Access Control Policy Composition for Resource Federation Networks Using Semantic Web and Resource Description Framework (RDF)

Vincent C. Hu, Stephen Quirolgico, Karen Scarfone
National Institute of Standards and Technology
vhu@nist.gov, steveq@nist.gov, karen.scarfone@nist.gov

Abstract-*The availability of global, pervasive information relies on seamless access to federated resources through sharing and trust between the participating members. However, most of the current architectures for federation networks are designed based on a centralized authorization management schema that limits the dynamic composing, organization, and reuse of federation access control policies. A schema for such environments has not been well thought out. In this paper, we present an innovative schema using Semantic Web technology that leverages the pervasive capability of semantic content and the fluency of machine understandable knowledge for access control policy in federated environments.*

Keywords: Access Control, Policy, Semantic Web.

1. Introduction

The availability of pervasive information will be greatly facilitated through the increase of globally distributed, interconnected information services. To support this global architecture, services residing within a group of local networks (or federations) interact with services residing in other federations. All member federations together form a federated network. To achieve networking in a global-computing framework, it is necessary to facilitate seamless access to federated services through inter-federation resource sharing and inter-trust between limited numbers of participating members of the global federation.

Federated resources such as software, data, and hardware components are managed by diverse organizations in widespread locations. The nodes, members, or computers of a federated network are able to act independently without centralized control, but the Trust Domain (TD) (i.e., the coverage of the authentication and authorization for the global access) is managed under a centralized system for most of the current federation architectures [1, 2, 3]. The reason is that the management of access control (AC) on a multi-organization global environment does not scale well, and it works only at the resource level, not the collective level [4, 5], making the centralized mechanism today's main solution of AC management. This approach not only inherits the limitations of centralized systems [6] but also restrict the pervasiveness of trust management in the federated network.

The difficulties lie under the usual case that the shared resources of a federation are available both locally and conditionally globally; to not violate the principle of reference monitor, both the local and global AC policies are integrated under one static AC management system. Therefore, it is challenging to: 1) specify AC rules that manage the dynamic trust relations among federated parties, 2) separate local resource AC policy from global (federation) policy, thus risking the possible leaking of authorization, and 3) share the AC profile among federated members providing similar services.

Many have researched the criticality and requirements [7] for the interaction between global (or federation) and local AC policies, but few have discussed practical approaches for solving the problems. One reason is that most AC mechanisms and models are not flexible enough to arbitrarily combine and compose AC policies [8]. In this paper, we present an innovative method of AC policy composition using Semantic Web technology that leverages the pervasive capability of semantic content and the fluency of machine understandable knowledge for the management of federated resource AC.

This paper contains seven sections. Section 1 introduces the motivation for the research. Section 2 defines the generic federation model our paper is based on. Section 3 describes AC policy components for resource federation. Section 4 discusses the Semantic Web and Resource Description Framework (RDF) for AC management followed by the application of RDF ontology. Section 5 reviews related works and discusses how they can include the proposed schema.

Section 6 is the conclusion, and Section 7 liste the references.

2. Generic Federation Model

Federated resources are distributed and shared by interoperating between three services:

• **Resource Provider** (RP) stores the information for sharing with federated members. The information is managed locally by the resource contributors or administrators of the RP. The availability and integrity of the resource is the central operation goal.

• **Resource Manager** (RM) is responsible for locating the resources in response to the access request from a Resource Consumer. The security and accessibility of communications between the Resource Providers and their connected Resource Consumers are the prime concerns of an RM.

• **Resource Consumer** (RC) is a client application that accepts user requests for resources and forwards those requests to an RM.

Ideally it is expected that there is only one RM that an RP has to communicate with, because the dissemination of shared resources is achieved by the RM. Only one connection between an RM and an RC is expected as well, because the discovery of resource locations should be done by an RM, as illustrated in Figure 1.

