Performance Assessment of XACML Authorizations for Supply Chain Traceability Web Services

Miguel L. Pardal[†], Mark Harrison[‡], Sanjay Sarma[§], José Alves Marques[†]

[†]Department of Computer Science and Engineering
Instituto Superior Técnico, Technical University of Lisbon, Portugal

[‡]Auto-ID Labs, Institute for Manufacturing,
University of Cambridge, UK

[§]Auto-ID Labs, Massachusetts Institute of Technology, USA

Email: miguel.pardal@ist.utl.pt, mark.harrison@cantab.net, sesarma@mit.edu, jose.marques@link.pt

***Abstract*—Service-Oriented Architecture (SOA) and Web Services (WS) offer advanced flexibility and interoperability capabilities. However they imply significant performance overheads that need to be carefully considered.**

Supply Chain Management (SCM) and Traceability systems are an interesting domain for the use of WS technologies that are usually deemed to be too complex and unnecessary in practical applications, especially regarding *security*.

This paper presents an externalized security architecture that uses the eXtensible Access Control Markup Language (XACML) authorization standard to enforce visibility restrictions on traceability data in a supply chain where multiple companies collaborate; the performance overheads are assessed by comparing 'raw' authorization implementations - Access Control Lists, Tokens, and RDF Assertions - with their XACML-equivalents.

***Keywords*-Web Services; Authorization; XACML; Performance; Supply Chain Traceability**

I. Introduction

Service-Oriented Architecture (SOA) [1] and Web Services (WS) [2] were envisioned to cope with constant software change and to simplify the interoperation of heterogeneous systems. WS-related standards[1] address non-functional concerns such as security, transactions, and reliable messaging. However, the complexity of these standards, alone and combined, raises significant performance concerns.

The main criticism against the use of WS technologies is that they are slow and difficult to use. The performance overheads have been measured in practical implementations. Juric at al. [3] compared the performance of SOAP with platform-specific technologies, namely, with Java Remote Method Invocation (RMI). They reported that a typical SOAP message is, on average, 4.3 times larger than an RMI message, and that the WS response time is 9 times slower than RMI. If WS-Security [4] technologies are used, then the overheads are even more significant: messages are 27 times larger, and response times are 15 times slower.

The main reason for these overheads was found to be the use of verbose XML data encoding for the message formats. The data access mode is also relevant. Bayardo et. al [5] found in their study that stream-based data access gives significantly better performance than document-based access.

In this paper, we investigate the performance of using WS authorizations in the *supply chain traceability* domain. Dynamic authorization is necessary because there are multiple companies involved, they are distributed across the world, and many of them do not have prior knowledge of the others.

A. Supply Chain Traceability

The world economy depends on countless supply chains that provide goods from producer to consumer. Supply Chain Management (SCM) solutions focus on planning and execution, integrated with companies' Enterprise Resources Planning (ERP) systems. Their purpose is to optimize the physical object flows so that goods travel in the least amount of time and at the lowest cost [6]. To achieve this goal, it is necessary to *keep track* of what is moving along the supply chain.

Radio Frequency Identification (RFID) technology [7] can significantly improve supply chain traceability, by allowing automatic data capture by RFID readers of identifiers stored inside RFID tags attached to items and/or pallets. Captured data is stored by each company at dispersed data silos.

The Electronic Product Code (EPC) Network architecture [8] defines standard data capture and query interface for repositories of RFID data, called the EPC Information Services (EPC IS) [9]. EPC Discovery Services (EPC DS) [10] [11] can index these scattered repositories and allow *traceability queries* [12], such as Track (*Where is the item?*) and Trace (*Where has the item been?*) to be efficiently answered.

Each supply chain participant can benefit from exchanging data with other companies, but information such as the levels

[1] WS is used here as an 'umbrella' term that refers to most variants of eXtensible Markup Language (XML) message formats and protocols built to provide data services, including SOAP messages containing headers (metadata) and body (payload), and also Representational State Transfer (REST) that provides granular access to resources.

of demand, inventory, and supplier identities should be kept private in a restricted circle of trust. Data access control is crucial if companies will be willing to share data [13].

