A Semantic Approach to Access Control and Credential Negotiation for Decentralized Online Repositories[*]

An OKKAM Project Use Case[**]

Hristo Koshutanski and Antonio Maña

University of Malaga
Campus de Teatinos, Malaga, Spain.
hristo@lcc.uma.es, amg@lcc.uma.es

Abstract. OKKAM project aims at enabling a web of entities by providing an infrastructure of decentralized online repositories, each owned by either a public or a private organization. A repository is designed to handle a large number of entries (as the Web identities are) where creators of entries are end-users.
In this paper we present a semantic approach to access control that naturally scales to the large number of entries in a repository and defines a flexible association of access policies and repository entries based on semantic attributes. An automated enforcement of access control policies is presented allowing users to automatically establish necessary access rights with online repositories, and interoperate their credentials based on semantics of credential interoperability.

1 Introduction

Online repositories comprise a vast catalog of database functionalities. The OKKAM project challenges the way entities are identified on the Web. The notion of entity in OKKAM is not restricted to computing entities. It refers to any entity such as people, places, real world objects, etc. The core OKKAM notion is the use of repositories for storage and retrieval of identity information on the Web. OKKAM provides a specific infrastructure for identity specification and a query process. Each identity in a repository has a unique identifier and a set of (minimal) attributes characterizing the identity. The attributes have different classification of protection and sensitivity. The project adopts an open attribute schema approach where the creators of identities define their own schema of attribute description.

[*] Hristo Koshutanski was supported by the Marie Curie EIF iAccess (#038978) fellowship of the 6th Framework Program of the European Commission. Antonio Maña was supported by the project EU-FP7-IP OKKAM (GA#215032) of the 7th Framework Program of the European Commission.

[**] EU-FP7-IP OKKAM project: http://www.okkam.org

Online distributed repositories. Potentially any company or organization can provide its own definition of identities specific for a given domain. Since the Web of entities is itself unbounded in terms of number of identifiable objects so OKKAM repositories inherit the large uncontrolled definition and expansion of identities. The repository infrastructure has to address the heterogeneity of organizations' requirements holding OKKAM repositories, scalability and manageability of large number of identities while, at the same time, it has to adapt to the dynamic nature (evolution) of the Web of entities.

OKKAM defines a decentralized peer-to-peer interaction model for repository storage, replication, entries creation, update and query. There are two types of interactions: repository-to-repository and user-to-repository. Repository-to-repository interactions occur because of query forwarding (e.g., from a private to a public node) or repository replication/synchronization. User-to-repository interactions occur when users query a repository or wish to perform an action on a repository (create, modify, or delete entries).

An OKKAM network is identified as having two types of nodes: public and private nodes. Public nodes are owned by a non-proprietary owner while private nodes are owned by proprietary organizations or companies. The public domain is open to potentially unknown users while the private domain has the specifics of being under the control of a proprietary organization.

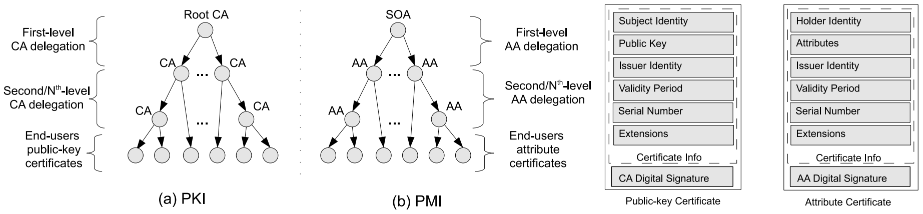
Unbounded number of entries in a repository. An OKKAM repository is expected to have unbounded number of entries. There is neither an upper-bound of expected number of entries nor a requirement to limit them. An OKKAM repository is designed to handle unbounded number of creators of entries. Any user is a potential creator of an entry or multiple entries. The use of resources, entries or identifiers (IDs) is interchangeable throughout the document and they all refer to identity elements in a repository.

We will focus on user-to-repository interactions as these comprise the main OKKAM functionality. Repository-to-repository interactions can be handled similarly to the user-to-repository interactions in the case of query forwarding (e.g., a private node becomes as a client when querying a public node).