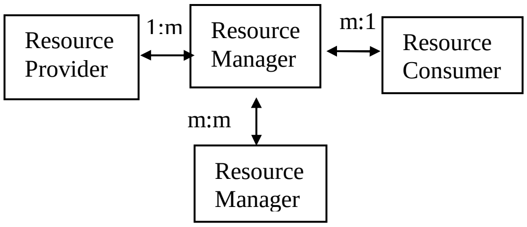

Figure 1. Generic resource federation model

In reality, a federated community may be networked in a variety of architectures. The three basic services may be incorporated or simplified such that more than one service is managed or hosted in one physical system. However, we assume these three services and their connections are essential for any resource sharing federation, and the resource sharing protocols between them are composed by interlacing and/or recursively by the following scenarios:

Scenario 1: The information request from an RC is sent to an RM and then relayed to an RP directly without passing the request to other RMs or RPs.

Scenario 2: A resource query cannot be satisfied by the connected RP, so the RM must collect and consolidate the partial results returned from more than one RP.

Scenario 3: An RM does not have a direct or static connection to any RP that is able to provide the information as requested, so the resource discovery protocols need to be invoked for exchanging information with other RMs that may have connections to other RPs that have locations for the resources.

3. Access Control Policy Components for Resource Federation

To support accessibility and maintain the integrity of resource sharing in the above model, the AC policies between the three services are required such that a service has its own policy for the federation. Figure 2 illustrates a generic scheme of a resource federation network and AC policies associated with each of the services.

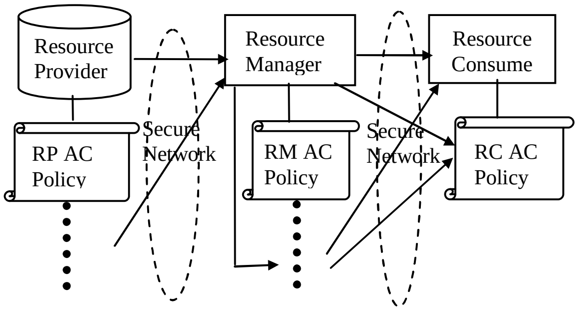

Figure 2. Resource Federation scheme

Besides the security between services in the lower level communication mechanism (e.g., through a PKI infrastructure), support of the federation according to the AC policies posted by the services requires AC functions to be implemented. These functions manage and manipulate the types of enforcement rules listed below. Here we assume that the policy for each service is maintained locally by the administration of the service.

RP –

(*p1*) share (or conditional) rules

(p2) non-share (or conditional) rules

RM –

(*m1*) list of trusted RPs

(*m2*) list of trusted RMs

(*m3*) credibility rules for *m1* and *m2* (ex. RP *A* has more credential than RP *B*)

(*m4*) priority rules for *m1* and *m2* (ex. RP *A* can be replaced by RP *B*—*A* "is a replacement" of *B*)

(*m5*) reference rules (information from RP *A* is composed of information from RPs *B*, *C*, and *D*—*A* "should be supplemented by" *B*, *C*, and *D*)

(*m6*) mediation rules (information from RP *A* cannot conflict with information from RP *B*)

RC –
(*c1*) reference rules (similar to the reference information in RM except at the application level such as logic operations (AND, OR, XOR) between collected information)
(*c2*) mediation rules (similar to the mediation information in RM except at the application level, such as data *a* from RM *X* cannot conflict with data *b* from RM *Y*)
(*c3*) constraint rules (for RMs, such as no information older than 10 days can be trusted)

Rules *p1*, *p2*, *m1*, *m2* and *c3* contain **resource availability** information while *m3, m4, m5, m6, c1,* and *c2* are information for **trust management**. Each rule is an AC policy assertion enforced upon two of the RPs, RMs, or RCs. Such a formal relation can be annotated as members of a set that contains the binary relations the rule set is enforced upon: *Rule_x* = {……(*Sx*,*Sy*),….}, where *Sx* service is related to service *Sy* by the enforcement of *Rule_x*, for example:
Credential = {….(*S1*,*S2*)….} says the resource from RP *S1* has more credential than RP *S2*, and
Replace = {….(*S1*,*S2*)….} says the resource from RP *S1* should be requested if RP *S2* is not available. Thus, by conventional set operations, **an AC trust management policy can be composed and combined through the Boolean or closure properties of the sets of trust management rules – Theorem 1.**