B. Overview

This paper compares the performance of *visibility restriction* approaches for supply chains using an *authorization* component of an *externalized security architecture* that relies on SOA principles and WS-related technologies; 'raw' implementations are compared with equivalent policies in a standard authorization language to measure the overheads.

In the next section, the traceability system architecture for the supply chain domain is described. Then the externalized security architecture is presented with specific focus on authorization. Next we describe the supply chain authorization implementations along with their conversion to a standard authorization language. The paper ends with the performance assessment, conclusions and future work prospects.

II. Traceability System

There are several architecture proposals to build traceability systems. Pardal and Alves Marques [14] summarized them using the criteria of *centralization* and *data integration*. For this work the Meta-Data Integration (MDI) architecture was chosen as the reference because it is aligned with the EPC Network architecture [8], defining two system layers:

- EPC IS servers collect the detailed event data;
- EPC DS servers store data links to EPC IS servers.

Using this approach, traceability data sharing policies can be authored by the data owner and then used both at EPC DS and IS layers, as depicted in Figure 1.

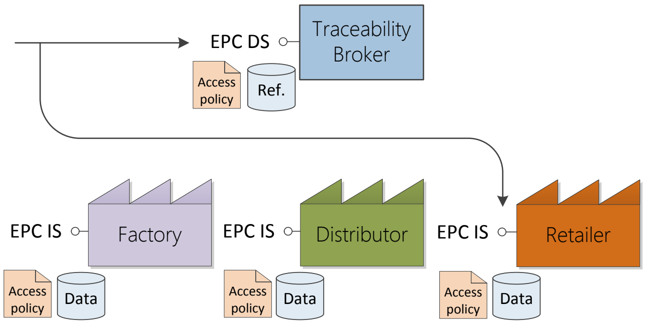

Fig. 1. XACML authorization policies protect both EPC DS and IS.

III. Externalized Security

The goal of the externalized security architecture is to unify the security management across applications so that business rules can be changed dynamically. For this reason, it is especially suited to virtualized cloud deployments [15].

Externalized security encompasses user management, authentication, authorization, logging and auditing. The security properties to be preserved from attacks are depicted in Figure 2. Companies have to be authenticated and data access must be authorized. The authentication can be achieved with identity providers [16] and the exchange of SAML assertions [17]. The authorization can be achieved with XACML [18] that is an XML vocabulary to represent authorization policies and requests that avoids hard-coded rules and allows improved consistency of policy enforcement. Hebig et al. [19] demonstrated how the different technologies needed for externalized security can work together.

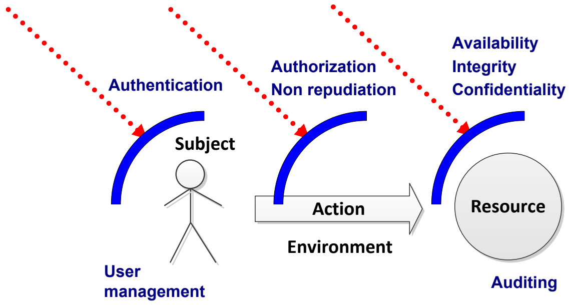

Fig. 2. Security properties.

A. eXtensible Access Control Markup Language

XACML [18] allows fine-grained access control and combination of (possibly conflicting) policies in an externalized security architecture. In general terms, an authorization is the verification of a subject's right to execute an action on a resource. The standard defines a policy format, follows a processing model and requires actual implementation.

1) Policy: Figure 3 presents a simplified XACML Policy structure with target and rules.

