabularies that describe a formal specification of a shared conceptualization [20]. They are used to capture knowledge about a certain domain. The knowledge gathered is the common and comprehensible meaning of the information and data, transformed into concepts and relationships between them.

Among the different ontology languages, we are focusing on the Web Ontology Language (OWL) [44], a recommendation of the World Wide Web Consortium (W3C). OWL is a markup language implemented on RDF and RDF Schema. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema, by providing additional vocabulary along with a formal semantics for describing concepts and properties (e.g. relations between concepts, cardinality, equality, richer typing of properties, etc).

There are three variations of OWL with different levels of increasing expressiveness: OWL Lite, OWL DL and OWL Full. Our work is developed using OWL DL, based on Description Logics. We aim for a decidable and computationally efficient system, since the aim of the mechanism is to evaluate and grant access to resources. For this reason, we discarded OWL Full, despite of the

great expressivity power compared with OWL DL. The expressivity provided by OWL is limited by tree-like structures [45], hence the new knowledge cannot be inferred from indirect relations between entities. A requirement for our proposal in section 4 must deal with inferred knowledge from indirect relations. The lack of expressivity is settled using Semantic Web Rule Language (SWRL) [46], which is designed as an extension of OWL DL. The problem which arises is that OWL DL becomes undecidable when the set of OWL axioms is extended to include Horn-like rules. To provide decidability when using SWRL, Motik et al. [45] define a subset of SWRL of DL-safe rules without affecting the expressivity of OWL. In SWRL, as any Horn Logic based language, rules are defined as a set of precedent and consequent states.

The following subsections define SecurOntology, the ontology developed, some vocabularies to be used, and a first set of rules to show how to guarantee access control to resources.

3.1 Semantic-based Access Control Decisions

Access control is strongly related to the structural definition of an organization. First, it is necessary to understand what an organization is. From our point of view, it is a set of Work Units such as departments, clusters, people, etc. In the lowest layer, the organization is comprised of people and analysis is focused on the relationships between these people. A number of questions must be addressed: Do they work in the same cluster?, Who is your supervisor?, Are you a manager?, What is the difference between being a director or an administrator?.

The issue which must be dealt with is fundamentally a type of social network. For social networks, there are several well-known ontologies and vocabularies. FOAF [47] is a common vocabulary that is extensively used nowadays, which enables users to define their contact information and social networks. The power of FOAF lies on its extensibility feature. Part of our work is based on this characteristic; trying to reuse the FOAF vocabulary in the access control domain. Also other extensions such as FOAFCorp [48], which extends FOAF, describing the structure and interconnections of corporate entities in more detail, take advantage of this feature.

Awareness of organizational requirements enables correct and dynamic management of information resources. For instance, in a company there may be several resources belonging to a department which can only be accessed by people working in that department; or resources which people are currently working on and only want to share with collaborating members and not with supervisors. Our ontology tries to define the support for access rights from domain proximity, eliminating the lack of expertise needed to manage security in complex systems. As was previously indicated, the organizational unit basic structure is a hierarchical tree, which is perfect regarding the tree-like structures of OWL ontologies.

Hence, each application can define its own structure based on subunits such as department, sub-department, projects, inter-departmental relationships or also partner relationships. For instance, an organization develops an ontology expressing its internal structure, indicating the role each employee, privileges taking into account all the expressive power that OWL offers. Once the structural definition of the organization is described, the use of vocabularies for describing relationships between people will enrich the model. A generic approach we reuse is

the RELATIONSHIP [49] vocabulary, defining terms like ancestorOf, employerOf, colleagueOf, hasMet, livesWith, among others. Nevertheless, SecurOntology can be extended to cover a specific domain. Let's propose a Collaborative Working Environment (CWE), where people collaborate together when doing their work. In this case, some more specific collaboration-related relationships between people are needed. In this example, there is a simple and general vocabulary focused on collaboration called CoVoc (COllaboration VOCabulary) developed by Nasirifard and Peristeras [50]. This vocabulary is proposed for annotating knowledge workers in a collaborative environment. Summarising, CoVoc is a set of terms tackling different collaborative relationships and social connections between individuals in a collaborative working environment.