4. Semantic Web and RDF for Access Control Management

Information on the Semantic Web has a simple structure that allows knowledge to be expressed as a set of descriptive statements that define the relationship between one thing and another (e.g., "item123 has price $9.95"). The Semantic Web links an enormous amount of dispersed knowledge in a mesh that is easily processable by machines on a global scale, such as those on the World Wide Web or a federated network.

The Semantic Web is generally built on languages and technologies that utilize Uniform Resource Identifiers (URIs) [9] to represent data, usually in triples-based structures called Resource Description Framework (RDF) syntaxes. RDF provides a general, flexible method to decompose any knowledge into small pieces with some rules about the semantic (meaning) surrounding those pieces [10]. The benefit of using RDF is that the information maps directly and unambiguously to a knowledge model, which is decentralized in publicly available sites and ready for common parsers available at any server system. [11]

The W3C has developed an XML serialization of RDF, which is considered to be the standard interchange format for RDF on the Semantic Web, although it is not the only format. For example, Notation3 [12], or the subset called Turtle, is a de facto standard for writing out RDF; it is widely deployed, commonly used by Semantic Web developers, and the most important RDF notation to understand because it most clearly captures the abstract graph [10] of knowledge. Thus, we use it for demonstrations in the rest of this paper. In Notation3, we can simply write out the URIs in a triple, delimiting them with "<" and ">" symbols. For example,
<http://www.abc.com/#x>,
<http://www.abc.com/#y>,
<http://www.abc.com/#z>,
where **subject** *x* relates **object** *z* by **predicate** *y*, and to use literal values, simply enclose the value in double quote marks as:
<http://www.abc.com/#item123 >
<http://www.abc.com/#has price>
"$9.95".

Trust management rules

A trust management rule can be expressed by a relation pair (*Sx*, *Sy*) in a set that contains the type of rule such that the pair in the set *Rule_x*, which are subject *Sx* and predicate *Rule_x*, and object *Sy* form an RDF triple of an AC policy rule. For example, *Replace* = {…(*Sx*,*Sy*), …..} is translated into *Sx can_replace Sy in* RDF. Combined with Theorem 1, we conclude that **a trust policy can be specified by sets of RDF statements – Theorem 2**.

As stated in Section 2, AC management for resource federation is enforced by incorporating AC policies of local services. The authorization knowledge rendered in the Semantic Web is maintained by the AC rules and associated network services expressed in RDF statements, which provide the flexibility in resolving the three issues described in Section 1 as follows:
1) In addition to AC rules, RDF specifies the trust information between two services, for example, server *A* does not trust server *B.*
2) AC policies composed in RDF for the federated resource can be separated from the domestic AC policies that only manage resources, which are not intended to be shared with other federation members.
3) An AC profile in RDF is available for broadcasting publicly among other services for immediate use.

The example in Figure 3 illustrates use of a trust management policy in RDF as represented by a

knowledge graph, which shows a federation policy from each service's point of view.

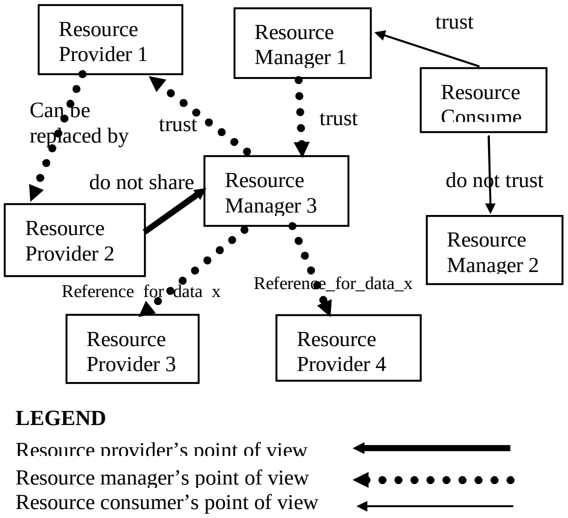

Figure 3. Resource Federation knowledge graph

Resource availability rules

The resource availability rules *p1*, *p2*, and *c3* are used to express restricting conditions for a user's access request. The format for such generic rules [13] is:

