Ubiquitous Access Control for SPARQL Endpoints: Lessons Learned and Future Challenges

Luca Costabello, Serena Villata, Nicolas Delaforge, Fabien Gandon
INRIA Sophia Antipolis, France
firstname.lastname@inria.fr

ABSTRACT

We present and evaluate a context-aware access control framework for SPARQL endpoints queried from mobile.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services

General Terms

Design, Algorithms

Keywords

SPARQL, Context Awareness, Access Control

1. INTRODUCTION

In the Web of Data [6], providers expose their content publicly, knowing that it is not safe. This may prevent further publication of datasets, at the expense of the growth of the Web of Data itself. Moreover, the mobile, ubiquitous Web is continuously evolving, enabling new scenarios in consuming and contributing to the Web of Data. We must therefore not ignore the mobile context in which data consumption takes place. In this paper, we propose a context-aware access control framework for protecting SPARQL endpoints, adopting exclusively Semantic Web languages. Two main features distinguish the policies in our framework from related research: (i) triple-level granularity (using Named Graphs [2]) and (ii) the support for context information, e.g. requester location, nearby people, device features, time of the day, etc. Other works with similar scope have been proposed. We differ from WAC[1] since we go beyond RDF document granularity and we do not rely on access control lists. Sacco and Passant [7] present the PPO vocabulary[2] to express access control policies for RDF documents. Flouris et al. [5] provide a fine-grained access control framework on top of RDF repositories coupled with a high level specification language translated into a SPARQL/SerQL/SQL query to enforce the policy. Finin et al. [4] consider attribute-based access control where, similarly to our proposal, the constraints are based on general attributes of an action. Context information is supported to some extent by Abel et al. [1]. They provide triple-level access control as a layer on top of RDF stores. Contextual conditions are pre-evaluated before expanding the queries. Toninelli et al. [8] adopt context-awareness and semantic technologies for access control but they do not apply their solution to the Web of Data.

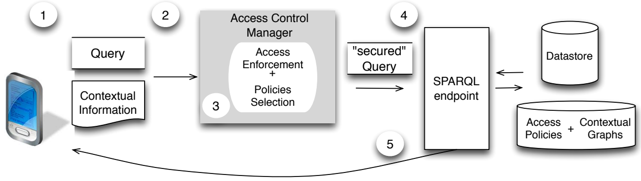


Figure 1: The access control framework architecture.

2. OUR PROPOSAL

Our system relies on two complementary lightweight vocabularies, `S4AC`[3] for access control, and `PRISSMA`[4] for modelling the mobile context.

Access Policies protect a named graph, thus targeting single triples, if needed. As seen in Figure 2, each Access Policy is associated to a privilege level and includes a set of context-aware Access Conditions, i.e. constraints that must be satisfied, conjunctively or disjunctively, to access the protected resources. Access Conditions are implemented as SPARQL ASK queries. At runtime, Access Policies are associated to the actual mobile context used to evaluate the set of Access Conditions.

For what concerns the mobile context, we agree with the widely-accepted proposal by Dey [3][5]. In our model, context is seen as an encompassing term, an information space defined as the sum of three different dimensions: the *User* model, the *Device* features and the *Environment* in which the request is performed.

Our Access Control Manager is designed as a pluggable component for SPARQL endpoints (Figure 1). The access control evaluation procedure is described below: (1) the mobile consumer queries the SPARQL endpoint. Contextual information is sent along with the query and saved as

[1] `http://www.w3.org/wiki/WebAccessControl`

[2] `http://vocab.deri.ie/ppo`

[3] `http://ns.inria.fr/s4ac`

[4] `http://ns.inria.fr/prissma`

[5] More specifically, we rely on `http://bit.ly/XGR-mbui`

```
:policy1 a s4ac:AccessPolicy;  ACCESS POLICY
           s4ac:appliesTo :alice_reviews; RESOURCE TO PROTECT
           s4ac:hasAccessPrivilege [a s4ac:Read];  ACCESS PRIVILEGE
           s4ac:hasAccessConditionSet :acs1.

:acs1 a s4ac:AccessConditionSet;
        s4ac:ConjunctiveAccessConditionSet;
        s4ac:hasAccessCondition :ac1,:ac2.   ACCESS CONDITIONS
                                                 TO VERIFY
:ac1 a s4ac:AccessCondition;
       s4ac:hasQueryAsk
       """ASK {?context a prissma:Context.
               ?context prissma:user ?u.
               ?u foaf:knows ex:alice#me.}""".

:ac2 a s4ac:AccessCondition;
       s4ac:hasQueryAsk
       """ASK {?context a prissma:Context.
               ?context prissma:environment ?env.
               ?env prissma:based_near ?p.
               FILTER (!(?p=ex:ACME_boss#me))}""".
```

Figure 2: A sample Access Policy

named graph using SPARQL 1.1 Update Language statements[6]. (2) The client query is filtered by the Access Control Manager instead of being directly executed on the SPARQL endpoint. (3) The Access Control Manager selects the set of policies affecting the client query and after their evaluation returns the set of accessible named graphs. (4) The client query is executed only on the accessible named graphs and (5) the result of the query is returned to the consumer.