Mainly, CoVoc covers any collaboration between people working in different projects, participating in different events, deals with University and industry environments, and has various social generic activities such as reading weblogs, watching online videos, listening to podcasts, etc. A few activities defined in CoVoc are: writeDocumentWith, reviseDocumentOf, WriteDeliverableWith, taskLeader, developer, tester, industrialPartner, academicPartner, metInConference, supervisor, etc.

Our ontology define two main concepts: Work Unit and Resource. Work Unit is extended with FOAF and CoVoc in order to enrich the model. Moreover, the Work Unit concept is extended with the ontology in which is defined the structure of the organization. Furthermore, Resource can be extended in the same way in order to distinguish between subclasses of resources, such as research papers, deliverables, etc. These characteristics establish who can access which resources, allowing the composition of groups to give access to certain resources, and distinguishing different access categories such as readable, writable, etc. between resources and work units. Also, we can extend the vocabulary in order to consider aspect such as isAuthorOf, isReviewerOf, isResponsibleFor, etc., to increase the expressiveness.

Summarizing, SecurOntology is responsible for describing the resources, making use of all the expressive and logic capabilities which semantic Web offers. Similar to organizations, resources can have a hierarchical structure so the definition of the privileges can make use of enhancements such as transitivity, inheritance or symmetry that allow inference of each user privileges in each resource with no need to define them explicitly. Nevertheless, a set of basic rules are necessary.

Nevertheless, a set of basic rules are necessary to guarantee the functionality needed by the framework. The rule set will be defined in the following section.

3.2 SecurOntology

In this section, we described the SecurOntology ontology, composed by classes, properties and finally, a set of rules. Firstly, the SecurOntology classes consist of a basic hierarchy with the following super classes:

- **Resources**: In this class we will have the resources of the system. If a resource has some children, the children will be established with the relation mentioned in ObjectTypeProperties.

- **Owners**: It will represent the possible owners of the resources. Inside this class will have instances to represent the different owners of the system.
- **Roles**: It establishes the possible role of some owner. It exists a current roles like "Administrator", "User", etc.
- **Permission**: It represents the permission that have "an owner" over "a resource". It has at least three instances that represents the most common permissions in UNIX systems:
    - **Read**: Establish that the "owner" has read permission over the "resource".
    - **Write**: Establish that the "owner" has write permission over the "resource".
    - **Execution**: Establish that the "owner" has execution permission over the "resource".
- **ResourceAndPermission**: Is a class that allows establishing a permission to a current resource in order that a owner, can adopt this "resource and permission". For example: User1 wants "read and write" permission over resource "Doc1". So, an instance of ResourceAndPermission will be created, adding the resource "Doc1" and permissions "read and write" to the instance. Lately, the owner can adopt this instance with the property "hasPermission", and will adopt the "read and write" permission over the resource "doc1". This is helpful because can be established more than one kind of permission over resources, and every user will take the permission/resource that need in each case.
- **ConsultInstanc**e: Is a class that allows creating instances to make the consults over the ontology.

Secondly, the ontology contains a number of ObjectTypeProperties in order to establish the relations between the instances of the classes in the ontology. In the next table the relationships are shown. Underlined properties mean they are functional so they do not take more than one value.

Name	Domain	Rank
hasRole	Owners ConsultInstance	Roles
isOwnerOf ($_1$)	Owners ConsultInstance	Resources
itsOwnerIs ($_1$)	Resources	Owners ConsultInstance
hasPermission	Owner Permissions ConsultInstance	Resources
hasChild ($_2$)	Resources	Resources

isChildOf (2)	Resources	Resources
resource	ResourceAndPermission	Resources
permission	ResourceAndPermission	Permissions

Table 1. ObjectTypeProperties of ontology

Brief descriptions about these properties are explained here:

- ***hasRole***: Specify the role that the owner have. It only can has one rol (functional property).
- ***isOwnerOf(1)***: Specify that the selected user is the owner of some resources. Symmetric property: itsOwnerIs.
- ***itsOwnerIs(1)***: Specify the selected resource, it's owner is .. whatver. Symmetric property: isOwnerOf.
- ***hasPermission***: Specify the current permission(s) that a current owner(s) have over a resource.
- ***hasChild(2)***: Specify that a resource have as a child another resource. It has a symmetric property: isChildOf.
- ***isChildOf(2)***: Specify that a resource is a child of another resource. It has a symmetric property: hasChild..
- ***resource***: Specify the resource that will have some "permissions" in the instance of ResourceAndPermission.
- ***permission***: Specify the permission associated to a resource in the instance of ResourceAndPermission.