Access control and certification management. Trust in the OKKAM infrastructure is based on certificate authorities that qualify OKKAM repositories and user privileges by means of digital certificates. The access control process will be based on digital certificates attesting users' roles and access rights. Digital certificates are well suited for decentralized peer-to-peer identification and authorization. Management of digital certificates as well as privacy of their usage is a key issue in OKKAM.

1.1 Paper contribution: Interactive semantic access control

In this paper we present a model that converges a semantic access control approach [1] with an automated credential negotiation process [2, 3]. The model is based on a semantic definition of access policies and resources being protected. An access policy refers to a set of semantic properties which are matched against the semantic properties of resources.



Hierarchical models

Certificates structure

Fig. 1. PKI/PMI

The interactive access control model is applied on top of the semantic access control specification for on-the-fly establishment and enforcement of necessary access rights between user-to-repository (or repository-to-repository) interactions. The automated trust negotiation process is extended with semantic interoperability of credentials [4] to handle when different credentials (coming from different domains) have a same level of access control (semantic equivalence) and provide efficient cross-domain access rights establishment.

Section 2 provides a background on existing identity and attribute certification. Section 3 reviews the core certification model of the OKKAM infrastructure. Section 4 overviews the semantic access control model. Section 5 describes the negotiation model and its association to semantic interoperability of credentials. Section 6 concludes the paper and outlines the future work.

2 Background on Certification Infrastructures

Public Key Infrastructure. A Public Key Infrastructure[1] (PKI) [5] provides a framework for managing identity information in a decentralized and open environment. At the core of PKI is the notion of a pair of keys (a public and a private one). Each pair of keys belongs to a unique individual and the private part of the key pair is kept secret by this individual. A key holder therefore has the unique capability of encrypting data using the private part of a key pair. The public part of the key pair allows any third party to verify that data have been encrypted with the corresponding private key.

A *public-key certificate* is a digitally signed document certifying that a particular individual owns a certain public key. A public-key certificate has a special data structure and is digitally signed by a third party called the Certificate Authority (CA). A public-key certificate serves to bind a public key to an individual holding the corresponding private key, and the correctness of the binding is guaranteed by the CA [6]. A third party trusting the CA that issued the public key certificate can assume the correctness of the identity information and can verify if an individual is the owner of the corresponding private key.

[1] http://www.ietf.org/html.charters/pkix-charter.html

In this framework, confidence regarding an unknown party identity is derived from existing trust relation between the verifying party and the CA. Since there are multiple CAs, a mechanism is needed to allow verification of an unknown CA integrity. This is achieved by distinguishing between two types of certificates: end-user certificates and CA certificates. An end-user certificate is a public-key certificate issued by a CA to a subject that cannot be an issuer of other public-key certificates. By contrast, a CA certificate is a public-key certificate issued (signed) by a CA to delegate to another party the right of issuing public-key certificates. Through this certificate, the delegate takes the function of a CA which (if the delegation policy allows) can further delegate this right to new parties.

With such a model, trust in a given CA can be built through the establishment of a *certification path* from an end-user certificate to a trusted CA's certificate. A certification path corresponds to an unbroken chain of CAs' public-key certificates that serves as a proof (of trust) to authenticate parties. Figure 1(a) shows a graphical representation of the hierarchical trust relationship established between CAs. In this hierarchy, the CA with the highest authority is called the root CA and has a self-signed certificate.

Privilege Management Infrastructure. A *Privilege Management Infrastructure* (PMI) [5] is a model similar to PKI but catering for attribute assertion instead of the binding between public key and identity. PMI has the same model as PKI regarding the delegation of authority for certificate issuance and end-user certificates. In a PMI context, a CA is called an Attribute Authority (AA), a root CA is called a Source Of Authority (SOA), and a public-key certificate is called an attribute certificate. Figure 1(b) shows the PMI hierarchical model. Note that an AA and a CA have different functional roles but may operate under a same physical authority.