"If (*Condition_a <Boolean> Condition_b*) then (*user <action> object*)"

This prototype rule can be specified by a set of RDF statements in the following principle:

A imply B, where A = (*Condition_a <Boolean> Condition_b*), *B* = (*user <action> resource*). *A* and *B* are then further decomposed into RDF triples in which *A* has subject *Condition_a*, predicate <*Boolean*>, and object *Condition_b*. *B* has subject *user*, predicate <*action*>, and object *resource*.

Formally, the rule is specified in a set of RDF statements as:

A has_subject Condition_a
A has_predicate <Boolean>
A has_object Condition_b
Condition_a <Boolean> Condition_b
B has_subject user
B has_predicate <action>
B has_object resource
user <action> resource
A imply B

As an example, the rule: "if user (*u*) is a member of both group *X* and group *Y*, she will be *grant*ed the read (*r*) access to file *f* at time period T", formally expressed in predicate calculus:

$(u \in X \cap Y) \wedge T \rightarrow grant(r, f)$

is specified in RDF as:

Q grant_r URI:f /* *Q* represent "*read* file *f*" */
X is_the_first_Intersect_argument_of Z /* $Z = X \cap Y$ */
Y is_the_second_Intersect_argument_of Z
u is_a_member_of Z /* $u \in X \cap Y$ */
P is_ true_when_there_is URI:u
P is_the_first_AND_operand_of R /* $R = P \cap T$ */
T is_the_second_AND_operand_of R
R imply *Q* /* $R \rightarrow Q$ */

All variables in the above RDF statement are local to the service, i.e. the AC policy is locally defined, except *u* and *f*, which are globally recognizable to the federated members (therefore the URIs are required).

Alternatively, access permission may be described as a subject of an RDF triple, such as a request for subject *s* access *a* to object *o* is granted for access if conditions *C1* and *C2* are met. Its RDF triples are:

Subject	Predicate	Object
Permission	*for_request*	*(S, A, O)*
Permission	*requires*	*Decision d*
Decision d	*required_1st_AND_ argument_from*	*C1*
Decision d	*required_2nd_AND_ argument_from*	*C2*

RDF ontology

RDF can also be used at a higher level to describe RDF predicates and classes of resources. Similar to XML Document Type Definitions and XML Schema, RDF ontologies, schemas, and vocabularies provide RDF information about other RDF information. DTDs and XML Schema specify what constitutes a valid document without indicating how a document should be interpreted, and without restricting the set of elements that can be used in any given file. However, RDF ontologies (RDF Schema (RDFS) [14], Web Ontology Language (OWL) [15]) provide relations between higher-level elements indicating how some information should be interpreted in applications. The Schema uses the notion **class** to describe a type of things that possess similar attributes. For instance, paper and pen are members of the class Stationery. RDF ontologies also do not restrict at all which predicates are valid where. Any statement is valid anywhere. [10]

Features listed below, presented by RDF predicates in the RDF ontologies, provide convenient ways to compose more efficient and granular AC policies.

- **`rdf:type`** predicate relates an entity to another entity that denotes the class of the entity. The purpose of this predicate is to indicate what kind of thing a resource is. For AC policy, this predicate can be used for the "group" attribute of a user or resource in an access request described as <user, action, resource>. For example:

policy_a:user_x rdf:type status:unclassified
which means *user_x* belongs to (has attribute of) *unclassified*.