In addition, the ontology also contains some DataTypeProperties to establish values like name, codes, etc, but these properties are not important for the study of this article so no more mention will be done.

3.3 Rules for SecurOntology and Use Case

Access control is accomplished by using SWRL, in order to infer new knowledge which does not exist in the knowledge base. The first issue is the translation and use of the knowledge base developed in OWL. The OWL triples are easily transformed into SWRL facts. These rules can be used in the inference process as the fact base. In our implementation of SecurOntology, rules are written and executed as Jena Rules.

In the following example, a user, Project Manager 1, wants to access DOC1.1. Both DOC1.1 and DOC1.2 are a subclass of DOC1. We hereby need to describe formally the following premises:

1. User "Project Manager 1", onwards, "User1" is the Owner of "DOC1" (superclass of DOC1.1 and DOC1.2)
2. The Role of user User1 is "Administrator".
3. As an "Administrator" it has a Read and Write permission on DOC1.

The logical characterization would be as follows:

*(“User1” **isOwnerOf** “DOC1”) && (“User1” **hasRol** “Admin”) && (“User1” **hasPermission** “perRAndWOverDoc1”) && (“perRAndWOverDoc1” permission “Read” and permission “Write”) && (“perRAndWOverDoc1” resource “DOC1”) → (“User1” isOwnerOf “DOC1.1”) && (“User1” isOwnerOf “DOC1.2”)*

A temporal instance called “perRAndWOverDoc1” must be set, specifying the Read and Write Permissions on DOC1. Hence, this instance is associated with “hasPermission” on User1, and subsequently, the values of this instance (Permission and Resource), taking into account that “User1” has complete access (isOwnerOf) to DOC1. Those Properties set in bold are those to be applied to the “ConsultInstances” class since it needs these properties to generate the “query instances”. As Jena Rules, the expression would be as follows:

@prefix ont: <URI_ONTOLOGY#>.
@include <RDFS>.

[rule_DOC1.1_NOT_REST_RESOURCES: (?i ont:isOwnerOf ?x) notEqual(?x, ont:DOC1) -> (?i ont:hasNegResources ont:DOC1.1_NOT_RESOURCES)]

This rule declares a premise that defines the property *hasNegResources* to these resources that are not "valid", that is, all the resources that are not *DOC1* are not valid. In case that the inference engine receives some resources that are not valid (some resources different from *DOC1*), the property *hasNegResources* will be established. Its function is to discriminate resources.
The next rule (that is in backward form) is used to allow to infer the resource that we are managing (*DOC1.1*), in the case that we are processing the resource "*DOC1*" but not any resource of the resources not allowed.

[rule_DOC1.1_DOC1_RES: (?i ont:results ont:DOC1.1) <- (?i ont:hasResource ont:DOC1) noValue(?i, ont: hasNegResources ont: DOC1.1_NOT_RESOURCE)]

With these two rules, we can infere “DOC1.1” when “any user” has the resource “DOC1” as his own resource. Now, we need to limit the resource to user “User1”. We made the same, but in this case, we will limit, instead the resources, the users:

*[rule_DOC1.1_NOT_REST_USERS: (?i ont: **itsOwnerIs** ?x) notEqual(?x, ont:USER1) -> (?i ont:hasNegUsers ont:DOC1.1_NOT_USERS)]*

*[rule_DOC1.1_DOC1_USERS: (?i ont:results ont:DOC1.1) <- (?i ont: **itsOwnerIs** ont:USER1) noValue(?i, ont: hasNegUsers ont: DOC1.1_NOT_USERS)]*

In this case, the property “itsOwnerIs” is a the relation to know the owner from a resource. Now, we need to limit the Role:

[rule_DOC1.1_NOT_REST_ROLES: (?i ont: ***hasRol*** *?x) notEqual(?x, ont:ADMIN) -> (?i ont: hasNegRoles ont:DOC1.1_NOT_ ROLES)]*