An attribute certificate binds an attribute statement (or a set of attribute statements) to an individual. Trust in such certificate is derived from the trust in the AA authority to issue specific attributes. For instance, the University of Trento has the authority to state that Mario Rossi is an associate professor in this university. If a party trusts the University of Trento, then it can trust its assertion about Mario Rossi being employed there or not. Figure 1 shows the core structure of the certificate document used in both models.

Currently, only few PKI/PMI compliant implementations have reached a very broad deployment states. The most widely used such standards today are X.509 [7], SPKI[2] [8] and SAML[3] [9]. They all support expression of identity and of attribute/value information following the general model presented.

3 Core PKI/PMI model of OKKAM

In the following we will review the core certification model suitable for the OKKAM public and private domains. Trust in OKKAM infrastructure and its

[2] Simple Public Key Infrastructure

[3] Security Assertion Markup Language

authenticity is an essential element that we will start from. In the model we will unify CAs and AAs under one physical authority, i.e. any entity identified as CA will have the authority to issue not only public-key certificates but also attribute certificates. Hereinafter, whenever we refer to a CA we will implicitly refer to its ability as an AA.

A root CA, as in any PKI model, is the entity that trust starts from. The root CA is the most sensitive authority which will set up trust in the OKKAM infrastructure. As such, the root CA will have the authority to identify subordinate CAs and delegate the right to be a CA. Each node has an OKKAM repository under its own administration. Each node has a CA that has the authority to manage public-key and attribute certificates only to the scope of its domain.

The root CA has the responsibility to certify any OKKAM node so that any third party can verify (trust) that node as being part of the infrastructure. Thus, the first level authority delegation forms the trust in the OKKAM infrastructure where each CA corresponds to only one OKKAM node. Note that the first level delegation includes public and private OKKAM nodes, which means even proprietary nodes are to be certified by the root of trust.

Next level of delegation includes the end-user certification. Within a CA's authority each entity should register as a specific user type. The root CA will never issue end-user certificates since its role is to serve as a root of trust and, as such, it will only authorize OKKAM nodes to be part of the infrastructure.

Across all public nodes there will be a common classification of users to roles and responsibilities that each CA should conform to. Regarding private nodes, the common classification of users to roles may differ due to specific organizational settings. However, if private nodes want to share part of their users with those of public nodes, or vice versa, they have to conform (interoperate) with the public domain of users to roles classification. Semantics of credential interoperability is discussed in the following section.

4 Semantic Access Control

The semantic access control model (SAC) [1] proposes the use of semantic properties of resources to decouple the standard syntactic relation of resources and their respective access policies. SAC access control specification deals with a semantic abstraction.

Semantic definition of access policies. The SAC model defines a Semantic Policy Language (SPL) in order to express access policies in terms of credential definition and requirements applicable to some semantic properties. A separate specification called Policy Applicability Specification (PAS) is used to dynamically relate policies to objects based on semantic information of objects. Additionally, each object (identifier) is described by a set of semantic properties so that when a request for accessing an object arrives, the access control module derives the relevant object properties and evaluates all PAS specification to select which semantic policies match the object properties. Both SPL policies and PAS use semantic information about objects, and other contextual information. Additionally, policies can be composed using imported elements from other policies without ambiguity. This compositional approach allows us to define an abstract meaning of policy elements which helps in reducing the complexity of management.

By using the semantic definition of policies one can specify independent policies, so that, for each newly generated entry the user does not need to explicitly define a policy for the entry but to specify some entry's properties protected by some of the predefined policies. The notion of predefined semantic policy definitions is relevant to the OKKAM's nature where potentially any user can create identities but the same is not expected to be aware of how to define (qualify level of protection) for the identities it creates. SAC will allow a policy developer to define even policies for fine grained actions on objects (e.g., modify, merge, delete, split etc.).

In the following we show simple (rather informal presentation of) SPL policies and a corresponding PAS specification. The first SPL policy defines the credential requirements for a trusted OKKAM user. The second policy defines the requirements for an empowered administrator (with a delete authorization permission). The third SPL policy (#OpenAccessPolicy) defines no access restrictions.