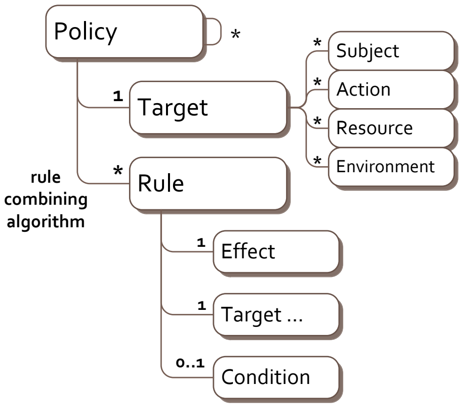

Fig. 3. XACML simplified Policy model.

The target defines a simplified set of conditions that determine if the policy is relevant for the request. Rules are conditions that evaluate to 'Permit', 'Deny', 'NotApplicable' (when no target matches), or 'Indeterminate' (when internal errors occur). XACML has a set of predefined functions that are used to transform and compare data attributes, to arrive at an outcome.

Policies are declarative and do not allow function definitions to keep algorithmic complexity low [20]. Additional functions can be provided by the library implementations, but this limits the compatibility of policy interpretation.

2) Execution: The authorization architecture that XACML assumes is defined by RFC 2753 [21] and 2904 [22], and defines several structural elements represented in Figure 4. The Policy Administration Point (PAP) is used to author and manage policies. The Policy Enforcement Point (PEP) is an application-specific component that intercepts requests to access a resource, and can also perform 'before' or 'after' actions, called *obligations*. The PEP checks with the Policy Decision Point (PDP), a generic component that takes any request along with a set of policies, and *evaluates* the request with respect to the applicable policies. The Policy Information Point (PIP) retrieves additional attribute values for the PDP not contained in the request, if required.

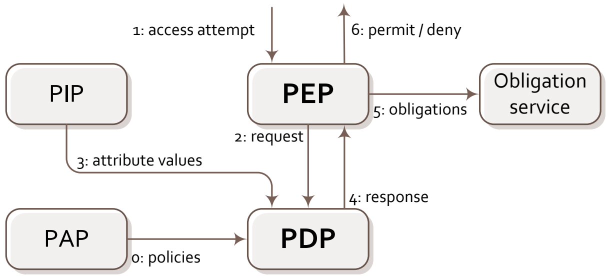

Fig. 4. XACML request processing.

3) Implementations: There are several available XACML libraries and tools, both open-source and commercial. The open-source implementations were surveyed and the findings are presented in Table I.

Impl.	Version	Last update	Sponsors
sunxacml	1.2	Jul 2004	Sun Microsystems (currently Oracle)
Heras-AF	1.0.0-M2	Sep 2010	U. Applied Sciences Rapperswil, Switzerland
enterprise -java-xacml	r258	Jan 2009	Zian Wang
PicketBox	3.0.0.Final	Feb 2011	JBoss, Red Hat
xEngine	beta 0.2	Aug 2010	Michigan State U., North Carolina State U.

TABLE I
OPEN-SOURCE XACML IMPLEMENTATIONS.

Sun XACML library is the reference implementation, sponsored by Sun Microsystems, but has not been updated since 2004. Since the project has been made publicly available, several "branches" were created by different sponsors. There is a 2.0 version under development but its sponsorship is not clear and it breaks source code compatibility.

HERAS-AF[2] [23] is a well documented library. It was developed in academia, but is currently also used in production. The open-source version has only an in-memory policy repository and the project has been inactive since 2010.

'enterprise-java-xacml' has the best reported performance [24] but has little documentation.

PicketBox is supported by JBoss and the code appears to be converging to production-quality but the library is insufficiently documented at the present time.

XEngine [25] also claims to have the best performance but it is insufficiently documented and is currently inactive.

HERAS-AF was picked for this research because of the breadth of available documentation, including open access to the source code.

IV. Supply Chain Authorizations

Access control for supply chains is different from traditional authorization because there is a lack of prior knowledge about who should be authorized due to the way that each physical object path emerges [26].

The Supply Chain Authorization (SC-Az) Application Programming Interface (API) has been proposed recently [27] to allow companies participating in a supply chain to express their authorizations using business concepts such as item, company, etc.; and then to enforce the resulting policies using alternative underlying mechanisms.