[rule_DOC1.1_DOC1_USERS: (?i ont:results ont:DOC1.1) <- (?i ont: ***hasRol*** *ont:ADMIN) noValue(?i, ont: hasNegRoles ont: DOC1.1_NOT_ ROLES)]*

The next step, is limit that the permissions allowed are "Read" and "Write" and that are associated with the resource "DOC1", so, we will create this rules (The instance that manage this data is "*perRAndWOverDoc1")*:

[rule_DOC1.1_NOT_REST_PERMISSIONS: (?i ont: ***permission*** *?x)* ***notEqual(?x, ont:READ) notEqual(?x, ont:WRITE)*** *-> (?i ont: hasNegPermissions ont:DOC1.1_NOT_ PERMISSIONS)]*

[rule_DOC1.1_DOC1_PERMISSIONS: (?i ont:results ont:DOC1.1) <- ***(?i ont: permission ont:READ) (?i ont: permission ont:WRITE)*** *noValue(?i, ont: hasNegPermissions ont: DOC1.1_NOT_ PERMISSIONS)]*

In this case, we can see that we are limiting the permissions to two values (READ and WRITE) so we can establish the rule with the two values. Finally, to limit the resource in this individual:

[rule_DOC1.1_NOT_REST_RESBIS: (?i ont: ***resource*** *?x)* ***notEqual(?x, ont:DOC1)*** *-> (?i ont: hasNegResBis ont:DOC1.1_NOT_ RESBIS)]*

[rule_DOC1.1_DOC1_RESBIS: (?i ont:results ont:DOC1.1) <- ***(?i ont: resource ont:DOC1)*** *noValue(?i, ont: hasNegResBis ont: DOC1.1_NOT_ RESBIS)]*

In this section, we have carefully worked on the SecurOntology classes, properties and relationships. We have also described the overall logical framework for intelligent access control decisions. Finally, we described a set of rules through a use case which unleash the potential of the SecurOntology approach.

4 SECO: A Software Architecture for Semantic-based Access Control Decisions

This section describes the analysis, design and implementation of a particular software architecture to make Semantic-based Access Control Decisions, based on the principles of loosely-coupled, self-described and layered composition. In this section, we will show the

SECO architecture by introducing a number of software components that use the technologies described in previous sections. Since we envisage a software architecture as the set of software components, connections and interfaces in which a particular software system is organized, we will elaborate on how the architecture supports a number of functionalities from that standpoint. Hence, the SECO architecture is composed by several self-contained software modules or subsystems as it is discussed in the following:

- Rule-based Semantic Policies: One of the major advantages of SecurOntology is formalizing access control procedures by means of rules, as described in section 3.3. In the SECO architecture and implementation we have used Jena rules, whose syntax has been detailed in the use case in the aforementioned section. In this component, those rules are stored and described, composing access control policies.

- Rule-based Semantic Engine: This component represents a logical follow-up of the rule-based semantic policies, since those rules must be interpreted and executed. We implemented a simple rule engine to evaluate the set of rules specified in the use case (that set of rules being a particular example), using the Jena framework.

- SECO Access Control Manager: Since the policies have been interpreted through rules in the previous components, the SECO Access Control Manager confronts both the rules with the SecurOntology instances retrieved from the Semantic Storage component (which will be subsequently specified) and transforms both policies and instances into actual permission to Resources from Owners with a particular set of Permissions in the following three Web applications:

    - Content Management System (CMS): In our implementation, we envisaged and used a CMS as a computer application used to manage work flow needed to collaboratively create, edit, review, index, search, publish and archive various kinds of digital media and electronic text. This CMS allows Owners the access through Resources based on the permission granted from the SECO framework. Particularly, this one of the advantages of the whole approach, where an external structure can be applied to day-to-day applications, while obtaining profit from formalization, soundness and expressivity.

    - Financial Statements: The SECO framework can also help to access a set of data, especially financial, or protected by the Sarbanes-Oxley Act regulations[1], where particular emphasis on security and privacy are critical but in a number of restricted cases. The complexity of this regulation can be expressed by means of the logical complexity provided by SECO.