SPL policy identifier	**Credential requirements**
#TrustedUserPolicy	registered user OR administrator
#PoweredAdminPolicy	administrator AND delete authorization
#OpenAccessPolicy	

The following PAS defines some policy applicability on objects' properties and operations.

Object property	**Property operations**	**Applicable policies**
Social_security_number	read	#TrustedUserPolicy
Social_security_number	modify	#TrustedUserPolicy
Social_security_number	delete	#TrustedUserPolicy
OKKAM_ID	read	#OpenAccessPolicy
OKKAM_ID	modify	#PoweredAdminPolicy
OKKAM_ID	delete	#PoweredAdminPolicy
#ANY#	read	#OpenAccessPolicy
#ANY#	modify	#TrustedUserPolicy
#ANY#	delete	#TrustedUserPolicy

For simplicity we only consider operations: read, modify and delete. The PAS states that a property Social_security_number on read, modify and delete operations is protected by a trusted user policy, meaning that only identifiable users are allowed to access the Social_security_number of objects in a repository.

The second protected attribute is an OKKAM_ID defining objects' identifiers, as generated by a repository system. OKKAM_ID is assigned uniquely to each identity object. Read access to an object's OKKAM_ID is unprotected (any user is allowed to see it), while modify and delete operations are restricted to only empowered administrators. Since a delete operation is very sensitive in

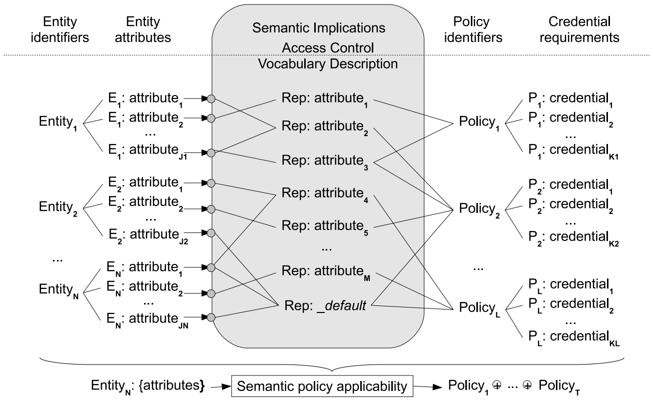


Fig. 2. A semantic policy applicability process

the OKKAM settings only administrators with an explicit delete authorization permission are allowed to perform the operation.

The third attribute identifier (#ANY#) serves as a default protection on attributes not explicitly protected by an access policy. The PAS here defines a default read operation to be unprotected, while modify and delete operations are only allowed to trusted users.

Now, if a user created an object in a repository with properties First_name, Last_name and Social_security_number (for example generating its own identity information), on receive of a request and action on that identity the following semantic policy composition would apply:

Object property	**Property operations**	**Credential requirements**
OKKAM_ID	read	
OKKAM_ID	modify	administrator AND delete authorization
OKKAM_ID	delete	administrator AND delete authorization
Social_security_number	read	registered user OR administrator
Social_security_number	modify	registered user OR administrator
Social_security_number	delete	registered user OR administrator
First_name	read	
First_name	modify	registered user OR administrator
First_name	delete	registered user OR administrator
Last_name	read	
Last_name	modify	registered user OR administrator
Last_name	delete	registered user OR administrator

Attribute-based access control vocabulary for policy applicability. One of the main project challenges is to allow end users to use their own attribute schema for describing identities on the Web. This open attribute schema poses an inherent challenges to the access control protection on identities in a repository. An approach adopted is to define a predefined access control vocabulary for generic attribute protection, so that creators of identifiers can use it to semantically describe the attributes of the identities they create. The motivation is that creators best know the exact semantics behind the attributes they define when generating identities. In this case, for each attribute of an identity, the creator can "describe" it by means of the pre-defined access control vocabulary, thus providing a semantic description of protection on the given attribute.

The predefined access control vocabulary is to be easy perceived by the end users and, at the same time, to be flexible in describing desired level of protection on a priori unknown attribute schema. To address the last feature we need to define an evolvable access policy vocabulary that best qualifies, at a given time, the protection requirements of identities' attributes.