- **`rdfs:domain`** and **`rdfs:range`** predicates relate a predicate to the class of elements that can serve as the subject or object of the predicate, respectively. Using this feature, an AC policy can restrict the *domain* of users (such as users with type *x* attribute) and *range* of resources for an access permission. For example:
policy_a:copy rdfs:domain status:unclassified
policy_a:copy rdfs:range status:unclassified_files
which means only users in the class *unclassified* are permitted to *copy* files in the class of *unclassified_files*.

Additional classes defined by the OWL let AC policy authors define more of the meaning of their policy predicates within RDF. Two standard classes of predicates defined by OWL include:

- **`owl:TransitiveProperty`** can be used to describe the transitive property of an access privilege or an attribute of two users/groups in an AC policy. For example,
policy_x:higher rdf:type owl:TransitiveProperty
which means if previous RDF statement: *top_secret higher secret* and *secret higher classified*, then *top_secret higher classified* is also true for policy *policy_x* (as in Multi-Level Security policy).
- **`owl:subClassOf`** describes privilege or attribute inheritance of two users/groups in an AC policy. For example, from the following two RDF statements:
policy_x:inherit rdf:type owl:subClassOf
policy_x:group_a policy_x:inherit policy_x:group_b
user group *group_a inherit* the attributes of *group_b* for some AC policy assignments such as in a Role-Based AC policy.

Each of the above OWL classes is `rdf:subClassOf rdf:Property.` AC managers can use these classes, by convention, to make inferences to each other. Besides the standard classes provided by RDF and OWL, AC authors can also define specific classes that fit the ontologies of the AC policies applied.

5. Related Work

Many have proposed global information sharing mechanisms, such as Paranoid [16], that allow users to selectively and easily share information with others securely, with or without specifying trust relations of the sharing parties. However, these mechanisms are geared to resource sharing on the peer-to-peer level, so they provide only a limited capability for AC management on a global scale. There exist several global resource sharing techniques that apply semantic (or context-aware) with ontology context for the high-level description and reasoning of AC policies such as [17, 18] however, they either lack AC rule composition details or do not address trust management in resource federation model. The AC concept rendered in Globus's Grid Security Infrastructure (GSI) [19] (and other grid applications) contains a library and a few utilities that are used as a standard mechanism for bridging disparate security mechanisms. It not only understands identity credentials of all federation members but also supports delegation and policy distribution by translating between other mechanisms and GSI as needed and converting from a GSI identity to a local identity for authorization. In contrast to the relatively homogenous approach of GSI, OGSA [20] security envisages translation and mapping of security parameters (e.g., credentials) between different domains [8]. However, to address the issues in Section 1, the TD information in the protocol should be included in the Identity mapping services (i.e., Trust, Attribute and Bridge/Translation Services).

XACML [21] based authorization mechanisms such as Virtual Organization Membership System (VOMS), Shibboleth (with appropriate PDP implementation), PRIMA, and Privilege and Role Management Infrastructure Standards (PERMIS) [22] provide a flexible and mechanism-independent representation of access rules that vary in granularity, allowing the combination of different authoritative domains' policies into one policy set for making AC decisions in a widely distributed system environment. However, the flexibility and expressiveness of XACML make it complex and verbose. It is hard to work directly with the language or policy files. Further, supporting XACML in a heterogeneous environment calls for fully specified data type and function definitions that produce a highly verbose document even if the actual policy rules are trivial. In contrast to the above methods, RDF is free from syntactic and semantic complexity, and has only the AND operation when describing hierarchical relations between attributes or policies. As for federation policies, XACML has to be bounded by the combining algorithm in the same PolicySet, otherwise the applications themselves have to be told where to find the IDs. So AC elements are defined by a DTD or Schema, which is not extensible, and all of the members in the federated system will need to agree to a DTD or Schema change. But then there must be some guide as to what the elements of the XML files mean, and thus a central authority for deciding these things. RDF solves this problem by making everything a global ID (except literals), so basically anything the RDF application sees is an ID that means something. Thus, multiple policies can be independently implemented by assigning them to different policy providers, allowing policy decisions to be processed independently.