    - Resource Repository: Knowledge management systems are very prone to store most of their documents (which are encompassed by the Resource ontological term), in a Sharepoint-like strategy. SECO also provides access control as an external structure

[1] Sarbanes Oxley Regulation Act: http://www.sarbanes-oxley.com/

- Sensitive Data: A crucial part of most Web applications, sensitive data is key to observe a number of privacy and security concerns.

- Semantic Storage This component deals with the Ontology Storage. Ontology Repositories are software components that deal with scaling, loading and inferencing of real ontologies. It deals both with the instances of the SecurOntology ontology and the SecurOntology schema itself. The SecurOntology ontology has been widely described in section 3. We have analyzed, designed and, finally, implemented the SecurOntology ontology to populate instances of this ontology associating the concepts of the hierarchy with the sets of structured data. The ontology is based on the hierarchy of classes showed in section 3 and implemented with the Ontology Web Language (OWL), a family of knowledge representation languages for authoring ontologies, endorsed by the World Wide Web Consortium. Particularly, since we will be using Description Logics as the underlying framework to reason about, we implemented the ontology in its OWL-DL flavor.

In the following figure, we depict the SECO architecture and each of the components whose functionality has been described. The actual architecture is shown as follows:

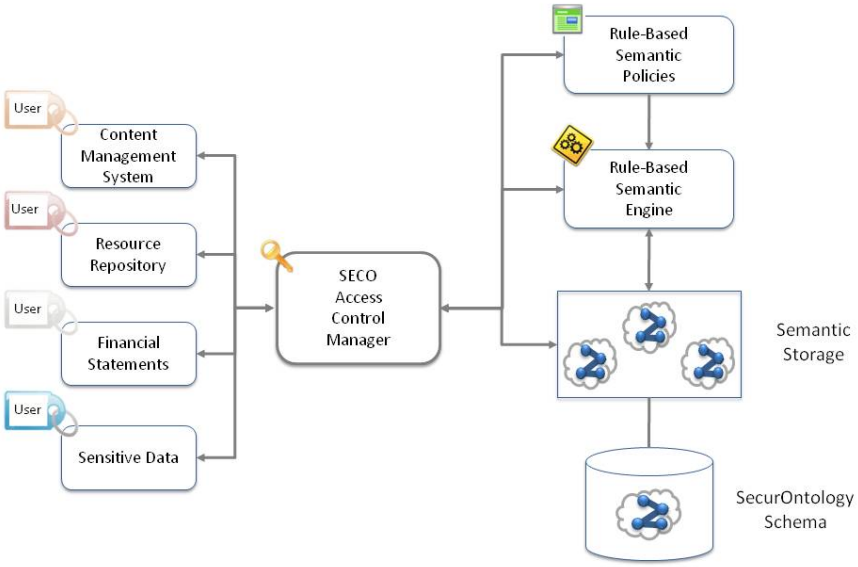

Figure 1. The SECO Architecture

In figure 2 the logical layer diagram in which SECO is divided can be observed.