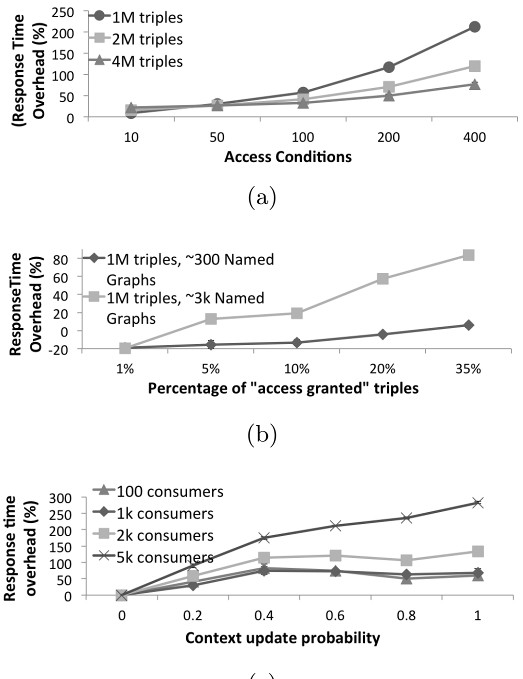
The Access Control Manager has been implemented as a Java EE component and plugged to the Corese-KGRAM RDF store and SPARQL 1.1 query engine[7]. Prototype evaluation with the Berlin SPARQL Benchmark dataset 3.1[8] shows that: (i) larger datasets are less affected by the delay introduced by our access control framework, as datastore size plays a predominant role in query execution time (Figure 3a), (ii) when the access is granted to a small fraction of named graphs, the query is executed faster than the case without control on the accesses (Figure 3b), and (iii) performance is affected by the number of active mobile consumers, each associated to a mobile context graph, i.e., the delay of the SPARQL 1.1 Update operations depends on the size of the triple store and on the number of named graphs (Figure 3c).

3. FUTURE CHALLENGES

The proposed access control framework is conceived as an easy-to-integrate pluggable filter for data servers that support the SPARQL query language. Our framework relies only on Semantic Web languages, since no other formalism has been added.

On the other hand, supporting mobile context for access control leads to several open problems. For instance, the trustworthiness of contextual information sent by mobile consumers should not be taken for granted. Context verification techniques are therefore needed, along with a mechanism to authenticate the consumer's identity[9].

We are aware that sensible data such as current location must be handled with a privacy-preserving mechanism. For instance, we may deal with access control and obfuscation rules for tracking mobile users.



(a)

(b)

(c)

Figure 3: Response time overhead

Our current framework works on top of standard-compliant SPARQL endpoints. However, the model consists in a set of general rules providing true/false answers. One further challenge is to generalize our framework to support other linked data access strategies such as *follow-your-nose*, thus decoupling our solution from SPARQL-wrapped RDF stores.

4. REFERENCES

[1] F. Abel, J. L. De Coi, N. Henze, A. W. Koesling, D. Krause, and D. Olmedilla. Enabling Advanced and Context-Dependent Access Control in RDF Stores. In *ISWC, LNCS 4825*, pages 1–14, 2007.

[2] J. J. Carroll, C. Bizer, P. J. Hayes, and P. Stickler. Named graphs. *J. Web Sem.*, 3(4):247–267, 2005.

[3] A. K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5:4–7, 2001.

[4] T. W. Finin, A. Joshi, L. Kagal, J. Niu, R. S. Sandhu, W. H. Winsborough, and B. M. Thuraisingham. R*OWL*BAC: representing role based access control in *OWL*. In *SACMAT, ACM*, pages 73–82, 2008.

[5] G. Flouris, I. Fundulaki, M. Michou, and G. Antoniou. Controlling Access to RDF Graphs. In *FIS, LNCS 6369*, pages 107–117, 2010.

[6] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.

[7] O. Sacco and A. Passant. A Privacy Preference Ontology (PPO) for Linked Data. In *LDOW*, 2011.

[8] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila. A semantic context-aware access control framework for secure collaborations in pervasive computing environments. In *ISWC-2006, LNCS 4273*, pages 473–486, 2006.

[6] http://www.w3.org/TR/sparql11-update

[7] http://tinyurl.com/corese-engine

[8] http://bit.ly/berlin-sparql

[9] http://www.w3.org/2005/Incubator/webid/spec