Thus, on entity creation time, the creator can use a semantic policy language, such as OWL [10] for describing the newly defined attributes by means of the access control vocabulary. In this way, all repository entries will have an additional descriptor of a semantic access control specification of their attributes.

Figure 2 shows a semantic policy applicability process. The semantic applicability layer evaluates for a given repository entry and its attributes what access control qualification it applies based on the entity semantic access control description. We have specifically defined vocabulary attributes managed by a repository administrator. An inportant aspect here is the resulting policy composition out of the semantic policy applicability process. A research direction is to explore how to approach a possible policy composition based on what semantic information current protected attributes may leak out of the identity when used together.

5 Automated Negotiation with Semantic Interoperability of Credentials

Most authorization approaches are based on locally issued credentials (attributes or privileges) bound to user identities. When these schemas are applied to open distributed settings, like the OKKAM nature, they result in limited and inconvenient credential management with a lack of interoperability of attributes. It is unlikely to expect that different heterogeneous systems would unify a common homogeneous set of authorization criteria.

The SAC model introduces an extension of the core semantic policy definition [4] to address potential semantic interoperability of credentials issued from different CAs. The notion of Source Of Authorization Description (SOAD) defines how third-party CA's roles (certifiable attributes) qualify to CA's own roles/attributes certifying its own users. The foundation of this approach is that trust in CA's authority to certify users with certain roles implies trust in that CA to define bilateral semantic implications of how other CAs roles inter-relates with its own roles. The SOAD metadata model provides an essential semantic information to be considered in a process of decision making.

For example, a user in a public domain could be either a registered user or an administrator user of a node under the CA's authority, while in the private domain a user could be qualified with different attributes, for example, being a PhD_student, or an assistant, or a professor at an Italian university.

Let assume that the OKKAM public domain wants to open (semantically expand) access to its repositories to professors at Italian universities by qualifying them as having an administrator privileges. To do so, each CA of a public node will include in its respective SOAD the following semantic implication:

$$Administrator@CA_{OKKAM} \leftarrow Professor@CA_{CRUI}.$$

The rule states if an entity is a recognized professor at CRUI[4] then the entity obtains semantically equivalent permissions of a local administrator at an OKKAM public node (under a given CA_{OKKAM}). The implication essentially *delegates trust* to CRUI as an external equivalence for recognizing Italian professors. Now, let the SOAD of CA_{CRUI} have the following implications identifying the role professor at University of Trento, University of Bari and University of Roma as a recognized professor role at CRUI:

$$\begin{aligned} Professor@CA_{CRUI} &\leftarrow Professor@CA_{UTrento}.\\ Professor@CA_{CRUI} &\leftarrow Professor@CA_{UBari}.\\ Professor@CA_{CRUI} &\leftarrow Professor@CA_{URoma}.\\ &\ldots \end{aligned}$$

If a professor at the University of Trento tries to access a public OKKAM repository, the enforcement module will recognize his position as a CRUI professor and will provide him with equivalent permissions to those of a local administrator. The semantic interoperability approach assumes that CA's SOADs are networked and known by entities trusting the CAs. In our example, the public repository obtains (or cached from previous connections) the SOAD of CRUI and all other SOADs of external trust implications.

Automated credential negotiation. OKKAM management policy postulates that all CAs under a public domain will have a same level of trust in each other, for example, a registered user/administrator at one node is recognized as a registered user/administrator at another node. This can be flexibly achieved by uniforming the role names among all public nodes. However, between public and private nodes trust relationships are established on a bilateral manner and often the necessary access rights for a resource are unknown at a request time.

The work in [2] proposes an interactive access control model (IAC) where a server interacts with a client requesting him for missing credentials (attributes) necessary to grant a resource. The client, on its turn, checks if it has the requested credentials and sends a response back to the server. The server re-evaluates an access policy to verify if the returned set of credentials grants access to the object. In case the client does not have all credentials from the first round, the server re-computes a new set of missing credentials and asks them to the client. Thus, a client and a server interact until either the client presents a set of credentials

[4] Conferenza dei Rettori delle Università italiane: http://www.crui.it

satisfying server's access policy or there is no missing set to be asked to the client and the server denies access.