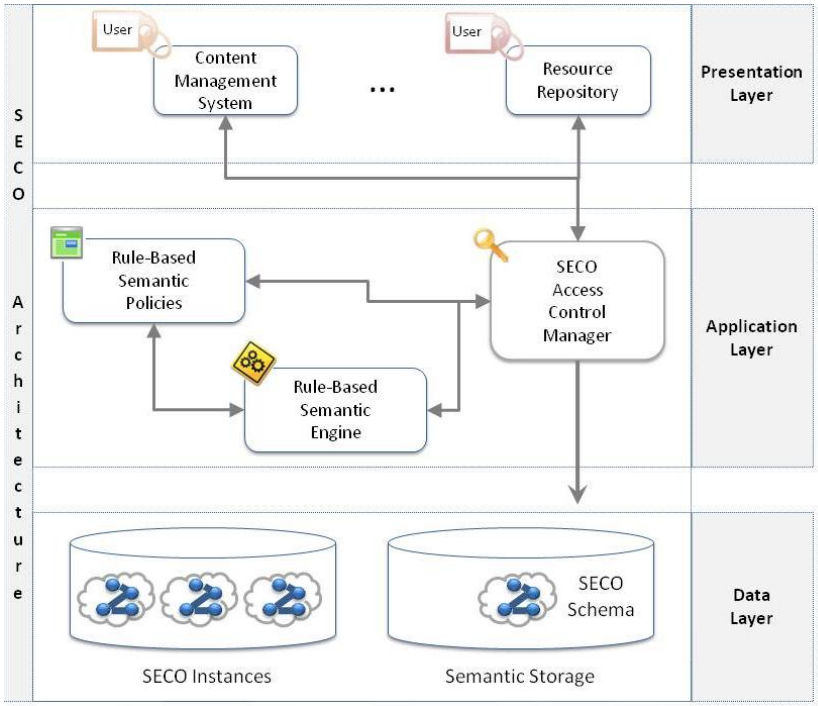

Figure 2. SECO logical layers

Figure 2 shows the logical structure of the SECO architecture. It consists of a 3-layered architecture, which was selected because of its adaptability, flexibility and reusability. The system was developed in an incremental and evolutionary manner that needed those main characteristics to grant a successful fulfillment. A three-layered architecture also offers the advantage of easing the localization of errors, since it avoids the transfer of errors between layers.

In the upper layer, the presentation, the applications where the SECO architecture had an immediate application and was tested are included. The Content Management System (CMS), the financial data silos or the sensitive data structures are located in this layer.

The Application Layer is the responsible for managing all the business logic, allowing a better decoupling of responsibilities in the final system. Fundamentally, the layer is composed by the Rule-based Semantic Policies and Rule-based Semantic Engine components providing an intelligent platform to apply the resource access control. Finally, the SECO Access Control Manager is the core software component applying and turning the policies rules execution into actual permissions. Last but not least, the Data layer is in charge of handling ontologies in a low level, which means, it manages the knowledge storage contained in the system.

5. Conclusions and future work

The SecurOntology approach has outlined a new solution based on a three-pronged strategy, namely a well-structured ontology, a logical declarative framework and a software architecture as a proof-of-concept, for the problem of syntactic, basic RBAC approaches being too trivial for

a number of complex Web applications. Our solution is based on Semantic Technologies in order to use its underlying formal properties to reason, validate and allow access to resources. In addition, our architecture can be applied to a set of critical data silos or to a Content Management System (CMS) providing an intelligent means of tackling with security concerns. This solution is also an interesting option to extend the expressivity and provide knowledge-driven decisions for resources access control which is not solved in previous works of access control frameworks using Semantic Web capabilities.

Our future research will focus in extending the functionalities and properties of the framework. This will consist on a three-step process. First of all, the SecurOntology will be validated and tested in a number of domains, being added more properties and potential rules to improve and customize its efficiency. Secondly, the logical framework will be extended by adding a set of new logical constructs and analyzing how other logical languages or formalisms could improve the formal correctness of the approach. Finally, our software architecture will be tested in a set of new domains.

7. References

1 Sandhu, R., Coyne, E.J., Feinstein, H.L. and Youman, C.E., Role-Based Access Control Models, IEEE Computer 29 (2) (2000) 38–47.

2 Joshi, J.B.D., Access-Control language for multidomain environments, IEEE Internet Computing 8(6) (2004) 40-50.

3 Bell, D. E. & LaPadula, L. J., Secure computer system: unified exposition and MULTICS. Technical Report ESD-TR-75-306, The MITRE Corporation, Bedford, MA, 1976.

4 Lampson, B. Protection. In Proceedings of the 5th Symposium on Information Sciences and Systems (Princeton, NJ, Mar.) 1974, 437–443.

5 Osborn, S., Sandhu, R. and Munawer, Q., Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies, ACM Transactions on Information and System Security 3(2) 2000, 85-106.

6 Bertino, E. RBAC models — concepts and trends. Computers & Security 22(6) (2003) 511-514.

7 Damiani, M.L., Bertino, E. and Perlasca, P., Data security in location-aware applications: an approach based on RBAC, International Journal of Information and Computer Security 1(1/2) (2007) 5-38.

8 Bhatti, R., Bertino, E., Ghafoor, A. and Joshi, J.B.D., XML-Based Specification for Web Services Document Security, IEEE Computer 37(4) (2004) 41-49.

9 Wainer, J., Kumar, A. and Barthelmes, P., DW-RBAC: A formal security model of delegation and revocation in workflow systems, Information Systems 32(3) (2007) 365-384.