The previous SC-Az version had two implementations: Enumerated Access Control (EAC) and Chain-of-Communication Tokens (CCT). EAC uses an Access Control List (ACL) implementation available on the Java virtual machine[3]. CCT represents access rights within object references, called tokens, implemented with custom code. For this research, we added a new implementation, Chain-of-Trust Assertions (CTA), based on the Apache Jena[4] Semantic Web [28] library.

A. Chain-of-Trust Assertions

CTA is different from EAC and CCT because its semantics can be extended. Access rights are expressed using logical statements, called *assertions*, issued by the multiple parties. Figure 5 shows the assertion-related operations.

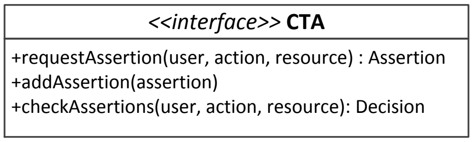
<<interface>> CTA
+requestAssertion(user, action, resource) : Assertion
+addAssertion(assertion)
+checkAssertions(user, action, resource): Decision

Fig. 5. CTA interface operations.

Access is granted if there is an explicit unbroken chain of trust assertions leading back to the owner of the data. Figures 6 and 7 show a simple CTA policy stated in Resource Description Framework (RDF) classes and properties, expressed as subject-predicate-object tuples.

In the example, policy0 created by company0 (data owner) grants read access to record0 about item0 to company0 and company1. The extensibility can be achieved by adding new properties e.g. cta:grantsWrite to grant *write* access.

[2] Holistic Enterprise-Ready Application Security Architecture Framework.

[3] Package sun.security.acl

[4] http://jena.apache.org/

```
:company0  a  cta:Organization .
:company1  a  cta:Organization .

:item0     a  cta:Identifier .
:record0   a  cta:Record .
:policy0   a  cta:Policy .

:company0  cta:publishes   :record0 .
:record0   cta:about       :item0 .

:company0  cta:creates     :policy0 .
:policy0   cta:protects    :item0 .
:policy0   cta:grantsRead  :company0 .
:policy0   cta:grantsRead  :company1 .
```

Fig. 6. CTA Policy in RDF Turtle format.

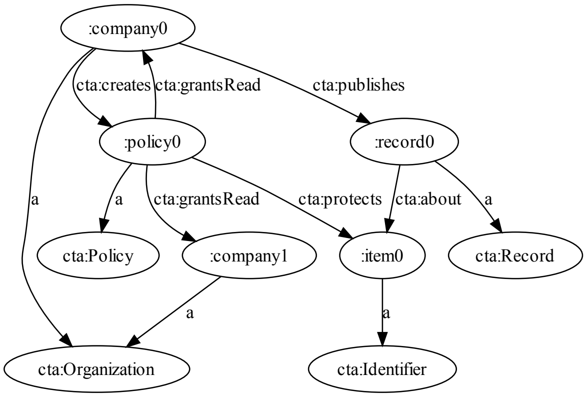

Fig. 7. CTA Policy graph.

B. Conversion to XACML

To convert a CTA policy to a XACML policy the RDF statements are navigated in the following fashion: first the objects of "pol protects *id*" statements are found. For each item identifier, a XACML policy is created. The policy target matches the item identifier. A permit rule is created to grant each access right – e.g. read – to the objects from "pol grantsRead *org*" statements. A final catch rule is created so that all other requests regarding the item are also denied. The rule combining algorithm is 'first-applicable' so the outcome of the matched first rule is the access decision.

This conversion approach is based on the policy conversions by Karjoth et al. [29], and it allows SC-Az policy instances to be represented in the standard XACML format[5].

C. Deployment

For the authorizations in the traceability system, XACML is used as the canonical format for representing data access policies, both at the EPC IS and DS levels, as depicted in Figure 1. The policy master copy is maintained at the EPC DS, but a local copy is maintained in each IS to also protect its records locally.