The work in [3] defines an extension to IAC where a client entity is also empowered as having its own IAC reasoning so that whenever a server asks a client for a set of missing credentials the client computes, according to its own credential control policy, what missing credentials the server has to present to see the client's credentials. The extension defines a negotiation protocol, on top of IAC model, allowing a client and a server to interact until an agreement is reached and the server provides access to the requested object, or one of the parties denies the negotiation process[5].

Let us take our own example of a repository (a server) that protects an OKKAM identity object having the attributes OKKAM_ID, Social_security_number, First_name and Last_name, and its associated semantic access policy as defined in the previous section. Figure 3 shows an example of a possible negotiation scenario between a user and a repository server.

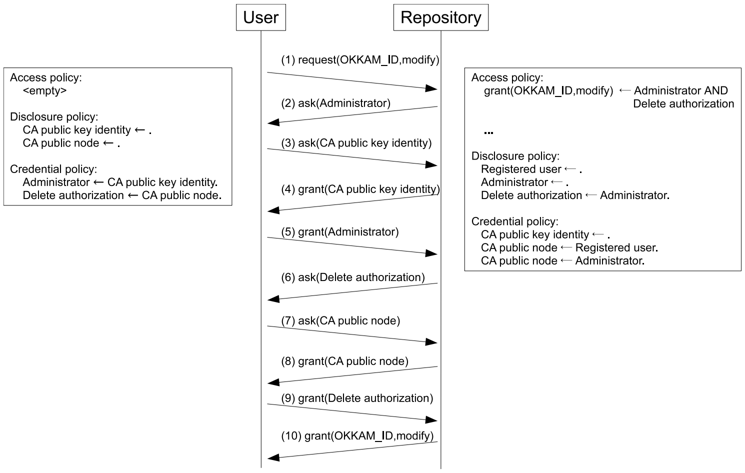

User
Repository
Access policy:
<empty>
Disclosure policy:
CA public key identity ← .
CA public node ← .
Credential policy:
Administrator ← CA public key identity.
Delete authorization ← CA public node.
(1) request(OKKAM_ID,modify)
(2) ask(Administrator)
(3) ask(CA public key identity)
(4) grant(CA public key identity)
(5) grant(Administrator)
(6) ask(Delete authorization)
(7) ask(CA public node)
(8) grant(CA public node)
(9) grant(Delete authorization)
(10) grant(OKKAM_ID,modify)
Access policy:
grant(OKKAM_ID,modify) ← Administrator AND Delete authorization
...
Disclosure policy:
Registered user ← .
Administrator ← .
Delete authorization ← Administrator.
Credential policy:
CA public key identity ← .
CA public node ← Registered user.
CA public node ← Administrator.

Fig. 3. Automated negotiation scenario

The disclosure policy on the repository site states that the need of roles Registered_user and Administrator is disclosed on demand to potentially any requestor, while the need of Delete_authorization permission is disclosed to those clients who have presented their Administrator role. Since a delete permission is sensitive information its need is only disclosed to administrator users.

The credential policy states that the repository own public-key certificate is given to anybody who needs to see it, while the repository own credential of being a public node is given only to registered users or administrator users, thus ensuring only legitimate system users can obtain the node public status.

[5] iAccess software: http://www.interactiveaccess.org

Let us look at the user side. We note that a user has no access control policy since it provides no resources/objects. The user has an Administrator certified privilege which is given only to repositories identified by their public-key certificates, and a Delete_authorization permission given only to repositories running under an OKKAM public domain.

Automated negotiation with semantic interoperability of credentials. An important aspect of a cross-domain semantic access control process is the ability of opponents to establish an agreement on necessary access rights whenever possible. The use of SOADs, as part of the SAC model, provides a scalable solution to inter-domain semantics of credentials. The challenge here is to customize a negotiation process to allow SOAD documents to be evaluated on the fly achieving a smooth and automated credential agreement.