10 Shafiq, B., Joshi, J.B.D., Bertino, E. and Ghafoor, A., Secure Interoperation in a Multidomain Environment Employing RBAC Policies, IEEE Transactions on Knowledge and Data Engineering 17(11) (2005) 1557-1577.

11 Breu, R., Popp, G. and Alam, M., Model based development of access policies, International Journal on Software Tools for Technology Transfer, 9(5) (2007) 457, 470.

12 Bertino, E., Bonatti, P. and Ferrari, E., TRBAC: a temporal role-based access control model, ACM Transactions on Information and System Security, 4(3) (2001) 191–233.

13 Wainer, J., Barthelmess, P. and Kumar, A., W-RBAC a workflow security model incorporating controlled overriding of constraints, International Journal of Cooperative Information Systems, 12(4) (2003) 455-485.

14 Damiani, M.L., Bertino, E., Catania, B. and Perlasca, P., GeoRBAC: A Spatially Aware Rbac, ACM Transactions on Information and System Security 10(1) (2007) 2.

15 Sohr, K., Drouineaud, M., Ahn,G.J. and Gogolla, M., Analyzing and Managing Role-Based Access Control Policies. IEEE Transactions on Knowledge and Data Engineering 20(7) (2008) 924-939.

16 Joshi, J.B.D., Aref, W.G., Ghafoor, A., Spafford, E.H., Security Models for Web-Based Applications, Communications of the ACM 44(2) (2001) 38-72.

17 Moses, T., OASIS eXtensible Access Control Markup Language 2.0, core specification. OASIS XACML Technical Committee StandarD, 2005.

18 Wu, D., Chen, X., Lin, J. & Zhu, M., Ontology-Based RBAC Specification for Interoperation in Distributed Environment. First Asian Semantic Web Conference, Beijing, China, September 3-7, 2006, pp. 179-190.

19 Berners-Lee, T., Hendler, J. and Lassila, O., The Semantic Web. Scientific American, 284(5) (2001) 34-43.

20 Gruber, T.R., A translation approach to portable ontologies, Knowledge Acquisition 5(2) (1993) 199-220.

21 Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L. and Patel-Schneider, P.F., OIL: An Ontology Infrastructure for the Semantic Web, IEEE Intelligent Systems 16(2) (2001) 38-45.

22 Shadbolt, N., Hall, W. and Berners-Lee, T. The Semantic Web Revisited, IEEE Intelligent Systems 21(3) (2006) 96-101.

23 Denker, G., Kagal, L. and Finin, T., Security in the Semantic Web using OWL, Information Security Technical Report 10(1) (2005) 51-58.

24 Di, W., Jian, L., Yabo, D. & Miaoliang, Z., Using Semantic Web Technologies to Specify Constraints of RBAC, In Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2005, 543-545.

25 Uszok, A., Bradshaw, J., Jeffers, R., Suri, N., Hayes, P., Breedy, M., Bunch, L., Johnson, M., Kulkarni, S. and Lott, J., KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement. In Proceedings of IEEE Fourth International Workshop on Policy (Policy 2003). Lake Como, Italy, 4–6 June, Los Alamitos, CA: IEEE Computer Society, 2003, pp. 93–98.

26 Kagal, L., Finin, T. and Johshi, A., A Policy Language for Pervasive Computing Environment. In Proceedings of IEEE Fourth International Workshop on Policy (Policy 2003). Lake Como, Italy, 4–6 June, Los Alamitos, CA: IEEE Computer Society, 2003, pp. 63–76.

27 Tonti, G., Bradshaw, J.M., Jeffers, R., Montanari, R., Suri, N. and Uszok, A., Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder, In Proceedings of the International Semantic Web Conference, 2003 419-437.

28 Damianou, N., Dulay, N., Lupu, E., Sloman, M., The Ponder Policy Specification Language. In proceedings of Workshop on Policies for Distributed Systems and Networks (POLICY 2001). Springer-Verlag, LNCS 1995, Bristol, UK, 2001.

29 Kwon, J. and Moon, C.J., Visual modeling and formal specification of constraints of RBAC using semantic web technology, Knowledge-Based Systems 20 (4) (2007) 350-356.