D. Related work

WS technologies are already widely used in RFID system implementations. The most prominent example is Fosstrak [30] that provides a SOAP endpoint to the EPC IS event repository. Also, Guinard [15] developed alternative REST-based interfaces and explored cloud-based deployment.

Shi et al. [31] implemented a secure DS and used an extended attribute-based access control to implement fine-grained access policies in detail. Their custom policy engine is specialized but does not realize the benefits of using a standard policy language, making auditing and policy validation harder. In any case, their engine can be extended to recognize XACML as an input format.

Kerschbaum and Chaves [32] propose an encryption scheme that allows different levels of fine-grained access control to be enforced on each item, also relying on the central role played by a DS.

V. Performance assessment

The aim of the performance assessment was to evaluate if the performance of the solution is suitable for potentially very large and complex supply chains.

The SC-Az assessment tool [27] was used to perform test runs to measure both from the 'raw' EAC, CCT and CTA mechanisms and from their XACML-equivalent forms.

The test machine was a Quad-core CPU[6] at 2.50 GHz, with 3.25 GB of usable RAM, and 1 TiB hard disk; running 32-bit Windows 7 (version 6.1.7601), and Java 1.7.0_04.

The policies were defined using the common SC-Az API to allow the same business needs to be represented internally by each implementation. The policies were then converted to XACML format and tested using the HERAS-AF implementation to assess the correctness and the performance.

A. Evaluation

The data sharing policies were correctly translated and enforced. The performance results are the average of repeated runs of the same experiments, to achieve statistical confidence in the data.

The number of items considered in the experiments were based on information collected by Ilic et al. [33]. We start with small number of items (100) and go up to medium number (10^4).

1) Raw: Figure 8 presents a plot of the 'request evaluation' time for increasing number of policies protecting items.

We can see that the performance of CCT implemented with custom code is clearly the worse. EAC using Java's ACLs and CTA using Apache Jena's RDF handle the loads much better with results below 0.1 ms. The reason for this difference could be that the CCT implementation is *stateful* whereas EAC and CTA are *stateless*. CCT needs to perform (unoptimized) searches in token collections to find the right token for each request while the other two receive all the values as arguments.

[5] The conversion of EAC and CCT to XACML is described in [27].

[6] Intel Core 2 Quad Central Processing Unit Q8300

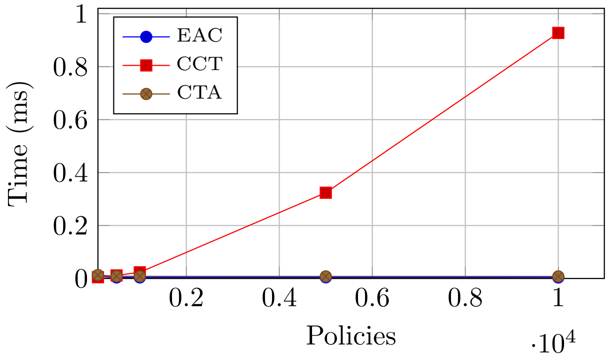

Fig. 8. Raw evaluation time with increasing number of policies.

2) XACML: Figure 9 presents a plot of the 'request evaluation' time, again for increasing number of policies converted to XACML format.

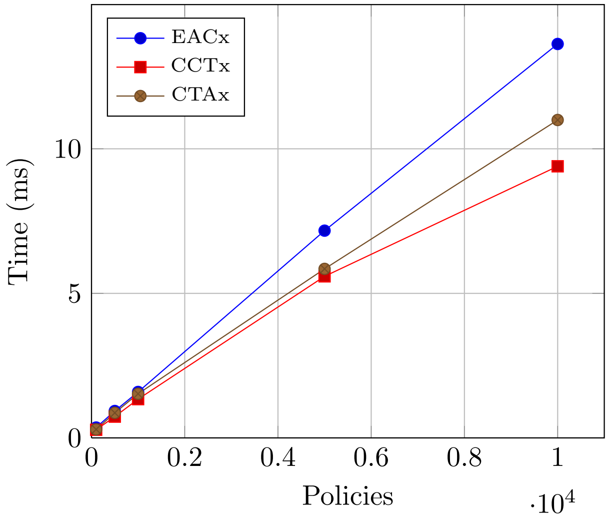

Fig. 9. XACML evaluation time with increasing number of policies.