If the user in Figure 3 is replaced with a user belonging to a private node, let us take the example of a professor at the University of Trento. In this case, a solution is to allow a repository server along with its request for an Administrator role, to communicate to the client the SOAD specification imported from the CA authoring the public repository. Since the user has no administrator role, the same can evaluate the attached SOAD specification, verify if the CRUI's SOAD qualifies professors at the University of Trento, and derive the semantic equivalence of his professorship with the local administrator authority.

Next, the user evaluates what counter requirements it has for giving its professorship certificate and counter-requests the repository. Analogously, the user will also send a SOAD specification of the University of Trento's CA along with the counter request. In that way, the repository evaluates what equivalent credentials it has with respect to the university's SOAD and proceeds accordingly. The negotiation process continues until a semantic credential agreement is reached or denied. Figure 4 shows the newly identified semantic interoperability layer for automated trust negotiations. The semantic layer is bootstrapped from SOAD specifications of CAs trusted to the two opponents.

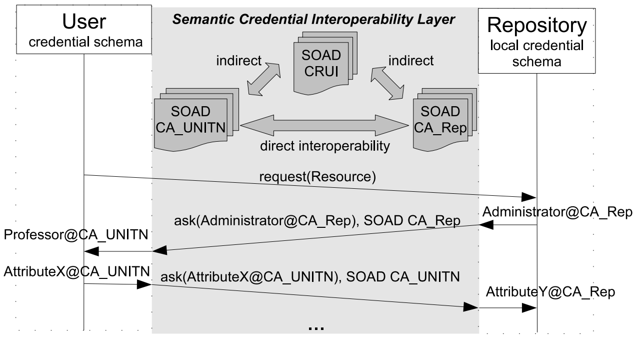


Fig. 4. Automated trust negotiation with a semantic interoperability layer

6 Conclusions and Future Work

We have presented a semantic access control approach, its extension for semantic interoperability, and its automated negotiation-based enforcement suitable for decentralized online repositories in the context of the OKKAM project. We have shown how the semantic access control model could scale to the large number of entries in a repository. We have introduced an extention to an automated credential negotiation model that provides entities with ability to bilaterally establish necessary access rights using semantic interoperability of credentials.

Future work will follow on the research directions below.

- Research on a suitable access control vocabulary and representation, and how to define policy compositions based on the semantics of repository entries' attributes.
- Integration of the semantic interoperability of credentials model into an automated negotiation process, and how to scale to deriving indirect interoperability of credentials. The use of the OWL as a description language of SOAD-based documents.

References

1. Yagüe, M., Maña, A., López, J., Troya, J.: Applying the semantic web layers to access control. In: Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03), IEEE Press (2003)
2. Koshutanski, H., Massacci, F.: Interactive access control for autonomic systems: from theory to implementation. ACM Transactions on Autonomous and Adaptive Systems (TAAS) **3**(3) (August 2008)
3. Koshutanski, H., Massacci, F.: A negotiation scheme for access rights establishment in autonomic communication. Journal of Network and System Management **15**(1) (March 2007)
4. Yagüe, M., Maña, A., Sánchez, F.: Semantic interoperability of authorizations. In: Proceedings of the 2nd International Workshop on Security In Information Systems (WOSIS'04), Porto, Portugal, INSTICC Press (2004) 269–278
5. ITU-T: The directory: Public-key and attribute certificate frameworks (2005) ITU-T Recommendation X.509:2005 | ISO/IEC 9594-8:2005.
6. Ferguson, N., Schneier, B.: Practical Cryptography. Number ISBN 0-471-22357-3. Wiley (2003)
7. ITU-T: The directory: Authentication framework - 08/05 (2005) ITU-T Recommendation X.509, available at http://www.itu.int/rec/T-REC-X.509-200508-I.
8. SPKI: SPKI certificate theory (1999) IETF RFC 2693.
9. SAML: OASIS Security Assertion Markup Language (SAML) (2005) www.oasis-open.org/committees/security.
10. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL web ontology language reference (February 2004) http://www.w3.org/TR/owl-ref.