6. Conclusion

In this paper, we presented a schema that includes RDF framework and ontology properties for AC policies that handle federation resources management without limitations posted by the static central authorization mechanism. The basic idea is to take advantage of the dynamic features in globally recognizable frameworks such as Semantic Web for knowledge dissemination. Instead of the central management ideas of the existing architectures, the proposed schema relies on the exchange of trust and availability information between federated members, which provides freedom in composing, organizing (separate local and global), and reusing AC policies. Our proposed method not only provides a new method in communicating the AC information in the federated environment, but also brings forth a new paradigm that allows freedom for AC management in distributed networking environments. Although not all detailed AC properties are included in this paper (left for future research), we believe this schema could be used for the next generation of federation network design.

7. References

[1]. “The Open Grid Services Architecture, Version 1.0”, *http://www.gridforum.org/documents/GWD-I-E/GFDI.030.pdf*.

[2] The Globus Security Team, “Globus Tookit Version 4 Grid Security Infrastructure: A Standards Perspective”, *www.globus.org/security/overview.html.*

[3] “The Open Source Architecture”, *http://www.apac.redhat.com/software/architecture/forward.php3*.

[4] Chien A. A., “Globus Grid Security”, *CSE225 Lecture note #13*, University of California, San Diego, 2004.

[5] Hu C. V., Ferraiolo D. F., and Scarfone K., “Access Control Policy Combinations for the Grid Using the Policy Machine”, *Seventh IEEE International Symposium on Cluster Computing and the Grid, CCGrid 2007*, Rio de Janeiro, Brazil, May 2007.

[6] Hu C. V. and Scarfone K, “Decentralized Trust Domain Management in Multiple Grid Environments”, *proceeding The 2007 Community Computing workshop (CommCom2007) at UCS 2007*, November 25-28, Tokyo Japan

[7] Enterprise Grid Alliance Security Working Group, “Enterprise Grid Security Requirements”, 2005.

[8] Hu C. V., Frincke D. A., and Ferraiolo D. F., “The Policy Machine For Security Policy Management”, *Proceeding ICCS Conference*, San Francisco, 2001.

[9] Berners-Lee T., Fielding R., and Masinter L., “Uniform Resource Identifier (URI)”. *http://www.ietf/org/rfc/rfc3986.txt*

[10] Tauberer J., “what is RDF and what is it good for?” http://www.rdfabout.com/intro/

[11] “The Semantic Web: An Introduction”, *http://infomesh.net/2001/swintro/*

[12] Berners_Lee T., “Notation 3 (N3) A readable RDF Syntax”, *http://www.w3.org/DesignIssues/Notation3.html*

[13] Hu C. V., D. Kuhn, D. R., and Ferraiolo D. F., “The Computational Complexity of Enforceability Validation for Generic Access Control Rules”, *proceeding IEEE SUTC2006 conference*, Taichung, Taiwan, 2006.

[14] *http://www.w3.org/TR/rdf-schema/*

[15] *http://www.w3.org/TR/owl-features/*

[16] Abel, F., et al, “Enabling Advanced and Context-Dependent Access Control in RDF Stores”, *The 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference*, Busan, Korea, 2007

[17] Kagal L. et al, “Using Smantic Web Technologies for Policy Management on the Web”, *American Association for Artificial Intelligence* 2006.

[18] Toninelli, A. Montanari R., Kagal L., and Lassila O., “A Semantic Context-Aware Access Control Framework for Secure Collaborations in Pervasive Computing Environments”, *Proceedings, 5th International Semantic Web Conference, ISWC 2006*, Athens, GA, USA, November 5-9, 2006.

[19] “Grid Security Infrastructure”, http*://www.globus.org/Security/*

[20] “Open Grid Forum”, *http://www.ogf.org/*

[21] Lockhart H., Parducci B., and Anderson A., “OASIS extensible Access Control Markup Language” (XACML) *http://www.oasis-open.org/committees/tc-home.php*, 2005.

[22] Lang et al, “A Multipolicy Authorization Framework for Grid Security” *Proceedings 5th IEEE International Symposium on Network Computing and Applications*, 2006.