The performance of CCT and CTA are the best, EAC is worse, but on the same order of magnitude.

3) Raw vs. XACML: Comparing the y-axis of Figures 8 and 9 it is visible that the XACML performance overheads are very significant. Table II presents each XACML time divided by the corresponding 'raw' time for chains handling increasing numbers of items protected by policies. We observed a 400-fold overhead, on average. Also, the performance overhead increases with the number of deployed policies.

Nr. Policies	EAC	CCT	CTA
$0.01 \cdot 10^4$	49.9	58.9	22.7
$0.05 \cdot 10^4$	240.0	63.9	103.5
$0.1 \cdot 10^4$	406.7	57.5	191.4
$0.5 \cdot 10^4$	1670.1	17.3	752.9
$1 \cdot 10^4$	3684.4	10.1	1429.2

TABLE II
XACML OVERHEAD WITH INCREASING NUMBER OF ITEM POLICIES.

VI. Conclusion

The first contribution of this paper is the *performance assessment* of the enforcement of supply chain data visibility policies, in the context of an externalized security architecture. First, we verified that the policies could be translated and enforced using a standard XACML infrastructure. Then we compared the performance of the 'raw' implementations with the XACML-converted policies for increasing number of items being protected. The XACML overheads were found to be very significant: 400 times on average, and over 1000 times in the worst cases.

Our results agree with previous research [3] [5] that shows that WS technologies overheads are usually very significant. There is a clear need to simplify and streamline the standards and there is room for performance improvements in the implementations. However, despite these drawbacks, WS technologies are being widely used today, meaning that their advantages – interoperability, flexibility, tooling support – add value for developers and make up for the additional cost. That is also the case for using XACML to protect supply chain data. XACML allows the policies to be exchanged in a standard format that can be properly interpreted in all the policy enforcement points and allows the use of other tools that comply with the standard (e.g. auditing tools).

The second contribution of this work is the *extension* of the Supply Chain Authorization (SC-Az) API with the Chain-of-Trust Assertions (CTA) implementation based on Semantic Web technologies. CTA has extensible semantics while EAC and CCT have predefined semantics and, because of this, CTA was expected to be slower. However, for the SC-Az baseline when all three mechanisms express exactly the same visibility restrictions there are no significant performance differences for evaluation of requests, and CTA is even better than EAC. Considering that the performance is similar and that CTA is extensible, moving forward CTA should be the default choice for expressing supply chain authorizations.

A. Future work

There are indications that the performance overheads reported using the HERAS-AF library could be significantly lowered with more optimized alternatives [34] [25]. This proposition needs to be measured in practice before more general conclusions can be drawn.

The evaluation job execution will be enhanced to measure the performance of XACML with a large number of items (more than 10^4 items [33]). The performance impact of representing object groupings – *batches* – and company sets – *groups* – will also be assessed.

The suitable performance of CTA opens possibilities for more expressive traceability data sharing policies. Trust can be *transitive* in some cases, as tested already with good preliminary results. Also, instead of issuing plain trust assertions, parties can issue *conditional* assertions, like reciprocal trust ("I trust you if you trust me"). Additionally, special predicates can be designed to express *dynamic* chain upstream/downstream conditions, allowing data sharing (or at least, initial data discovery) between parties that did not have previous interactions. It can also support the *delegation* of administrative rights from one organization to another. All of these can be particularly useful in scenarios like *recalls* [35].

The authorization challenges of externalized security in the supply chain traceability system illustrated how WS can assist in complex and dynamic business environments involving multiple organizations. Many more WS research can be done in this domain as traceability systems requirements are a worthy match for Web Services' advanced capabilities.