30 Chen, T.Y. Knowledge sharing in virtual enterprises via an ontology-based access control approach., Computers in Industry 59(5) (2008) 502-519.

31 Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W. H. and Thuraisingham, B. (2008). ROWLBAC - Representing Role Based Access Control in OWL. In Proceedings of the 13th Symposium on Access Control Models and Technologies (73-82). ACM Press, June 2008.

32 Bonatti, P.A. and Olmedilla, D., Rule-Based Policy Representation and Reasoning for the Semantic Web. Reasoning Web 2007 LNCS, vol. 4636, 2007, 240-268.

33 Antoniou, G., Baldoni, M., Bonatti, P.A., Nejdl, W. and Olmedilla, D., Rule-based policy specification. In: Yu, T., Jajodia, S. (Eds.) Secure Data Management in Decentralized Systems. Advances in Information Security, vol. 33, Springer, Heidelberg, 2007.

34 Bonatti, P.A., Duma, C., Fuchs, N., Nejdl, W., Olmedilla, D., Peer, J. & Shahmehri, N., Semantic web policies - a discussion of requirements and research issues. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS 4011, Springer, Heidelberg, 2006.

35 Olmedilla, D., Security and privacy on the semantic web. In: Petkovic, M., Jonker, W. (eds.) Security, Privacy and Trust in Modern Data Management, Springer, Heidelberg, 2007.

36 Nejdl, W., Olmedilla D. and Winslett, M., Peertrust: automated trust negotiation for peers on the semantic web, Proceedings of Workshop on Secure Data Management in a Connected World in Conjunction with the 30th International Conference on Very Large Data Bases, 2004, pp. 118–132.

37 Bonatti P. & Olmedilla, D., Driving and monitoring provisional trust negotiation with metapolicies, POLICY '05: Proceedings of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05), IEEE Computer Society, Washington, DC, USA, 2005, pp. 14–23.

38 Artz, D. and Gil, Y., A survey of trust in computer science and the Semantic Web, Web Semantics: Science, Services and Agents on the World Wide Web, 5(2) (2007) 58-71.

39 Blanco, C., Lasheras, J., Valencia-García, R., Fernández-Medina, E., Toval, A. and Piattini, M. A Systematic Review and Comparison of Security Ontologies. Third International Conference on Availability, Reliability and Security, 2008, pp. 813-820.

40 Lozano-Tello, A. & Gómez-Pérez, A., ONTOMETRIC: A Method to Choose the Appropriate Ontology, Journal of Database Management 15(2) 2004 1-18.

41 Chowdhury, M.M.R., Elahi, N., Alam, S. and Noll, J., A framework for privacy in social communities, International Journal of Web Based Communities 5(2) (2009) 293-312.

42 Carminati, B., Ferrari, E. and Perego, A., Rule-Based Access Control for Social Networks. OTM Workshops. Springer-Verlag, LNCS 4278, 2006, pp. 1734-1744

43 Carminati, B. & Ferrari, E., Access control and privacy in web-based social networks, International Journal of Web Information Systems, 4(3) (2008) 395-415.

44 Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., OWLWeb Ontology Language Reference. http://www.w3.org/TR/owl-ref/ (2004)

45 Motik, B., Sattler, U. and Studer, R., Query Answering for OWL-DL with Rules. International Semantic Web Conference 2004, SpringerLink, 2004, pp. 549-563.

46 Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B. and Dean, M., SWRL: A Semantic Web Rule Language Combining OWL and RuleML. http://www.w3.org/Submission/SWRL/ , 2004.

47 Brickley, D. & Miller, L., FOAF Vocabulary Specification 0.91. http://xmlns.com/foaf/spec/20071002.html, 2007.

48 Brickley, D., FOAFCorp - Corporate Friends of Friends. http://rdfweb.org/foafcorp/intro.html , 2003.

49 Davis, I. and Vitiello, E., RELATIONSHIP: A vocabulary for describing relationships between people. http://purl.org/vocab/relationship/rel-vocab20050810, 2005.

50 Nasirifard, P. & Peristeras, V., CoVoc: A vocabulary for describing collaborations and e-Professionals. http://purl.oclc.org/vocabulary/covoc/covoc-20080331, 2008.