Acknowledgment

Miguel L. Pardal is supported by a PhD fellowship from the Portuguese Foundation for Science and Technology FCT (SFRH/BD/45289/2008).

References

[1] T. Erl, *SOA Principles of Service Design*. Prentice Hall, July 2007.

[2] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services: Concepts, Architectures and Applications*. Springer Verlag, 2004.

[3] M. B. Juric, I. Rozman, B. Brumen, M. Colnaric, and M. Hericko, "Comparison of performance of Web Services, WS-Security, RMI, and RMI-SSL," *Journal of Systems and Software*, vol. 79, no. 5, pp. 689–700, 2006.

[4] K. Lawrence, C. Kaler, A. Nadalin, R. Monzillo, and P. Hallam-Baker, *Web Services Security: SOAP Message Security 1.1*, OASIS Std., February 2006.

[5] R. J. Bayardo, D. Gruhl, V. Josifovski, and J. Myllymaki, "An evaluation of binary XML encoding optimizations for fast stream based XML processing," in *Proc. of the 13th Int'l Conf. on World Wide Web*, ser. WWW. New York, NY, USA: ACM, 2004, pp. 345–354.

[6] K. Laudon and J. Laudon, *Management Information Systems - 12th edition*. Prentice Hall, January 2011.

[7] E. W. Schuster, S. J. Allen, and D. L. Brock, *Global RFID: The value of the EPCglobal network for supply chain management*. Springer, 2007.

[8] F. Thiesse, C. Floerkemeier, M. Harrison, F. Michahelles, and C. Roduner, "Technology, Standards, and Real-World Deployments of the EPC Network," *IEEE Internet Computing*, vol. 13, no. 2, pp. 36–43, 2009.

[9] EPCglobal, *EPC Information Services (EPCIS) 1.0.1 Specification*, GS1 Std., September 2007.

[10] J. J. Cantero, M. A. Guijarro, A. Plaza, G. Arrebola, and J. Baos, "A Design for Secure Discovery Services in the EPCglobal Architecture," in *Unique Radio Innovation for the 21st Century*, D. C. C. Ranasinghe, Q. Z. Z. Sheng, and S. Zeadally, Eds. Springer Berlin Heidelberg, 2010, pp. 183–201.

[11] C. Kürschner, C. Condea, O. Kasten, and F. Thiesse, "Discovery Service Design in the EPCglobal Network, Towards Full Supply Chain Visibility," *Internet of Things*, pp. 19–34, 2008.

[12] R. Agrawal, A. Cheung, K. Kailing, and S. Schonauer, "Towards Traceability across Sovereign, Distributed RFID Databases," in *Int'l Database Engineering and Applications Symp. (IDEAS)*, 2006.

[13] M. Eurich, N. Oertel, and R. Boutellier, "The impact of perceived privacy risks on organizations' willingness to share item-level event data across the supply chain," *Journal of Electronic Commerce Research*, vol. 10, no. 3-4, pp. 423–440, December 2010.

[14] M. L. Pardal and J. A. Marques, "Cost Model for RFID-based Traceability Information Systems," in *IEEE Int'l Conf. on RFID Technology and Applications*, September 2011.

[15] D. Guinard, C. Floerkemeier, and S. Sarma, "Cloud computing, REST and Mashups to simplify RFID application development and deployment," in *Proc. of the 2nd Int'l Workshop on Web of Things (WoT)*. ACM, 2011, pp. 9:1–9:6.

[16] D. Baier, V. Bertocci, K. Brown, E. Pace, and M. Woloski, *A Guide to Claims-Based Identity and Access Control*. Microsoft, 2010.

[17] S. Cantor, J. Kemp, R. Philpott, and E. Maler, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, OASIS Std., March 2005.

[18] B. Parducci, H. Lockhart, and E. Rissanen, *eXtensible Access Control Markup Language (XACML) Version 3.0*, OASIS Std., August 2011.

[19] R. N. Hebig, C. Meinel, M. Menzel, I. Thomas, and R. Warschofsky, "A Web Service Architecture for Decentralised Identity- and Attribute-Based Access Control," in *Proc. IEEE Int'l Conf. Web Services (ICWS)*, 2009, pp. 551–558.

[20] C. Alm and R. Illig, "Translating High-Level Authorization Constraints to XACML," in *Proc. 6th World Congress Services (SERVICES-1)*, 2010, pp. 629–636.

[21] R. Yavatkar, D. Pendarakis, and R. Guerin, *RFC 2753 – A Framework for Policy-based Admission Control*, IEFT, Internet Engineering Task Force Std., January 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2753.txt

[22] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence, *RFC 2904 – AAA Authorization Framework*, IEFT, Internet Engineering Task Force Std., August 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2904.txt

[23] F. Huonder, "Conflict Detection and Resolution of XACML Policies," Master's thesis, University of Applied Sciences Rapperswil, July 2010.

[24] F. Turkmen and B. Crispo, "Performance evaluation of XACML PDP implementations," in *Proc. of the 2008 ACM Workshop on Secure Web Services*, ser. SWS. New York, NY, USA: ACM, 2008, pp. 37–44.

[25] A. Liu, F. Chen, J. Hwang, and T. Xie, "Designing Fast and Scalable XACML Policy Evaluation Engines," *IEEE Transactions on Computers*, no. 99, 2010.

[26] M. L. Pardal, M. Harrison, and J. A. Marques, "Assessment of Visibility Restriction Mechanisms for RFID Data Discovery Services," in *IEEE Int'l Conf. on RFID*, April 2012, p. 7.

[27] M. L. Pardal, M. Harrison, S. Sarma, and J. A. Marques, "Enforcing RFID Data Visibility Restrictions Using XACML Security Policies," in *IEEE Int'l Conf. on RFID Technology and Applications*, November 2012.

[28] D. Allemang and J. Hendler, *Semantic Web for the Working Ontologist, Second Edition: Effective Modeling in RDFS and OWL*, 2nd ed. Morgan Kaufmann, June 2011.

[29] G. Karjoth, A. Schade, and E. V. Herreweghen, "Implementing ACL-Based Policies in XACML," in *Annual Computer Security Applications Conf. (ACSAC)*, December 2008, pp. 183–192.

[30] C. Floerkemeier, C. Roduner, and M. Lampe, "RFID Application Development with the Accada Middleware Platform," *IEEE Systems Journal, Special Issue on RFID Technology*, December 2007.

[31] J. Shi, D. Sim, Y. Li, and R. Deng, "SecDS: a secure EPC discovery service system in EPCglobal network," in *Proc. of the 2nd ACM Conf. on Data and Application Security and Privacy*, ser. CODASPY. New York, NY, USA: ACM, 2012, pp. 267–274.

[32] F. Kerschbaum and L. W. F. Chaves, "Encryption-enforced access control for an RFID discovery service," in *Proc. of the 17th ACM Symp. on Access Control Models and Technologies*, ser. SACMAT. New York, NY, USA: ACM, 2012, pp. 127–130.

[33] A. Ilic, A. Grssbauer, F. Michahelles, and E. Fleisch, "Understanding data volume problems of RFID-enabled supply chains," *Business Process Management Journal*, vol. 16, no. 6, 2011.

[34] B. Butler, B. Jennings, and D. Botvich, "XACML policy performance evaluation using a flexible load testing framework," in *Proc. of the 17th ACM Conf. on Computer and Communications Security*, ser. CCS. New York, NY, USA: ACM, 2010, pp. 648–650.

[35] L. W. F. Chaves and F. Kerschbaum, "Industrial Privacy in RFID-based Batch Recalls," in *Proc. of the 12th Enterprise Distributed Object Computing Conf. Workshops*, ser. EDOCW. Washington, DC, USA: IEEE Computer Society, 2008, pp. 192–198.